Quantifying the Risk of Wormhole Attacks on Bluetooth Contact Tracing

Stefan Czybik
Institute of System Security
Technische Universität Braunschweig
Braunschweig, Germany

Daniel Arp
Machine Learning Group
Technische Universität Berlin
Berlin, Germany

Konrad Rieck
Institute of System Security
Technische Universität Braunschweig
Braunschweig, Germany

ABSTRACT

Digital contact tracing is a valuable tool for containing the spread of infectious diseases. During the COVID-19 pandemic, different systems have been developed that enable decentralized contact tracing on mobile devices. Several of the systems provide strong security and privacy guarantees. However, they also inherit weaknesses of the underlying wireless protocols. In particular, systems using Bluetooth LE beacons are vulnerable to so-called *wormhole attacks*, in which an attacker tunnels the beacons between different locations and creates false contacts between individuals. While this vulnerability has been widely discussed, the risk of successful attacks in practice is still largely unknown.

In this paper, we *quantitatively* analyze the risk of wormhole attacks for the exposure notification system of Google and Apple, which builds on Bluetooth LE. To this end, we dissect and model the communication process of the system and identify factors contributing to the risk. Through a causal and empirical analysis, we find that the incidence and infectivity of the traced disease drive the risk of wormhole attacks, whereas technical aspects only play a minor role. Given the infectious delta variant of COVID-19, the risk of successful attacks thus increases and may pose a threat to digital contact tracing. As a remedy, we propose countermeasures that can be integrated into existing contact tracing systems and significantly reduce the success of wormhole attacks.

CCS CONCEPTS

• **Networks** → **Mobile and wireless security**; • **Applied computing** → **Health care information systems**.

KEYWORDS

Exposure Notification Systems, Wireless Attacks, COVID-19

ACM Reference Format:
Stefan Czybik, Daniel Arp, and Konrad Rieck. 2022. Quantifying the Risk of Wormhole Attacks on Bluetooth Contact Tracing. In *Proceedings of the Twelveth ACM Conference on Data and Application Security and Privacy (CODASPY '22), April 24–27, 2022, Baltimore, MD, USA.* ACM, New York, NY, USA, 12 pages. *https://doi.org/10.1145/3508398.3511496*

CODASPY '22, April 24–27, 2022, Baltimore, MD, USA

ACM ISBN 978-1-4503-9220-4/22/04…$15.00
https://doi.org/10.1145/3508398.3511496

1 INTRODUCTION

Contact tracing is a key strategy for slowing down and containing the spread of infectious diseases. While public health services traditionally carry out this strategy through personal interviews, different automated tracing systems have been recently proposed to support the mitigation of the COVID-19 pandemic. Technically, these systems build on mobile devices, such as smartphones, that track contacts between individuals either centrally or decentrally. A prominent example of these approaches is the decentralized scheme by Troncoso et al. [20] that forms the basis of the Google/Apple Exposure Notification (GAEN) system and, by now, is deployed on millions of mobile devices worldwide [6, 20].

The GAEN system is based on a clever interplay of the Bluetooth LE (BTLE) protocol and a privacy-preserving scheme for proximity tracing. In this scheme, the proximity of mobile devices is measured by broadcasting BTLE beacons with temporary pseudonyms, which enables notifying users of exposure to infected individuals without revealing personal data [7, 8]. The system provides strong privacy guarantees in comparison to related approaches and is resistant against different types of attacks [4, 9]. Consequently, several countries have adopted it for implementing national Contact-Tracing-Apps (CTAs) for COVID-19, such as "SwissCovid", "Immuni", and "Corona-Warn-App".

Unfortunately, however, the GAEN system inherits well-known weaknesses of the underlying Bluetooth protocol. In particular, it is vulnerable to so-called *wormhole attacks*, in which an adversary tunnels beacons between locations and induces false contacts, potentially leading to increased self-isolation of individuals. This vulnerability was anticipated during the design of the system and has since been experimentally validated by Baumgärtner et al. [2]. However, the practical risk of wormhole attacks on GAEN has not been quantified yet. It is unclear what factors contribute to the success of the attacks and whether they enable weakening the efficacy of decentralized contact tracing in practice.

In this paper, we *quantitatively* analyze the risk of wormhole attacks for digital contact tracing and the GAEN system. To this end, we dissect the communication process of the system and identify factors contributing to the risk of attacks. These factors cover technical parameters, such as the communication bandwidth and pseudonym lifetime, as well as properties of the traced disease, such as its incidence and infectivity. Based on a causal analysis, we quantify the influence of these factors on the success of attacks. By collecting GAEN beacons over several months at different locations, we further complement our model with empirical measurements. Our analysis reveals that the *incidence* and *infectivity* of the traced disease primarily impact the risk of wormhole attacks, while technical aspects only play a minor role.

In particular, we find that an increasing incidence linearly raises the chances of an adversary to tunnel beacons of infected persons between locations. Moreover, an increased infectivity of the disease reduces the effort to trigger false notifications, thus further fuelling the attack. During the COVID-19 pandemic, the tracing parameters had to be changed multiple times to cope with more infectious virus variants. Thus, we conclude that the risk of wormhole attacks currently increases and may jeopardize digital contact tracing. While a few fake alerts are easily compensated, high-incidence regions and infectious variants render attacks more dangerous and can lead to manipulated self-isolation of several people.

As a remedy, we propose two countermeasures that can be easily integrated into existing systems for contact tracing with GAEN. First, we recommend adjusting the parameters of the system to reduce the risk of successful wormhole attacks. Second, we present a simple detection scheme that identifies unusual bursts of beacons. Our empirical analysis demonstrates that especially the second scheme significantly reduces the success of wormhole attacks while preserving the effectivity of proximity tracing.

The rest of this paper is organized as follows: We discuss the background of digital contact tracing and wormhole attacks in Section 2. Our model of the GAEN system is introduced in Section 3. We present its causal and empirical analysis in Section 4 and 5, respectively. Limitations and countermeasures are then discussed in Section 6 and 7. Finally, Section 8 concludes the paper.

2 BACKGROUND

During the COVID-19 pandemic, different concepts for contact tracing have been developed and extensively studied [e.g., 1, 13, 15, 19]. We focus our analysis on the Google/Apple Exposure Notification (GAEN) system that is widely employed in practice and provides strong privacy guarantees. In GAEN, the contact of individuals and the risk of infections are determined locally on mobile devices. A central service is only needed to pass anonymous information about positive diagnoses to the devices, limiting the risk of revealing personal data.

The local calculation of the infection risk is carried out using a Contact-Tracing-App (CTA) that interfaces with the GAEN framework. For our analysis, we focus on the *Corona-Warn-App (CWA)*, an open-source implementation of a CTA developed in Germany. While this mobile application performs a risk estimation conceived by the German Robert Koch-Institute (RKI), we assume that other CTAs use similar mechanisms for determining the risk on mobile devices. Hence, we restrict our analysis to modeling this estimation and assessing its influence on the success of wormhole attacks.

2.1 Contact Tracing with GAEN

The GAEN system builds on the DP3T proposal [20] and uses BTLE advertisements for the communication between devices. An undirected advertisement beacon is broadcasted every 200-270 ms to the nearby surrounding of a mobile device [7]. The beacon is called a *Rolling Proximity Identifier (RPI)* and contains all information for estimating the proximity of mobile devices and propagating information about infections in retrospection.

Technically, the RPI is composed of a Rolling Proximity Identifier Key (RPIK) and Associated Encrypted Metadata (AEM). The RPIK is derived from a Temporary Exposure Key (TEK) that is generated daily on the mobile device. The AEM contains metadata encrypted with the TEK, such as the signal strength of the sender [7, 8]. After receiving beacons, GAEN calculates the attenuation of the BTLE signal path to estimate the distance of the contact. The attenuation is the difference between the signal strength of the receiver and the sender. If the signal strength is lower at the receiver, the attenuation is positive. The attenuation is given in decibel (dB) and thus expressed on a logarithmic scale.

