TraceBand: Privacy-Preserving Contact Tracing on Low-Power Wristbands

Patrick Rathje[1], Olaf Landsiedel[1, 2]
[1]Kiel University, Kiel, Germany
[2]Chalmers University of Technology, Gothenburg, Sweden
{pra,ol}@informatik.uni-kiel.de

Abstract

TraceBand is a low-power and versatile wristband for contact tracing during pandemic outbreaks. Running on a 64 MHz Cortex-M4 microprocessor with 256 KB of memory, TraceBand targets a low-cost design. Still, it seamlessly integrates into the smartphone-based tracing, extending the reach and, therefore, the effectiveness of the Exposure Notification protocol by Google and Apple. As the wristband is restricted to Bluetooth Low Energy for tracing and communication, companion devices and gateways serve as synchronization points and allow the device to offload the recorded contacts for an off-device risk analysis or download the information of infected contacts to perform the risk analysis on the device itself. The presented design ensures compatibility from the ground up and tackles occurring challenges due to resource reduction. Evaluations cover the applicability and energy consumption of the Exposure Notification protocol on resource-constrained devices and the on-device identification of risk contacts. The results show that the contact exchange suits battery-driven, resource-constrained devices, as TraceBand requires 2 mAh per day. At the expanse of battery life, up to 5,000,000 risk keys can be checked on the device daily, increasing consumption to roughly 80 mAh for a single day.

Categories and Subject Descriptors

J.3 [**Computer Applications**]: LIFE AND MEDICAL SCIENCES; K.4.2 [**Computing Milieux**]: COMPUTERS AND SOCIETY, Social Issues

General Terms

Design, Human Factors, Measurement

Keywords

Contact Tracing, Wearable, Low-Power

1 Introduction

Contact tracing, i.e., identifying persons who may have been in contact with an infected person, is an essential method for controlling infectious disease outbreaks [7, 14]. Manual contact tracing is, however, a time-consuming task. To automate and scale the process of contact tracing during the Covid-19 pandemic, Google and Apple unveiled the Exposure Notification protocol (GAEN) for Android and iOS devices and brought contact tracing to billions of smartphones [4]. Building on Bluetooth Low Energy (BLE), this GAEN protocol enables automated and privacy-preserving contact tracing, silently exchanging beacons without user intervention. The large-scale availability is essential for contact tracing, as the tracing gets more effective when its user base increases. In turn, an increased efficacy attracts additional users and further intensifies social incentives [11].

However, the focus on smartphones excludes significant and often vulnerable groups in our society from utilizing automated contact tracing: Some just cannot afford a modern smartphone required for contact tracing, elderly people often lack the knowledge to operate a smartphone, young children rarely have smartphones and in many jobs, such as in retail and public safety, the use of (personal) smartphones is prohibited. In this paper, we argue that there is a need to include low-cost devices, such as a simple wristband, in the contact tracing process. Such a simple device, compatible with the GAEN protocol, could offer a versatile and low-cost alternative to smartphones. Just like a smartwatch or a fitness tracker, it would be usable at work, affordable, and easy to use for both elderly and children.

Prioritizing an affordable, low-cost design, we reduce hardware expenses and build on constrained computing resources with sparse memory and storage (i.e., 256 KB memory and 8 MB storage). In contrast to the fully-fledged smartphone operating systems that the GAEN tracing framework builds on, the fundamentally reduced and different hardware architecture, without the possibility to build on Android or iOS, requires an adapted implementation. At the same time, we cut down communication to the bare necessities, i.e., in this case BLE. Hence, ensuring compatibility goes beyond correctly implementing the underlying cryptographic fundamentals and the tracing protocol itself; it requires proper integration with BLE as the sole communication protocol.

In this paper, we introduce TraceBand (TB) which brings contact tracing to cheap and resource-efficient BLE tags.

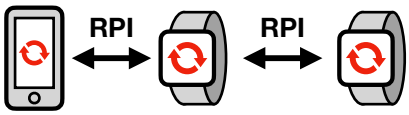


Figure 1. The low-cost TraceBand (TB) integrates seamlessly into the Exposure Notification protocol by Apple and Google, exchanging rolling proximity identifiers (RPIs) as pseudonyms with smartphones and other TB devices. The release of identifiers upon an infection allows others to identify risk contacts.

