Towards Secure and Safe Industrial Systems using BLE's Broadcast Isochronous Streams

Sebastian Dorn†, Fikret Basic*, Rainer Hofmann†, Michael Spörk†, and Carlo Alberto Boano*
*Institute of Technical Informatics, Graz University of Technology, Austria
†DEWINE Labs, Austria
Email: {basic, cboano}@tugraz.at, {sebastian.dorn, rainer.hofmann, michael.spoerk}@dewinelabs.com

***Abstract*—Wireless technology is becoming increasingly prominent in industrial environments today, particularly for controlling automated guided vehicles such as mobile robots. Beyond conventional communication requirements, wireless solutions must also comply with functional safety standards such as IEC 61508 and IEC 62745. However, the susceptibility of wireless systems to RF interference and cyberattacks raises questions about their suitability for safety-critical environments. In this paper, we investigate the integration of one of the latest features of Bluetooth Low Energy, namely *Broadcast Isochronous Streams* (BIS), which enable efficient one-to-many broadcast communication, into industrial systems. Furthermore, we present an application-layer framework built on top of BIS that provides essential security and safety features such as encryption, decryption, lightweight key exchange, transmission of periodic keepalive signals, and emergency stop functionality. Experimental results demonstrate that the proposed framework adds only 33% latency overhead, while ensuring compliance with key safety and security requirements. The framework also supports network scaling to hundreds of devices and enables timely execution of safety procedures even in harsh RF conditions, demonstrating that BIS can be securely and reliably employed in industrial environments.**

***Index Terms*—Bluetooth, BIS, IoT, Security, Safety, Low-power wireless, Industry, Automation, IEC 61058, IEC 62745, Zephyr.**

I. Introduction

Industry, much like every other aspect of daily life, has invested significant effort and interest in the digitalisation of its various processes. With the advent of small yet powerful embedded systems within the Internet of Things (IoT) context, both industry and academia have started developing sophisticated architectures capable of supporting modern production processes. This evolution has been visible through the technological movement known as *Industry 4.0*, and more recently, through *Industry 5.0*, which expands the focus toward sustainability and human-centric paradigms [1].

Today, IoT technologies are prevalent across many industrial environments, particularly in warehouses, where they serve as a foundational component of automation technologies, for example, through agents such as robots or automated guided vehicles (AGV) [2]. Communication between these agents typically relies on wireless technologies. However, an important distinction in an industrial setting, compared to other environments, is that the devices and processes must comply with numerous functional safety standards, such as IEC 61508 [3] and IEC 62745 [4]. Meeting these standards can be challenging, as conventional wireless technologies, commonly used in less-critical environments, might not satisfy communication reliability requirements posed by the standards, which are necessary to prevent harm to the equipment or personnel. Consequently, several application-layer protocols have been developed specifically for industrial applications, for instance, WirelessHART [5] and OPC UA [6], which aim to provide stable and reliable communication even under harsh environments; however, they may remain heavily limited in terms of performance compared to their wired counterparts.

Beyond safety considerations, security has become an equally vital concern in modern wireless deployments, as wireless communication significantly increases the attack surface. Successful attacks targeting devices or communication channels can result not only in information loss and privacy breaches but, in the worst case, in compromised system safety [7]–[9]. While safety and security are distinct concepts, they coexist within industrial systems, and compromised security frequently leads to safety degradation. Therefore, contemporary industrial systems often face a trade-off between reliability, safety, and security, resulting in power-hungry and highly application-specific solutions.

An alternative to traditional industrial wireless technologies is Bluetooth, particularly Bluetooth Low Energy (BLE), which is well-established and widely used across diverse IoT applications and low-power wireless networks [10]. BLE offers broad applicability across diverse contexts and use cases, providing efficient short-range wireless communication with low latency and high reliability. Moreover, with the introduction of *isochronous channels* in the Bluetooth v5.2 Core Specification, BLE has been extended to support time-synchronized data transmission. In particular, *Broadcast Isochronous Streams* (BIS) enable one-to-many broadcast communication with bounded latency and reliability guarantees, thereby supporting real-time communication scenarios required by emerging industrial applications. To date, BIS have been primarily adopted in LE Audio applications [7], where safety-critical considerations are typically not addressed.

In this work, we aim to bridge this gap by investigating the use of BLE BIS technology within industrial settings through the design and implementation of an application-layer framework providing key safety and security modules. Our objective is to provide a robust and adaptable framework suitable for a range of automated industrial use cases, while considering industrial security and safety requirements.

Contributions. We present the following key contributions:

- A thorough threat analysis for safety-critical industrial applications relying on BLE BIS communication (§ III).
- A security system model that incorporates essential principles from IEC 61508 and IEC 62745 (§ IV).
- The design and implementation of an application-layer framework built on top of BLE BIS technology, enabling the design of secure and safe industrial systems (§ V).
- An evaluation of the proposed design using off-the-shelf hardware demonstrates the feasibility of our approach, which incurs only a 33 % increase in end-to-end latency. Our results also show that the proposed solution correctly executes operational safety/security procedures and maintains a communication reliability between 96% and 99% despite the presence of sustained RF interference (§ VI).

