6.5610 Final Report - BLE

Caden Gradek cgradek@mit.edu
Yiqing Du yiqingdu@mit.edu
Rishab Parthasarathy rpartha@mit.edu
Arnold Su arnoldsu@mit.edu

Abstract

As wearable health devices become increasingly integrated into daily life, the security of their wireless communications is of growing concern—especially given the sensitive nature of the physiological data they transmit. Many of these devices rely on Bluetooth Low Energy (BLE) for connectivity, but often use insecure pairing methods like Just Works, which sacrifice authentication for ease of use. In this study, we investigate the vulnerabilities of BLE pairing protocols, focusing on the feasibility of a Man-in-the-Middle (MitM) attack. Using a Nordic nRF52840 USB dongle configured with BLE sniffer firmware, we intercepted and analyzed live BLE traffic between a commercial heart rate monitor and a mobile device. We demonstrate that, under Just Works pairing, an attacker can fully decrypt and relay data between two devices without detection. Our analysis reveals that even encrypted BLE sessions can be compromised when authentication is absent or improperly implemented. To address these risks, we discuss hardware-constrained alternatives, such as low-power input methods and Out-of-Band (OOB) pairing mechanisms, that offer stronger security guarantees while maintaining usability for health-focused wearables.

1 Introduction

In recent years, wearable health devices, such as heart rate monitors and fitness trackers, have gained widespread adoption. These devices enable users to track key health metrics such as heart rate, step count, and sleep patterns. However, with the growing prevalence of these health monitors comes an important consideration: security. Many of these wearable health items operate under significant power and computational constraints, which often necessitate the use of communication protocols that prioritize efficiency over reliability and security.

Bluetooth Low Energy (BLE) has become the most widely used communication protocol for Internet of Things (IoT) devices like health monitors, and is currently used in billions of devices worldwide (1). Many of these devices rely on BLE to facilitate seamless data transfer, which, while effective for quick and low-power data exchange, can be vulnerable to various security threats like eavesdropping and Man-in-the-Middle (MitM) attacks, which we exploit in this study. Although newer Bluetooth standards offer enhanced security features designed to mitigate these risks, their effectiveness is often undermined by improper implementation or non-adoption in device firmware and software (1). Many researchers have already raised concerns about the reliance on BLE protocols, and in particular on the security and privacy of BLE itself(2) (1).

The trade-off between ease-of-use and robust security becomes even more critical as the sensitivity of the data transmitted grows (3) (4). Specifically, health information is regarded as the most confidential of all types of personal information, with surveys showing that a majority of people believe that health information is sensitive or confidential in nature. Furthermore, less than 50% of people who use wearable health monitors understand the device's information security policies and how the companies are protecting their privacy. Even fewer people are familiar with how their device, transmits, stores, and handles their sensitive information (5). Existing studies have already posited that the information collected by these devices, when combined with other sources of data and

6.5610 Applied Cryptography Final Project (Spring 2025).

publicly available information, can reveal sensitive information about individuals, breaching their privacy (6). We thus believe it is worthwhile to examine these devices more closely, so as to spread awareness and make clear the exact security and privacy risks.

In this study, we are therefore interested in investigating the security of these health wearables—for example, answering questions like: is it possible for us to extract personal health data simply by looking at Bluetooth packets? Secondly, we aim to study alternative cryptographic strategies compatible with BLE constraints that can be used to prevent attacks and bolster data protection for users of health monitors.

2 Related Works

2.1 BLE legacy vs current BLE methods

Bluetooth Low Energy (BLE) connection security has evolved significantly since its inception. The initial approaches, used in BLE versions 4.0 and 4.1, are known as Legacy Connections (7). This method utilizes a custom, multi-phase key exchange protocol, which has been identified as a source of significant security concerns (1). Understanding the distinction between these older connections and the newer Secure Connections (BLE v4.2+) is crucial for appreciating the security implications for devices like health wearables.

2.1.1 BLE Legacy Connections (Versions 4.0 & 4.1)

Legacy Connections rely on a custom-designed, three-phase process for pairing devices and establishing an encrypted link. Unfortunately, this process contains significant security weaknesses, primarily stemming from the transmission of critical information without encryption (in plaintext) (1).

