Pwning Samsung:
An Attempt at Hardware and Bluetooth Hacking

Ali Chowdhurry
alichy@umich.edu
University of Michigan - Dearborn

Andrew Schwartz
csandrew@umich.edu
University of Michigan - Dearborn

Jose Vasquez
josev@umich.edu
University of Michigan - Dearborn

Abstract

This report documents our exploratory efforts to hardware hack the Samsung Galaxy SmartTag 2, released on October 11, 2023. As one of the first attempts to investigate potential vulnerabilities in this device, our work contributes foundational observations to a largely unexplored area. Despite its recent release, we found no prior publications or documented discoveries regarding security weaknesses in this tag.

Our primary focus included both software-based and hardware-based attacks. On the software side, we employed Bluetooth sniffing techniques to intercept communications. However, these attempts did not yield actionable insights or vulnerabilities. On the hardware side, we probed various test pads on the device's PCB[1]. Although we successfully captured the signal readings, the data remain unparsed and its significance is currently undetermined.

This study highlights the challenges of reverse-engineering modern IoT devices, particularly those with limited public research. Although we did not uncover specific exploits or vulnerabilities, our findings underscore the complexity of analyzing proprietary technologies like the Samsung SmartTag 2. Future work will involve deeper investigation into the captured signals, potential UART[2] or SPI[3] access, and further refinement of our Bluetooth sniffing methodology. This report serves as a starting point for researchers interested in expanding the security analysis of this IoT[4] device.

1 Introduction

How secure is your tracking device? *"A smart way to keep track of important things in your life,"* [1]—a promise made by Samsung and many other manufacturers of devices designed to safeguard your valuable possessions. In this research, we investigate the security of the Samsung Galaxy SmartTag 2 to determine if it shares vulnerabilities commonly found in other IoT devices, such as Apple's AirTag. As the latest addition to the ecosystem of smart tracking devices, the SmartTag 2 builds upon its predecessor with enhanced tracking capabilities, positioning itself as a significant competitor to Apple's AirTag and its ecosystem. Like other smart trackers, the SmartTag 2 relies on Bluetooth Low Energy (BLE) and Ultra-wideband (UWB) technology for communication and location tracking through a network of nearby connected devices.

Given the wide adoption of IoT devices, security and privacy concerns have become increasingly critical. Due to the recent release of the SmartTag 2, little to no research has been documented, leaving it largely unexplored and potentially vulnerable. In contrast, devices like Apple's AirTag have undergone extensive analysis. This lack of investigation presents an opportunity to examine the SmartTag 2's design and assess its resilience against both hardware and software-based attacks.

The primary focus of this research is to explore the SmartTag 2 for potential security weaknesses. On the software side, we employed Bluetooth sniffing techniques to intercept and analyze communications. On the hardware side, we probed the printed circuit board (PCB) to identify accessible test pads and captured signal data for further examination.

This paper contributes to the foundational analysis of the largely unexplored area of SmartTag 2 security and proposes methodologies for future research. The main contributions of our work include:

- Capturing signal data from the PCB and identifying potential access points for further analysis.
- Highlighting the challenges of probing a device with limited public documentation.
- Demonstrating Bluetooth sniffing techniques on the SmartTag 2 and evaluating their effectiveness.

SmartTags remain very untested still as our research was limited by time and expertise. We believe there to be multi-

[1] Printed Circuit Board
[2] Universal Asynchronous Receiver-Transmitter
[3] Serial Peripheral Interface
[4] Internet of Things

ple attack vectors but were unable to successful exploit one. Hopefully, our work here can form a basis for further research on the device as the consequences of flawed security are immense with a location tracking centric device like the SmartTag 2 [2]. Our research will all be made available to use by others to continue the effort [5].

The rest of this paper is structured as follows. We explain the hardware components of the tag and show how the Samsung Find My Mobile (FMM) [3] works in section 2. Using this gathered information we will create a threat model for the tags in section 3. Section 4 will talk about the possible hardware attacks and we new information we gathered. In section 5 will be about the Bluetooth aspect of the tags and explain possible attacks there. Finally, we will discuss related and future work in section 6 and our conclusion in section 7.