II. Background

A. Industrial use cases relying on wireless technology

The industrial use of wireless technologies is rapidly expanding as part of broader digitalization trends such as Industry 4.0 and Industry 5.0. This growth is driven by the need to improve efficiency and enable smarter production environments. Central to this transformation is the *smart factory*, where sensors, actuators, and autonomous systems collect data and perform tasks independently, helping to optimize workflows and ensure safe operation in environments that are dangerous or difficult for humans to access. To expand on existing use cases and enhance their functionality, smart factories increasingly rely on wireless communication technologies. Key applications include industrial automation, asset tracking, and safety monitoring. Assets such as tools, materials, and personnel are monitored using technologies such as RFID, IEEE 802.15.4, IEEE 802.11ah, 5G URLLC, and LoRaWAN, while safety-critical functions, including emergency stops and human–robot collaboration, depend on reliable, low-latency communication [11]. However, the adoption of such systems is slowed by concerns regarding cybersecurity, investment costs, and compliance with stringent functional safety standards, such as IEC 61508 and IEC 62745. This work investigates how to combine safety-critical requirements with appropriate security mechanisms when using BLE technology.

B. Safety and security requirements and standards

In industrial environments, safety and security are essential requirements for dependable system operation. Functional safety provides dedicated mechanisms to prevent hazardous outcomes, with the IEC 61508 defining requirements for electrical, electronic, and programmable electronic systems. It also introduces Safety Integrity Levels (SILs) based on a system's Probability of Failure on Demand (PFD) [3]. Reliable and low-latency communication is critical, as message corruption, loss, or excessive delays can increase the PFD and jeopardize compliance with the targeted SIL [12]. Alongside safety, industrial systems must also satisfy core security attributes, namely confidentiality, integrity, and availability,

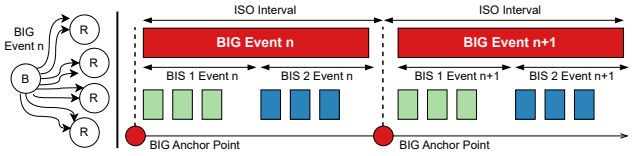

Fig. 1: Illustrative example: a broadcaster (B) transmits two Broadcast Isochronous Streams (BIS) to several receivers (R) as part of a single Broadcast Isochronous Group (BIG). Each BIG event – the actual transmission of all BIS in the group – is scheduled at the so-called BIG anchor point, which occurs periodically according to the ISO interval.

to protect against eavesdropping, tampering, and denial-of-service attacks. Furthermore, authentication, access control, and non-repudiation ensure that only trusted entities interact with the system and that actions cannot be falsely denied [8], [9]. Together, these safety and security measures form the foundation for protecting industrial wireless systems while supporting safety-critical functions.

C. BLE and BIS

Bluetooth Low Energy (BLE), introduced in Bluetooth v4.0, was designed to provide significantly lower power consumption while benefiting from the widespread adoption of the overall Bluetooth ecosystem. Bluetooth v5.2 introduced the concept of *Isochronous Channels* (ISO): these include the connection-less *Broadcast Isochronous Streams* (BIS), which enable efficient one-to-many broadcast communication [10]. BIS data is transmitted within *Broadcast Isochronous Group* (BIG) events, as illustrated in Fig. 1. Originally designed for audio broadcasting – for example, for public announcements at airports or sports venues – a typical BIG contains two BIS, corresponding to the left and right stereo channels of wireless earbuds [7]. The low-latency nature of BIS transmission, combined with synchronized delivery across multiple receivers, ensures coherent and simultaneous data reception on all subscribed devices.

Notably, audio broadcast applications show structural similarity to industrial broadcast applications, which also require low-latency and highly-reliable communication. We therefore study in this work how to integrate BIS into industrial environments. In an industrial context, for example, one could employ this technology to establish a BIG, where safety information for all AGV devices is broadcasted. Within this BIG, as shown in Fig. 1, BIS 1 could carry *emergency stop signals*, while BIS 2 could contain data for *AGV fleet coordination*. Receivers can choose which BIS to subscribe to, allowing, for example, some devices to only monitor emergency stop signals.

However, BIS technology is at its infancy, and still faces several security challenges. BIS, in fact, rely on a pre-shared key (PSK) embedded as a broadcast code: this was shown to be vulnerable to attacks, as demonstrated by BISON [7]. By obtaining the public broadcast code, the attacker can overshadow the real broadcasters using higher transmission power and forge channel map updates to redirect legitimate listeners to attacker-controlled channels. We address this weakness by providing an application-layer with a key-rotation mechanism to make BIS suitable for safety-critical applications.

III. Threat Model and Requirements

A. Requirements specification

As a targeted use case, we will focus on the general mobile agents that are part of a fleet management system (FMS). The central FMS's responsibility is to enable effective coordination among agents in an industrial environment. The system architecture must be scalable, supporting the simultaneous operation of hundreds of nodes.

Balancing safety, security, and industrial efficiency introduces inherent trade-offs. While encryption and authentication enhance data confidentiality, integrity, and availability, they can also introduce computational overhead that compromises real-time performance. Furthermore, cryptographic failures due to hardware or network issues may compromise system safety. Industrial needs such as scalability and low maintenance often conflict with stringent security measures, e.g., regular key rotations recommended by NIST can increase maintenance costs and downtime [13].

Based on these factors, we derive the following safety and security requirements (SR) that we consider in our design:

- *SR1*: Establish secure communication channels ensuring confidentiality, integrity, and availability.
- *SR2*: Maintain minimal computational overhead to meet time-critical deadlines in compliance with IEC 61508.
- *SR3*: Provide system reliability corresponding to at least Safety Integrity Level (SIL) 2, as defined by IEC 61508.
- *SR4*: Support rapid deployment and minimize maintenance complexity and cost.
- *SR5*: Enable long-term reuse of securely managed cryptographic key material without degrading the system's security level.