- **Phase 1**: Feature Exchange: To determine the appropriate pairing method (see Section 2.2), the devices begin by exchanging details in plaintext about their capabilities—such as authentication requirements, preferred key sizes, and whether they support input/output features like a display or keyboard. (1) (7).
- **Phase 2**: Key Generation: This is the phase where security is meant to be established, but it remains at the core of many attacks (1). Devices first agree on a Temporary Key (TK) using a pairing method chosen based on the capabilities exchanged in Phase 1. The security of the entire connection heavily relies on this TK (7). Next, each device generates a 128-bit random number (Mrand for the controller, Srand for the periphery). Critically, these random numbers are also exchanged completely unencrypted (1) (7). Using the TK, the plaintext random numbers, and other parameters (like device addresses), the devices send signatures, which are calculated confirmation values (Mconfirm, Sconfirm), which they exchange to verify they possess the same TK (7). Once signatures have confirmed that the devices agree on the TK, the devices use the TK and the exchanged random numbers to compute an encryption key for the current pairing session, which is called the Short-Term Key (STK) (1).

  However, this system is vulnerable to attacks. The biggest issue here is the plaintext exchange of Mrand and Srand. If an attacker can guess or brute-force the TK (which can be trivial in the "Just Works" pairing model where TK is 0, or feasible if a short, guessable PIN is used in Passkey Entry (8)), they have everything needed to calculate the exact same STK as the legitimate devices (1). They simply perform the same cryptographic function using the known TK and the eavesdropped plaintext random numbers. Once the attacker has the STK, they can decrypt all the supposedly secure communication for that session and potentially inject malicious packets (9). The entire session's security hinges on the secrecy of the TK, which is often not well protected depending on the pairing method selected based on device capabilities (1).
- **Phase 3**: Bonding (Optional): If devices decide to remember each other for future connections, they exchange longer-term keys in this phase, such as the Long-Term Key (LTK), Identity Resolving Key (IRK), and Connection Signature Resolving Key (CSRK). While these keys are encrypted, they are encrypted using the STK derived in Phase 2 (1). Consequently, if an attacker compromised the STK in Phase 2, they can also decrypt and steal

these vital long-term keys (9). This compromises not just the current session, but potentially all future connections and device identification capabilities (1).

During Phase 1, an attacker passively eavesdropping can immediately learn about the devices involved and how they intend to pair since this information is sent in plaintext (1). This gives the attacker valuable intelligence for planning subsequent attacks, such as predicting the pairing method that will be used or preparing for a Man-in-the-Middle (MitM) attack (9) (1).

2.1.2 BLE Secure Connections (Versions 4.2 and newer)

Secure Connections were specifically designed to overcome the vulnerabilities inherent in the Legacy method (10). However, we note that devices with BLE Secure enabled can still use Legacy pairing techniques to connect to Legacy devices (1). Overall, the primary improvement in BLE Secure is a complete overhaul of Phase 2 (Key Generation). Secure Connections discard the flawed custom protocol involving TK, plaintext random numbers, and STK generation. Instead, they implement the industry-standard, Elliptic Curve Diffie-Hellman (ECDH) key exchange algorithm, specifically using the P-256 curve (10).

ECDH is a form of asymmetric (public-key) cryptography used for key agreement (11). Each device generates a temporary, secret ECDH private key and a corresponding public key. Using its own private key and the other device's public key, each device performs an ECDH computation. Due to the mathematical properties of elliptic curves, both devices will independently arrive at the exact same shared secret value (11). This shared secret is then used along with nonces exchanged between the devices to derive the Long-Term Key (LTK) using a key derivation function (10). However, the devices still exchange their public keys openly, which means that an attacker can observe these public keys (10).

The key benefit of ECDH is that the shared secret (and thus the LTK) is generated without ever transmitting sensitive keying material directly over the air. An eavesdropper observing the public key exchange cannot compute the resulting shared secret without knowing at least one of the private keys, which is computationally infeasible for the standardized P-256 curve (11). This provides strong protection against passive eavesdropping attacks that could easily compromise Legacy Connections (10).

This securely generated LTK is then used directly to encrypt the communication link. Pairing methods (like Passkey Entry, Out of Band) are still employed in Secure Connections, but they are used to authenticate the key exchange, ensuring that the devices are indeed communicating with the intended partner and preventing MitM attacks during the pairing process (10)(9). This authentication step binds the identity of the devices to the securely generated key.