2 Background

Hardware Components

Ultra-Wideband (UWB) Chip

The Samsung SmartTag 2 employs the SR040 UWB transceiver, which provides precise ranging capabilities through time-of-flight (ToF) and time-difference-of-arrival (TDoA) measurements. UWB operates on the IEEE 802.15.4z High Rate Pulse (HRP) Physical Layer (PHY) standard which is used for high data rate wireless communication and precision ranging applications [4]. Therefore, it utilizes frequencies from 6 to 8.5 GHz, making it well suited for high-accuracy location tracking in IoT environments.

Key features of the UWB chip include the following:

- **Precision Ranging:** Achieves an accuracy of ±10 cm even in non-line-of-sight conditions.
- **Low Power Operation:** Optimized for IoT applications powered by coin cell batteries, with current-limiting mechanisms to extend battery life.
- **Security Features:** Incorporates Scrambled Timestamp Sequences (STS) for secure data exchange, compliant with NIST SP 800-90A [5].
- **Integrated Processor:** Built on an ARM Cortex-M33 platform with TrustZone support, enabling secure firmware updates and data handling.

The combination of high precision, low power consumption, and secure communication of the UWB chip ensures reliable and efficient operation for advanced asset tracking in SmartTag 2. [6]

Bluetooth Low Energy (BLE) Chip

[5] https://github.com/csandrew-dev/smarttag-2

The SmartTag 2 also integrates the ATM33 BLE SoC, a Bluetooth 5.3 compliant system-on-chip designed for extreme low-power applications. BLE facilitates seamless communication and device connectivity over short ranges, complementing the UWB chip's high-accuracy features with wide compatibility and lower energy requirements [7].

Key features of the BLE chip include:

- **Power Efficiency:** Consumes just 0.85 mA in receive mode and 2.5 mA in transmit mode at 0 dBm output, extending the device's operational life [7].
- **Advanced Capabilities:** Supports features like Angle of Arrival (AoA) and Angle of Departure (AoD) for direction finding, enabling enhanced location tracking [7].
- **Security and Processing:** Includes ARM TrustZone, AES-256 encryption, and hardware cryptographic accelerators for secure communication and low-latency operations [7].

The BLE chip serves as the primary communication interface for the SmartTag 2, offering broad device connectivity while maintaining efficient power use.

UWB and BLE

The combination of UWB and BLE in the SmartTag 2 enhances its functionality by leveraging the strengths of both technologies. While UWB delivers high-precision spatial data for precious locations, BLE ensures device compatibility and efficient operation in everyday use. This dual-chip approach allows the SmartTag 2 to provide robust tracking and secure communication across a range of scenarios.

Samsung Find My Mobile

Samsung's Find My Mobile network utilizes a crowd-sourced Bluetooth Low Energy (BLE) tracking system to locate devices like SmartTags, even without direct internet access. When a SmartTag or other registered device is lost, it enters a "lost mode," broadcasting a unique, rotating identifier called a "privacy ID" via BLE. Nearby helper devices, such as Samsung phones or tablets, detect these signals and relay the detected privacy ID, along with their current location determined by GPS or WiFi, to a central server managed by Samsung. The owner of the lost device can then use the Samsung Find My Mobile application to query the server and retrieve the last known location of the lost item. This system relies on cryptographic mechanisms to protect privacy, such as regularly rotating the privacy ID and limiting the information accessible to helper devices. However, vulnerability in the implementation can pose risks, such as potential misuse of location data or manipulation. Figure 1 is from another research paper "Privacy Analysis of Samsung's Crowd-Sourced Bluetooth Location Tracking System" [8]:

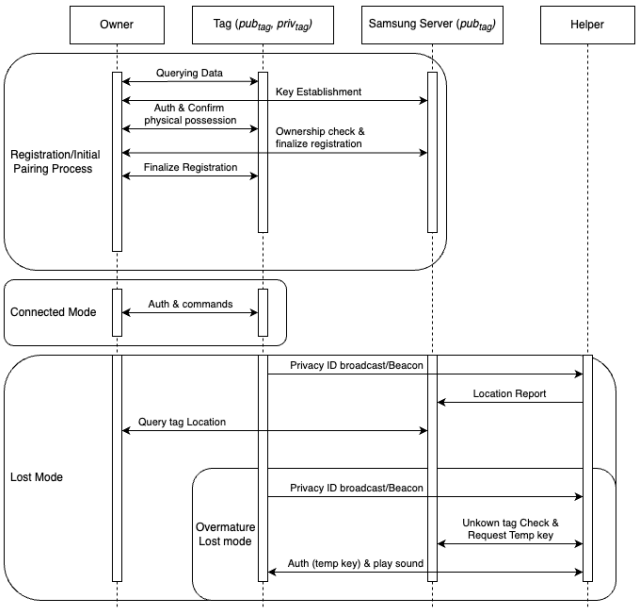

Figure 1: An overview of Samsung OF protocol for SmartTags [8]