B. Security analysis

Based on the derived requirements, we model a hypothetical scenario of an industrial service deployment using an FMS, modeled with a data flow diagram (DFD) as shown in Fig. 2. In this model, we consider three distinct environments: (i) the external system environment, which includes a *"Deployer"* device responsible for providing secure configurations, certificates, and cryptographic keys to the BLE controllers, namely the *"Broadcaster"* and the *"Receiver"*. Each BLE controller incorporates a security module containing a trusted zone referred to as *"Key Storage"*, where key material is securely stored and used to perform encryption and signing operations on messages. The *"User"* interacts with the system by sending commands to the *"Broadcaster"*, which subsequently relays these commands to the *"Receiver"*, i.e., the *"End Device"*.

We conduct the security analysis by first identifying the assets we want to protect: (A1) security configuration data, (A2) system operability, (A3) system communication data, and (A4) system reliability and timeliness. We now list potential threats (T) with their targeted assets (A) and suggested countermeasures (C), alongside any residual risks (R). We classify each threat based on the STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service,

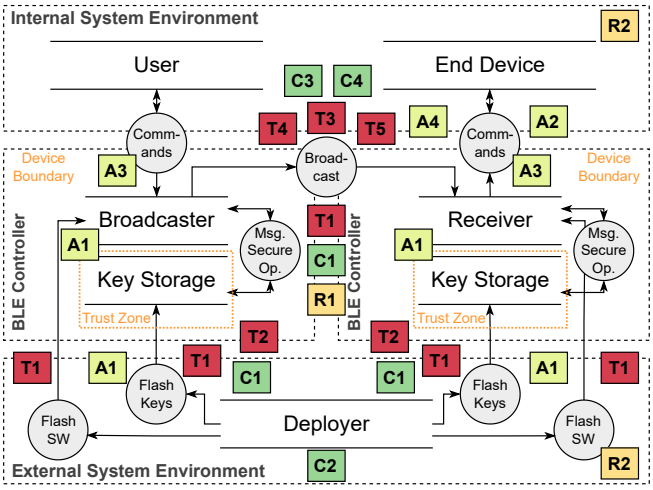

Fig. 2: Data flow diagram (DFD): system overview with core elements, data flow; assets, threats, and countermeasures points.

Elevation of privilege) threat classification [14], alongside a risk classification based on the risk assessment matrix standard [15] $\{$*Likelihood*, *Impact*, *Rating*$\}$. The values are established based on our expert analysis and the assessment of both the likelihood and probability of risk and threat occurrence. In general, we aim to put emphasis on those threats that have a high (or critical) risk rating:

- [T1] [S,T,E] MitM, spoofing, spying on the communication channels $\mapsto (A1), (A3)$; (C1) secure protocol design; (R1) software and hardware vulnerabilities.
  Risk assessment: $\{$*Unlikely*, *Major*, *High*$\}$.
- [T2] [S,T,R,I,E] Security key material compromise $\mapsto (A1), (A3)$; (C1) & (C2) deployment protocols for software; (R2) unaccounted security of operational environment & employees' access.
  Risk assessment: $\{$*Likely*, *Major*, *Critical*$\}$.
- [T3] [D] Signal interference $\mapsto (A2)$; (C3) prevent hardware access by physical means and external exposure.
  Risk assessment: $\{$*Possible*, *Major*, *High*$\}$.
- [T4] [T, D] Delay or tampering of safety-critical messages $\mapsto (A4)$; (C4) signal interference protection, e.g., with channel hopping, message resending, timestamping, and channel analysis.
  Risk assessment: $\{$*Possible*, *Catastrophic*, *Critical*$\}$.
- [T5] [S, T, E] Fake injection of safety-critical messages $\mapsto (A4)$; (C1), (C4); (R1).
  Risk assessment: $\{$*Possible*, *Major*, *High*$\}$.

Based on the results of the security and risk analysis, it can be concluded that the primary threats and corresponding countermeasures focus on protecting the command messages transmitted over the wireless channels. This is achieved through the implementation of dedicated security encryption layers designed to ensure data confidentiality and integrity. Furthermore, enhancing the system's robustness and resilience against malicious tampering with safety-critical messages, particularly obstruction or intentional delays, is crucial, as such disruptions could lead to catastrophic consequences.

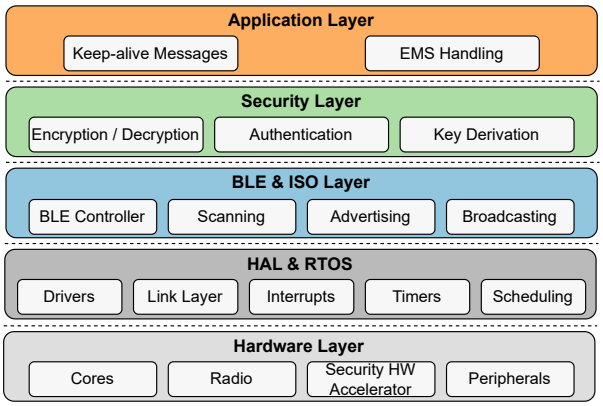

Fig. 3: Proposed layered software architecture.