In essence, Secure Connections replace the vulnerable, custom key generation mechanism of Legacy Connections with a strong, standardized public-key cryptography approach (ECDH) coupled with authenticated pairing methods. This vastly improves protection against the eavesdropping, MitM, and key-recovery attacks present in earlier BLE versions (1) (9). However, we note that Just Works pairing in Secure Connections provides no authentication, making it vulnerable against a potential MitM attack, as discussed below (12).

2.2 BLE Pairing Methods

The BLE standard defines several pairing methods, which are chosen based on the input/output (I/O) capabilities of the connecting devices (e.g., presence of a display, keyboard, or confirmation button) (13). These methods are used in both Legacy and Secure Connections, but their role and security implications differ significantly between the two modes. The primary goals are typically authentication (verifying the identity of the peer device to prevent Man-in-the-Middle attacks) and, in Legacy Connections, the generation or exchange of the Temporary Key (TK).

2.2.1 Just Works

Just Works is used when neither device has a display or keyboard for entering/confirming codes. The devices connect without requiring user interaction beyond perhaps an initial confirmation prompt.

Legacy Connections: This is highly insecure. The Temporary Key (TK) is set to zero (14). This provides no protection against passive eavesdropping (as the key used to derive the STK is known) and no protection against Man-in-the-Middle (MitM) attacks (12). An attacker can easily compute the STK and decrypt communication, or impersonate either device.

Secure Connections: Secure Connections use ECDH to generate the LTK, protecting against passive eavesdropping, however, the Just Works authentication step itself still offers no MitM protection (12). An attacker could potentially intercept the pairing process before authentication completes, although the link encryption uses the strong ECDH-derived key. It remains the least secure option and is not recommended when MitM protection is needed (12).

2.2.2 Passkey Entry

Passkey Entry is used when one device has a display and the other has a keyboard (or equivalent input method). One device displays a 6-digit passkey, and the user must enter this same passkey on the other device (1).

Legacy Connections: The 6-digit passkey becomes the TK (1). This provides MitM protection because the attacker doesn't know the passkey being entered by the user. However, it's still vulnerable to passive eavesdropping if the attacker can guess the 6-digit passkey (1 million possibilities, potentially brute-forceable offline) (8). It's also vulnerable if an attacker can observe the user entering the key (e.g., "shoulder surfing").

Secure Connections: The passkey is not used directly as the TK. Instead, it is processed with a nonce before being used in the ECDH key exchange. This provides MitM protection during authentication, and the underlying link encryption relies on the strong ECDH key, which when combined with the nonce, makes it much more secure against eavesdropping than the Legacy version (1).

2.2.3 Numeric Comparison

Numeric Comparison requires both devices to have a display and at least a confirmation button (e.g., Yes/No). Both devices independently calculate and display the same 6-digit number. The user must verify that the numbers match on both devices and confirm on each. This method was introduced with Secure Connections (BLE v4.2), and hence is not available in Legacy Connections (10). It provides MitM protection during authentication. If an attacker tries to intercept the connection, the numbers displayed on the two legitimate devices will not match (as the attacker would be performing separate ECDH exchanges with each device), and the user should reject the pairing (12). This is considered a secure method, provided the user diligently checks the displayed numbers, as it builds upon the Just Works method by adding a numerical verification layer (1).

2.2.4 Out of Band (OOB)

OOB uses a secondary communication channel (e.g., Near Field Communication - NFC, QR codes) to exchange cryptographic data used in the BLE pairing process (15) (16). This avoids transmitting sensitive pairing data over the potentially insecure BLE radio channel during the initial setup.

Legacy Connections: The OOB channel can be used to securely exchange a TK (up to 128 bits) or other necessary pairing values (7) (15). If the OOB channel itself is secure against eavesdropping and MitM, this is considered one of the most secure methods for Legacy Connections (12).

Secure Connections: The OOB channel can be used to exchange information needed for authenticated ECDH, such as public keys or confirmation/random values derived from them (10) (15). Again, the security relies heavily on the security properties of the OOB channel itself. It offers the potential for the highest security level if implemented correctly (13).