If a person participating in digital contact tracing gets a positive COVID-19 diagnosis, they can upload the TEKs from the last 14 days to a central service. These uploaded keys are called Diagnosis Keys (DKs) and provide the basis for determining contacts in retrospection. In particular, CTAs, such as the German CWA, download the DKs from the central service in regular intervals and check whether a locally retrieved RPI belongs to one of the DKs. If a match is detected, the user of the mobile device has been in the proximity of an infected individual in the past 14 days and a risk estimation needs to be started. In the following, we refer to a matching RPI as *positive RPI (pRPI)*, as it plays a key role in wormhole attacks.

2.2 Timing and Risk Estimation

While previous work has investigated the cryptographic scheme underlying GAEN for security and privacy issues, we take a different perspective in this paper. We focus on the *timing* and *risk estimation* of the protocol, as these aspects impact the success of wormhole attacks. Figure 1 shows an overview of the system with its different timing parameters.

Timing. On the side of the sender, an RPIK is derived from the TEK every *RPI lifetime* t_{rl}. On the receiving side, the mobile devices are listening for beacons by scanning for a short moment[1] in opportunistically intervals. This listening is characterized by two time parameters, the duration of the *scan window* t_{w} and the minimum *scan interval* t_{i}. As we see in Section 3, these time parameters also play a crucial role when estimating the efficacy of wormhole attacks on the GAEN system and the CTA.

Risk estimation. Once an exposure has been detected through a matching RPIK and DK, the CTA estimates the risk of an infection. This risk calculation combines information from the GAEN system, parameters from the CTA, and a transmission risk level provided by the central service together with the DKs. The transmission risk level describes the risk a person is transmitting the COVID-19 virus to another and is determined based on multiple epidemiological information [3].

For our analysis, we focus on the risk calculation implemented by CWA since version 1.9 from December 2020 [11]. This calculation is also known as the *ExposureWindow mode* and used by several other CTAs with the GAEN system. Starting from a matching DK, the system constructs *exposure windows* with a maximum length of 30 minutes. Each window contains a set of sightings of pRPIs along with information on their attenuation during the scan window. The

[1] *https://github.com/google/exposure-notifications-internals/blob/aaada6ce5cad0ea1493930591557f8053ef4f113/exposurenotification/src/main/java/com/google/samples/exposurenotification/ble/scanner/BleScannerImpl.java#L218-L222*

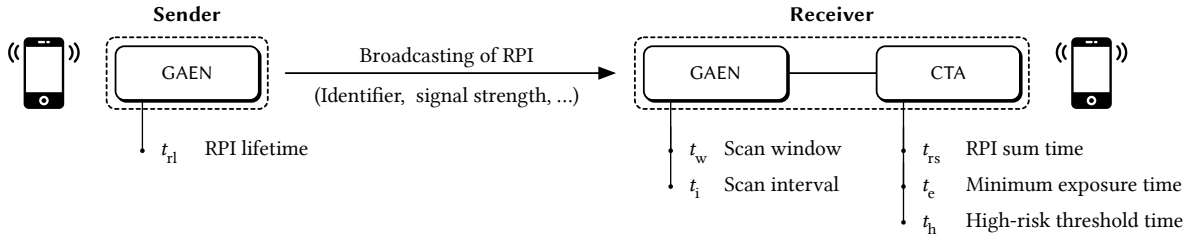

Figure 1: Overview of timings in the GAEN system.

sightings are only kept in the exposure window if they satisfy the following four conditions[2]:

(1) The RPI is technically valid, which is checked in the implementation of GAEN.
(2) The sighting lies within the exposure window with a tolerance of 2 hours, compensating clock drift.
(3) The time from the first sighting of the RPI is not more than t_{rs}, the so-called *RPI sum time*.
(4) The sender's signal strength of the sighting is between -50 dB and 20 dB. Unreasonable values are ignored.

The *RPI sum time* describes how long an RPIK is counted in an exposure window. The GAEN system drops later sightings of the RPIK. t_{rs} is twice t_{rl}, which is hard-coded in the implementation.[3]

Minimum exposure time. The CWA then conducts computations on the exposure windows returned by the GAEN framework. First, the application drops all windows where a close contact could not be measured over a given time span. That is, all exposure windows with an attenuation below 79 dB and a duration of less than the *minimum exposure time* t_e are ignored. Second, all windows with a transmission risk level lower than a given threshold are dropped. The particular threshold is defined in the CWA implementation.[4]

High-risk threshold time. After this filtering, the application computes the total contact time as a sum of the exposures. This summation process inside an exposure window has also changed since the roll-out of CWA.[5] The configuration we refer to is that scan windows with a mean attenuation below 63 dB are weighted with a factor of 0.8, between 63 dB and 73 dB with a factor 1.0, and between 73 dB and 79 dB with a factor of 0.1. Values above 79 dB are dropped.[6] The *total exposure time* per exposure window is multiplied with a normalization factor from the transmission risk level between 0.6 and 1.6[7] to get the *weighted exposure time*. A high-risk warning is finally displayed to the user if the sum of all weighted exposure times of the exposure windows is higher than *high-risk threshold time* t_h.

2.3 Wormhole Attack

The privacy protection offered by the GAEN system comes at a price: It is impossible to discriminate an authentic RPI from one that is recorded and replayed. This weakness results from the core principle of privacy-preserving proximity tracing and has been acknowledged in the design of the framework [4]. Mechanisms enabling to differentiate between authentic and replayed RPIs would either provide clues to identify the sender or induce a costly communication overhead.

As a result, GAEN is vulnerable to so-called *wormhole attacks* [10]. In this class of attacks, an adversary records network packets at one location and replays them at another. In the case of BTLE, this relaying of packets allows tunneling beacons between different locations. To carry out the attack, an adversary needs two or more BTLE devices at different places connected through a communication backend. All devices send and receive BTLE beacons. On the receiving side, the beacons get a timestamp and are transmitted to the backend. On the sending side, the beacons are retrieved from the backend and replayed, as long as they are valid and useful for the attacker. Baumgärtner et al. [2] demonstrate that such a wormhole attack is technically feasible for GAEN and allows creating fake contacts within the CWA.

3 QUANTIFYING THE RISK

We proceed to introduce our model for quantifying the risk of successful wormhole attacks. In order to develop a generic risk model for the GAEN, we first define the capabilities of an adversary carrying out the attack under ideal conditions. Based on these assumptions, we then progressively develop a model that allows us to analyze the risk of attacks and their impact in practice.

3.1 Threat Model

To quantify the risk of an attack, we require a threat model covering a precise definition of the adversary's goals and capabilities. Moreover, we need to specify technical restrictions that limit these capabilities in practice.

[2] *https://github.com/google/exposure-notifications-internals/blob/aaada6ce5cad0ea1493930591557f8053ef4f113/exposurenotification/src/main/java/com/google/samples/exposurenotification/matching/ExposureMatchingTracer.java#L573*
[3] *https://github.com/google/exposure-notifications-internals/blob/aaada6ce5cad0ea1493930591557f8053ef4f113/exposurenotification/src/main/java/com/google/samples/exposurenotification/matching/ExposureMatchingTracer.java#L663-L813*
[4] *https://github.com/corona-warn-app/cwa-server/blob/ebcf43773fe82e85cebb69c3ec2ae49999335016/services/distribution/src/main/resources/main-config/v2/risk-calculation-parameters.yaml*
[5] *https://github.com/corona-warn-app/cwa-server/commit/e88332397d4010c3854bd91de5b1a60b52658c58#diff-0e35d95ed502fde88e197efd495faf0cd4af7c215c377e5a183612a75d489ded*
[6] *https://github.com/corona-warn-app/cwa-server/blob/ebcf43773fe82e85cebb69c3ec2ae49999335016/services/distribution/src/main/resources/main-config/v2/risk-calculation-parameters.yaml*
[7] *https://github.com/corona-warn-app/cwa-server/blob/ebcf43773fe82e85cebb69c3ec2ae49999335016/services/distribution/src/main/resources/main-config/v2/presence-tracing-parameters.yaml*

Attacker's goal. For our analysis, we consider the wormhole attack as described by Baumgärtner et al. [2]. In this scenario, the attacker connects multiple locations through a wormhole to induce false warnings. The attacker succeeds if a high-risk warning in the CWA is shown [17] to a selected person, the victim. As a consequence of this notification, the person should self-isolate, which might, for instance, lead to financial losses or mental stress for the victim. The affected person is not forced to isolate, but not to do so would undermine the purpose of the CWA. We focus on a single victim, yet the adversary can easily scale the attack to several persons at all ends of the wormhole. This would lead to a more destructive threat. Still, the attack setup would be technically identical.