Compatible with Apple's and Google's Exposure Notification protocol, TraceBand devices offer a low-cost, versatile alternative to smartphones for GAEN contact tracing at the same level of privacy (see Figure 1).

Overall, this work contributes the following:

- We present TraceBand (TB), a versatile, low-cost Exposure Notification-compatible wristband running on an nRF52840, a 64 MHz Cortex-M4 microprocessor with 256 KB memory and BLE [1].
- TB supports risk analysis on companion devices as well as the identification of risk contacts on the device itself and was the subject of trials with 140 participants.
- We evaluate the energy consumption and the number of manageable contacts to check on-device. With an average consumption below 2 mA, TB could run approximately a hundred days off a coin battery. For on-device risk analysis, at the expanse of up to 80 mAh, TB can check around 5,000,000 contacts daily.

2 Background

For the necessary background, this section explains the role of contact tracing during pandemics, Bluetooth Low Energy as a communication protocol for contact tracing, and the GAEN protocol.

2.1 Contact Tracing

In an infectious disease outbreak, vaccines, medication, or even testing possibilities are unlikely to be available right from the start: keeping a dangerous outbreak under control is thus the top priority. While measures like social distancing or wearing masks can reduce a direct transmission, they cannot entirely prevent it. However, authorities can notify recent contacts before they get infectious due to the incubation time. Thus, a quick notification allows the preventive isolation of exposed contacts. As such, tracing contacts can help to break infection chains and increase control over the outbreak [7, 14]. Factors such as proximity or the contact duration determine the exposure level and the associated risk of transmission.

2.2 Bluetooth Low Energy

Bluetooth Low Energy (BLE) is a wireless communication protocol with a particular focus on low costs and low power. Today, billions of devices in the Internet of Things (IoT) are equipped with BLE radios, ranging from smart home devices, headphones, and fitness trackers to smartphones and personal computers [3]. Within the BLE protocol, devices can announce their presence by periodically broadcasting advertisements. The advertisements support additional custom data fields and can be received by all nearby devices. On reception, devices determine the received signal strength indicator (RSSI). If the transmission power is known, the difference between the transmission and reception power approximates the path loss of the transmitted signal, which in turn allows an estimate of the distance between the devices[6]. BLE-based contact tracing builds on this distance metric for the risk analysis.

2.3 Exposure Notification

At the beginning of the Covid-19 pandemic, contact tracing was one of the available first response measures. As a result, different tracing options exist, each balancing utility and privacy [1]. In the Exposure Notification protocol by Apple and Google (GAEN) [2], devices use BLE advertisements to broadcast rolling proximity identifiers (RPI) frequently. The RPIs are temporary pseudonyms and rely on cryptographic primitives to preserve users' privacy: At the start of the day, each device randomly generates a new, secret temporary exposure key (TEK). Devices then derive the RPIs based on a TEK for 10-minute intervals, i.e., 144 RPIs per day. The decentralized protocol runs in the background and does not require user interactions: each device stores received RPIs locally. In addition to the RPI, the broadcasted packet contains associated encrypted metadata (AEM). This metadata includes the actual transmission power and is only readable with access to the corresponding TEK. If users get infected, they can upload the relevant daily keys (e.g. last 14 days), which the health authorities publish. All other users download the TEKs and derive the individual RPIs to check for possible exposure. The devices estimate the exposure's intensity based on the duration and distance. Though the low precision of BLE distance estimation results in a less selective quarantine, the overall Exposure Notification protocol supports the containment [7].

3 Design

Bringing the GAEN protocol to cheap and versatile devices makes contact tracing accessible to more users and thus, increases the overall effectiveness of contact tracing. However, with its focus on low costs, the design faces the challenge of restricted hardware with no direct support for the Google and Apple Exposure Notification protocol (GAEN). The primary design challenges are (1) GAEN compatibility, (2) storage and energy requirements, and (3) the subsequent risk and contact analysis.

GAEN Compatibility: To ensure interoperability with smartphones employing the Exposure Notification protocol, TraceBand needs to implement it properly. However, with hundreds of KB of memory and less than 10 MB of storage, the target devices cannot run even a reduced Android or iOS. Hence, we devise a system design and implementation of GAEN for resource-efficient IoT devices with the integration of BLE and cryptographic primitives.