3 Threat Model

Samsung has not publicly released a detailed threat model for the SmartTag 2. However, based on the hardware, firmware design, and its integration into Samsung's ecosystem, we can deduce several security assumptions. While the SmartTag 2 boasts various security features, some vulnerabilities and threats may not have been fully accounted for, such as stalking or advanced hardware attacks, as these require ongoing mitigations. In this section, we outline attacker capabilities (Section 3-A) and potential threats (Section 3-B). Additionally, we discuss existing attacks on SmartTag 2 and related systems (Section 3-C).

A. Attacker Capabilities

We consider two types of attackers in the context of Samsung SmartTag 2:

Physical Access Attacker: An attacker with physical access to the SmartTag can attempt to modify the hardware by soldering wires to its PCB, detaching or replacing components, or removing the SmartTag entirely. These actions may damage the device or leave traces but can enable firmware extraction or identity modification. Physical access attacks are often considered outside the scope for consumer-grade trackers but remain a realistic threat vector.

Over-the-Air Attacker: An attacker within proximity can interact with the SmartTag 2 through Bluetooth Low Energy (BLE) or Ultra-Wideband (UWB). These attackers are limited to exploiting over-the-air protocols but could potentially bypass protections in the SmartTag's communication stack, such as beacon rotation or location sharing.

Our focus is on threats unique to Samsung SmartTag 2 and its integration with Samsung's SmartThings ecosystem. The role of the broader Samsung Find network in enabling tracking and its potential for misuse are also considered.

B. Threats and Mitigations

Stalking: The SmartThings protocol ensures that only the legitimate owner of the SmartTag can access its location information through their Samsung account. However, an attacker could misuse the device by covertly attaching it to a victim's belongings. Samsung has implemented privacy features to combat stalking, such as alerting users of unknown SmartTags traveling with them and playing sounds after a certain period of disconnection from the owner. However, these mitigations can be bypassed. For instance:

Firmware updates have improved anti-stalking features, including randomizing alert times and increasing sound volume. Tampering: Disabling the speaker makes detection harder, and modified SmartTags with disconnected speakers could be misused by stalkers or thieves.

Stolen Hardware: When an attacker gains physical access, they can remove the battery and steal the item associated with the SmartTag. However, Samsung links SmartTags to SmartThings accounts, making it impossible to pair a stolen SmartTag to a new account without removing it from the original owner's account. This binding mechanism reduces the incentive to steal the device itself but does not mitigate the theft of the tagged item.

Firmware Readout and Modification: Samsung employs basic protections against firmware readout and over-the-air modification:

Readout: While firmware updates are encrypted, hardware access could potentially enable attackers to dump firmware from the internal flash. The absence of advanced protections, such as secure boot or encrypted firmware storage, could expose vulnerabilities. Modification: SmartTag firmware updates are digitally signed and verified. Over-the-air attacks are unlikely without the corresponding private keys. Additionally, secure boot mechanisms prevent modified firmware from running.

Identity Modification: Each SmartTag has a unique identifier tied to the owner's Samsung account. Identity tampering, such as altering this identifier, could enable bypassing account locks or concealing malicious use. Secure mechanisms in the SmartTag firmware aim to protect this identifier, but physical access might still enable modification if the attacker bypasses protections.

Outdated Firmware: Regular firmware updates are critical for addressing vulnerabilities and improving privacy features. SmartTags verbosely update in the SmartThings app when

connected to a Samsung device. If a SmartTag is disconnected for extended periods or the User always selections "Later" for the update, it might miss crucial updates, leaving it vulnerable.

C. Known Attacks

There are no known attacks for this generation SmartTag, only the first generation. The SmartTag 2s have different chips so the exact same attacks my not apply but the basic concepts might be transferable to this domain.