Attacker's capabilities. To be able to bound the practical risk for wormhole attacks appropriately, we assume a strong yet realistic attacker. In our threat model, the attacker has the technical equipment to receive an arbitrarily high amount of RPIs at multiple locations within the limits of the BTLE protocol. All these RPIs can be forwarded to another location, where the attack is carried out. The RPIs can be replayed at this location also within the limitations of BTLE. The attacker can send RPIs to the victim at any desired signal strength and as long as needed for a successful attack.

Attacker's restrictions. The attacker must take into account the limitations imposed by the design of the CWA and the underlying protocol. In particular, she cannot make changes to a user device or the central service. Therefore, despite tunneling beacons, she must adhere to the original protocol of the GAEN system. In addition, the adversary cannot determine whether an RPIK belongs to a DK, that is, is sent from a device of an infected person. Consequently, the attack can only increase the chance for a high-risk warning at the victim but never directly trigger one.

Summary of threat model

- ⊕ Receive, send, and replay RPIs
- ⊕ Manipulate the sending signal strength
- ⊕ Send RPIs long enough to the victim
- ⊖ Manipulate devices or the backend service
- ⊖ Access information about the diagnosis state

3.2 Improving the Attack

Our risk model assumes that each BTLE advertisement sent by the adversary is considered by the victim's CTA. At first glance, this seems like an unrealistic assumption, as the original attack proposed by Baumgärtner et al. [2] does not guarantee this behavior. However, in the following, we show that an adversary only needs to slightly adapt her attack strategy to meet this assumption.

Adapting the attack. To modify the attack, we first need to understand the role of the *Received Signal Strength Indication (RSSI)* for risk calculation. The RSSI, expressed in dBm, indicates the signal strength that is reaching the receiver.

Note that the RSSI does not allow deriving the signal strength used by the sender, but instead represents an independent metric. The CTA uses the RSSI and the sender's signal strength stored in an RPI to calculate the attenuation, which, in turn, allows computing the final risk score. From Section 2, we already know that the CWA only considers RPIs with an attenuation lower than 79 dB for the risk estimation. Consequently, we need to ensure that the relayed RPIs in a wormhole attack match this condition.

When a victim approaches the attacker's device, the attacker collects RPIs broadcasted by the victim's smartphone. From these RPIs, she then extracts the meta information of the RSSI and uses it to estimate the distance between both devices. Because the attenuation of a signal path is always the same in both transmitting directions, the attacker can calculate and increase the sender's signal strength for the BTLE advertisement, allowing her to compensate for the attenuation. As a result, the attacker ensures that the risk calculation of the victim's device considers all RPIs sent by the attacker, increasing the effectiveness of the attack.

While this modification of the attack seems straightforward, the attacker must overcome a non-trivial problem first. As previously discussed, the attacker needs to know the sender's signal strength and the RSSI to derive the attenuation, which is used to calculate the overall risk of infection. Unfortunately, the attacker has no information about the replayed RPI's signal strength, as it is stored in the AEM and, therefore, not available to the adversary at attack time. Furthermore, it is not possible to derive the sender's signal strength from the RSSI, as there exists no standardized relationship between the RSSI and any particular physical parameter.

As a remedy, the attacker can perform calibration experiments with test devices in advance. If the attacker knows the victim's smartphone, she can use a similar device to estimate the required signal strength. Note that previous research has already successfully demonstrated the feasibility of such an approach to estimate unknown parameters [5, 12, 14]. In the following, we thus assume that all RPIs received by the victim's device are considered by the CTA for risk estimation.

3.3 Modelling the Risk

Based on our threat model and the improved attack, we can now introduce a model for bounding the risk of a wormhole attack. To this end, we first describe the underlying assumptions and discuss why these are sufficient for bounding the risk in practice. Afterward, we present our risk model to compute the risk of a successful attack in a worst-case setting.

Model Assumptions. As we consider a worst-case scenario, we can omit various factors that would otherwise need to be taken into account. Further, we make several simplifying assumptions that reduce the complexity of the resulting risk model. Note that these assumptions do not affect the outcome of our analysis, as we are solely interested in bounding the risk of wormhole attacks. We discuss the effect of the assumptions in Section 6.

In total, we make four assumptions:

A.1 pRPIs are equally distributed under all RPIs
A.2 All RPIs are recognized by the risk calculation
A.3 Movement of people is equally distributed
A.4 All people report their diagnosis

The first assumption allows us to work with mean values of several model parameters and omit high-order statistical moments. The second assumption enables neglecting the impact of the scan interval, packet loss and attenuation. Moreover, we do not model

the delay of reported infections in practice. All other parameters like the transmission risk are assumed as worst-case values for the construction of our model.

A simple risk model. Let us start by designing a simple model for the risk of a successful wormhole attack. On an abstract level, a successful attack mainly depends on two parameters: The first variable is the number of received pRPIs by the victim, which we denote as n_a. The second variable is the minimum number of pRPIs required to trigger a high-risk warning in the CWA denoted as n_i. That is, an attacker obviously succeeds, if she can ensure that

$$n_\mathrm{a} \geq n_\mathrm{i} \tag{1}$$

Note that we do not model *when* high-risk warnings appear, and hence both variables are time-independent in our analysis.

While the above described relation between n_a and n_i is straightforward, it is so far unclear how values or bounds for these variables can be analytically determined. The following sections thus discuss which factors influence n_a and n_i, as well as how we can use this knowledge to bound the overall risk of successful attacks.

3.4 Modeling the Variable n_i

The number of needed pRPIs for a successful attack n_i depends mainly on values from the risk calculation of the GAEN and CWA. If the sum of all weighted exposure times of the exposure windows is higher than the high-risk threshold time t_h, a high-risk warning is displayed. The maximum amount of time one pRPI can contribute to the risk calculation needs to consider the RPI sum time and the maximum transmission risk level factor. From this follows that the maximum amount of time is $t_\mathrm{rs} \cdot \tau$. We need to divide t_h by $t_\mathrm{rs} \cdot \tau$ and ceil the result to get n_i:

$$n_\mathrm{i} = \left\lceil \frac{t_\mathrm{h}}{t_\mathrm{rs} \cdot \tau} \right\rceil \tag{2}$$

In Equation (2), there is no influence of the minimum exposure time t_e so far. This formulation is correct, but a constraint is missing now. If the GAEN needs more than one RPI from the same TEK in the risk calculation, the exposure of only one RPI will be ignored. This happens when $t_\mathrm{rs} < t_\mathrm{e}$. We can calculate the number of RPIs n_c we need for recognition in the risk calculation with Equation (3). This number tells us how many continuous RPIs from the same TEK are needed for a successful attack:

$$n_\mathrm{c} = \left\lceil \frac{t_\mathrm{e}}{t_\mathrm{rs} \cdot \tau} \right\rceil. \tag{3}$$

We assume that the attacker is replaying an RPI always with t_rs. We discuss this assumption later in Section 3.6.

3.5 Modeling the Variable n_a

Only a tiny fraction of the received RPIs at the victim are pRPIs. The size of the fraction depends on two factors: The total amount of RPIs that the victim's device can receive during a single scan window, denoted as n_r, and the incidence of the traced disease, that is, the number of infected individuals participating in digital contact tracing using GAEN.