Storage and Energy Requirements: A person comes into BLE communication range of potentially hundreds of devices each day. The respective RPIs of these contacts need to be stored on the limited storage space of an IoT de-

[1] available as open-source at https://github.com/ds-kiel/TraceBand

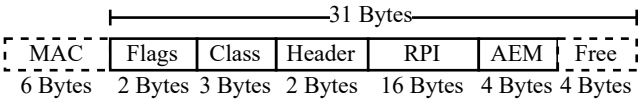


Figure 2. The BLE advertisements within the GAEN protocol contain the rolling proximity identifier (RPI) and the associated encrypted metadata (AEM).

vice for contact tracing. This quickly accumulates to several MBs of storage which exceeds the memory capacity of the constrained devices. Utilizing flash memory, we show how traces can be stored. Further, as we target mobile devices, energy consumption determines the battery life, a vital usability factor. Here, optimizing power consumption helps to reach a lifetime of several days on a single battery charge.

Risk and Contact Analysis: For risk and contact analysis, our device needs to download the TEKs of infected users, i.e., their daily keys, via the Internet, compute all 144 temporary identifiers of a person and match these with its locally stored contacts. This brings two challenges, (1) access to the Internet and (2) the computational complexity of this matching task. Direct Internet access via cellular or Wi-Fi is not an option in our design space, as this would greatly increase both energy consumption and the costs of such a device. Instead, we opt for a gateway approach, where other BLE-enabled devices such as personal computers and smartphones function as gateways. In our design and evaluation, we focus on two options where the contact matching is either executed (a) on-device, enabling full-autonomy of TB, or (b) off-device, e.g., on a trusted device. To address the challenge of on-device computational complexity, we devise a bloom filter-based approach to identify contacts.

After addressing the main challenges and presenting our approach, we dive into the design components, elaborate on the soft- and hardware implementation and close this section with details about the trials.

3.1 Advertising

For each interval, a device derives a new RPI as a temporary identifier from the interval number and the temporary exposure key (TEK). It broadcasts this temporary identifier as a BLE advertisement every 250 ms. While Android generates a new RPI after 650 s and 600 s, iOS and TB randomize the generation interval between 500 s and 1250 s. This randomization hinders device tracking across intervals. Matching the requirements of the GAEN protocol, advertisements contain the mandatory flags, a 16-bit service class identifier, and finally, 20 Bytes of protocol payload data. The payload data consists of the 16 Byte rolling proximity identifier (RPI) and 4 Bytes of associated encrypted metadata (AEM), see Figure 2. As the GAEN protocol is part of the firmware in Android and iOS and not all parameters are open, we exemplarily recorded the GAEN advertisements from devices and compared them to our implementation. Table 1 displays the results of this compatibility test.

3.2 Scanning

For the reception of advertisements, TB scans for two consecutive seconds every 5 minutes and stores temporary identifiers for the subsequent risk analysis. However, flash memory is not only limited in size but further wears down as pages are cleared.

Table 1. Comparison of Advertising Behavior to match the Exposure Notification requirements. TB's communication is restricted to BLE and does not support classical Bluetooth (BR/EDR) like Android and iOS.

Property	TB	iOS	Android
Broadcast Interval	250 ms	220 ms	250 ms
Adv. per Interval	1	3	1
RPI Interval	500-1250 s	500-1250 s	650 s
Sim. LE and BR/EDR	false	true	true
BR/EDR not supported	true	false	false

Table 2. TB receives around 12 records per neighbor each hour. Based on estimates by the Bluetooth SIG, TB collects roughly 2304 records a day.

Activity	Devices	Records per Hour	Duration Estimate	Expected Records
Home	4	48	12	576
Work	12	144	8	1152
Commute	14	168	2	336
Social	10	120	2	240

TB saves recorded advertisements in a circular buffer, resetting the oldest memory page before reusing it. This record buffer supports concurrent and thread-safe access by timestamp and sequence number, persistence, and deletion. Each advertisement is saved and stays identifiable by a 24-bit sequence number. In addition to the sequence number, the received RPI (16 Bytes) and AEM (4 Bytes), TB saves the reception timestamp (4 Bytes) and the RSSI value (1 Byte). Another Byte contains a 7-bit checksum and a deletion bit. If entries need to be deleted, TB sets the entire entry to 0 (or 1 depending on the type of the flash), removing the information and toggling the deletion flag simultaneously.

