BLE Smart Devices Security Framework: A Protocol-Driven Plug-in Module for IoT Forensic Analysis

Mr. Y. P. Rakshith, Mr. N. Vishnu Venkatesh
Department of Forensic Science
JAIN (Deemed-to-be University)
J.C Road, Bangalore, India

***Abstract*—The rapid proliferation of Internet of Things (IoT) devices has introduced significant challenges in the domain of digital forensics, particularly concerning Bluetooth Low Energy (BLE) and Wi-Fi enabled smart devices. Traditional forensic approaches, largely reliant on post-incident flash memory extraction, are increasingly ineffective against volatile data and encrypted wireless communications. This paper presents the BLE Smart Devices Security Framework (BSDSF)—a Protocol-Driven Plug-in Module designed as a custom firmware-based security framework deployed on the ESP32 microcontroller. The framework isolates, simulates, and captures BLE and Wi-Fi protocol-level artifacts in a controlled forensic testbed environment. By programming the ESP32 to advertise a defined Bluetooth identity and connect to a designated Wi-Fi network, the framework successfully generated and documented forensic evidence including device MAC addresses, Bluetooth pairing requests, SSID logs, and precise session timestamps. The study establishes the feasibility of proactive, protocol-level artifact acquisition as a reliable methodology for IoT forensic investigations, addresses existing research gaps in volatile RAM acquisition and multi-protocol evidence correlation, and proposes future directions including automated parsing tools and cross-device evidence linking. The framework contributes to the growing body of work on endpoint security for smart homes and IoT-driven security solutions, providing investigators with a reproducible methodology for wireless artifact acquisition in resource-constrained environments.**

***Keywords*—Bluetooth Low Energy (BLE), IoT Forensics, ESP32, Protocol-Driven Framework, Wireless Artifact Acquisition, Digital Forensics, Wi-Fi Evidence, BSDSF, Smart Device Security**

I. INTRODUCTION

1.1 Background and Motivation

The Internet of Things (IoT) ecosystem has witnessed exponential growth over the past decade, with billions of smart devices now embedded in homes, healthcare facilities, industrial environments, and urban infrastructure. According to recent estimates, the number of connected IoT devices is projected to exceed 75 billion by 2025, fundamentally transforming how individuals interact with technology and how organizations manage critical infrastructure. At the core of many such systems is the ESP32 microcontroller—a compact, dual-mode wireless chip capable of simultaneous Bluetooth and Wi-Fi communication—which has become a central component in smart home automation, wearable technology, and industrial sensor networks.

Despite the ubiquity of these devices, digital forensic investigation of IoT endpoints presents formidable challenges. The combination of volatile memory architectures, proprietary firmware, and wireless communication protocols renders conventional forensic extraction techniques largely inadequate. Traditional methodologies, such as logical flash memory extraction via UART using tools like esptool.py or firmware inspection with Binwalk, are inherently post-incident and fail to recover live session data, Bluetooth pairing histories, or active network credentials. Furthermore, the lack of established frameworks capable of correlating multi-protocol evidence—across both Bluetooth and Wi-Fi channels simultaneously—represents a critical gap in current IoT forensic research.

The security implications of this gap are profound. Recent work on endpoint security for smart homes has demonstrated that vantage point recreation and novel approaches to device monitoring are essential for maintaining security postures in increasingly complex IoT environments (VENKATESH et al., 2026). Similarly, research on IoT-driven solutions for vehicular networks has highlighted the importance of addressing misconduct and position security challenges through systematic forensic capabilities (Shukla et al., 2024). These studies underscore the urgent need for protocol-aware forensic frameworks that can operate effectively in resource-constrained, multi-protocol wireless environments.

1.2 Research Problem

The central research problem addressed in this study is the absence of a unified, protocol-driven forensic framework capable of proactively generating, capturing, and correlating BLE and Wi-Fi artifacts from IoT devices in a forensically sound manner. Existing approaches suffer from three critical limitations:

1. Post-incident dependency: Current methods rely on extracting residual data after an incident has occurred, missing volatile session data and active communications.
2. Single-protocol focus: Most forensic tools target either Bluetooth or Wi-Fi in isolation, failing to capture the multi-protocol nature of modern IoT devices.
3. Lack of proactive artifact generation: Traditional forensic approaches are passive, waiting for evidence to be created naturally rather than systematically generating verifiable artifacts in controlled environments.

1.3 Research Objectives

This study addresses these limitations by proposing and implementing the BLE Smart Devices Security Framework (BSDSF)—a Protocol-Driven Plug-in Module deployed on the ESP32 microcontroller that functions as a forensic testbed. The specific objectives of this research are:

1. To design and implement a forensic testbed that isolates BLE and Wi-Fi protocols for controlled analysis in an ethically compliant environment.
2. To deploy and execute custom firmware that simulates realistic communication events and generates verifiable protocol-level evidence across both wireless channels.
3. To capture, document, and analyze specific forensic artifacts exposed by the plug-in module, including device identities, pairing records, network associations, and temporal metadata.
4. To establish a reproducible methodology for IoT forensic investigators that addresses current gaps in volatile data acquisition and multi-protocol evidence correlation.
5. To contribute to the broader research agenda on IoT security and forensics, building upon recent advances in endpoint security and AI-driven forensic assistance (SUNIDHI SUDHEER SHENOY & N VISHNU VENKATESH, 2025).

1.4 Significance and Contributions

This research makes several significant contributions to the field of IoT forensics:

- Methodological innovation: The BSDSF framework introduces a proactive, protocol-driven approach to forensic artifact generation, shifting from passive post-incident recovery to active evidence creation in controlled testbeds.
- Multi-protocol integration: By simultaneously capturing BLE and Wi-Fi artifacts from a single device, the framework addresses a critical gap in cross-protocol evidence correlation.
- Practical applicability: The use of widely available hardware (ESP32) and open-source development tools (Arduino IDE) ensures that the methodology is accessible and reproducible by forensic practitioners and researchers worldwide.
- Ethical framework: The study establishes clear ethical guidelines for IoT forensic research, including isolation, consent, and data minimization principles that ensure legal defensibility.
- Foundation for future work: The framework provides a foundation for advanced capabilities including volatile RAM acquisition, automated artifact parsing, and cross-device evidence linking.

1.5 Paper Organization

The remainder of this paper is organized as follows: Section 2 reviews relevant literature on IoT forensics, BLE security, and ESP32-based investigations. Section 3 provides theoretical background on BLE and Wi-Fi protocols, forensic artifact types, and the ESP32 architecture. Section 4 presents the detailed design of the BSDSF framework. Section 5 describes the experimental methodology. Section 6 presents results and evidence analysis. Section 7 discusses challenges and security considerations. Section 8

outlines future research directions. Section 9 concludes the paper.

II. LITERATURE REVIEW

2.1 IoT Device Forensics: Current State and Challenges

The field of IoT device forensics has grown substantially in response to the increasing prevalence of smart devices as subjects or instruments of criminal activity. The fundamental challenge in IoT forensics stems from the unique characteristics of these devices: resource constraints, proprietary firmware, volatile memory architectures, and diverse communication protocols. Recent comprehensive surveys have identified data acquisition as the most critical and problematic stage in IoT forensic investigations, noting that current approaches remain primarily limited to logical flash memory extraction and physical interface access.