IV. Secure System Architecture Framework for BLE BIS

The proposed design for our BLE BIS industrial system network is based on the requirements derived in § III. The system needs to comply with the *SR2 & SR3*, i.e., provide a low latency and high reliability (specifically, the SIL 2 aims for a PFD smaller than 10^{-3}). The required latency corresponds to the necessity of safety-function activation and is application-specific. Since we want to provide a general design, we will base our targeted assumption on the industrial experience and already existing solutions. Our design model aims to achieve a *Safety Function Response Time* (SFRT) below $1\,s$, by considering real-world FMS cases and IEC 61508 [16].

Software architecture. The software follows a layered architecture, shown in Fig. 3, as required by IEC 61508, ensuring modularity, testability, and maintainability. At the base, the *HAL & RTOS Layer* handles hardware abstraction and real-time task scheduling. Above it, the *BLE & ISO Layer* periodically transmits keep-alive messages and control commands. The key derivation inside the *Security Layer* ensures consistent and parameterized key material across all devices. The encryption block uses the security coprocessor to encrypt and decrypt messages, while the authentication block performs device authentication operations as required. At the top, the *Application Layer* enforces system safety through keep-alive handling and executes low-priority application logic such as command parsing and execution.

A. Safety design

The system architecture incorporates multiple fail-safes in line with IEC 62745 to maintain safe operation in the presence of RF interference or denial-of-service attacks [4], as shown in Fig. 4. These systems aim to fulfill two main requirements from § III-A, *SR2 & SR3*. Specifically, sent messages should not add extra overhead to interfere with the required timely executions of the safety-critical functions, and predictable hard deadlines, i.e., where a system can only be safe if safety-critical functions can be performed on time with a specific and predictable deadline. For this to work, the broadcaster transmits a *keep-alive message* every $keepalive\_time$, which

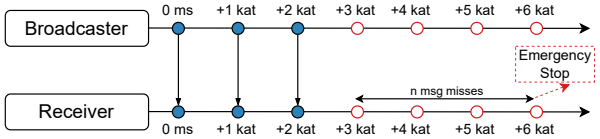

Fig. 4: Behavior of the safety design protocol: the receiver observes keep-alive messages based on the static keepalive_time (kat). If no message is received after a certain time (iterated with 'n_msg_misses'), the protocol initiates an automatic emergency stop. In this example, we have four message misses.

is defined by the application and typically measured in milliseconds. The keep-alive message is retransmitted after every $keepalive\_time$ interval to confirm active communication. If a receiver misses all *keep-alive messages* for n_msg_misses, it automatically triggers an "Emergency Stop" (EMS). The number of n_msg_misses is application-specific, but it is intended to be less than the SFRT with enough delta time in between to allow devices to halt within the mandated time. The benefit of the wireless system is that it allows personnel to stop the system via EMS remotely from anywhere.

In addition to keep-alive traffic, standard vehicle commands are broadcast on the same channel. If a user initiates an EMS, the broadcaster sends a stop command and then ceases keep-alive transmission. Thus, any device that misses the EMS command will still halt after n_msg_misses without receiving any further communication. Restarting the system requires each device to receive a dedicated reactivation command, preventing unintended reactivation following transient signal loss.

B. Security design

The security architecture balances strong protection to address the threats identified in § III and operational simplicity, targeting the simultaneous operations and deployment to multiple co-located system instances. Each independent system uses a unique symmetric master key for encryption and an asymmetric key pair for authentication, ensuring isolation between parallel deployments. We focus on mitigating the primary threat of malicious message injection.

Native BIS security was deemed insufficient and hence expanded. BIS provides no authentication, violating the system's countermeasure requirements, and it does not support runtime key rotation. Additionally, vulnerabilities in the static *Broadcast Code* identified by the BISON attack conflict with long-term secure operation [7]. To comply with key management recommendations from NIST [17], we design a security layer with custom key derivation, encryption, and authentication blocks. Specifically, we allocate the key management between the (i) broadcaster keys: asymmetric *private key*, and symmetric *master key*, and the (ii) receiver keys: asymmetric *public key* of the broadcaster, and symmetric *master key*. Furthermore, we design a *key rotation protocol* and a *synchronisation protocol* for use with BIS, shown in Figs. 5 and 6, respectively.

The key derivation protocol periodically generates a new salt S to derive fresh session keys. The broadcaster cryptographically signs the new salt with its private key (pr_{key}) and distributes it to all receivers along with a timestamp T at the time of key switching. Each receiver verifies the signature

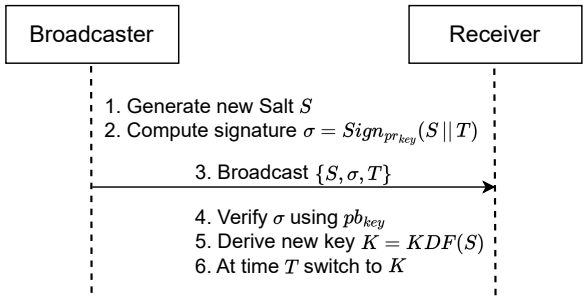

Fig. 5: Key rotation protocol: executed periodically to update session keys and maintain fresh cryptographic material, thereby mitigating attacks resulting from potential key leakage.