In total, each record requires 32 Bytes of storage. Hence, a 2 MB storage can hold up to 65,536 entries. With 2304 expected records per day (see Table 2), the storage has just enough capacity for the estimated 55,296 records for 14 days.

3.3 Synchronization

As TraceBand devices do not have direct Internet access, they access the lists of keys of infected users released by the relevant health authorities via companion devices and gateways. While companion devices are particularly useful for personal use, e.g., a smartphone of a parent as a companion for the wristbands of the kids in a family, gateways enable contact tracing especially for institutions and events: Placed at strategic points like an entrance, they provide BLE access points to central servers and thus provide synchronization and over-the-air updates without the need for additional devices.

3.4 Off-Device Risk Analysis

As space and computation are limited on resource-constrained devices, TB can offload the risk analysis to trusted companion devices and gateways (see Figure 3). A TraceBand device uploads its received identifiers, i.e., contacts, of the last days. Based on this contact history and infected keys released by health authorities, the companion

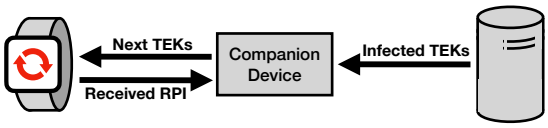


Figure 3. A trusted companion device allows easy administration as it synchronizes records and analyzes the risk on behalf of the wristband. Alternatively, a BLE gateway allows the synchronization with remote servers, enabling low-cost contact tracing for, e.g., institutions.

Figure 4. For full autonomy and increased privacy, TB downloads infected keys using BLE gateways and identifies risk contacts on the device. Digitally signed by the health authorities, the data can be distributed over untrusted devices, possibly even other wristbands. TB notifies the user about possible risk contacts.

device analyses the risk of exposure on behalf of the device. For communication, Bluetooth SIG drafted a Wearable Exposure Notification System (WENS), which enables wearables to share tracing data with devices like smartphones using BLE [4]. TB communication builds on this ENS protocol. In addition, companion devices and gateways serve as reliable sources for time and software updates.

3.5 On-Device Risk Analysis

GAEN-capable smartphones compare the keys released by the relevant health authorities with their contact history. To enable an on-device risk analysis, TB supports identifying risk contacts on the device, see Figure 4. This fosters full autonomy and further increases the level of privacy compared to the off-device risk analysis.

Like the off-device risk analysis, TB relies on gateways to access the infected TEKs published by the health authorities. However, with potentially thousands of entries released each hour, TB needs an efficient matching mechanism. To support the vast number of TEKs, TB builds on Bloom filters and stores all recorded RPIs into a bloom filter. Then, for each received TEK to check, TB derives the corresponding RPI from the TEK and its starting interval. The bloom filter allows an efficient but probabilistic check if a test RPI could be present in the storage, speeding up the identification and reducing the time and energy of costly flash memory access. While entries in bloom filters are usually hashed with different hash functions to set respectively test individual bits in the Bloom filter vector, we reuse the inherently random structure of RPIs, split its 16 Bytes, and use these as individual hash values, saving precious computation time and power. For example, for the typical 14 days, we expect to have less than 65,536 records kept in 2 MB of flash storage. We split the 16 Bytes into 4 hash values with 4 Bytes each. A 64 KB bloom filter, for example, then allows a false-positive rate of just 2.6%, i.e., out of 39 matches, just one matched contact was not met. If the bloom filter does not fit into memory, we can reuse the validity period of records ($\pm$2 hours) and create multiple bloom filters for specific periods, e.g., each day.

Figure 5. The TB prototype used in trials features a low-power nRF52840 SoC with additional 2 MB storage. With costs below 10€, it enables low-cost participation in the contact-tracing process. Designed as a wristband, the circular core can also be used as a key fob or a necklace. Picture by Benjamin Walczak.