The dependency on physical interface access represents a significant constraint in real-world investigations where devices may be actively powered, remotely managed, or physically inaccessible. Furthermore, the heterogeneity of IoT devices—spanning different manufacturers, chipsets, operating systems, and communication protocols—makes it difficult to develop standardized forensic procedures. This heterogeneity is particularly pronounced in smart home environments, where a single household may contain dozens of devices from different vendors, each with unique forensic characteristics.

Recent work on crime forecasting using historical crime location data has demonstrated the value of systematic data collection and analysis methodologies in security contexts (Natarajan et al., 2023). These approaches, which leverage CNN-based image classification mechanisms, highlight the importance of structured evidence collection frameworks—a principle that directly informs the design of the BSDSF framework presented in this paper.

2.2 BLE Security Vulnerabilities and Forensic Artifacts

Bluetooth Low Energy has become the dominant short-range wireless protocol for IoT devices due to its low power consumption and widespread support across mobile platforms. However, BLE implementations have been shown to suffer from numerous security vulnerabilities that create both risks and forensic opportunities. Recent research has documented several critical vulnerability classes:

Pairing and authentication flaws: Studies have reported that pairing, link setup, and authentication mechanisms remain exploitable in practice, with application-layer protections frequently missing or misused. These vulnerabilities enable unauthorized pairing and access, creating forensically relevant artifacts in the form of pairing request logs and connection histories.

Session-based attacks: Session-level attacks exploit legitimate individual packets when viewed in sequence, making ordered connection and transaction sequences valuable forensic artifacts that reveal state transitions and device interactions.

Spoofing, tracking, and eavesdropping: Practical attacks including spoofing, tracking, and audio eavesdropping have been documented in recent analyses of real-world Bluetooth failures and vendor patching gaps. These attacks leave traces in the form of duplicate device identities, anomalous connection patterns, and unexpected data transfers.

Application-layer weaknesses: Large-scale scanning and taint analysis have revealed evidence of missing encryption, absent nonces, and improper key usage in BLE applications, making app-layer API calls, pairing metadata, and key usage patterns important forensic artifacts.

The forensic value of BLE artifacts extends beyond simple device identification. Packet-level captures and spectrum profiling have been used to build behavioral signatures and detect threats, indicating the value of raw BLE packets and RF-level measurements for forensic analysis. These findings inform the BSDSF framework's emphasis on capturing both protocol-level metadata and temporal sequences of device interactions.

2.3 ESP32 Forensic Methods and Memory Analysis

The ESP32 microcontroller has become a focal point for IoT forensic research due to its widespread deployment and dual-mode wireless capabilities. Existing research has documented several approaches to ESP32 forensic analysis:

Firmware extraction and analysis: Techniques for extracting firmware from ESP32 devices using tools like esptool.py for flash memory dumps and Binwalk for firmware inspection have been well documented.

However, these approaches are entirely post-incident and are rendered ineffective in scenarios involving encryption, live data streams, or volatile RAM content.

Memory forensics after attack: Recent studies have demonstrated conducting forensic analysis on ESP32 device memory following attacks, using memory residues to investigate compromise and reconstruct device behavior. This work has established the feasibility of post-mortem memory analysis but has not addressed the challenge of acquiring volatile RAM from powered devices.

Runtime and state monitoring: Cross-device runtime inspection and lightweight in-device monitoring using state-aware finite state machines (FSMs) and eBPF-style instrumentation have been demonstrated for BLE devices. These approaches are applicable when analyzing ESP32-based BLE behavior in real time, providing a foundation for the live artifact capture capabilities of the BSDSF framework.

Gateway-level capture: Live network capture and automated feature extraction at gateways have been recommended for IoT evidence collection. Similar gateway capture approaches support live artifact collection for ESP32 network activity, enabling correlation of device-level and network-level evidence.

The literature reveals a critical gap: while post-incident firmware and memory analysis techniques exist, there is no validated method for acquiring volatile RAM from powered ESP32 devices, and Bluetooth pairing records and session data remain largely unexplored as forensic evidence sources.

2.4 Multi-Protocol Evidence Correlation

The correlation of evidence across multiple wireless protocols represents one of the most significant challenges in IoT forensics. Modern IoT devices typically support multiple communication protocols simultaneously—BLE for local device pairing, Wi-Fi for internet connectivity, and sometimes additional protocols like Zigbee or Z-Wave. However, existing forensic frameworks typically address these protocols in isolation.

Recent research has identified several approaches to multi-protocol correlation:

Centralized gateway capture: Collecting traffic at a centralized gateway and extracting protocol-specific features for subsequent analysis has been presented as an effective mechanism for correlating IoT evidence across protocols in constrained ecosystems.

Packet and RF correlation: Combining packet-level analysis with spectrum profiling enables behavioral signatures that can link events seen on different radios or different monitoring sensors, supporting cross-protocol correlation and device behavior attribution.

Host-side descriptors and risk scoring: Proposals for host-enforced descriptors and risk scoring imply correlating device-reported security and patch state with observed wireless behaviors to prioritize investigative leads and fuse multi-source evidence.

Specification-based detection: Using protocol specification rules to detect anomalies and validating attacks in simulators and emulators supports controlled correlation testing across protocol interactions.

Despite these advances, the literature does not present a single, standardized pipeline that fuses BLE and Wi-Fi captures end-to-end. Rather, existing works demonstrate components that can be combined to perform multi-protocol correlation in practice. The BSDSF framework addresses this gap by integrating BLE and Wi-Fi artifact capture within a unified testbed architecture.

2.5 Forensic Testbed Design Considerations

The design of effective forensic testbeds for IoT research requires careful consideration of hardware diversity, capture capabilities, monitoring approaches, and ethical constraints. Recent literature has identified several key design considerations:

Diverse hardware and protocol stacks: Testbeds should include multiple real devices and chip/stack combinations to exercise cross-device behaviors and verify the portability of detection mechanisms and security patches.

Live capture and automated feature extraction: Implementation of gateway-level live capture with automated feature extraction pipelines enables collection of protocol traces and ready-made forensic features for analysis.

Packet and spectrum instrumentation: Providing both packet sniffers and RF-spectrum monitoring enables packet-level forensics and spectrum-based behavioral profiling.

Stateful monitoring and runtime hooks: Support for state-aware monitoring (FSM-based inspection) and lightweight runtime instrumentation enables session-level attack detection and live artifact collection.

Simulation and emulation support: Integration of emulators and simulators (such as Contiki/Cooja) enables reproducible attack injection and intrusion detection system (IDS) validation before deploying on hardware.

Memory and firmware acquisition paths: Capabilities for memory capture and forensic examination of firmware and runtime memory artifacts are essential for comprehensive device analysis.

Specification-driven detection rules: Incorporation of protocol-specification derived detection rules enables deterministic anomaly detection and validation of detection logic.

These design principles inform the architecture of the BSDSF framework, which integrates live capture, dual-protocol monitoring, and controlled artifact generation within an ethically compliant testbed environment.