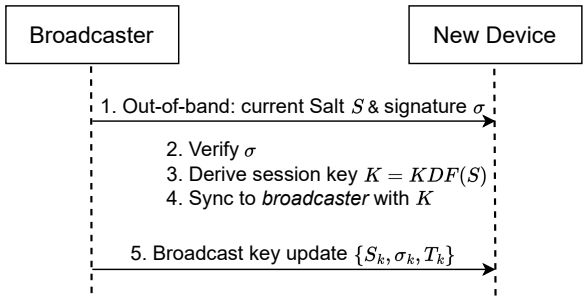

Fig. 6: Synchronization protocol: enables the system to operate reliably under the dynamic conditions of industrial environments by allowing new devices to join and automatically obtain the current session key.

using the broadcaster's public key (pb_{key}), and derives the new key K from the salt S and its *master key* using a Key Derivation Function (KDF). The receiver switches to the new key at timestamp T[1]. Using salt and timestamp, we achieve the *ephemeral attribute*, meaning that each key is uniquely derived for the dedicated session. Even if compromised, an attacker would be able to interact only with the current session, not with prior sessions that predate the key leak. New devices synchronize by obtaining the currently valid signed salt S and the signature from the broadcaster via an out-of-band channel, such as private company networks or work phones and stations. They verify its authenticity, derive the active session key, and subsequently participate in regular salt updates to maintain continuous key rotation.

V. Proof-of-Concept Implementation

To showcase the viability of our proposed design model, we implement the full system using off-the-shelf devices and software stacks. The chosen implementation system was aimed to fulfill all safety and security requirements specified in § III.

Hardware platform. The system uses the nRF5340 from Nordic Semiconductor, selected for its industrial-grade reliability and extensive development ecosystem. It supports Bluetooth 5.4, including BLE BIS. The integrated *CryptoCell-312* provides hardware-accelerated cryptography, a true random number generator (TRNG), and secure key management. Combined with Arm TrustZone, it enables isolated secure function execution and protected key storage.

Software stack. The system software is built on the Zephyr RTOS via the Nordic SDK. An RTOS was required to ensure deterministic scheduling, low latency, and predictable deadlines for safety-critical tasks. Zephyr's ongoing IEC 61508 SIL 3 certification aligns with the system's SIL 2 requirement and provides native BLE and security APIs.

For encryption, the ChaCha20 cipher was selected due to its high performance and for fulfilling the key management guidance [17]. For authentication operations, the elliptic curve-based digital signature algorithm with a 256-bit curve was chosen. While these algorithms have been selected for a prototype implementation, it is worth noting that any security primitives can be used instead, as long as they meet all the cryptographic and performance design requirements.

Wireless technology. Communication is based on BLE BIS. BIS were selected because they natively provide timestamping, duplicate rejection, and bounded latency, which directly address repetition, re-sequencing, and delay errors required for functional safety specified by IEC 61508. A built-in CRC ensures message integrity, and configurable retransmissions mitigate packet loss under interference. In this work, we use one BIG with a single BIS containing all relevant safety data.

A. BLE ISO layer and protocol

As already established, we use the BLE BIS to transmit commands and periodically send a *keep-alive message*. In one close industrial network, we distinguish between a broadcaster device that disseminates all commands, while all other devices act as receivers. BIS relies on periodic advertising to distribute BIG information and allow receivers to synchronize with the broadcast. The advertising packets also include a signed KDF salt, enabling new devices to derive the current session key. The advertising is configured as non-connectable and non-scalable to reduce the load on the advertising node, which could otherwise impact latency and reliability.

For the proof-of-concept implementation, a single broadcast isochronous stream is used. The broadcast parameters are selected to meet real-time safety constraints (listed in § VI-A), balancing latency and update frequency. The additional BIS features *Packing* and *Framing* are disabled to simplify the transmission pipeline and ensure ordered message delivery, in accordance with the relevant functional safety requirements. Safety-related handling at the protocol level follows the design outlined in § IV-A.

Messages transmitted over the broadcast channel follow a compact and extensible structure. Each message contains a header with versioning information, control flags, and metadata required for secure processing at the receiver. The payload length may vary depending on the message type. Security-relevant fields required for cryptographic processing are included as part of the header, while the payload itself is encrypted and authenticated. Integrity and ordering guarantees are provided by the underlying BIS mechanism, eliminating the need for additional application-layer countermeasures.

[1]As part of the proof-of-concept implementation, each receiver derives the timestamp by counting ISO Intervals. In an actual implementation, a clock synchronization algorithm is preferable.

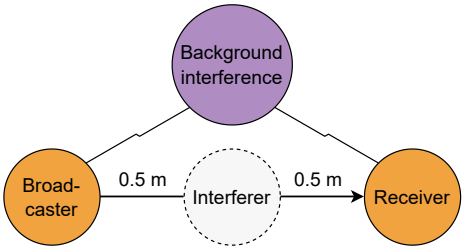

Fig. 7: Experimental setup with optional additional interference.

VI. Evaluation

In this section, we perform test measurements and performance analysis using the proof-of-concept implementation system introduced in § V. We analyse the feasibility of our system under deployed conditions and in the presence of RF interference, focusing on latency and reliability parameters as the important requirements for our safety-critical use cases.

A. Experimental setup

All test measurements were performed with 1,000 packets. The tests were repeated three times, taking the mean results. For all our tests, we used two nRF5340 DKs, one acting as a *broadcaster*, while the other board acted as the *receiver*. They are configured with 2M PHY mode and 0 dBm transmission power. For the BIG configuration, we use a BIS interval of 250 ms, with a maximum latency of 20 ms. The devices were set up in an office environment, one meter apart, so some background RF interference from surrounding devices was possible. All timing measurements were done using precise observation via GPIO pins and a Saleae Logic 8 logic analyzer. For reliability tests, an nRF7002-DK was used as a dedicated interferer device. It was placed between the two test devices, as illustrated in Fig. 7.