3 Methodology

3.1 Experiment Overview

With the understanding that most heart rate monitors operate using the Just Works pairing method, we are interested in the practical security afforded by these devices. Specifically, we are interested in the possibility of reading heart rate data packets without alerting the device user. For our experiments,

we tested on the CooSpo H6 Chest Strap Heart Rate Monitor. In order to carry out the MitM attack, we used the Nordic Semiconductor nRF52840 USB Dongle. Using the dongle's integration with Wireshark, we attempted to intercept the packets between the CooSpo Heart Rate Monitor and an iPhone.

3.2 nRF52840 Dongle

Nordic Semiconductor's nRF52840 USB Dongle is low-cost ($10), compact, and USB-powered. The dongle supports communication across multiple protocols, including Bluetooth 5.4, Bluetooth mesh, Thread, or Zigbee. In addition, Nordic Semiconductor provides tutorials and Python code for integration support with Wireshark, an open-source packet analyzer.

Although the nRF52840 Dongle is intended primarily for Bluetooth developers, this technology can easily be used by adversaries interested in sniffing a private BLE connection. The fact that we were able to collect such sensitive data with a very cheap and commercial device is alarming to say the least.

3.3 How Does the Attack Work?

In BLE Secure Connections using the Just Works pairing method, the Man-in-the-Middle (MitM) attack involves active interception. In this setup, the attacker positions themselves between the two devices at the moment of pairing and initiates two separate Just Works pairings: one with the peripheral device (e.g., heart-rate monitor) and one with the central device (e.g., smartphone). Since neither side has any mechanism to verify the identity of the peer (no numeric comparison or passkey entry), both devices believe they are securely paired—when in fact, they are each paired with the attacker. The attacker now becomes an invisible relay, decrypting, modifying, and re-encrypting all traffic in real time using the two independently negotiated STKs. This attack is visualized in Figure 1

Figure 1: Illustration of a Man-in-the-Middle (MitM) attempt during BLE Secure Connections. Alice and Bob each perform an Elliptic Curve Diffie-Hellman (ECDH) key exchange, but Mallory intercepts the pairing process and initiates separate ECDH exchanges with both parties. Without authentication methods like Numeric Comparison or Passkey Entry, Mallory can establish two secure-looking links and relay messages between Alice and Bob undetected

This attack is especially viable in environments where BLE devices automatically reconnect to previously paired devices, or where users are unaware of the pairing security implications. Moreover, the low power and I/O constraints of many health-monitoring wearables make it common for manufacturers to choose Just Works due to its convenience and simplicity, even though it provides no MitM resistance.

The result is that sensitive physiological data, such as live heart rate measurements, can be intercepted and potentially modified without detection. Our experimental setup confirms that, once pairing is established using Just Works, a properly positioned attacker can fully impersonate both communicating devices and gain access to all exchanged data, highlighting the critical need for authenticated pairing methods in BLE health devices.

3.4 Identifying Heart Rate Devices

Before we can intercept heart rate communication between devices, we must first identify the MAC address of the target heart rate monitor. Thankfully, Nordic Semiconductor's nRF Connect software provides some convenient tooling for identifying Bluetooth devices. Using *nRF Connect for Desktop Bluetooth Low Energy*, we were able to configure the dongle to read out all devices advertising BLE

connections. For any advertising BLE device, we could read the device name, MAC address, and device services without interacting with the device. It is extremely easy to identify heart rate devices since these devices include *Heart Rate* services within their advertisement. Once we have identified the target device using either the device name or services, we can pass the corresponding MAC address along to the Wireshark integration. Since these MAC addresses never change, we never have to repeat this step after identifying the MAC address for a device once.

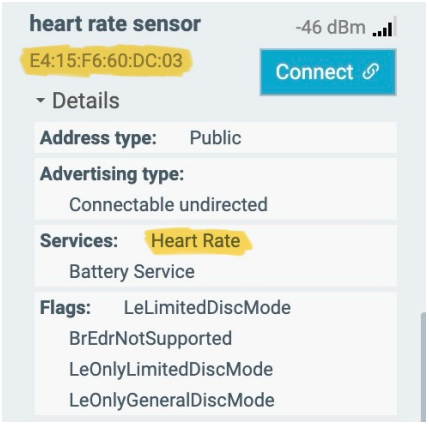

Figure 2: nRF Connect output for the CooSpo Heart Rate Monitor. While the heart rate monitor is looking to pair with a device, it advertises details about its connection. Included is the MAC address, device name, and device services.