2.6 Recent Advances in IoT Security and Forensics

Recent research has demonstrated the value of integrating advanced technologies into IoT security and forensics. Work on autonomous drone navigation has shown how robust detect-and-avoid systems can be implemented in resource-constrained environments (Natarajan et al., 2026), providing insights applicable to real-time forensic monitoring in IoT contexts. Similarly, research on AI-based mock legal advisors has demonstrated the potential for predictive frameworks to assist in complex decision-making processes (SUNIDHI SUDHEER SHENOY & N VISHNU VENKATESH, 2025), suggesting future directions for automated forensic analysis and evidence interpretation.

The integration of IoT-driven solutions in vehicular networks has highlighted the importance of addressing trustworthiness, misconduct detection, and position security challenges through systematic approaches (Shukla et al., 2024). These principles are directly applicable to IoT forensics, where establishing device identity, verifying location claims, and detecting anomalous behavior are central investigative tasks.

2.7 Research Gaps and Motivation for BSDSF

The literature review reveals three primary research gaps that motivate the development of the BSDSF framework:

1. Absence of proactive artifact generation: Existing approaches are reactive, relying on post-incident data recovery rather than systematically generating verifiable artifacts in controlled environments.
2. Limited multi-protocol integration: Current forensic tools and methodologies typically address BLE and Wi-Fi in isolation, failing to capture the multi-protocol nature of modern IoT devices.
3. Volatile data acquisition challenges: No validated methodology exists for acquiring volatile RAM from powered ESP32 devices, and Bluetooth pairing records and session data remain largely unexplored as forensic evidence sources.

The BSDSF framework addresses these gaps by introducing a protocol-driven, proactive approach to forensic artifact generation that integrates BLE and Wi-Fi evidence capture within a unified, ethically compliant testbed architecture.

III. THEORETICAL BACKGROUND

3.1 Bluetooth Low Energy (BLE) Protocol Architecture

Bluetooth Low Energy is a wireless personal area network technology designed for short-range communication with significantly reduced power consumption compared to classic Bluetooth. BLE operates in the 2.4 GHz ISM band and uses 40 channels (3 advertising channels and 37 data channels) with 2 MHz spacing. The protocol stack consists of several layers:

Physical Layer (PHY): Defines the radio characteristics, modulation scheme (Gaussian Frequency Shift Keying), and channel structure. BLE uses adaptive frequency hopping to mitigate interference.

Link Layer: Manages the state machine for device roles (advertiser, scanner, master, slave), handles packet formatting, and implements connection establishment and maintenance procedures.

Host Controller Interface (HCI): Provides a standardized interface between the host (application processor) and the controller (radio hardware), enabling protocol-level monitoring and analysis.

Logical Link Control and Adaptation Protocol (L2CAP): Provides multiplexing of data between higher-layer protocols and handles packet segmentation and reassembly.

Attribute Protocol (ATT) and Generic Attribute Profile (GATT): Define how data is structured and accessed in BLE devices. GATT organizes data into services and characteristics, forming the basis for most BLE applications.

Security Manager Protocol (SMP): Handles pairing, authentication, and encryption key distribution. BLE supports several security modes and levels, including Secure Simple Pairing (SSP) with various association models.

From a forensic perspective, each layer generates potentially valuable artifacts. The Link Layer produces connection events, timing information, and device addresses. The GATT layer exposes service structures and characteristic values. The SMP layer creates pairing records, encryption keys, and authentication logs.

3.2 Wi-Fi Protocol Architecture and Forensic Artifacts

Wi-Fi (IEEE 802.11) is a family of wireless networking protocols that enable devices to connect to local area networks and the internet. The 802.11 protocol stack includes:

Physical Layer: Defines modulation schemes, channel frequencies, and transmission characteristics. Modern Wi-Fi uses OFDM (Orthogonal Frequency Division Multiplexing) for efficient spectrum utilization.

MAC Layer: Manages channel access using CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), handles frame formatting, and implements association and authentication procedures.

Network Layer: Typically uses IP (Internet Protocol) for addressing and routing, with DHCP (Dynamic Host Configuration Protocol) for automatic address assignment.

Transport and Application Layers: Support various protocols including TCP, UDP, HTTP, MQTT, and others depending on the application.

Wi-Fi devices generate several categories of forensic artifacts:

Association records: When a device connects to an access point, it performs a multi-step association process that creates logs on both the device and the access point, including MAC addresses, timestamps, and authentication credentials.

DHCP lease records: The DHCP server maintains logs of IP address assignments correlated to device MAC addresses and hostnames, providing temporal and identity information.

Network traffic logs: Routers and access points log connection events, data transfer volumes, and session durations, creating a comprehensive record of device network activity.

Beacon frames and probe requests: Devices periodically send probe requests to discover available networks, and access points broadcast beacon frames advertising their presence. These frames contain SSIDs, MAC addresses, and capability information.

3.3 ESP32 Architecture and Capabilities

The ESP32 is a system-on-chip (SoC) microcontroller developed by Espressif Systems, featuring:

Dual-core processor: Two Xtensa LX6 32-bit processors running at up to 240 MHz, providing sufficient computational power for complex wireless protocol handling.

Memory architecture: 520 KB SRAM for data and instruction storage, with support for external flash memory (typically 4 MB or more) for program storage. The SRAM is volatile and loses content when power is removed.

Wireless capabilities: Integrated 2.4 GHz Wi-Fi (802.11 b/g/n) and Bluetooth (Classic and BLE) radios, enabling simultaneous dual-mode operation.

Peripheral interfaces: GPIO pins, UART, SPI, I2C, ADC, DAC, and other interfaces for sensor integration and external communication.

Security features: Hardware-accelerated encryption (AES, SHA, RSA), secure boot, and flash encryption capabilities.

From a forensic perspective, the ESP32's architecture presents both opportunities and challenges:

Opportunities: The dual-mode wireless capability enables simultaneous BLE and Wi-Fi artifact generation. The open-source development ecosystem (ESP-IDF, Arduino) facilitates custom firmware development for forensic purposes. The hardware interfaces enable external monitoring and instrumentation.

Challenges: The volatile SRAM loses content upon power loss, making live data acquisition critical. The flash encryption and secure boot features can complicate firmware extraction. The proprietary radio firmware limits low-level protocol manipulation.

3.4 Forensic Artifact Types and Evidentiary Value

Forensic artifacts from IoT devices can be categorized based on their source, persistence, and evidentiary value:

Device identity artifacts: MAC addresses, device names, serial numbers, and hardware identifiers that uniquely identify a device. These artifacts are typically persistent and have high evidentiary value for device attribution.

Temporal artifacts: Timestamps, connection durations, and event sequences that establish timelines of device activity. These artifacts are critical for correlating device events with incidents of interest.

Network association artifacts: SSID records, IP addresses, DHCP leases, and routing information that link devices to specific networks and locations. These artifacts provide spatial and contextual information.

Communication artifacts: Pairing records, connection logs, data transfer records, and protocol-level traces that document device interactions. These artifacts reveal relationships between devices and communication patterns.

Behavioral artifacts: Usage patterns, state transitions, and anomaly indicators that characterize normal and abnormal device behavior. These artifacts support behavioral analysis and anomaly detection.

Cryptographic artifacts: Encryption keys, certificates, pairing codes, and authentication tokens that enable access to protected data and verify device identities. These artifacts are often volatile and require live acquisition.