Find My Network Exploitation: Reverse engineering of the SmartThings Find protocol could enable attackers to create rogue SmartTags or track legitimate ones. Custom implementations might mimic legitimate devices while bypassing privacy protections. Additionally, the dense SmartThings network could be exploited for unintended purposes. As seen in the the previous generation tag research. [8]

Device Tampering for Silent Tracking: Modified SmartTags with disconnected speakers bypass detection mechanisms relying on sound alerts. While Samsung discourages the use of SmartTags for locating stolen items, disabling the speaker makes the device useful for such purposes.

Voltage Glitching Probing and applying properly timed voltages to a microchip on the PCB of the SmartTag allowed an APPProtect bypass on the first generation of the SmartTags. This attack might be possible on the SmartTags 2 but the microchips and design of the tag has changed so the attack is only speculative. [9]

Ghost Peak Tricking Ultra-Wideband devices into thinking something is closer than it actually is. This attack is possible on the NXP SR040 according to Secure Positioning 2022 paper, "Ghost Peak: Practical Distance Reduction Attacks Against HRP UWB Ranging" [10]. We were however unable to test this attack with the proper equipment and knownledge. We recommend checking out that paper and the accompanying presentation/video.

4 Hardware Analysis

An initial hardware analysis gave us a basic understanding of the smart tag and lead us to testing specific areas of the board in the hopes of a basic hardware security failure. The two microchips identified on the board are the Atmosic ATM3325 and the NXP SR040. We were able to get the data sheets and other technical documents for the chips [7] [5]. Figure 2 shows us what the PCB looks like:

Open Test Pads

The NXP SR040 has four exposed pads that correspond to the four SPI pins. At first glance we thought this might be to program the SmartTags the NXP chip is not the main CPU on the board, so we concluded that the SPI pads are for interfacing between the two chips or were used for debugging

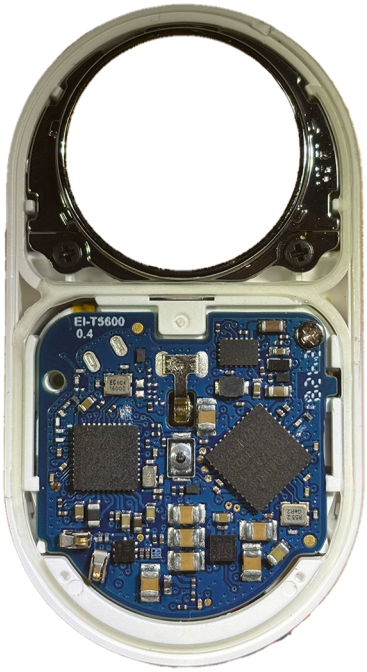

Figure 2: An initial look at the SmartTag's PCB.

when implementing Ultra-Wideband as it was not present in the previous generation. See Figure 3:

The higher up pins in Figure 3 did not produce anything when probed, the datasheet labeled these pins as "GPIO INT_N - Interrupt output (active low) in 6-wire SPI operation and GPIO RDY_N - Ready output (active low) in 6-wire SPI operation". The other four pins were SPI (SCLK, SDI, SDO, and CS). Unfornately these PINS did not produce anything of value upon probing and testing them. Our testing environment consisted of a BitMagic Logic Analyzer and a Macbook Pro running Pusleview. Over multiple tests nothing of value was extracted from the SmartTag's boot and pairing process. We deemed these pins disabled or Samsung is using a "non-standard" version of the SPI protocol. There are SWDCLK[6] and SWDIO[7] on the NXP chip but they have no traces on the PCB so we were unable to probe these pins. Our prediction is these are disabled as well.

Scratched pads

The Atmosic ATM3325 chip that is on the left side of the PCB has no exposed test pads like the NXP chip but the datasheet lists a couple possible pins for UART. We were able to scratch off the PCB solder mask and probe the pins. Please see Figure

[6]Serial Wire Debug Clock
[7]Serial Wire Debug Input/Output

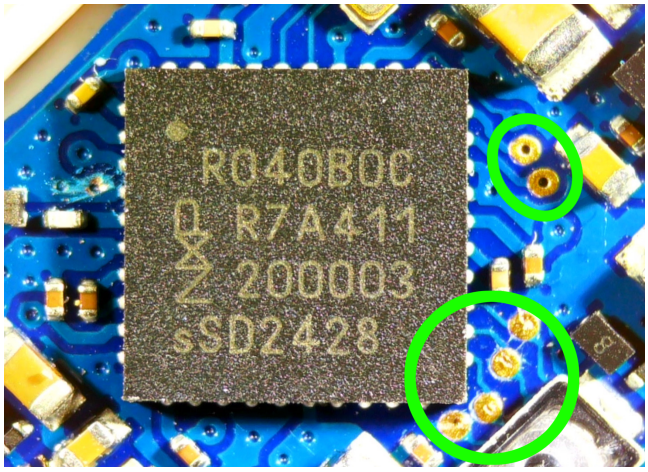

Figure 3: Highlight exposed test pads we probed on the NXP SR040.