3.6 Software Implementation

On the software side, we build on the Zephyr real-time operating system (Zephyr RTOS) for resource-constrained IoT devices and its certified BLE stack. For the cryptographic primitives, we build on mbedTLS, a small and portable SSL library. We tested our cryptographic subroutines, i.e, the random creation of daily keys, deriving the temporary identifiers for the intervals, and metadata encryption/decryption against available test vectors for the GAEN protocol. For the companion device, we provide an exemplary Android App. Programmed in Java, this open-source application enables an easy connection to TB and allows seamless RPI extraction based on the WENS draft by the Bluetooth SIG [4]. In addition, we use the same API connector in Python scripts, allowing the deployment on e.g., a Raspberry Pi as a gateway. TraceBand is available as open-source[2].

3.7 Hardware Implementation

For our prototype, we rely on the widely used nRF52 series. Equipped with Bluetooth 5.2, we choose the nRF52840 SoC, offering a 64 MHz Cortex-M4 microprocessor with 256 KB memory and 1 MB of internal flash storage. We package this SoC with 2 MB of external SPI storage, a status LED, 80 mAh battery, and a USB port for charging on a custom board and enclose everything in a simplistic case (see Figure 5). Even for the prototype, the overall costs per device amount to less than 10€.

With its versatile software implementation, TraceBand is not limited to the low-cost prototype but fosters integration into existing devices and firmware. For one, the TraceBand software supports the PineTime smartwatch offering an open-source hardware design with an nRF52832 chip and a square 1.3-inch IPS capacitive touch display.

[2]https://github.com/ds-kiel/TraceBand

3.8 Trial Deployment

Our prototype design has been deployed in trials during the Covid-19 pandemic: Over six months in the summer of 2021, 140 participants used TB for contact tracing in Kiel, Germany. Central servers handled the corresponding risk analysis with Raspberry-Pis distributed throughout the city as BLE-gateways. In an accompanying survey, 45 out of 48 people were convinced by the general concept. Short battery life and the need to visit synchronization points were identified as major barriers to user acceptance. Consequently, we further optimized energy consumption and investigated on-device risk identification.

4 Evaluation

Because we ensure compatibility with our design, the evaluation covers energy consumption and battery life as a critical factor for user acceptance and, therefore, the adaption of TB. First, this section analyzes the fundamental energy consumption running the GAEN background exchange, i.e., generation, advertising, and scanning of RPIs. Then, we evaluate the on-device risk analysis scenario and determine the time and energy requirements regarding battery life, in particular.

For evaluation, we deploy the Zephyr-based TB software on an nRF52840-DK [9]. The nRF52840-DK is equipped with the same SoC as our prototype and features a 64 MHz Cortex-M4 microprocessor with 256 KByte memory and BLE. In addition, 8 MB of external flash memory are available, of which we use 2 MB for record storage. We place the Bloom filter within 64 KB of memory, but the filter could be split and stored on flash for smaller devices. While the whole nRF52-family includes AES hardware acceleration, the nRF52840 specifically supports the ARM TrustZone CryptoCell 310 with hardware acceleration for keyed hash functions. Both the derivation of the identifier and the metadata encryption key would benefit from this additional acceleration. As this optimization is limited and has only a minor impact compared to the derivation of all temporary identifiers, we limit the evaluation to hardware accelerated AES. We measure currents and timings of the nRF52840-DK using a Power Profiling Kit I by Nordic.

4.1 Energy Usage of Contact Tracing

Running continuously in the background, the Exposure Notification advertising and scanning requires periodic actions from the processor, which idles otherwise. The first experiment evaluates the overall energy consumption per state. The device derives the daily cryptographic primitives for itself and runs the basic GAEN protocol. We determine the average value based on at least ten measurements and extrapolate the expected consumption.

As displayed in Figure 6, the idle state draws the least current with just 0.0026 mA. The average values result from 0.04 seconds for advertisements, 2.081 seconds for scanning, and 0.22 seconds for the cryptographic primitives of the full day. Table 3 displays the detailed times. Because the processor spends around 98% idling, the overall consumption is reduced to 1.967 mAh per day. The device derives its keys easily and without impacting the overall consumption.

Table 3. Timings of the Exposure Notification primitives.