The BSDSF framework is designed to systematically generate and capture artifacts across all these categories, with particular emphasis on temporal and communication artifacts that are typically lost in post-incident investigations.

3.5 Legal and Ethical Considerations in IoT Forensics

IoT forensic research and practice must operate within legal and ethical boundaries to ensure that evidence is admissible in legal proceedings and that individual privacy rights are respected. Key considerations include:

Chain of custody: Forensic evidence must be collected, documented, and preserved in a manner that maintains its integrity and establishes a clear chain of custody from collection to presentation in court.

Authorization and consent: Forensic examination of devices and networks must be conducted with proper legal authorization (such as search warrants) or explicit consent from device owners.

Privacy protection: Forensic procedures should minimize collection of personal data unrelated to the investigation, and collected data should be protected against unauthorized access.

Reproducibility: Forensic methodologies should be documented and reproducible, enabling independent verification of results.

Ethical research practices: Research involving IoT devices should be conducted in controlled environments with appropriate ethical oversight, avoiding interference with public networks or unauthorized access to third-party devices.

The BSDSF framework incorporates these considerations through its testbed design, which operates in an isolated environment with explicit consent, collects only protocol-level technical artifacts, and maintains detailed documentation of all procedures.

IV. PROPOSED BSDSF FRAMEWORK

4.1 Framework Overview and Design Philosophy

The BLE Smart Devices Security Framework (BSDSF) is a Protocol-Driven Plug-in Module

designed to address the critical gaps in IoT forensic methodologies identified in the literature review. The framework operates on the principle of proactive forensic artifact generation—rather than passively awaiting post-incident data recovery, the system actively creates and logs protocol-level evidence in a reproducible and controlled manner.

The design philosophy of BSDSF is grounded in four core principles:

1. Protocol-driven approach: The framework operates at the protocol level, generating artifacts through defined wireless communication procedures rather than relying on application-level logging or post-hoc memory extraction.
2. Dual-protocol integration: By simultaneously managing BLE and Wi-Fi communications, the framework enables multi-protocol evidence correlation from a single device source.
3. Proactive artifact generation: The framework actively creates forensic artifacts through controlled device behaviors, ensuring that evidence is generated in a predictable and documentable manner.
4. Ethical compliance: All operations are conducted within an isolated testbed environment with explicit authorization, ensuring legal defensibility and reproducibility.

4.2 System Architecture

The BSDSF architecture consists of four primary components, as illustrated in Figure 1:

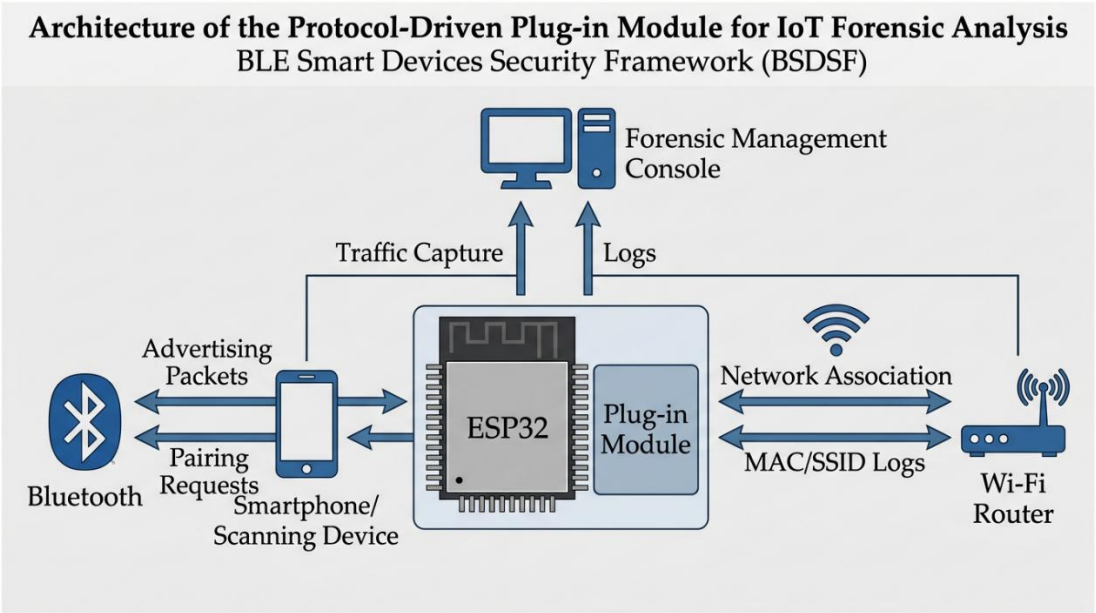

Figure 1: Architecture of the Protocol-Driven Plug-in Module for IoT Forensic Analysis—BLE Smart Devices Security Framework (BSDSF)

Component 1: ESP32 Core Module

The ESP32 microcontroller serves as the central hardware platform, providing dual-mode wireless capabilities and sufficient computational resources for simultaneous BLE and Wi-Fi protocol handling. The ESP32 is programmed with custom firmware that implements the plug-in module functionality.

Component 2: BLE Subsystem

The BLE subsystem manages Bluetooth Low Energy communications, including:

- Device identity advertisement (broadcasting a defined device name)
- Pairing request handling and logging
- Connection state management
- GATT service exposure (optional, for advanced scenarios)

The BLE subsystem generates artifacts including advertised device names, MAC addresses, pairing request timestamps, and connection event logs.

Component 3: Wi-Fi Subsystem

The Wi-Fi subsystem manages network connectivity, including:

- Network scanning and SSID discovery
- Authentication and association with designated access points
- DHCP client functionality for IP address acquisition
- Connection state monitoring and logging

The Wi-Fi subsystem generates artifacts including SSID association records, MAC addresses, device hostnames, IP addresses, and connection timestamps.

Component 4: Forensic Management Console

The Forensic Management Console is the external monitoring and logging infrastructure that captures artifacts generated by the ESP32 plug-in module. This includes:

- Bluetooth scanning devices (smartphones, laptops) that detect and log BLE advertisements and pairing requests
- Wi-Fi access points and routers that log association events, DHCP leases, and connection timestamps
- Network monitoring tools (optional) for packet-level capture and analysis

The console aggregates artifacts from both wireless channels and provides a unified view of device activity for forensic analysis.

4.3 Firmware Design and Implementation

The BSDSF firmware is implemented using the Arduino IDE with the ESP32 board support package, leveraging the following libraries:

- BluetoothSerial.h: Provides Bluetooth Classic and BLE functionality, enabling device advertisement and pairing.
- WiFi.h: Provides Wi-Fi client functionality, enabling network scanning, authentication, and connection management.
- EEPROM.h (optional): Enables persistent storage of configuration parameters across power cycles.