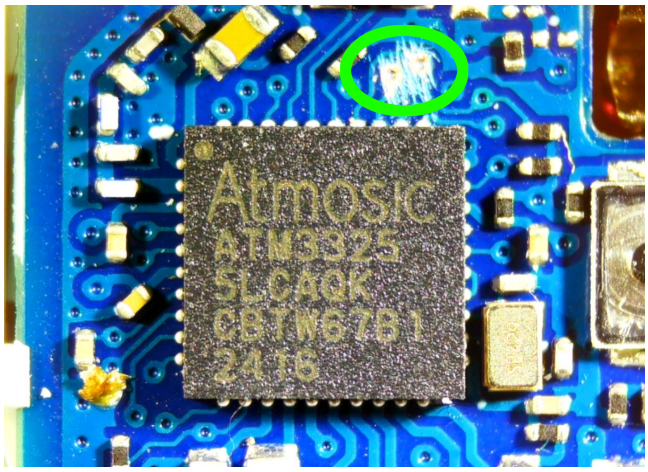

Figure 4: Highlight scraped pins for possible UART access.

4

These pins did produce data while probing, there was nothing recognizable by us and we were not able to serial to the chip using a tool like Screen[8] or Picocom[9]. Further analysis may be able to yeild something useful on this front but unfornately we were not able to produce much of value from testing these pins.

We had the highest hopes for a hardware security flaw with the SmartTag 2 but we were unable to find much of value. Further analysis of the SPI test pads and other pins on both NXP and Atmosic chips would provide even more value to furthering the security research for the SmartTag. A notable thing to mention is that taking the PCB out of the SmartTag casing and powering it on was still viable which means there is a method to bypass the speaker. This is a potential stalking threat, this same kind of attack was/is possible with the Apple Airtags as well. Although we are not supporters of ewaste, a possible solution to this disable the tracker if the speaker is not detected.

[8] https://www.gnu.org/software/screen/manual/screen.html
[9] https://github.com/npat-efault/picocom

5 Bluetooth Analysis

The analysis of Bluetooth communication between the Samsung SmartTag 2 and its paired devices was conducted to evaluate potential vulnerabilities that could be exploited. This investigation involved capturing and analyzing data transmitted during pairing, command execution, and general operation. The primary goal was to identify weaknesses that could allow external manipulation or disruption of the SmartTag's functionality without requiring the Samsung tablet application.

To facilitate this research, we employed an NRF52840 dongle [11] [12] connected to a MacBook Pro running macOS 14, along with Wireshark [13] for real-time packet analysis. Additionally, Android Debug Bridge (ADB) logs from a Samsung S8 tablet were used to capture application-level interactions. This setup enabled the capture of both HCI-level Bluetooth communications and higher-level commands sent via the Samsung ecosystem. These tools were utilized to monitor pairing sequences, test various commands, and record the resulting data.

Initial experiments focused on the pairing process. During these tests, we observed a series of HCI commands and events facilitating the connection between the tag and the Samsung tablet. The captured logs demonstrated the use of encrypted payloads and proprietary authentication protocols, which likely prevent unauthorized pairing attempts. Although we were unable to identify vulnerabilities in this handshake process, the recorded sequences provided valuable insights into the secure nature of Samsung's BLE stack.

Following the pairing analysis, we conducted a comprehensive examination of commands sent to the SmartTag 2. Commands such as playing sounds, adjusting volume, and locating the tag were initiated through the Samsung application, with their corresponding Bluetooth traffic captured and analyzed. Each command generated unique BLE traffic, allowing us to examine specific payloads for exploitable patterns. However, the data revealed no clear vulnerabilities, as all payloads appeared to be encrypted or obfuscated, effectively limiting any straightforward replication or modification attempts.