The interferer transmits random data at 20 dBm on Wi-Fi channel 1 with a data rate of 6 Mbit/s. We generate three classes of interference based on their intensity:

- *Low interference:* 150-byte packets with a gap of $4\,ms$, blocking approx. 5% of the Wi-Fi channel.
- *Medium interference:* 800-byte packets with a gap of $1.3\,ms$, blocking approx. 45% of the Wi-Fi channel.
- *High interference:* 4000-byte packets with a gap of $0.6\,ms$, blocking approx. 90% of the Wi-Fi channel.

B. Security overhead

To determine the computational overhead introduced by the proposed security layer on top of the baseline BIS communication, we first measure the execution time of the added security operations. We want to examine the trade-off between security strength and the safety compliance imposed by the functional safety standards.

The results in Table I show that symmetric encryption and decryption run during the general session communication introduce negligible overhead, as the CryptoCell-312 hardware accelerator efficiently handles these operations. In particular, ChaCha20-Poly1305 provides highly deterministic and predictable execution times, making it well-suited for safety-critical applications. In contrast, digital signature generation and verification introduce significant overhead, due to the inherent computational complexity of elliptic-curve cryptography. These results justify the system design decision to sign only critical control messages, rather than all transmitted data, in order to preserve low latency while maintaining strong security guarantees for safety-relevant actions.

TABLE I: Execution times of the operations in the Security Layer.

Operation (in ms)	Mean	Std. Dev.	Min.	Max.
Encryption	0.814	0.001	0.813	0.815
Decryption	0.844	0.001	0.843	0.854
Signing	16.398	0.201	15.786	16.896
Verification	22.449	0.312	21.437	23.110

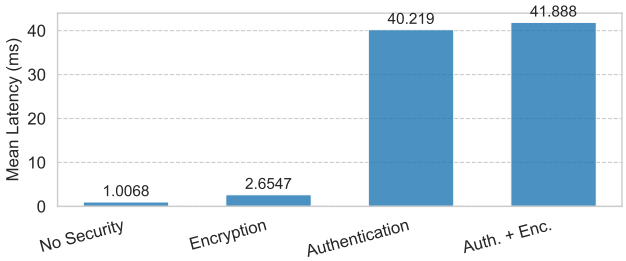

Fig. 8: Optimal latency for different security modes.

C. Latency analysis

We now measure the latency of a signed message broadcast from the broadcaster to the receiver to determine the full communication overhead. The tests are done with four security configuration options, with different operations and messages:

- *No Security*: plain mode, no encryption or authentication of the messages, i.e., no nonce or signature appended.
- *With Encryption*: only the encryption for the broadcast session is enabled. Messages now contain a nonce, increasing their payload size.
- *With Authentication*: only the authentication layer is enabled, with encryption disabled. Messages are sent unencrypted, but contain a digital signature.
- *Authentication + Encryption*: both the encryption and authentication layers are enabled, with full message sizes.

Optimal latency. We first analyze the latency under an idealistic system in which newly generated data are always ready before the next BIG anchor point, i.e., with no waiting time. We show the results in Fig. 8. Under these conditions, the encryption layer introduces only negligible latency. In contrast, the authentication layer introduces a significantly higher delay, resulting in an optimal message latency of approximately $42\,ms$. While this overhead appears substantial in isolation, it does not reflect real-world operation, as EMS events are not synchronised with the BIS interval. Therefore, we need to perform additional end-to-end latency measurements.

End-to-end latency. Since BIS transmission only occurs at the discrete BIG anchor points (in our configuration set at

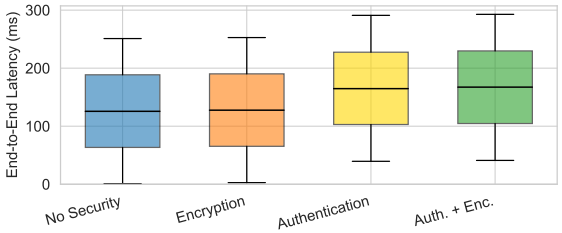

Fig. 9: End-to-end latency of a full BIS message for different security modes.

250 ms), an EMS event may occur at any time within a BIS interval. After encryption and authentication, the system must wait until the next anchor point before transmission. We evaluated the end-to-end latency by randomly triggering EMS events and measuring the time until reception at the receiver. Fig. 9 illustrates the results. Without security, or with encryption only, latency ranged from $0.2\,ms$ to $252\,ms$, with 50% of values between $65\,ms$ and $191\,ms$. With full authentication and encryption enabled, latency increased to a range of $42\,ms$ to $293\,ms$, whereas in 50% of cases, latency was between $104\,ms$ and $230\,ms$. The worst-case latency measured was $292.737\,ms$, occurring when the EMS event coincided exactly with a BIG anchor point. Despite the increased latency fluctuations from the BIS interval and security processing, the system satisfies the targeted safety requirements and SFRT of less than $1\,s$, discussed in § IV.

BIS latency. We analyze next the additional latency overhead that BIS adds to the measured end-to-end latency. To evaluate the latency of BIS added by Zepyhr and the over-the-air time, we measure the time from packet processing at the controller through wireless transmission until the receiver's application layer begins processing the packet. Table II shows the results. We can see that the latency introduced by BIS is consistently low in our configuration. For the complete security layer preparation of adding nonce, tag, and signature to the transmission, BIS adds approximately $0.4\,ms$ to the mean transmission latency. This overhead is negligible compared to the cryptographic processing and BIS scheduling time. Importantly, the latency deviates in a predictable manner, which is essential for meeting strict real-time deadlines.