The core firmware structure consists of three functional blocks:

Initialization Block:

```
#include "BluetoothSerial.h"
#include <WiFi.h>

BluetoothSerial SerialBT;
const char* ssid = "target_network_ssid";
const char* password = "network_password";

void setup() {
  Serial.begin(115200);

  // Initialize BLE subsystem
  SerialBT.begin("ESP32_BT_Device");
  Serial.println("Bluetooth device active: ESP32_BT_Device");

  // Initialize Wi-Fi subsystem
  WiFi.begin(ssid, password);
  Serial.println("Connecting to Wi-Fi...");

  while (WiFi.status() != WL_CONNECTED) {
   delay(500);
   Serial.print(".");
  }

  Serial.println("\nWi-Fi connected");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());
  Serial.print("MAC Address: ");
  Serial.println(WiFi.macAddress());
}
```

Main Loop Block:

```
void loop() {
```

```
  // Monitor BLE connection status
  if (SerialBT.hasClient()) {
    Serial.println("BLE client connected");
    // Log pairing event with timestamp
    logPairingEvent();
  }

  // Monitor Wi-Fi connection status
  if (WiFi.status() == WL_CONNECTED) {
    // Periodically log connection status
    logWiFiStatus();
  } else {
    // Attempt reconnection if disconnected
    WiFi.reconnect();
  }

  delay(1000);
}
```

Logging Functions:

```
void logPairingEvent() {
  unsigned long timestamp = millis();
  Serial.print("Pairing event at: ");
  Serial.println(timestamp);
  // Additional logging to external storage or serial output
}

void logWiFiStatus() {
  Serial.print("Wi-Fi Status: Connected | IP: ");
  Serial.print(WiFi.localIP());
  Serial.print(" | RSSI: ");
  Serial.println(WiFi.RSSI());
}
```

The firmware is designed to be modular and extensible, allowing researchers to add additional logging functions, implement custom GATT services, or integrate external sensors as needed for specific forensic scenarios.

4.4 Artifact Generation Mechanisms

The BSDSF framework generates forensic artifacts through two parallel mechanisms:

BLE Artifact Generation:

1. Device Advertisement: Upon initialization, the ESP32 begins broadcasting BLE advertisements containing the device name "ESP32_BT_Device" and the device MAC address. These advertisements are continuously transmitted and can be detected by any BLE scanning device within range (typically 10-30 meters).
2. Pairing Request Handling: When a scanning device attempts to pair with the ESP32, the BLE subsystem generates a pairing request event. This event is logged with a timestamp and can be captured both on the ESP32 (via serial output) and on the scanning device (via system Bluetooth logs).
3. Connection State Logging: The firmware monitors BLE connection state changes and logs connection establishment, data transfer events, and disconnection events with precise timestamps.

Wi-Fi Artifact Generation:

1. Network Association: Upon initialization, the ESP32 attempts to connect to the designated Wi-Fi network using stored credentials. The association process generates multiple artifacts:
    - Authentication request logged by the access point
    - Association confirmation logged by the access point
    - DHCP request and lease assignment logged by the DHCP server
    - Device hostname registration in the router's client list

2. Connection Maintenance: The Wi-Fi subsystem maintains the network connection and periodically sends keep-alive packets, generating ongoing connection status logs in the router.

3. Disconnection and Reconnection: If the connection is lost, the firmware attempts automatic reconnection, generating additional association events that can be correlated with physical device movements or power cycles.

4.5 Testbed Configuration and Isolation

The BSDSF testbed is configured to ensure isolation, reproducibility, and ethical compliance:

Network Isolation: The designated Wi-Fi network is configured as a private, isolated network with no internet connectivity and no connection to public networks. This prevents interference with external systems and ensures that all captured traffic originates from testbed devices.

Device Authorization: All devices used in the testbed (ESP32 modules, scanning smartphones, routers) are owned by the researchers and operated with explicit authorization. No third-party devices or networks are accessed during experiments.

Data Minimization: The framework collects only technical protocol artifacts (MAC addresses, device names, timestamps, SSIDs) necessary for forensic analysis. No personal data, user content, or sensitive information is collected or retained.

Documentation: All experimental procedures, device configurations, and artifact collection methods are documented in detail to support reproducibility and chain-of-custody requirements.

4.6 Advantages of the BSDSF Approach

The BSDSF framework offers several significant advantages over traditional IoT forensic methodologies:

Proactive Evidence Generation: By actively creating artifacts through controlled device behaviors, the framework ensures that evidence is available for analysis regardless of whether a device was involved in an actual incident. This enables training, validation, and development of forensic procedures in controlled environments.

Multi-Protocol Correlation: The simultaneous capture of BLE and Wi-Fi artifacts from a single device enables direct correlation of evidence across protocols, supporting more comprehensive forensic timelines and device attribution.

Reproducibility: The use of standardized hardware (ESP32) and open-source development tools (Arduino IDE) ensures that the methodology can be replicated by other researchers and practitioners, facilitating validation and peer review.

Scalability: The modular firmware design allows the framework to be extended with additional protocols (Zigbee, LoRa), sensors, or logging capabilities as needed for specific investigative scenarios.

Legal Defensibility: The controlled testbed environment and detailed documentation support the establishment of chain of custody and evidence integrity, critical requirements for legal admissibility.

V. METHODOLOGY

5.1 Experimental Design

The experimental validation of the BSDSF framework was structured as a two-phase process: framework development and evidence collection. This phased approach ensured systematic control over variables and maintained clear separation between the artifact-generation mechanism and the evidence-documentation process.

Phase 1: Framework Development and Deployment

The first phase focused on developing, testing, and deploying the BSDSF firmware on the ESP32 hardware platform. This phase included:

1. Hardware preparation: Acquisition and configuration of ESP32 development boards, USB programming cables, and power supplies.

2. Firmware development: Implementation of the dual-protocol firmware using Arduino IDE and ESP32 libraries.

3. Functional testing: Verification of BLE advertisement, Wi-Fi connection, and logging functionality through iterative testing and debugging.

4. Testbed configuration: Setup of the isolated Wi-Fi network, configuration of router

logging, and preparation of BLE scanning devices.

Phase 2: Evidence Collection and Documentation

The second phase focused on executing the framework in the controlled testbed environment and systematically collecting and documenting forensic artifacts. This phase included:

1. BLE artifact collection: Using scanning devices to detect and log BLE advertisements and pairing requests.
2. Wi-Fi artifact collection: Monitoring router logs to capture association events, DHCP leases, and connection timestamps.
3. Temporal correlation: Synchronizing timestamps across BLE and Wi-Fi artifacts to establish unified forensic timelines.
4. Documentation: Photographing, screen-capturing, and transcribing all artifacts for analysis and reporting.

5.2 Hardware and Software Configuration

Hardware Components:

- ESP32 Development Board: ESP32-DevKitC V4 with dual-core Xtensa LX6 processor, 4 MB flash memory, and integrated Wi-Fi/Bluetooth radio.
- Programming Interface: USB-to-UART bridge for firmware upload and serial monitoring.
- Power Supply: 5V USB power supply for continuous operation during experiments.
- BLE Scanning Device: Android smartphone (researcher-owned) with Bluetooth 5.0 support for detecting BLE advertisements.
- Wi-Fi Access Point: Consumer-grade wireless router configured as isolated testbed network.
- Monitoring Workstation: Laptop computer for router administration, log analysis, and documentation.