One particularly intriguing log entry was the initial Sent Reset command observed during pairing. This command, captured as follows:

```
Frame 1: 4 bytes on wire (32 bits), 4 bytes captured (32 bits)
Bluetooth
    [Source: host]
    [Destination: controller]
Bluetooth HCI H4
```

```
    [Direction: Sent (0x00)]
    HCI Packet Type: HCI Command (0x01)
Bluetooth HCI Command - Reset
```

Prompted a hypothesis about its potential utility as an attack vector. If this reset command could force the tag into a constant reset state, it might render the device non-functional. Such an exploit would involve repeatedly issuing the reset command using a BLE transmitter, effectively preventing the tag from performing its intended functions. This could disrupt the functionality of any nearby tags for as long as the command is continuously broadcast. While theoretically plausible, our attempts to implement this approach were inconclusive due to encryption [14] mechanisms and the inability to validate whether the reset command was received and executed by the tag outside of the authenticated environment.

The challenges encountered during this analysis highlight the robust security measures employed by Samsung. The encryption of BLE communications and the apparent reliance on proprietary protocols serve as significant barriers to unauthorized interactions. Furthermore, the dependency on the Samsung application for command initiation adds another layer of protection, as it likely incorporates additional authentication steps that cannot be bypassed using standard tools or methods. Despite these challenges, the investigation revealed the potential for disruptive techniques, such as the reset command hypothesis, which warrants further exploration.

Future research should focus on refining this hypothesis by employing advanced BLE transmission tools capable of bypassing authentication mechanisms. Additionally, hardware-level debugging of the SmartTag 2 through interfaces such as JTAG or UART may reveal critical information about its firmware and response to reset commands. Developing a custom BLE stack tailored to mimic the Samsung tablet's behavior could also prove instrumental in directly interacting with the tag and validating potential exploits.

This investigation represents an initial step toward uncovering Bluetooth vulnerabilities in the Samsung SmartTag 2. While no definitive exploits were identified, the findings underscore the complexity of modern BLE implementations and the need for sophisticated tools and methodologies to advance this field of research. [15] [16] [17]

6 Related Work

The security of IoT devices has been a topic of significant interest, with studies focusing on Bluetooth Low Energy (BLE) vulnerabilities, Ultra-Wideband (UWB) technology, and hardware reverse engineering. This section summarizes key findings from prior research, highlighting existing gaps and their relevance to the Samsung SmartTag 2.

Security of BLE/UWB IoT Devices

BLE and UWB-based IoT devices, such as Apple's AirTags, Samsung SmartTags, Tile trackers, etc, have been the subject of extensive security analysis due to their widespread adoption and potential for misuse. The researchers of the paper *"AirTag of the Clones: Shenanigans with Liberated Item Finders"* examined AirTags [18] in depth, uncovering vulnerabilities in both hardware and software. Key findings included the ability to extract firmware via voltage glitching and modify it to clone AirTags or manipulate NFC functionality [19]. A common vulnerability in these UWB devices is an exploit called Ghost Peak [10]. This is where an attacker can manipulate the UWB signals of devices by affecting the signal strength and tricking devices into thinking the other device is closer than it actually is.

This research provides a foundational understanding of BLE vulnerabilities and demonstrates effective methodologies for analyzing hardware and firmware security. This provided us with ample knowledge in understanding the hardware and security componets of an BLE IoT device before diving into researching Samsung SmartTag 2.

Reverse Engineering IoT Devices

The reverse engineering of IoT hardware often involves identifying and probing key interfaces, such as SPI, JTAG, and UART. In the AirTag study, researchers exploited debug test points and bypassed protections like Access Port Protection (APPROTECT [20]) using voltage glitching [19]. These methods enabled firmware extraction and modification, providing insights into the device's security design. [21]

Our exploration of Samsung SmartTag 2 similarly focuses on hardware-level analysis, including probing PCB test pads for SPI communication signals. The methodologies used in the AirTag research have guided our approach, particularly in setting up a cost-effective probing environment.

Samsung SmartTag 1

The Samsung SmartTag 1 has been subjected to hardware-level reverse engineering, revealing vulnerabilities that allowed researchers to dump its firmware and analyze its internal operations. The repository *Samsung-SmartTag-Hack* provides detailed images of the device's PCB and its components, along with descriptions of the fault injection techniques used to bypass protections on the Nordic nRF52833 chip. These methods involved manipulating the device's power supply to induce glitches [22], enabling access to otherwise secure memory regions [9].