D. Reliability analysis

Wireless communication in industrial environments is subject to electromagnetic interference, which may arise from neighboring devices or deliberate jamming attacks. Here, we aim to evaluate how effectively our system maintains its reliability and safety functions under adverse conditions. All experiments were conducted with the full security layer enabled to reflect real-world deployment.

We use the setup with an interferer discussed in § VI-A, with the various classes of RF interference. We observe the results in Fig. 10. Each dot represents the number of successfully received packets after 100 packet transmissions with no retransmissions. Under low-interference conditions, the system maintains a continuous 100% reliability. Under medium interference, reliability drops to 96%. With high interference, which affects approximately 90% of all BLE channels, reliability degrades significantly, reaching a minimum of 59%.

TABLE II: Transmission latency of BIS for each security mode.

Mode (in ms)	Mean	Std. Dev.	Min.	Max.
No Security	1.01	0.01	0.99	1.02
Encryption	1.06	0.01	1.04	1.08
Authentication	1.33	0.01	1.29	1.33
Auth. + Enc.	1.38	0.01	1.33	1.38

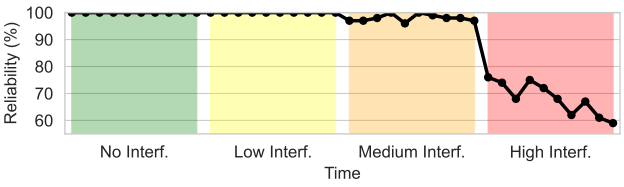

Fig. 10: Reliability over time with different interference classes.

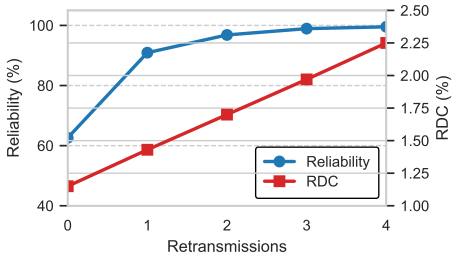

Fig. 11: Reliability and RDC in relation to the number of retransmissions.

For the safety requirements, our target is the reliability of over 99% under medium interference with low energy consumption. We want to determine the appropriate number of BLE link-layer retransmissions as a trade-off between the communication reliability and the increased radio activity. To quantify radio usage, we define radio-duty cycle (RDC) as the percentage of time the BLE radio is active within one-second intervals. We want to minimize the RDC because radio activity directly impacts energy consumption. We broadcast 1,000 packets and count only those that are also received and decrypted successfully under medium interference.

As shown in Fig. 11, increasing the number of retransmissions leads to a linear increase in RDC and a noticeable improvement in reliability. For this setup, we settled on four link-layer retransmissions, since we achieve an average reliability of 99.53% and an RDC of 2.25%, which meet our *SR2 & SR3* requirements at a reasonable trade-off.

These results demonstrate that BIS remains a highly robust broadcast mechanism under realistic industrial noise conditions. Even under high interference, a substantial number of packets are still delivered successfully. If the interference exceeds the capabilities of BIS to transmit data reliably, the implemented safety mechanism would correctly transition the system into a safe shutdown state after a fixed number of missed keep-alive packets.

VII. Related Work

Wireless communication offers key advantages over wired systems, but industrial settings impose challenges w.r.t. safety, reliability, and security, each calling for research [18].

Widely used industrial protocols. Two widely used protocols in wired and wireless industrial settings are WirelessHART [5] and OPC UA [6]. WirelessHART offers a wired industrial protocol over IEEE 802.15.4 using TDMA time-slot scheduling and interference mitigation via link-quality-driven channel hopping, enabling up to 99.999% reliability with integrated security. Although widely deployed, it is primarily designed for communication among a limited number of nodes rather than broadcast communication, such as with BLE BIS. OPC UA is a more general protocol intended for both large-scale systems and embedded industrial devices. Since it does not define a physical layer, designers can select technologies such as Wi-Fi or 5G and integrate security extensions as needed. However, OPC UA introduces larger protocol headers and multi-step session and security handshakes, increasing configuration complexity for safety-critical applications [19].

BLE safety. Several studies have examined the suitability of traditional BLE for safety-critical industrial applications. Rondon et al. [20] argued that BLE meets theoretical requirements due to its high reliability and low latency. Pang et al. [21] analyzed the trade-off between reliability and throughput, offering insights for defining safety-critical parameters such as EMS responsiveness. From a practical perspective, Park et al. [22] developed a BLE-based safety detection system for warehouse environments, while Gomez et al. [23] implemented a BLE solution to verify proper use of personal protective equipment. Both studies demonstrated BLE's reliability and energy efficiency in safety applications. However, despite this attention, research specifically addressing safety applications using BIS remains limited, a gap that this work addresses.

BLE security. Security in BLE systems has been widely investigated, with proposed architectures addressing both usability and lightweight constraints [8], [9]. Yet, BLE continues to face practical vulnerabilities, as shown by attacks such as KNOB [24] and BlueMirror [25]. BIS applications share these concerns and further require mitigations for the vulnerabilities identified in BISON [7]. BACON [26] addresses several of BISON's limitations, but does not yet meet the needs of highly complex systems that require discrete key updates and robust authorization mechanisms. Instead of adopting complex and system-specific approaches, we propose a simple yet maintainable solution that preserves both safety and security.