3.5 Capturing Packets using Wireshark

Once we have identified the MAC address of the target device, we can begin to utilize Nordic Semiconductor's Wireshark integration for the dongle. Before pairing occurs, we specify in the dongle-Wireshark interface to target only packets corresponding to the MAC address. As pairing begins, the dongle hijacks the connection and acts as a Man-in-the-Middle as explained in Section 3.3. If the dongle successfully intercepts the connection, the decrypted heart rate communication begins to appear in the Wireshark packet window. Meanwhile, the communication between the iPhone and heart rate monitor proceeds as normal, and the heart rate user is unable to discern that their data is now exposed. In our small testing sample, we found that the dongle successfully intercepted the pairing connection more than half the time.

4 Results

In the course of our experiments, we successfully conducted a packet sniffing session between two paired BLE devices. Through careful observation and analysis of the captured traffic, we were able to identify both the MAC addresses involved and the specific protocol-level interactions that correspond to the transmission of heart rate data. Because of the built-in support for BLE packets in Wireshark, we found that the heart-rate data was transmitted using a Handle Value Notification (Opcode 0x1b) on attribute handle 0x0012. See Figure 3 to find a screenshot of how Wireshark was able to successfully identify packets that are transmitting heart-rate data.

This observation allowed us to construct a filter for isolating the relevant packets in Wireshark:

```
btle.length != 0 && btatt && btatt.handle == 0x0012 && btatt.opcode == 0x1b
```

This filter ensures that only non-empty BLE packets containing ATT notifications from the heart rate characteristic are displayed. Within these packets, we observed that the second byte of the value field consistently represents the actual heart rate value measured by the peripheral device. See Figure 4 where we were able to see which byte of the data corresponds to "Value", which is the heart-rate. We were able to confirm this when sniffing future transmissions, even when Wireshark was unable to fully parse the information for us.

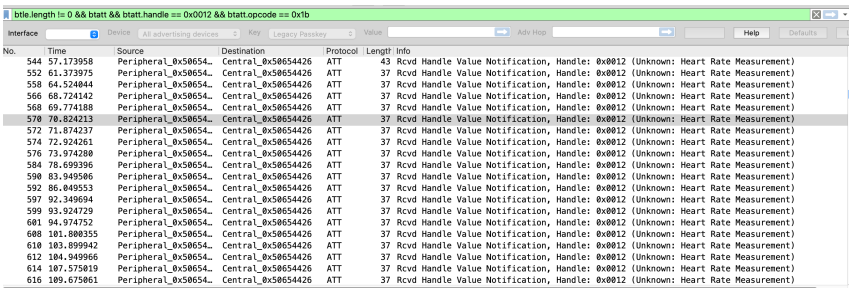

Figure 3: Screenshot of Wireshark successfully parsing and analyzing packet data between the heart rate monitor and mobile device. We can see that the packets associated with packets using handle value notification (opcode 0x1b) on attribute handle 0x0012 is heart-rate measurement data.

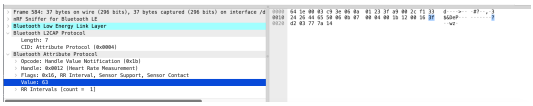

Figure 4: Individual packet data being transmitted from the heart-rate monitor to the paired mobile device. In this packet in particular, Wireshark was able to successfully parse the data and identify which byte corresponds to the heart-rate value itself.

The filter has enabled us to reliably extract real-time heart rate data from BLE communication streams. In Figure 5, we can see that by applying the filters and processing out that second byte of the value field, we were able to successfully (almost identically) recreate the heart-rate data being transmitted to the officially supported mobile app.