Interval	Function	Time [ms]
Daily	Generate Random Secret	0.300
Daily	Derive Identifier Key	0.288
Daily	Derive Metadata Encryption Key	0.288
Each Period	Derive Temporary Identifier	0.062
Each Period	Encrypt Metadata	0.066
Check Key	Derive All Temporary Identifiers	8.912

4.2 Enabling On-Device Risk Identification

The basic GAEN protocol leaves the microprocessor idling for most of the time. During this time, the device could identify risk contacts directly on the device: a fundamental step for independent risk analysis. We now evaluate this on-device scenario in terms of its expected energy consumption and overall capacity, i.e., how many TEKs can be checked per day. We assume a filled record storage of 2 MB, so 65,535 records in total are already present in the Bloom filter. Further, we suppose that the TEKs and their respective timestamps (32 Bytes per TEK) are transmitted over BLE with 800 kbps and a current of 8 mA [5]. Varying the number of TEKs to check daily, we simulate a varying number of new infections.

With an average of 15 ms to derive and check all RPIs for a single TEK, the device can potentially test over 200,000 TEKs in one hour using the Bloom filter. For comparison, the corresponding transmission of the necessary TEK data (6.4 MB) takes about a minute. Figure 7 presents the combined influence on energy consumption. Around 5,000,000 could be transported and checked during idle times each day, but the battery life would be reduced from e.g., 40 days to just a single day with an 80 mAh battery. The 300,000 new Covid cases per day in Germany during March 2022 would therefore challenge the computation and battery lifetime as this corresponds to 4,200,000 TEKs each day.

5 Related Work

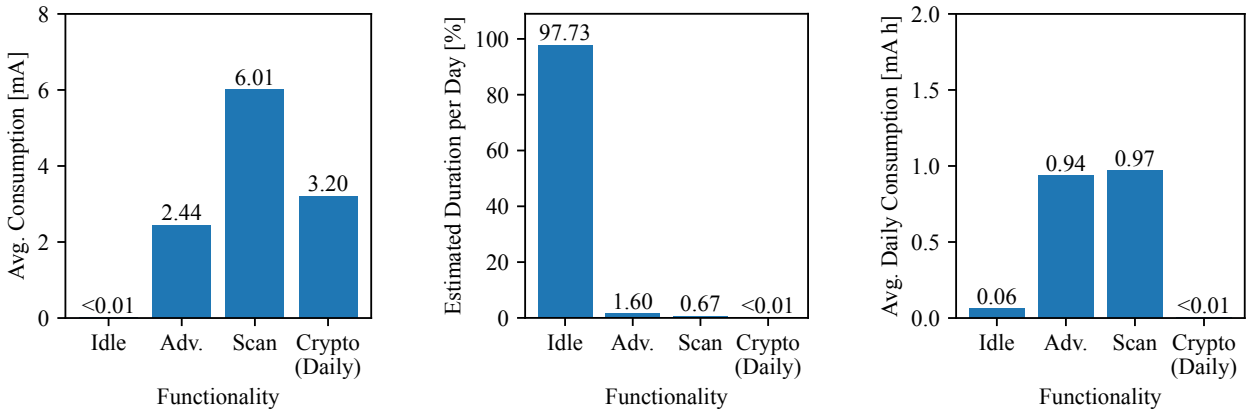
Beyond tracing on smartphones, a recent work analyzes the perspective of wearables and identifies subsequent challenges [10]. With the draft of the Wearable Exposure Notification Service, the Bluetooth SIG specifies a communication protocol that aims to deeply integrate wearables into the contact tracing process [4]. Individual solutions are further developed:

The EasyBand [12] is a proposal for a BLE-based contact tracing wearable. Close contacts to other EasyBands should be recorded and sent to a central server, marking devices at risk. As a proposal, no further implementation details or an evaluation of the system are given.

Another work focuses on a low-power solution [13] and uses battery-powered circuit boards equipped with 80 MHz ESP8266 microcontrollers that exchange UUIDs. A centralized tracing system notifies devices at risk when the boards connect over USB. No evaluation is given.

The P^3CT [8] approach leverages smartwatches (1 GB memory, 8 GB of storage) for a custom tracing protocol. The protocol is experimentally validated.

In contrast, this work presents and evaluates the design of a low-cost wristband with full compatibility with the widely available GAEN protocol.



(a) Average Current per Functionality **(b) Estimated Durations per Day** **(c) Extrapolated Daily Consumption**

Figure 6. Current Measurements, Usage Times, and Consumption for the basic GAEN functionality: While the basic consumption is high for the individual functionalities (a), their respective execution times are minor (b). Overall, this aggregates to an average consumption of 2 mAh each day (c), including the generation of TEKs and derivation of RPIs.