VIII. Conclusion and Future Work

In this work, we have presented a secure application-layer protocol built on top of BLE BIS to enable its use in safety-critical systems. The proposed model is derived from industrial requirements based on IEC 61508 and IEC 62745 standards. The implemented system achieves a high reliability while ensuring system security and safety, satisfying industrial conditions, and enforcing a fail-safe system control strategy. Experiments demonstrate a mean end-to-end response time of $167.320\,ms$, with the security layer increasing the overall system latency by 33.15%, while addressing BIS security limitations and providing countermeasures. This work demonstrates that BLE BIS, when combined with appropriate safety and security mechanisms, is a viable wireless technology for safety-critical industrial systems. Future work includes: (i) a more comprehensive evaluation at larger distances between devices and more challenging RF environments, e.g., with nearby operating machines, harsher RF interference, and stronger multi-path fading effects; (ii) a practical verification of the system's compliance with SIL 2 requirements; (iii) the extension of the system design to support bidirectional communication (e.g., for telemetry data and system monitoring); and (iv) the use of multiple BIS to separate safety and control data.

References

[1] M. Golovianko *et al.*, "Industry 4.0 vs. Industry 5.0: Co-existence, Transition, or a Hybrid," *Procedia Computer Science*, vol. 217, 2023.

[2] M. Ghobakhloo, "Industry 4.0, digitization, and opportunities for sustainability," *Journal of Cleaner Production*, vol. 252, 2020.

[3] IEC, "IEC 61508-1," 2022. https://webstore.iec.ch/en/publication/5515.

[4] IEC, "IEC 62745:2017 — Safety of Machinery – Requirements for Cableless Control Systems of Machinery," 2017.

[5] IEC, "Industrial communication networks – Wireless communication network and communication profiles – WirelessHART," 2016. https://webstore.iec.ch/en/publication/24433.

[6] OPC Foundation, "OPC Unified Architecture." https://opcfoundation.org/about/opc-technologies/opc-ua/, 2008.

[7] T. Gasteiger, C. Boano, and K. Römer, "BISON: Attacking Bluetooth's Broadcast Isochronous Streams," in *Proceedings of the 20th EWSN Conf.*, ACM, 2023.

[8] M. Cäsar *et al.*, "A survey on Bluetooth Low Energy security and privacy," *Computer Networks*, vol. 205, 2022.

[9] E. Kalinin *et al.*, "IoT Security Mechanisms in the Example of BLE," *Computers*, vol. 10, no. 12, 2021.

[10] Bluetooth SIG, "Bluetooth Core Specification, v6.0." 2024.

[11] A. Seferagić *et al.*, "Survey on Wireless Technology Trade-Offs for the Industrial Internet of Things," *Sensors*, vol. 20, no. 2, 2020.

[12] A. M. Zanchettin *et al.*, "Safety in human-robot collaborative manufacturing environments: Metrics and control," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, 2016.

[13] V. Lesi *et al.*, "Integrating Security in Resource-Constrained Cyber-Physical Systems," *ACM Trans. Cyber-Phys. Syst.*, vol. 4, no. 3, 2020.

[14] OWASP, "Threat Modeling Process." https://owasp.org/www-community/Threat_Modeling_Proces, 2025. Accessed: 03.11.2025.

[15] AcqNotes LLC., "Risk Assessment Matrix – Overview." https://acqnotes.com/acqnote/tasks/risk-reporting-matrix, 2024. Accessed: 03.11.2025.

[16] "Overview Of IEC 61508 & Functional Safety," tech. rep., IEC, 2022.

[17] E. Barker and W. Barker, "Recommendation For Key Management," tech. rep., National Institute Of Standards And Technology, 2018.

[18] F. Foukalas *et al.*, "Dependable Wireless Industrial IoT Networks: Recent Advances and Open Challenges," in *Proc. of IEEE ETS*, 2019.

[19] A. Burger *et al.*, "Bottleneck Identification and Performance Modeling of OPC UA Communication Models," in *Proc. of the ICPE Conf.*, 2019.

[20] R. Rondón, M. Gidlund, and K. Landernäs, "Evaluating Bluetooth Low Energy Suitability for Time-Critical Industrial IoT Applications," *International Journal of Wireless Information Networks*, vol. 24, 2017.

[21] B. Pang *et al.*, "Modeling the Trade-off between Throughput and Reliability in a Bluetooth Low Energy Connection," in *Proceedings of the 20th EWSN Conf.*, 2023.

[22] J. Park, Y. Cho, and S. Timalsinac, "Direction Aware Bluetooth Low Energy Based Proximity Detection System for Construction Work Zone Safety," in *Proceedings of the 33rd ISARC Symp.*, IAARC, 2016.

[23] J. M. G. de Gabriel *et al.*, "A Safety System based on Bluetooth Low Energy (BLE) to prevent the misuse of Personal Protection Equipment (PPE) in construction," *Safety Science*, vol. 158, 2023.

[24] D. Antonioli, N. O. Tippenhauer, and K. B. Rasmussen, "The KNOB is Broken: Exploiting Low Entropy in the Encryption Key Negotiation Of Bluetooth BR/EDR," in *28th USENIX Symp.*, USENIX Assoc., 2019.

[25] T. Claverie and J. L. Esteves, "BlueMirror: Reflections on Bluetooth Pairing and Provisioning Protocols," in *Proc. of the IEEE SPW*, 2021.

[26] T. Gasteiger *et al.*, "BACON: Improving Broadcast Audio Authentication," in *Proceedings of the IEEE MILCOM Conf.*, 2024.