The insights from this research provided a foundation for understanding the hardware architecture of Samsung SmartTags. By identifying key interfaces and components, such as test points for SPI and UART communication, the repository not only highlights the device's susceptibility to physical tampering but also guides further investigation into the security of SmartTag 2. The availability of detailed documentation, images, and datasheets has significantly contributed to our understanding of SmartTag vulnerabilities and informed the methodology used in our own exploration.

7 Conclusion

Samsung SmartTag 2s are a great piece of technology to use but show signs that they have some implementation flaws such as the "broken" speaker attack and the Ghost Peak attack. These attacks are mainly in the software implementations on the SmartTag so they could be fixed in the future. As far as our testing goes, we concluded that the hardware implement of the two chips was secure to the most basic of attackers. We can not conclude that the chips may not be vulnerable to a voltage glitch attack, we were unable to conduct that attack.

The viability and usability of these attacks are pretty low. An average user would not encounter this in the wild. These attacks require a lot of effort for little in return. Realistically, if an attacker wanted to track someone or something, they could get another device that doesn't notify users that something is following them. Our research helps all users understand more about their technology.

We established a great foundation for additional research and attacking to take place. If we had more time, resources, and knowledge we are sure we could create something for substantial, but we hope that others take what we have established here and conduct their own research.

Appendix: Team Contribution Breakdown

This section provides a detailed breakdown of the contributions made by each team member during the project.

Ali Chowdhurry

He coordinated the overall structure and flow of the report, ensuring adherence to the IEEE format. He also analyzed and interpreted the findings from both hardware and Bluetooth investigations.

Andrew Schwartz

He led the Bluetooth data collection, including setting up the NRF dongle and capturing BLE communication logs on the secondary DUT[10]. He also conducted experiments involving pairing, command execution, and reverse-engineering of payloads. He conducted hardware data collection and analysis alongside Jose.

Jose Vasquez

He conducted the hardware analysis of the SmartTag 2 PCB, including identifying and probing the exposed test pads.He reviewed datasheets and technical documentation for the NXP SR040 and Atmosic ATM3325 chips. He researched Bluetooth and made the bluetooth analysis conclusion in the report.

[10]Device Under Testing

Collaborative Efforts

All team members contributed to identifying relevant prior work and reviewing related research in Section 6. Discussions and brainstorming sessions were held to formulate hypotheses, design experiments, and analyze results.We all provided proofreading and revisions of the final report were performed collaboratively.

Acknowledgments

We would like to thank all the previous research that has gone into devices like the Samsung SmartTag 2 and the technologies behind it. A huge thank you to the AirTag of the Clones [19] guys for inspiring us to conduct this project. A couple more thanks to the previous smarttag hack people [9] and anyone who makes their research and information available for free on the internet.

References

[1] Samsung Newsroom, "Introducing the New Samsung Galaxy SmartTag2: A Smart Way to Keep Track of Important Things in Your Life," 2023, last accessed 5 December 2024. [Online]. Available: https://news.samsung.com/us/introducing-new-samsung-galaxy-smarttag2-a-smart-way-to-keep-track-of-important-things-in-your-life/

[2] Samsung Support, "Preventing SmartTag 2 Location Tracking," 2024, last accessed 5 December 2024. [Online]. Available: https://www.samsung.com/ae/support/mobile-devices/preventing-smartag-2-location-tracking/

[3] Samsung, "Samsung find: Locate and manage your devices," https://samsungfind.samsung.com/, 2024, last accessed 5 December 2024.

[4] RF Wireless World, "Difference between UWB LRP PHY and HRP PHY Mode," 2024, last accessed 5 December 2024. [Online]. Available: https://www.rfwireless-world.com/Terminology/Difference-between-UWB-LRP-PHY-and-HRP-PHY-mode.html

[5] NXP Semiconductors, "Trimension SR040: Reliable UWB Solution for IoT," 2024, last accessed 5 December 2024. [Online]. Available: https://www.nxp.com/products/wireless-connectivity/trimension-uwb/trimension-sr040-reliable-uwb-solution-for-iot:SR040

[6] M. Weller, J. Classen, F. Ullrich, D. Waßmann, and E. Tews, "Lost and found: stopping bluetooth finders from leaking private information," in *Proceedings of the 13th ACM Conference on Security and Privacy*

in Wireless and Mobile Networks, ser. WiSec '20. ACM, Jul. 2020, p. 184–194. [Online]. Available: http://dx.doi.org/10.1145/3395351.3399422