Maximum RPIs per scan window. We start by bounding the maximum number of RPIs that can be sent during a scan window to a device n_r. The upper bound of this number shown in Equation (4) depends mainly on the limitations of the BTLE protocol and the scan window t_w of the GAEN. From the GAEN specification, we know the data size of the advertisement is $D = 376$ Bit. BTLE advertisements have a data rate of $C_{BT} = 1$ MBit/s per channel. Between two advertisements, there needs to be an inter-frame space $IFS = 150$ Bit. This means that there is an overhead of $O_{BT} = \frac{D+IFS}{D} = 1.40$ [18]. Consequently, we can bound the maximum number of receivable RPIs per scan window as follows

$$n_\mathrm{r} = \left\lfloor \frac{C_{BT}}{D \cdot O_{BT}} \cdot t_\mathrm{w} \right\rfloor. \tag{4}$$

Infection Incidence. As a second factor contributing to the variable n_a, we consider the infection incidence I. This epidemiological value is typically defined as the mean 7-day incidence per 100,000 population of the traced disease. As only people using the CWA are vulnerable to wormhole attacks, however, we simply model the incidence of the CWA devices, that is, the ratio of devices associated with an infection per 100,000. Thus, our model is comparable to the current 7-day incidence of the SARS-CoV-2 virus published by the RKI for Germany.

Like n_i, we have also an influence to n_a if n_c is bigger than 1 or not. If n_c is bigger, this means that continuous RPIs over at least t_e are needed. Otherwise, their exposure windows are not recognized by the risk calculation. RPIs are often not continuous due to the movement of mobile devices from which they are collected. To model this issue, we introduce the parameter M that represents the percentage of people fluctuating at the collection site, where 1 means all people are exchanged during the RPI lifetime. If we put this together, we obtain Equation (5) for the number of pRPIs transmitted during the attack.

$$n_\mathrm{a}(I, M) = n_\mathrm{r} \cdot \frac{I}{10^5} \cdot \begin{cases} 1 & \text{if } n_\mathrm{c} \leq 1 \\ (1 - M) & \text{if } n_\mathrm{c} > 1 \end{cases} \tag{5}$$

3.6 Sending and Collecting RPIs

Equipped with bounds for Equation (1), we can continue to model the practical operation of a wormhole attack. From Equation (4), we know how many RPIs need to be collected by an adversary to fill a whole scan window. In practice, however, it is also useful to know how many RPIs an attacker need to send to reach n_i pRPIs in a scan window. We can calculate this number, denoted as n_m, as follows:

$$n_m(n_\mathrm{i}, I, M) = n_\mathrm{i} \cdot \frac{10^5}{I} \cdot \begin{cases} 1 & \text{if } n_\mathrm{c} \leq 1 \\ \frac{1}{(1-M)} & \text{if } n_\mathrm{c} > 1 \end{cases} \tag{6}$$

This number plays an important role when we later investigate different practical attack scenarios and conditions.

Attack Strategies. Once the adversary has collected RPIs, there are different strategies for sending these to the victim. In a classic wormhole attack, a packet is simply relayed to another location. An improvement of this attack is to replay collected RPIs several

times. For the GAEN system, this repetition is helpful, as it increases the chances of RPIs to be recognized by the victim during a scan window. An ideal replay time is the RPI sum time, enabling an attacker to optimally use collected RPIs. If the attacker acquires only a limited amount of RPIs, a *burst replay* provides good performance within this time. The RPIs are collected in a two-hour window and replayed in bursts to the victim in intervals of the RPI sum time. By contrast, if a large amount of RPIs is available, the attacker can also conduct a *flooding replay*. At places where people stay around the RPI sum time, RPIs can be continuously replayed for two hours for maximum utilization.

An attacker can combine and mix both strategies arbitrarily. Based on our threat model, we focus on a targeted attack on one victim. That is, we consider an adversary aiming at creating false high-risk warnings for one person at the sending site. Since our model is time-independent, the concrete strategy thus has no real influence on our analysis and model.

Order of RPIs. If we have the case that $n_c > 1$, the attacker needs to ensure that RPIs from the same TEK are replayed in order. A solution for this problem is to replay RPIs every t_{rl} instead of t_{rs}. This acceleration is easy to conduct but doubles n_c. A replay of t_{rs} in order is not trivial, because there is no obvious information about which RPIs belong to the same TEK. A stretched replay might be a solution for this, where the sequence of collected RPIs are replayed in a double amount of time. This stretched replay will ensure the order and increase the replay time from t_{rl} to t_{rs}.

Collecting RPIs. Several parameters affect the collection of RPIs at the endpoints of the wormhole attack. Theoretically, each device can collect n_r per scan window t_w. However, this optimal collection is usually not feasible in a real scenario due to the limited availability of devices near the collection point. For this reason, we do not create an analytical model for collection and instead conduct a field study to fill in the missing information. This empirical analysis is presented in Section 5.

4 CAUSAL ANALYSIS

Our risk model allows us to study the influence of various factors on the success of a wormhole attack for a arbitrary CTA using the GAEN. In the following, we examine the effects of those five factors that mainly determine the effectiveness of the attack. For this causal analysis, we focus on the current configuration of the German CWA app. Since RPI sum time is the only variable in our model that does not depend on the disease being tracked, it is plotted on the x-axis in the following figures. Note that all figures in this section refer to the basic equation (1).

Table 1: CWA configuration at different times.

Date	t_h	t_e
November 2020[8]	15 min	10 min
February 2021[9]	15 min	5 min
March 2021[10]	13 min	5 min
April 2021[11]	9 min	5 min

Current CWA configuration. The configuration of the app has changed several times in the past. In particular, the values of t_h and t_e have been adjusted to account for the increasing infectivity of the COVID19 virus. Table 1 lists these different configurations. For our causal analysis, we use the values defined in April 2021 [14]. Based on these values, we arrive at the following configuration of our risk model which captures the behavior of the GAEN system and the CWA in a worst-case scenario:

$$t_w = 4\,s \quad t_i = 3\,min \quad t_e = 5\,min$$
$$t_h = 9\,min \quad t_{rl} = 10\,min \quad t_{rs} = 20\,min$$
$$n_c = 1 \quad n_i = 1 \quad n_r = 7787$$
$$\tau = 1.6$$

The value $n_c = 1$ simplifies the attack a lot because each collected RPI can be replayed for t_{rs} and will be recognized in the risk calculation. Further, M is not relevant in this current configuration and thus omitted in the following analysis.

4.1 Influence of Incidence

First, we examine the influence of the incidence of the CWA devices. From Equation (5) we can see that n_a has a linear dependency on I. Figure 2 shows this dependency. Although the plot looks trivial at a first glance, we observe that only for I equal or higher than 12.84, at least one positive RPI can be transmitted during the attack. This is a lower bound for a successful attack in general for the current configuration.

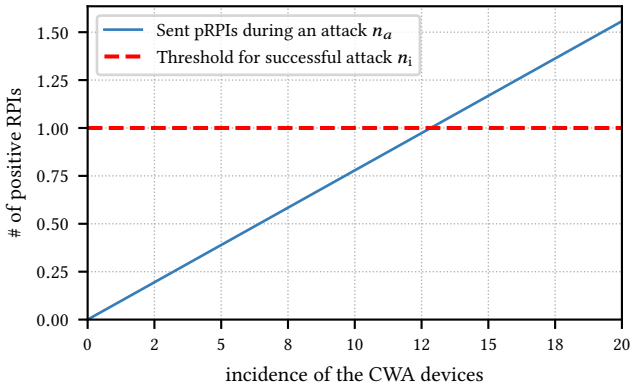

Figure 2: Number of sent positive RPIs during an attack depending on the incidences at the current CWA configuration

[8] *https://github.com/corona-warn-app/cwa-server/commit/4f4da45c99ad311363b42b16d6950152eb8f65c6*
[9] *https://github.com/corona-warn-app/cwa-server/commit/f5b3490b5ecca692661e26a98b8b7af9ed149153*
[10] *https://github.com/corona-warn-app/cwa-server/commit/e88332397d4010c3854bd91de5b1a60b52658c58*
[11] *https://github.com/corona-warn-app/cwa-server/commit/9e3b0990fc8e82929c0c7e060a4d8a8e5ebaf326*

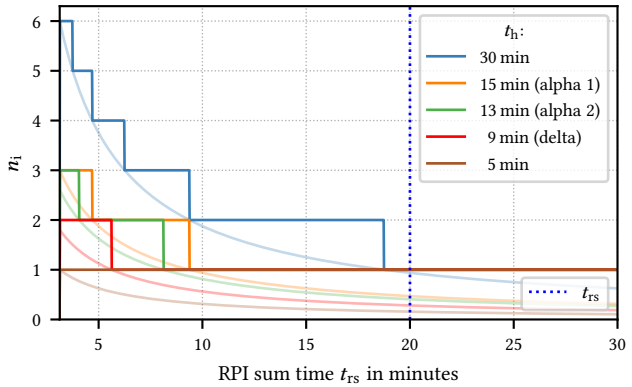

Figure 3: Number of needed positive RPIs for a successful attack depending on the RPI sum time for different high-risk threshold times t_h.