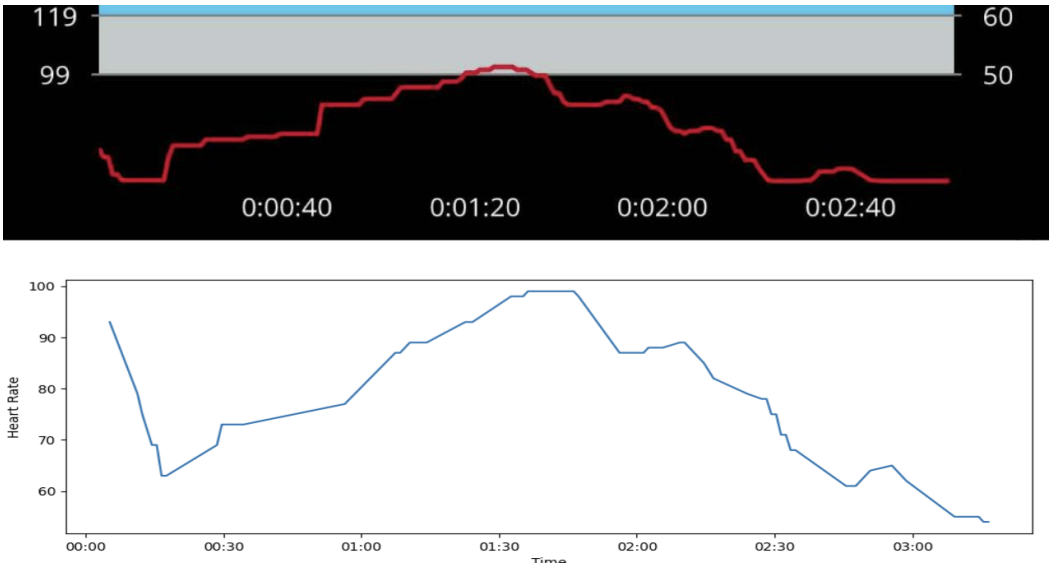

Figure 5: The top plot is the captured heart rate from the heart rate monitor from the officially supported mobile app for the device. The bottom is created by us by parsing packet data captured in Wireshark. The capturing window is the same for both plots.

5 Discussion

Our findings raise important concerns regarding the security and privacy of Bluetooth Low Energy (BLE) heart rate monitors. The relative ease with which we were able to sniff heart rate data between two paired BLE devices highlights a significant vulnerability. Despite the presence of more

sophisticated and secure pairing mechanisms in BLE, heart rate monitors often rely on lightly secured communication channels, which allow third parties to intercept sensitive physiological data without detection or authorization.

This vulnerability is especially troubling in medical contexts because it involves deeply personal and sensitive information that can have lasting consequences for an individual's privacy and safety. Breaches can lead to discrimination in employment, insurance, or social contexts. The sensitivity of this data makes safeguarding it imperative.

To address this issue, we evaluated several user-friendly and technically feasible security enhancements aimed at strengthening the device pairing process and protecting data integrity, recognizing that the attack and security vulnerabilities come from the limitations on the pairing process between devices. As a whole, it is often an engineering decision to use these insecure pairing methods because of the energy needs of more sophisticated methods. These devices often also value convenience and ease-of-use for their consumers, especially during the pairing process.

However, as explained in Section 2.2.2 and Section 2.2.3, a secure solution against the MitM attack we implemented is a passkey-based-system. The concern, however, is maintaining the low-energy requirements of these BLE devices. For instance, implementing a screen onto the heart-rate monitor would be too energy intensive, and make it no longer a low-energy device. However, by selectively choosing an I/O device such as a dial that is low-energy, we believe that we can significantly increase security, without compromising usability and energy consumption.

Alternatively, we also want to propose two solutions that utilize a secondary communication channel, or Out of Band (OOB) connection (see Section 2.2.4). One such method is using QR codes as a pairing method. QR codes are secure, easy to use, and easily implementable on health care devices. However, one key concern is people who use health care monitors (especially those who use them as fitness trackers) often put the monitors under more strenuous conditions. Environmental factors like dust, sweat, and water could degrade a QR code, affecting its long-term readability. Outside of QR codes, near field communication (NFC) technology is another OOB pairing method that is secure against MitM attacks. NFC is already widely implemented in the most secure devices such as credit cards. We believe it is worth considering porting over this technology to healthcare devices.

For all of these methods, the cost is low (around $10 at maximum) along with being low power. Even for the only powered device, which is NFC, the power consumption is supplied through magnetic eddy currents from the active device (such as the phone trying to tap). These currents are independent of the low-energy BLE device and also are at most of the order of 10 milliamps, which is within the energy range of standard BLE (10; 17).