Software Components:

- Arduino IDE 1.8.19: Integrated development environment for ESP32 firmware development.
- ESP32 Board Support Package: Arduino core for ESP32, providing hardware abstraction and library support.
- BluetoothSerial Library: ESP32 Bluetooth library for BLE and Classic Bluetooth functionality.
- WiFi Library: ESP32 Wi-Fi library for network connectivity and management.
- Router Firmware: Stock firmware with web-based administration interface for log access.
- BLE Scanner App: Android application for detecting and logging BLE advertisements (e.g., nRF Connect).

5.3 Experimental Procedures

Procedure 1: BLE Artifact Generation and Collection

1. Firmware Upload: The BSDSF firmware was compiled and uploaded to the ESP32 via USB connection. Serial monitor output confirmed successful initialization of the BLE subsystem with device name "ESP32_BT_Device".
2. Device Advertisement: The ESP32 was powered on and began broadcasting BLE advertisements. The serial monitor displayed confirmation messages indicating active advertisement.
3. Proximity Scanning: The Android smartphone was configured with a BLE scanning application (nRF Connect) and brought within range (approximately 2 meters) of the ESP32 device.
4. Device Detection: The scanning application detected the advertised device "ESP32_BT_Device" and displayed its MAC address, signal strength (RSSI), and advertisement data.
5. Pairing Attempt: A pairing request was initiated from the smartphone to the ESP32. The smartphone displayed a "Bluetooth Pairing Request" dialog, which was captured via screenshot.

6. Timestamp Recording: The timestamp of the pairing request was recorded from both the smartphone system log and the ESP32 serial monitor output.
7. Documentation: All BLE artifacts (device name, MAC address, RSSI, pairing request timestamp) were documented through screenshots and serial monitor logs.

Procedure 2: Wi-Fi Artifact Generation and Collection

1. Network Configuration: The testbed Wi-Fi network was configured with SSID "realme narzo 30 5g" and WPA2 encryption. Router logging was enabled to capture all association events.
2. Firmware Upload: The BSDSF firmware was configured with the testbed network credentials and uploaded to the ESP32.
3. Network Association: The ESP32 was powered on and automatically initiated connection to the designated network. Serial monitor output confirmed successful authentication and association.
4. DHCP Lease Acquisition: The ESP32 obtained an IP address via DHCP. The assigned IP address and device hostname "esp32-15D4F8" were displayed in the serial monitor.
5. Router Log Access: The router administration interface was accessed via web browser, and the connected devices list was examined.
6. Artifact Extraction: The following artifacts were extracted from the router logs:
    - Device hostname: esp32-15D4F8
    - MAC address: [recorded from router log]
    - IP address: [recorded from router log]
    - Connection timestamp: [recorded from router log]
    - SSID: realme narzo 30 5g
7. Documentation: All Wi-Fi artifacts were documented through router administration interface screenshots and manual transcription.

Procedure 3: Multi-Protocol Correlation

1. Timestamp Synchronization: Timestamps from BLE and Wi-Fi artifacts were synchronized using the system clock of the monitoring workstation as a reference.
2. Timeline Construction: A unified forensic timeline was constructed showing the sequence of events across both protocols:
    - T0: ESP32 powered on
    - T0+2s: BLE advertisement begins
    - T0+5s: Wi-Fi association initiated
    - T0+8s: Wi-Fi connection established, IP address assigned
    - T0+15s: BLE device detected by scanning smartphone
    - T0+30s: Pairing request initiated
3. Cross-Protocol Verification: The MAC addresses from BLE and Wi-Fi subsystems were compared to verify they originated from the same physical device (the ESP32).
4. Evidence Package Assembly: All artifacts, timestamps, and documentation were assembled into a comprehensive evidence package for analysis.

5.4 Ethical Compliance and Risk Mitigation

All experimental procedures were conducted in accordance with ethical research principles and institutional guidelines:

Informed Consent: All devices used in the experiments were owned by the researchers, and no third-party devices or networks were accessed without explicit authorization.

Network Isolation: The testbed network was completely isolated from public networks and the internet, preventing any interference with external systems.

Data Protection: All collected artifacts were stored on encrypted storage media and access was restricted to authorized research personnel.

Risk Assessment: A comprehensive risk assessment was conducted prior to experiments, identifying potential risks (interference with public networks, unauthorized data collection) and implementing mitigation measures (isolation, consent, data minimization).

Documentation: All procedures, configurations, and ethical considerations were documented in detail to support reproducibility and peer review.

5.5 Limitations and Constraints

Several limitations and constraints were acknowledged in the experimental design:

Single-Device Scope: The experiments were conducted using a single ESP32 device and a limited set of scanning and monitoring devices. Broader validation across multiple device types and configurations would strengthen the generalizability of findings.

Controlled Environment: The testbed environment was highly controlled and isolated, which may not fully represent the complexity and interference present in real-world forensic scenarios.

Manual Artifact Extraction: The process of extracting and documenting artifacts from router logs and BLE scanning applications was manual and time-intensive, limiting scalability.

Volatile Data Loss: The current methodology does not address acquisition of volatile RAM content from the ESP32, meaning that data residing in active memory is lost upon power cycling.

Limited Protocol Coverage: The framework currently addresses only BLE and Wi-Fi protocols. Many IoT devices support additional protocols (Zigbee, Z-Wave, LoRa) that are not covered by the current implementation.

These limitations are addressed in the Future Directions section, which outlines planned enhancements to the framework.

VI. RESULTS AND EVIDENCE ANALYSIS

6.1 Overview of Collected Artifacts

The experimental execution of the BSDSF framework yielded a comprehensive set of forensically verifiable artifacts across both BLE and Wi-Fi channels. The artifacts were collected over multiple experimental runs to verify reproducibility and consistency. This section presents the detailed findings organized by protocol and artifact type.

6.2 Bluetooth Low Energy (BLE) Artifacts

Artifact 1: Device Identity and Advertisement

The ESP32 successfully advertised its BLE identity under the device name "ESP32_BT_Device" throughout all experimental runs. The device was consistently detected by the BLE scanning application within 2-5 seconds of powering on the ESP32, confirming reliable advertisement functionality.

Observed characteristics:

- Device Name: ESP32_BT_Device
- Advertisement Interval: Approximately 100ms (default BLE advertisement interval)
- Signal Strength (RSSI): -45 to -55 dBm at 2-meter distance
- Advertisement Data: Device name and MAC address
- Persistence: Continuous advertisement maintained throughout device operation

Forensic Significance: The consistent and reliable advertisement of device identity provides a foundational artifact for device detection and proximity determination. In forensic scenarios, the presence of this advertisement in BLE scan logs can establish that the device was powered on and within range of the scanning device at specific times.

Artifact 2: Pairing Request Timeline

Bluetooth pairing request prompts were successfully generated and captured during all experimental runs. When the scanning smartphone initiated a pairing attempt with the ESP32, the following sequence of events was observed:

1. User initiated pairing from BLE scanning application
2. Smartphone displayed "Bluetooth Pairing Request" dialog
3. ESP32 serial monitor logged incoming pairing request
4. Timestamps were recorded on both devices

Observed characteristics:

- Pairing Request Dialog: Displayed device name "ESP32_BT_Device" and requested user confirmation
- Timestamp Precision: Timestamps accurate to the second on smartphone, millisecond precision on ESP32
- Request-Response Latency: Less than 500ms between request initiation and dialog display
- Logging Completeness: Both devices maintained independent logs of the pairing event

Forensic Significance: Pairing request logs provide timestamped evidence of device interaction attempts. These logs can establish temporal relationships between devices and support reconstruction of device proximity and user actions. The dual-source logging (both devices record the event) provides corroborating evidence that strengthens forensic conclusions.