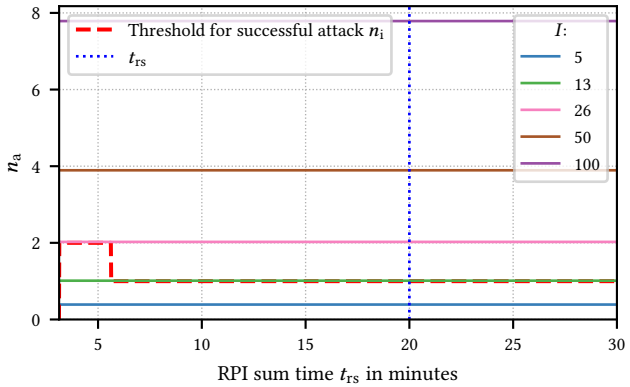

Figure 4: Number of send positive RPIs during an attack depending on the RPI sum time for different incidence of the CWA devices I

Take-away message 1

For an incidence of the CWA devices below 12.84, wormhole attacks are ineffective in the current configuration and do not pose a risk.

Depending on the configuration of the CWA, however, sending a pRPI does not necessarily mean that an attack is successful. In the following, we show how the value of t_h can impact the effect of successful attacks.

4.2 Influence of Threshold Time

From Equation (2), we know that the number of needed pRPIs for a successful attack n_i is correlated with the high-risk threshold time t_h. That is, if we lower t_h, we automatically decrease n_i. Figure 3 shows n_i for different values of t_h, where the x-axis represents the RPI sum time t_{rs}. The values of t_h are plotted in different colors and the fine lines are the underlying continuous functions. The blue dashed line marks the current configuration of t_{rs} in CWA.

Interestingly, even at a high-risk threshold time of 32 minutes, only a single pRPI is needed to perform a successful attack. This low threshold is mainly because of the high infectivity of the current virus variant. Even at a lower t_{rs} only an increment of n_i to two would be possible. Given the increasing infectivity of COVID-19, wormhole attacks thus become more effective.

Take-away message 2

Only a single positive RPI is needed for a successful wormhole attack if the high-risk threshold time in CWA is below 32 minutes, as in the current configuration.

4.3 Interplay of Incidence and RPI Lifetime

Since we know that there is a lower bound for the incidence of the CWA devices I, we can plot n_i and n_a for different values of I. In Figure 4, we see n_a on the y-axis for varying incidences in different colors. The threshold for a successful attack is the red dashed line and is the same as in Figure 3.

From this plot, we can derive the risk if the intersection between I and t_{rs} is above the threshold of n_i. If the intersection of the value of incidence of the CWA devices and the current RPI sum time is above the threshold, a successful attack becomes possible. The larger the distance, the easier an attack can be performed, since fewer RPIs need to be tunneled. Hence, we can deduce that the current configuration is vulnerable to wormhole attacks. In addition, we observe that a lower RPI sum time t_{rs} would reduce the risk at lower incidences.

Take-away message 3

The current configuration of CWA is vulnerable to wormhole attacks, whose effectivity linearly depends on the incidence of the CWA devices.

4.4 Further Influences

Minimum exposure time. Another value that the infectivity of the virus could impact is the minimum exposure time. In Figure 5, the number of needed pRPIs for a successful attack is shown for different values of the minimum exposure time t_e, similar to Figure 3. We observe that the influence of the minimum exposure time is inversely related to the high-risk threshold time shown in Figure 3. Furthermore, the point where n_c is larger than 1 is increasing together with t_e. Note that we do not show values for $n_c > 1$, because they are not comparable. Instead we mark the threshold for $n_c > 1$ with a vertical line to zero.

We deduce that the larger the minimum exposure time t_e is, the higher is the threshold, where $n_c > 1$. Similarly, the lower t_e is, the more difficult an attack would be at a lower RPI sum time t_{rs}. However, we are far away from a point where t_e impacts the current configuration, leading to the following result.

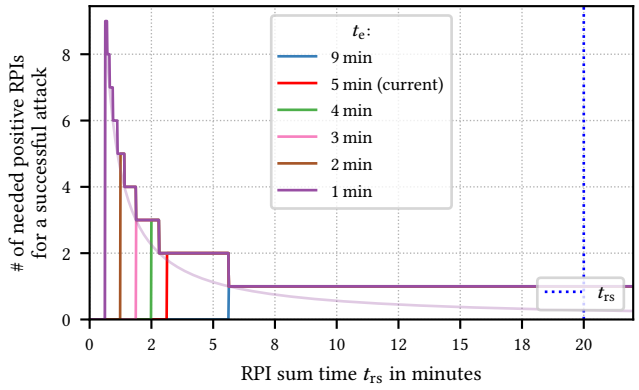

Figure 5: Number of needed positive RPIs for a successful attack depending on the RPI sum time for different minimum exposure times.

Take-away message 4

The minimum exposure time plays a negligible role in the success of a wormhole attack in the current configuration of CWA.

Movement of people. As the last aspect in this causal analysis, we want to look at the impact of people's movement on our model. This movement happens at the places where the RPIs are collected and depends on the particular location and occupation of the people. For example, at a train station people are moving fast between gates, while in a café they are sitting for a longer time.

From Equation (5), we know that the movement is only relevant for $n_c > 1$. Then the attacker needs to collect continuous RPIs for a successful attack. M is a linear factor to n_a because of our assumption A.3. In view of the current infectivity of COVID-19 and the RPI sum time, the movement of people thus does not impact the effectivity of wormhole attacks.

Take-away message 5

The current configuration of CWA prevents an influence of people's movement on the success of a wormhole attack. However, for $n_c > 1$, it becomes a significant factor.

5 EMPIRICAL ANALYSIS

Our model of the GAEN system shows that various factors can contribute to the success of wormhole attacks. So far, however, our model is not linked to practice and hence we fill this gap now. To this end, we conduct a field experiment and collect real-world RPIs using two different strategies: First, we use a regular smartphone and capture BTLE beacons by moving around at different places. This *dynamic* collection provides us with information for scenarios where the adversary creates a non-stationary wormhole. Second, we position recording devices at different locations over a period of several weeks. From this *static* collection, we extrapolate the effort required to collect different amounts of RPIs in practice.

5.1 Ethical Considerations

Due to the design of GAEN and its privacy guarantees, the collected BTLE beacons do not contain any identifiable, private information and can be considered anonymous [16]. Therefore, this collection does not fall under private data regulations. Still, we followed ethical practices as defined in the ACM Publications Policies and defined an experimental process for working with the data. We obtained permission from the static site owners to acquire data and informed the participants through signs about the collection. To protect the data, we encrypted it at the collection sites and decrypted it only temporarily during the experiments, thus limiting potential abuse.

5.2 Dynamic Collection

The dynamic collection of our field experiment has been conducted in August 2021. We use the GAEN backend in an Android smartphone to capture data during daily activity. This includes commuting between home and work locations, moving around at public places, and traveling by train to different big cities in Germany. This collections aims at mimicking the typical usage of a CTA and the data recorded is identical to what the application would have collected in practice.