6 Contribution

Caden identified relevant health devices and profiled their BLE capabilities, Yiqing did background research and discovered the attack vector, Rishab researched the hardware aspect of Bluetooth sniffing and implemented the final version of the attack, and Arnold analyzed the results, synthesized plots, and proposed solutions. All team members contributed equally.

References

[1] A. Barua, M. A. Al Alamin, M. S. Hossain, and E. Hossain, "Security and privacy threats for bluetooth low energy in iot and wearable devices: A comprehensive survey," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 251–281, 2022.

[2] A. K. Das, P. H. Pathak, C.-N. Chuah, and P. Mohapatra, "Uncovering privacy leakage in ble network traffic of wearable fitness trackers," in *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*, HotMobile '16, (New York, NY, USA), p. 99–104, Association for Computing Machinery, 2016.

[3] S. Seneviratne, Y. Hu, T. Nguyen, G. Lan, S. Khalifa, K. Thilakarathna, M. Hassan, and A. Seneviratne, "A survey of wearable devices and challenges," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2573–2620, 2017.

[4] T. Zhang, J. Lu, F. Hu, and Q. Hao, "Bluetooth low energy for wearable sensor-based healthcare systems," in *2014 IEEE healthcare innovation conference (HIC)*, pp. 251–254, IEEE, 2014.

[5] L. Cilliers, "Wearable devices in healthcare: Privacy and information security issues," *Health Information Management Journal*, vol. 49, no. 2-3, pp. 150–156, 2020. PMID: 31146589.

[6] K. Montgomery, J. Chester, and K. Kopp, "Health wearables: ensuring fairness, preventing discrimination, and promoting equity in an emerging internet-of-things environment," *Journal of Information Policy*, vol. 8, pp. 34–77, 2018.

[7] Bluetooth Special Interest Group (SIG), "Bluetooth Core Specification Version 4.0, Vol 3, Part H, Security Specification," tech. rep., Bluetooth Special Interest Group (SIG), Jun 2010. Available online at https://www.bluetooth.com/.

[8] M. Ryan, "Bluetooth: with low energy comes low security," in *Proceedings of the 7th USENIX Conference on Offensive Technologies*, WOOT'13, (USA), p. 4, USENIX Association, 2013.

[9] D. Antonioli, N. O. Tippenhauer, and K. Rasmussen, "The KNOB is Broken: Exploiting Low Entropy in the Encryption Key Negotiation of Bluetooth BR/EDR," in *Proceedings of the 29th USENIX Security Symposium (USENIX Security '20)*, pp. 1447–1464, USENIX Association, Aug. 2020.

[10] Bluetooth Special Interest Group (SIG), "Bluetooth Core Specification Version 4.2, Vol 3, Part H, Security Specification," tech. rep., Bluetooth Special Interest Group (SIG), Dec 2014. Available online at https://www.bluetooth.com/.

[11] National Institute of Standards and Technology (NIST), "Special Publication 800-56A Revision 3: Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography," Tech. Rep. NIST SP 800-56Ar3, U.S. Department of Commerce, Apr 2018.

[12] N. Semiconductor, "Legacy pairing vs le secure connections." Nordic Developer Academy, n.d. Accessed: 2025-04-24.

[13] T. Instruments, "Bluetooth low energy (le) - security fundamentals." SimpleLink Academy for CC23xx, n.d. Accessed: 2025-04-24.

[14] Y. Horbenko, "How secure is the ble communication standard?." Lemberg Solutions Blog, Mar 2019. Accessed: 2025-04-24.

[15] ndeflib developers, "Bluetooth secure simple pairing." ndeflib 0.3.2 documentation, n.d. Accessed: 2025-04-24.

[16] M. Brabham and M. Watson, "Pairing Bluetooth devices via QR code," Defensive Publication 4589, Technical Disclosure Commons, Sep. 2021. Accessed on 2025-04-26.

[17] NXP Semiconductors, "NTAG213, NTAG215, NTAG216 - NFC Forum Type 2 Tag compliant ICs." `https://www.nxp.com/products/rfid-nfc/nfc-hf/ntag/ntag213-ntag215-ntag216-nfc-forum-type-2-tag-compliant-ics:NTAG213_215_216`, 2014. Accessed: 2025-05-12.