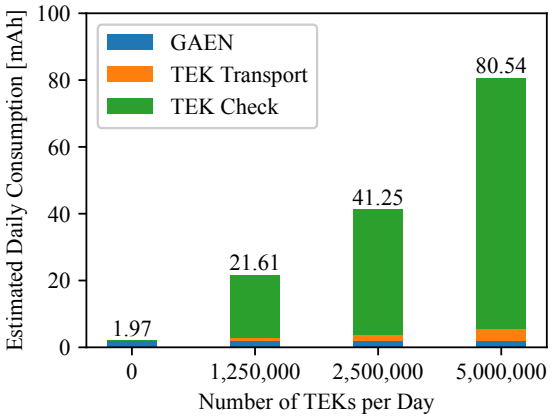


Figure 7. On-Device Matching of Risk Contacts: Identifying risk requires more resources the more cases need to be checked. A device can check around 5,000,000 TEKs each day at the cost of battery lifetime.

6 Conclusion

This work presents TraceBand as a low-cost extension of the Exposure Notification protocol, increasing its reach and effectiveness. Neglected participants such as the elderly, children, or employees who may not want or are not allowed to carry a smartphone could benefit from the wristband's versatility and low costs. The evaluation indicates that the GAEN protocol is well suited for battery-driven, resource-constrained devices, using only 2 mAh per day on the nRF52840 SoC. As a vital step in decentralized contact tracing, we show on-device identification of risk contacts with up to 5,000,000 infected keys per day and evaluate the corresponding energy consumption of up to 80 mAh for a single day.

Acknowledgements

This work is supported by the Federal Ministry of Health based on a resolution by the German Bundestag.

7 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] Apple Inc. Privacy-preserving contact tracing. https://covid19.apple.com/contacttracing, 2020 (accessed Nov 28, 2020).

[3] S. Bluetooth. Bluetooth market update, 2021.

[4] Bluetooth SIG, Inc. Ens wearables. https://www.bluetooth.com/learn-about-bluetooth/bluetooth-technology/bluetooth-ens/, 2020 (accessed Dec 03, 2020).

[5] P. Bulić, G. Kojek, and A. Biasizzo. Data transmission efficiency in bluetooth low energy versions. *Sensors*, 19(17):3746, 2019.

[6] 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. IEEE, 2021.

[7] E. Hernández-Orallo, P. Manzoni, C. T. Calafate, and J.-C. Cano. Evaluating how smartphone contact tracing technology can reduce the spread of infectious diseases: The case of covid-19. *Ieee Access*, 8:99083–99097, 2020.

[8] P. C. Ng, P. Spachos, S. Gregori, and K. Plataniotis. Epidemic exposure notification with smartwatch: A proximity-based privacy-preserving approach. *arXiv preprint arXiv:2007.04399*, 2020.

[9] Nordic Semiconductor. nrf52840 dk. https://www.nordicsemi.com/Software-and-Tools/Development-Kits/nRF52840-DK, 2020 (accessed Dec 03, 2020).

[10] V. Shubina, A. Ometov, and E. S. Lohan. Technical perspectives of contact-tracing applications on wearables for covid-19 control. In *2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 229–235. IEEE, 2020.

[11] S. Trang, M. Trenz, W. H. Weiger, M. Tarafdar, and C. M. Cheung. One app to trace them all? examining app specifications for mass acceptance of contact-tracing apps. *European Journal of Information Systems*, 29(4):415–428, 2020.

[12] A. K. Tripathy, A. G. Mohapatra, S. P. Mohanty, E. Kougianos, A. M. Joshi, and G. Das. Easyband: A wearable for safety-aware mobility during pandemic outbreak. *IEEE Consumer Electronics Magazine*, 9(5):57–61, 2020.

[13] Y. Verbelen, S. Kaluvan, U. Haller, M. Boardman, and T. B. Scott. Design and implementation of a social distancing and contact tracing wearable. In *2020 6th IEEE Congress on Information Science and Technology (CiSt)*, pages 466–471. IEEE, 2021.

[14] World Health Organization. Contact tracing in the context of covid-19. https://www.who.int/publications/i/item/contact-tracing-in-the-context-of-covid-19, 2020 (accessed Nov 28, 2020).