During the collection time, we observe different densities of RPIs, reaching peaks of up to 59 unique RPIs per scan window, that is, per 4 seconds in the current configuration. Figure 6 shows the distribution of the number of unique RPIs per scan window over the collection period. The highest value has been monitored at a central train station of a German city during rush hour. At this location, an attacker can capture a notable number of unique RPIs and tunnel them to other places. Because the RPIs are changing every 10 minutes, multiple RPIs might belong to the same TEKs. Hence, we cannot extrapolate concrete values from this result.

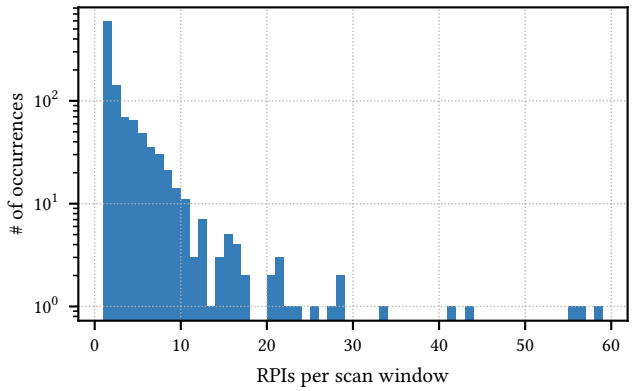

Figure 6: Seen RPIs per scan window during the data collection in August 2021 with a smartphone

To refine our analysis, we use a dedicated mobile application to measure BTLE data *continuously*. With this application, we visit the central train station of a German city and record RPIs during typical gate changes. Our highest collected amount is 126 unique

RPIs in 07:54 minutes during the rush hour. This corresponds to 0.268 identifiers per second and provides a promising basis for an attack. In the same data, the peak per scan window is 45 RPIs, which matches the scale of our previous measurements shown in Figure 6. Note that due to the lifetime of the RPIs, we only consider measurements up to 10 minutes in this experiment.

5.3 Static Collection

To monitor the amount of RPIs over a longer period of time, we collect BTLE data between August and November 2020 at different locations in a German city. For this experiment, we use five Raspberry PI Zero W as recording devices. We place the devices in a book store, a public library, a supermarket, and a café. The places differ in the visitor dynamics and thus allow for analyzing the effectiviy of wormhole attacks in different settings.

Table 2: Description of our collection locations.

Location	Size	Duration
Big book store	2.000 m²	43 days
Library entrance	4.300 m²	92 days
Library sitting areas	4.300 m²	92 days
Big supermarket	3.500 m²	58 days
Café	100 m²	33 days

In the book store, we place the recording device at the point of sale near the entrance, which is connected to the pedestrian zone of the city center. In the library, we position one device near the entrance to catch visitors, and another device near the workplace where people sit and read. In the supermarket, we place the device at the point of sale in the middle of the entrance area. Finally, the device in the café is set up near the entrance, which also leads to the city center. Table 2 provides more details about the five locations and the individual collection periods. Due to different agreements with the owners of the sites, the collection periods vary between 33 and 92 days.

Table 3: Average stay duration and number of unique RPI at the collection locations.

Location	Max RPIs per t_{rl}	Average stay per RPI in minutes t_{s}
Big book store	168	3:39
Library entrance	56	6:01
Library sitting areas	46	8:58
Big supermarket	124	6:31
Café	62	5:15

Table 3 lists the amount of collected RPIs and the average stay duration for the different places. At the library sitting areas, the average RPI stay time t_{s} reaches the maximum with over 8 minutes. People usually spend multiple hours at this place, so we see most RPIs for the whole RPI lifetime of 10 minutes. The book store is in a pedestrian zone where many people are walking by. The recording device at the store recognizes several RPIs with an average time of roughly 4 minutes. We assume that not only beacons are collected from visitors, but also from passers-by at this point. Moreover, the number of unique RPIs also differs between the locations, where the book store and the supermarket yield the highest values with 124 and 168 identifiers per RPI lifetime, respectively.

Compared to a dynamic acquisition, an installation at a fixed location has the advantage that no person is needed to record the data. A device can simply be placed somewhere hidden and serve as an endpoint for a wormhole attack by transmitting and receiving tunneled beacons.

5.4 Estimating the Attack Effort

Equipped with the results of a dynamic and static data collection, we are ready to link these measurements to our model of the GAEN system. From Section 3, we know how many RPIs are required for a successful attack, that is, tunneling at least one pRPI through the wormhole. Therefore, to evaluate the attack effectiveness, we can now calculate how many RPIs can be recorded in two hours and how many simultaneous endpoints are required to render the attack successful for a given incidence level.

Let us start by investigating the number of unique collected RPIs in two hours. As the RPIK changes every RPI lifetime, multiple identifiers can belong to the same TEK. To account for this, we estimate the adjusted collected RPIs with Equation (7). In particular, we calculate a lower estimate for the number of suitable RPIs from different TEKs as the ratio of average RPI stay time and RPI lifetime as follows,

$$n_{\mathrm{a}} = n_{\mathrm{t}} \cdot \left(1 - \frac{t_{\mathrm{s}}}{t_{\mathrm{rl}}}\right). \tag{7}$$

To perform this estimation over two hours, we use a sliding window with a step of one minute and calculate the maximum value of all windows.

Table 4 shows the results of this estimation for the five places. Although we observe a considerable number of identifiers, we also find that it is impossible to collect enough data in two hours to fill the scan window from a single place. Recall that the scan window can hold up to 7787 RPIs. If we assume a worst-case scenario, multiple recording locations are necessary to reach an optimal attack that only depends on external factors, such as the incidence or infectivity.

Table 4: Maximum amount of collected RPIs in two hours in the different locations

Location	Collected in 2h	
	n_{t}	n_{a}
Big book store	1208	767.08
Library entrance	306	121.89
Library sitting areas	241	24.90
Big supermarket	703	244.88
Café	390	185.25

Table 5 provides an overview of the number of locations needed at different incidence levels. The numbers are calculated so that the attacker achieves at least one pRPI per full scan window. The most efficient location is the book store in the pedestrian zone.

A successful attack requires between 63 and 2 instances of such a location, depending on the incidence. The other places are less effective and need up to 3.6 times more recording spots. However, for a high incidence of 500, all places require less than 10 instances, making an attack clearly feasible in practice. Moreover, the book store and the train station provide similar attack capabilities already for an incidence of 100.

Table 5: Number of needed places to carry out a successful attack for different incidences.

Location	Needed for n_r				
Incidence I:	13	26	50	100	500
Big book store	63	32	17	9	2
Library entrance	188	94	49	25	5
Library sitting areas	228	114	60	30	6
Big supermarket	85	43	22	11	3
Café	170	85	44	22	5
Train station	~66	~33	~17	~9	~2

Take-away message. The incidence is a key factor contributing to the success of wormhole attacks in practice. While the infectivity of the tracked disease affects the risk estimation within the CTA, the incidence is an external factor that negatively correlates with the number of attack endpoints. Our measurements show that at a high incidence, the number of endpoints becomes small, allowing an adversary to carry out the attack with little effort. At a moderate incidence of less than 100, wormhole attacks are still possible, but require a significant number of sites to be successful.

6 LIMITATIONS & DISCUSSION

Our risk model of decentralized contact tracing characterizes a complex process that spans multiple network and processing layers. To deal with this complexity, we make several assumptions in our threat analysis for simplification. While we are generally careful to always assume a more powerful adversary and consider a worst-case scenario, our assumptions also impose limitations, which we discuss below.

6.1 Movement Dynamics

Our empirical measurements are affected by the movement of people. The amount of RPIs varies depending on the time of data collection and the location of recording. Figure 7 shows the mean distribution of collected data in the book store as an example. Many RPIs are collected during opening hours between 11:00 and 18:00, when people are also in the nearby pedestrian area. Moreover, we see typical differences between weekdays and weekends. The other locations have different distributions, but the variance of the data is comparable, reflecting the typical movement of people.

An adversary needs to account for these dynamics, as RPIs cannot be replayed over a period longer than 2 hours. For example, attacks on the book store are less effective at night than during business hours. However, to avoid underestimating the attacker's capabilities, we focus on the period with the highest activity and base our conclusions on this worst-case scenario.