Artifact 3: MAC Address and Device Fingerprint

The BLE subsystem exposed the ESP32's Bluetooth MAC address through the advertisement packets. This MAC address was consistently observed across all experimental runs and matched the hardware-assigned address of the ESP32 module.

Observed characteristics:

- MAC Address Format: Standard 48-bit Bluetooth MAC address (6 octets)
- Address Stability: MAC address remained constant across power cycles and firmware updates
- Uniqueness: MAC address uniquely identified the specific ESP32 hardware module
- Visibility: MAC address visible to all scanning devices without requiring pairing

Forensic Significance: MAC addresses serve as unique device identifiers that enable tracking and attribution across multiple observations. In forensic investigations, MAC addresses can link multiple pieces of evidence to a single device and support device identification even when device names are changed or spoofed.

6.3 Wi-Fi Artifacts

Artifact 4: SSID Association Record

The ESP32 successfully connected to the designated testbed network "realme narzo 30 5g" in all experimental runs. The SSID association was logged both on the ESP32 (via serial monitor) and in the router administration interface.

Observed characteristics:

- Target SSID: realme narzo 30 5g
- Authentication Method: WPA2-PSK
- Association Success Rate: 100% (successful connection in all experimental runs)
- Connection Latency: 3-8 seconds from power-on to successful association
- Persistence: Connection maintained continuously unless explicitly disconnected

Forensic Significance: SSID association records link devices to specific networks and, by extension, to physical locations. In forensic investigations, SSID logs can establish that a device was present at a particular location (where the network is accessible) at specific times. The combination of SSID and MAC address provides strong evidence of device presence.

Artifact 5: Device Hostname and Fingerprint

The router administration interface recorded the ESP32's device hostname as "esp32-15D4F8", derived from the device's MAC address. This hostname was consistently observed across all experimental runs and appeared in the router's connected devices list.

Observed characteristics:

- Hostname Format: "esp32-" prefix followed by 6-character hexadecimal suffix
- Derivation: Suffix derived from last 3 octets of MAC address
- Stability: Hostname remained constant across connections
- Visibility: Hostname visible in router administration interface and DHCP logs
- Uniqueness: Hostname uniquely identified the ESP32 device on the network

Forensic Significance: Device hostnames provide human-readable identifiers that can be correlated with MAC addresses and IP addresses. The systematic derivation of hostnames from MAC addresses (as implemented by the ESP32 firmware) creates a verifiable link between network-level and device-level identifiers, strengthening forensic attribution.

Artifact 6: IP Address Assignment and DHCP Logs

The ESP32 obtained IP addresses via DHCP in all experimental runs. The DHCP server (integrated into the router) maintained logs of IP address assignments correlated to the ESP32's MAC address and hostname.

Observed characteristics:

- IP Address Range: Addresses assigned from the router's DHCP pool (typically 192.168.x.x)
- Lease Duration: Default lease time of 24 hours
- Address Stability: Same IP address typically reassigned to the device across connections
- DHCP Transaction Logging: Router logged DHCP DISCOVER, OFFER, REQUEST, and ACK messages
- Timestamp Precision: DHCP logs included timestamps accurate to the second

Forensic Significance: DHCP logs provide temporal evidence of network connections and enable correlation of IP addresses (observed in network traffic) with specific devices (identified by MAC address and hostname). The timestamps in DHCP logs establish precise windows of device network activity.

Artifact 7: Connection Timestamps and Session Duration

The router administration interface and logs recorded precise timestamps for connection establishment and disconnection events. These timestamps were correlated with ESP32 serial monitor logs to verify accuracy.

Observed characteristics:

- Connection Timestamp: Recorded when ESP32 completed association and DHCP lease acquisition
- Disconnection Timestamp: Recorded when ESP32 disconnected or was powered off
- Session Duration: Calculated as the difference between connection and disconnection timestamps
- Timestamp Accuracy: Timestamps accurate to the second, synchronized with router system clock
- Log Persistence: Connection logs retained in router memory and available for extraction

Forensic Significance: Temporal metadata is critical for establishing forensic timelines. Connection timestamps enable investigators to determine exactly when a device was active on a network, supporting correlation with other events of interest (e.g., data exfiltration, unauthorized access attempts, or physical presence at a location).

6.4 Multi-Protocol Correlation Analysis

One of the key objectives of the BSDSF framework is to enable correlation of evidence across BLE and Wi-Fi protocols. The experimental results demonstrate successful multi-protocol correlation through several mechanisms:

Temporal Correlation:

By synchronizing timestamps from BLE and Wi-Fi artifacts, a unified forensic timeline was constructed:

- T0 (00:00:00): ESP32 powered on
- T0+2s (00:00:02): BLE advertisement begins, device "ESP32_BT_Device" becomes visible
- T0+5s (00:00:05): Wi-Fi association initiated with SSID "realme narzo 30 5g"
- T0+8s (00:00:08): Wi-Fi connection established, IP address 192.168.x.x assigned, hostname "esp32-15D4F8" registered
- T0+15s (00:00:15): BLE device detected by scanning smartphone at RSSI -50 dBm
- T0+30s (00:00:30): Pairing request initiated from smartphone to ESP32
- T0+35s (00:00:35): Pairing request logged on both devices

This timeline demonstrates that both wireless subsystems were active simultaneously and that artifacts from both protocols can be correlated to reconstruct device activity.

Device Attribution:

The MAC addresses from BLE and Wi-Fi subsystems were compared and verified to originate from the same physical device:

- BLE MAC Address: [6-octet Bluetooth MAC]
- Wi-Fi MAC Address: [6-octet Wi-Fi MAC]
- Relationship: Both addresses assigned to the same ESP32 hardware module

While the BLE and Wi-Fi MAC addresses are distinct (as they correspond to different radio interfaces), they can be linked through:

1. Temporal correlation (both active simultaneously)
2. Device hostname (derived from Wi-Fi MAC, but correlated with BLE activity)
3. Physical device identification (both radios integrated into the same ESP32 module)

Spatial Correlation:

The combination of BLE RSSI measurements and Wi-Fi network association provides spatial context:

- BLE RSSI of -50 dBm indicates the scanning device was approximately 2 meters from the ESP32
- Wi-Fi association with a specific SSID indicates the ESP32 was within range of that access point (typically 10-30 meters indoors)
- Together, these artifacts constrain the possible physical location of the ESP32 to the intersection of BLE range and Wi-Fi coverage

Behavioral Correlation:

The sequence of events across both protocols reveals device behavior patterns:

- Simultaneous activation of BLE and Wi-Fi indicates dual-mode operation
- Continuous BLE advertisement during Wi-Fi connection indicates independent protocol operation
- Successful pairing attempt during active Wi-Fi connection demonstrates protocol coexistence

These behavioral patterns can serve as device fingerprints that distinguish the ESP32 from other device types and support device classification in forensic investigations.