[7] Atmosic Technologies, "ATM33: Ultra-Low Power Bluetooth SoC for IoT and Wearables," 2024, last accessed 5 December 2024. [Online]. Available: https://atmosic.com/products_atm33/

[8] Tingfeng Yu and James Henderson and Alwen Tiu and Thomas Haines, "Privacy Analysis of Samsung's Crowd-Sourced Bluetooth Location Tracking System," *ArXiv*, vol. abs/2210.14702, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:253116608

[9] whid injector, "Samsung-SmartTag-Hack," 2021, last accessed 5 December 2024. [Online]. Available: https://github.com/whid-injector/Samsung-SmartTag-Hack

[10] Secure Positioning, "Ghost Peak: Practical Distance Reduction Attacks Against HRP UWB Ranging," 2021, last accessed 5 December 2024. [Online]. Available: https://securepositioning.ch/ghost-peak/

[11] Nordic Semiconductor, "nrf52840 dongle," 2024, last accessed 5 December 2024. [Online]. Available: https://www.nordicsemi.com/Products/Development-hardware/nrf52840-dongle

[12] Nordic Semiconductor Academy, "nrf sniffer for bluetooth low energy," 2024, last accessed 5 December 2024. [Online]. Available: https://academy.nordicsemi.com/courses/bluetooth-low-energy-fundamentals/lessons/lesson-6-bluetooth-le-sniffer/topic/nrf-sniffer-for-bluetooth-le/

[13] Wireshark Foundation, "Wireshark: Go Deep," https://www.wireshark.org/, 2024, last accessed 5 December 2024.

[14] Darrel Hankerson, Alfred J. Menezes, and Scott Vanstone, *Guide to Elliptic Curve Cryptography*. Springer Publishing Company, Incorporated, 1st edition, 2010.

[15] Fenghao Xu, Wenrui Diao, Zhou Li, Jiongyi Chen, Kehuan Zhang, "BadBluetooth: Breaking Android Security Mechanisms via Malicious Bluetooth Peripherals," *ndss-symposium*, 2019, last accessed 5 December 2024. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019_06B-4_Xu_paper.pdf

[16] A. Heinrich, M. Stute, and M. Hollick, "penHaystack: A framework for tracking personal Bluetooth devices via Apple's massive Find My network," *ACM*, 2021. [Online]. Available: https://dl.acm.org/doi/10.1145/3448300.3468251

[17] Yue Zhang, Jian Weng, Rajib Dey, Yier Jin, Zhiqiang Lin, and Xinwen Fu, "Breaking Secure Pairing of Bluetooth Low Energy Using Downgrade Attacks," *Usenix*, 2020. [Online]. Available: https://www.usenix.org/system/files/sec20-zhang-yue.pdf

[18] Travis Mayberry and Ellis Fenske and Dane Brown and Jeremy Martin and Christine Fossaceca and Erik C. Rye and Sam Teplov and Lucas Foppe, "Who Tracks the Trackers?: Circumventing Apple's Anti-Tracking Alerts in the Find My Network," in *WPES '21: Proceedings of the 20th Workshop on Privacy in the Electronic Society*. ACM, 2021, pp. 181–186.

[19] Thomas M. Roth and Fabian Freyer and Matthias Hollick and Jiska Classen, "AirTag of the Clones: Shenanigans with Liberated Item Finders," *2022 IEEE Security and Privacy Workshops (SPW)*, pp. 301–311, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251078522

[20] Limited Results, "nRF52 Debug Resurrection: APPROTECT Bypass," 2020, last accessed 5 December 2024. [Online]. Available: https://limitedresults.com/2020/06/nrf52-debug-resurrection-approtect-bypass/

[21] J. van Woudberg and C. O'Flynn, *The Hardware Hacking Handbook – Breaking Embedded Security with Hardware Attacks*. San Francisco, 2022.

[22] Klee, Tyler and Hudson, Michael and Orlikowski, Dave and Wright, Matthew, "Exploiting the Unpatchable: Microcode as a Persistent Platform for Hardware Implants," in *Proceedings of the 14th USENIX Workshop on Offensive Technologies (WOOT 20)*, 2020, last accessed 5 December 2024. [Online]. Available: https://www.usenix.org/system/files/woot20-paper-klee.pdf