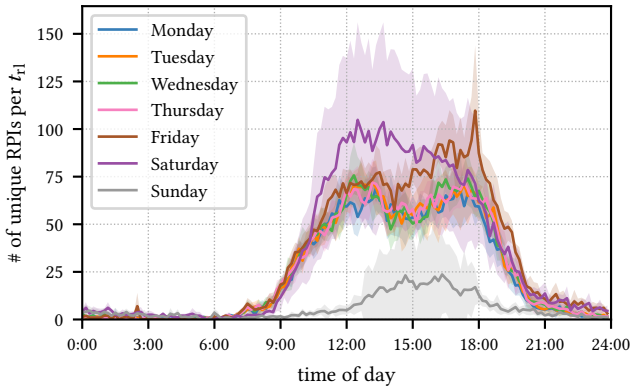

Figure 7: Number of collected RPIs per RPI lifetime (10 minutes) in the big book store

6.2 Transmission Risk Level

The risk estimation of CWA includes a transmission risk level that is determined externally and varies over time. Shortly after a positive diagnosis, the level gets the highest value and then gradually decreases. For more information, see the CWA documentation [3] that describes how the transmission risk level is calculated.

The success of a wormhole attack depends on the level of the transmission risk of the involved pRPI. Therefore, to gain insights into this parameter, we download the DKs of 20 days and analyze the contained level values. In Figure 8, we see the distribution of the transmission risk levels.

We observe a high transmission risk at level 1, which is ignored in the risk calculation. The other levels 3 to 8 are considered with a factor τ between 0.6 and 1.6. For our analysis, we focus on the 1.6 factor, which covers 9.71% of cases, as this reflects a worst-case attack scenario. Nevertheless, we also perform our calculation with $\tau = 0.6$. Because $t_{rs} \cdot 0.6$ is bigger then t_h, the factor of 0.6 will not lower the threshold for a possible attack at a frequency of 12.84 and does not change n_i. Consequently, in the current configuration, all pRPIs with a transmission risk level of 3 and higher can be used for an attack, which corresponds to over 63% of the collected data.

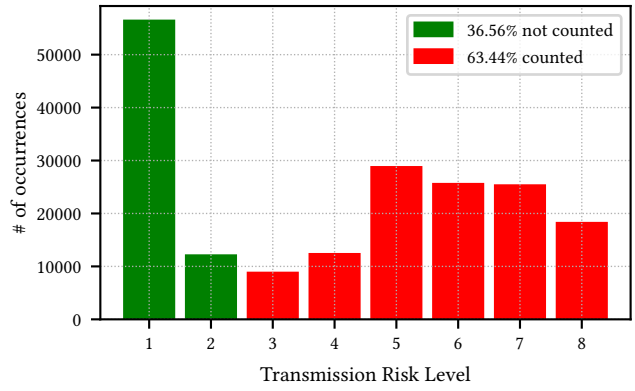

Figure 8: Distribution of the transmission risk during 20 days in September 2021

6.3 Variance of Incidence

During the development of the COVID-19 pandemic, the spread in different population groups varied significantly, reaching values above 1,000 under certain conditions. An attacker can exploit this variance to strengthen its attack, for example, by targeting regions and groups with high incidence. Since the incidence is a very complex parameter whose dynamics are difficult to characterize, we assume a uniform distribution for our analysis. That is, we focus on an average-case scenario for this parameter.

6.4 Packet Loss

In our calculation, we do not consider packet loss in BTLE transmission. BTLE uses the 2.4 GHz ultra-high frequency band, which is shared with other wireless technologies like Wifi. The protocols use distributed access to the medium, so collisions and packet loss can naturally occur. To handle these problems, the GAEN system uses the defined scan interval and scan window.

For an attack, the packet loss influences n_r as a factor of η. It has a linear influence on the number of RPIs an adversary can use during an attack but does not change the number of needed pRPIs for a successful attack. If we assume a packet loss of 50%, an attack is possible at an incidence of the CWA devices of 25.68 and 3894 RPIs can be used during a scan window. Still, we assume no packet loss when investigating wormhole attacks in our analysis, thus reflecting a worst-case scenario with respect to this factor and perfect transmission conditions.

6.5 Attack Success Delay

Finally, we do not explicitly consider the delay of a successful attack in our analysis. This delay results from the diagnostic process of COVID-19 and the deployment of the corresponding DKs. The delay spans a few days, so estimating the incidence and amount of pRPI becomes non-trivial due to the dynamic development of the pandemic. However, this uncertainty also cannot be exploited by an adversary and thus is irrelevant for our analysis.

7 COUNTERMEASURES

Our analysis does not only uncover factors contributing to this risk of wormhole attacks, but also provides starting points for constructing countermeasures. In this section, we explore these points and introduce two countermeasures, one of which focuses on improving the configuration of the GAEN system and the CWA, while the other is based on detecting ongoing attacks.

7.1 Configuration Improvements

It is evident from our analysis that many parameters are relevant to the success of wormhole attacks. Most of these are dictated by technical properties of the protocol and characteristics of the traced disease. For example, the scanning interval of the system is tailored to the noisy transmission of BTLEs, and the risk estimation is carefully adapted to the infectivity of the disease. As a result, there is limited room for optimizing the configuration of GAEN towards robust proximity tracing.

Improved RPI lifetime. The only major parameter we can change to reduce the risk of a wormhole attack is the RPI lifetime t_rl. If the lifetime is reduced, the attacker is forced to collect *more* and *continuous* beacons from the same devices, making tunneling from crowded locations difficult. At the same time, however, a reduced RPI lifetime requires more battery power and memory storage for proximity tracing. Hence, we seek a trade-off between attack robustness and resource efficiency.

From Section 3.4, we know that for $t_\mathrm{rs} \leq \frac{t_\mathrm{e}}{\tau}$ the number of RPIs needed for triggering the risk calculation is larger than 1. We can reach this point by setting $t_\mathrm{rl} = \frac{t_\mathrm{rs}}{2 \cdot \tau}$. An attack now becomes impossible for an incidence of the CWA devices smaller than 25.68. Furthermore, the attacker's effort for collecting RPIs considerably rises, since the RPIs must be continuous and from the same devices. As a result, the adversary is forced to move away from dynamic and crowded locations to places with longer dwell time, thereby reducing the attack effectivity.

Reduced clock drift. Another possible improvement can be devised by reducing the clock-drift compensation in the risk estimation. Currently, GAEN employs a two-hour window for this compensation. As the CTA needs an Internet connection for the risk calculation anyway, it is possible to synchronize the clocks of all participating devices and thereby reduce the drift window significantly. This reduction increases the attacker's effort, since RPIs can now only be replayed for a shorter period of time. Nevertheless, an attack remains possible, and only its effectiveness is reduced. Additionally, to compensate for a delayed change of the RPIK, there is the difference between the RPI lifetime and RPI sum time. In contrast to the two-hour window, reducing this difference would directly reduce the risk of an attack.

Take-away message. Adjusting the RPI lifetime and clock-drift compensation are two simple but effective strategies to reduce the chances of a successful wormhole attack on GAEN. The remaining attack risk depends mainly on the incidence and infectivity of the tracked disease. Consequently, additional countermeasures are required when GAEN is deployed in regions of high incidence or when more infectious variants are traced.

7.2 Attack Detection

Several methods have been developed for detecting and mitigating wormhole attacks in wireless networks [see 22]. However, these methods cannot be directly applied to the GAEN system, as it lacks authentication primitives and distance-bounding protocols by design [21]. As a consequence, existing detection methods can neither identify the relay nor the replay of beacons in an attack.

Our focus is to introduce easy implementable countermeasures. From Section 5.2, we know how many RPIs are received during a scan window in normal or even crowded circumstances. In our experiments, the peak has been 59. If we compare this number with the necessary amount of RPIs for an attack, we observe a huge difference. This difference reduces with the incidence of the traced disease, as less RPIs are necessary for inducing one pRPI. At an incidence of 100 we need 1000, at 500 200, and at 1000 still 100. Even if we look at the extreme cases in Section 5 where the density of devices is very high, we never reach RPI bursts comparable to a wormhole attack.

Thus, from our empirical measurements, a simple threshold is enough to reliably detect wormhole attacks. If we choose a threshold of 100 RPIs per scan window the detection works correctly up to an incidence of the CWA devices of 1000. If the incidence further increases, it becomes difficult to detect an attack with a simple threshold alone. During our study, we have also experimented with more complex countermeasures, such as learning-based approaches for detecting attacks that might be effective in this scenario. For example, we can check if there are statistical anomalies in the RSSI distribution. If the GAEN identifies 100 mobile devices within a radius of 2 meters, an attack is likely to happen. We leave these approaches for future work.

Take-away message. A simple attack detection is easy to implement at the cost of slightly increased power consumption. If an attack is detected, the RPIs in the current scan window can be discarded and the user informed of an anomalous situation. This detection response renders successful attacks ineffective while preventing false positives, such as when the user intentionally comes into close contact with hundreds of other people.

8 CONCLUSION

Digital contact tracing has proven to be a useful tool in containing the spread of COVID-19 and could also play an important role in combating future pandemics. However, recent research suggests that decentralized BTLE-based tracing systems are inherently vulnerable to wormhole attacks that can lead to unnecessary self-isolation and undermine acceptance of the system. In this paper, we investigate this problem and quantitatively assess the risk of such attacks on the GAEN system by Google and Apple.

By modeling the timing and risk estimation of GAEN, we can precisely determine the influence of different variables on the feasibility of wormhole attacks. Our analysis reveals that the incidence and infectivity of the traced disease primarily impact the risk of wormhole attacks, while other factors are of less relevance. Furthermore, we show that the effort for such an attack is realistic in practice, as the attacker only needs to tunnel RPIs from highly-frequented locations, such as train stations or supermarkets.

To counter this threat, we explore different directions to thwart wormhole attacks. First, we analyze the configuration of the underlying system and propose changes that reduce the effectiveness of attacks. Second, we propose a detection method that identifies anomalous bursts of RPIs. In combination, both defenses significantly raise the bar for adversaries to conduct wormhole attacks. While our work cannot generally rule out relay and replay attacks on digital contact tracing, we show with this work that careful analysis of the protocol elements helps harden a system and quantify its risk to attacks.

ACKNOWLEDGEMENTS

The authors would like to thank Robin Heinbockel for implementing a collection device and acquiring data for our empirical studies. The authors also acknowledge funding from the German Federal Ministry of Education and Research (BMBF) under the project BIFOLD (Berlin Institute for the Foundations of Learning and Data, ref. 01IS18025A and ref 01IS18037A) and the Deutsche Forschungsgemeinschaft (DFG) under the project 456292433.

REFERENCES

[1] N. Ahmed, R. A. Michelin, W. Xue, S. Ruj, R. Malaney, S. S. Kanhere, A. Seneviratne, W. Hu, H. Janicke, and S. K. Jha. A survey of covid-19 contact tracing apps. *IEEE Access*, 8:134577–134601, 2020.

[2] 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. In *Proc. of the International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020.

[3] CWA Team. Epidemiological Motivation of the Transmission Risk Level. *https://raw.githubusercontent.com/corona-warn-app/cwa-documentation/5f45237279d4b92c71e98d747958ff0f550ebce9/transmission_risk.pdf*, 10 2020. (visited September, 2021).

[4] DP-3T Project. Privacy and security attacks on digital proximity tracing systems. *https://github.com/DP-3T/documents/blob/master/Security%20analysis/Privacy%20and%20Security%20Attacks%20on%20Digital%20Proximity%20Tracing%20Systems.pdf*, 2020. (visited September, 2021).

[5] B. Etzlinger, B. Nußbaummüller, P. Peterseil, and K. A. Hummel. Distance estimation for ble-based contact tracing – a measurement study. In *2021 Wireless Days (WD)*, pages 1–5, 2021.

[6] Google Inc. and Apple Inc. COVID-19 Exposure Notifications. *https://www.google.com/covid19/exposurenotifications*, Apr. 2020. (visited September, 2021).

[7] Google Inc. and Apple Inc. COVID-19-Notification, Bluetooth Specification, Version 1.2. *https://blog.google/documents/70/Exposure_Notification_-_Bluetooth_Specification_v1.2.2.pdf*, Apr. 2020. (visited September, 2021).

[8] Google Inc. and Apple Inc. COVID-19-Notification, Cryptography Specification, Version 1.2. *https://blog.google/documents/69/Exposure_Notification_-_Cryptography_Specification_v1.2.1.pdf*, Apr. 2020. (visited September, 2021).

[9] Y. Gvili. Security analysis of the COVID-19 contact tracing specifications by Apple inc. and Google inc. Technical Report 428, IACR Cryptology ePrint Archive, 2020.

[10] Y.-C. Hu, A. Perrig, and D. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, volume 3, pages 1976–1986 vol.3, 2003.

[11] R. Koch-Institut. Corona-warn-app version 1.9 greift auf version 2 des exposure notification framework zurück, 2020. URL *https://www.coronawarn.app/de/blog/2020-12-16-corona-warn-app-version-1-9/*. (visited September, 2021).

[12] D. J. Leith and S. Farrell. Measurement-based evaluation of google/apple exposure notification api for proximity detection in a commuter bus. *PLOS ONE*, 16(4): 1–16, 04 2021.

[13] J. Li and X. Guo. Covid-19 contact-tracing apps: a survey on the global deployment and challenges. Technical report, Imperial College London, 2020.

[14] S. Meyer, T. Windisch, A. Perl, D. Dzibela, R. Marzilger, N. Witt, J. Benzler, G. Kirchner, T. Feigl, and C. Mutschler. Contact tracing with the exposure notification framework in the german corona-warn-app. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2021.

[15] N. Min-Allah, B. A. Alahmed, E. M. Albreek, L. S. Alghamdi, D. A. Alawad, A. S. Alharbi, N. Al-Akkas, D. Musleh, and S. Alrashed. A survey of covid-19 contact-tracing apps. *Computers in Biology and Medicine*, 137:104787, 2021. ISSN 0010-4825.

[16] Robert Koch-Institut. Privacy notice corona-warn-app. *https://www.coronawarn.app/assets/documents/cwa-privacy-notice-1.3-en.pdf*, 2020. (visited September, 2021).

[17] Robert Koch-Institut. Infektionsketten digital unterbrechen mit der corona-warn-app. *https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/WarnApp/Warn_App.html*, 9 2021. (visited September, 2021).

[18] B. SIG. Bluetooth core specification. *https://www.bluetooth.com/specifications/specs/core-specification-5-2/*, 2019. (visited September, 2021).

[19] R. Sun, W. Wang, M. Xue, G. Tyson, S. Camtepe, and D. C. Ranasinghe. An empirical assessment of global covid-19 contact tracing applications. In *In proceedings of the 43rd International Conference on Software Engineering (ICSE 2021)*, 2020.

[20] C. Troncoso, M. Payer, J.-P. Hubaux, M. Salathé, J. Larus, E. Bugnion, W. Lueks, T. Stadler, A. Pyrgelis, D. Antonioli, L. Barman, S. Chatel, K. Paterson, S. Čapkun, D. Basin, J. Beutel, D. Jackson, M. Roeschlin, P. Leu, B. Preneel, N. Smart, A. Abidin, S. Gürses, M. Veale, C. Cremers, M. Backes, N. O. Tippenhauer, R. Binns, C. Cattuto, A. Barrat, D. Fiore, M. Barbosa, R. Oliveira, and J. Pereira. Decentralized privacy-preserving proximity tracing. Technical Report abs/2005.12273, arXiv, 2020.

[21] S. Vaudenay. Analysis of dp3t. Cryptology ePrint Archive, Report 2020/399, 2020. *https://ia.cr/2020/399*.

[22] K. S. Win. Analysis of detecting wormhole attack in wireless networks. In *World Academy of Science, Engineering and Technology*, pages 422–428, 2008.