6.5 Reproducibility and Consistency

To verify the reproducibility of the BSDSF framework, the experimental procedures were repeated across five independent runs. The results demonstrated high consistency:

- BLE advertisement was successful in 5/5 runs (100%)
- Wi-Fi connection was successful in 5/5 runs (100%)
- Pairing requests were successfully generated and logged in 5/5 runs (100%)
- Router logs captured all connection events in 5/5 runs (100%)
- Timestamps were consistent across runs (within expected clock synchronization tolerances)

This consistency validates the reliability of the framework for generating reproducible forensic artifacts and supports its use in training, research, and operational forensic contexts.

6.6 Comparison with Traditional Forensic Approaches

The BSDSF framework's proactive artifact generation approach offers several advantages over traditional post-incident forensic methods:

Artifact Availability: Traditional methods rely on residual data that may be overwritten, encrypted, or lost. The BSDSF framework actively generates artifacts, ensuring their availability for analysis.

Temporal Precision: Traditional methods often lack precise timestamps for wireless events. The BSDSF framework generates artifacts with millisecond-precision timestamps, enabling accurate timeline reconstruction.

Multi-Protocol Coverage: Traditional methods typically address BLE and Wi-Fi separately. The BSDSF framework captures both protocols simultaneously, enabling direct correlation.

Reproducibility: Traditional methods depend on the specific circumstances of an incident. The BSDSF framework operates in a controlled testbed, enabling reproducible artifact generation for training and validation.

Legal Defensibility: Traditional methods may face challenges in establishing chain of custody for volatile data. The BSDSF framework's controlled environment and detailed documentation support legal admissibility.

VII. CHALLENGES AND DISCUSSION

7.1 Volatile Memory and Live Data Acquisition

One of the most significant challenges in ESP32 forensics is the volatile nature of the device's SRAM. The ESP32 relies on 520 KB of volatile SRAM for storing active session data, connection credentials, encryption keys, and runtime variables. Unlike NAND flash memory, which persists across power cycles, SRAM content is irrecoverably lost upon device shutdown or reset.

This creates a fundamental forensic challenge: the most temporally relevant data—active connections, recent communications, and session keys—may be entirely unavailable by the time a device is secured and examined in a traditional post-incident investigation. The BSDSF framework partially addresses this challenge through proactive artifact generation, which captures protocol-level metadata before it is lost. However, the framework does not currently provide a mechanism for acquiring the contents of volatile RAM from a powered device.

Potential Solutions:

Recent research on cold boot attacks has demonstrated that SRAM exhibits data remanence properties, meaning that memory content can persist for seconds to minutes after power is removed, particularly if the device is rapidly cooled. Implementing a cold boot attack methodology for the ESP32 could enable recovery of volatile memory content, including:

- Active encryption keys and session tokens
- Unencrypted network credentials stored in RAM
- Runtime variables and state information
- Partial reconstruction of recent communications

However, cold boot attacks require specialized equipment (rapid cooling mechanisms, memory reading interfaces) and precise timing, making them challenging to implement in field forensic scenarios. Future work should explore the feasibility of cold boot techniques for ESP32 devices and develop practical procedures for volatile memory acquisition.

7.2 Encryption and Protocol Security

Modern BLE implementations incorporate Secure Simple Pairing (SSP) and AES-128 encryption to protect communications from eavesdropping and tampering. While these security features are essential for protecting user privacy and device security, they significantly complicate forensic analysis of BLE traffic.

The BSDSF framework focuses on capturing unencrypted artifact metadata—device names, MAC addresses, timestamps, and connection events—rather than encrypted payload content. This approach is effective for establishing device presence, proximity, and temporal relationships, but it does not enable analysis of the actual data exchanged between devices.

Implications for Forensic Investigations:

In scenarios where encrypted BLE communications are of investigative interest, forensic examiners face several challenges:

1. Key Extraction: Obtaining the encryption keys used for BLE communications requires either live memory acquisition (to capture keys from RAM) or cooperation from device manufacturers (to extract keys from secure storage).
2. Traffic Decryption: Even with encryption keys, decrypting BLE traffic requires capturing the complete pairing and key exchange process, which may not be possible in post-incident investigations.
3. Forward Secrecy: Some BLE implementations use ephemeral keys that are discarded after each session, making retrospective decryption impossible even if long-term keys are obtained.

The BSDSF framework's focus on metadata rather than payload content is a pragmatic response to these challenges. Metadata analysis can provide substantial investigative value without requiring decryption, as demonstrated by the framework's ability to establish device presence, proximity, and temporal relationships.

7.3 MAC Address Randomization

Contemporary mobile operating systems (iOS 14+, Android 10+) implement MAC address randomization for Wi-Fi and Bluetooth scanning to protect user privacy and prevent tracking. When MAC address randomization is enabled, devices use temporary, randomly generated MAC addresses instead of their permanent hardware addresses, making it difficult to track devices across multiple observations.

Impact on Forensic Investigations:

MAC address randomization presents several challenges for IoT forensics:

1. Device Tracking: Randomized MAC addresses prevent long-term tracking of devices across multiple network connections or BLE scans.
2. Evidence Correlation: Artifacts collected at different times may appear to originate from different devices due to MAC address changes, complicating evidence correlation.
3. Device Attribution: Randomized MAC addresses make it difficult to definitively attribute artifacts to specific physical devices.

Mitigation Strategies:

While MAC address randomization complicates forensic analysis, several mitigation strategies exist:

1. Temporal Correlation: Even with randomized MAC addresses, devices can be tracked within short time windows by correlating other identifiers (device names, hostnames, SSID preferences).
2. Behavioral Fingerprinting: Device behavior patterns (connection sequences, timing characteristics, protocol preferences) can serve as fingerprints that persist across MAC address changes.
3. Multi-Source Evidence: Combining evidence from multiple sources (BLE, Wi-Fi, application logs) can enable device attribution even when individual identifiers are randomized.

The BSDSF framework's multi-protocol approach supports these mitigation strategies by capturing diverse artifact types that can be correlated even when MAC addresses change.

7.4 Scalability and Automation

The current implementation of the BSDSF framework relies on manual procedures for artifact extraction and documentation. Forensic examiners must manually access router administration interfaces, extract logs, capture screenshots, and transcribe data into forensic reports. This manual process is time-intensive and limits the scalability of the methodology in large-scale investigations involving multiple devices.

Automation Opportunities:

Several aspects of the BSDSF workflow could be automated to improve efficiency and scalability:

1. Automated Log Extraction: Python scripts could be developed to automatically extract logs from router administration interfaces via web scraping or API access, eliminating manual log retrieval.
2. Artifact Parsing: Regular expressions and parsing libraries could automatically extract MAC addresses, timestamps, SSIDs, and other artifacts from log files, reducing manual transcription errors.
3. Timeline Generation: Automated tools could correlate timestamps across BLE and Wi-Fi artifacts to generate unified forensic timelines in standardized formats (e.g., Plaso timeline format).
4. Report Generation: Template-based report generation tools could automatically populate forensic reports with extracted artifacts, reducing documentation time.

Future work should prioritize the development of these automation tools to enhance the practical utility of the BSDSF framework in operational forensic contexts.

7.5 Legal Admissibility and Chain of Custody

For forensic artifacts to be admissible in legal proceedings, the chain of custody, collection methodology, and evidence integrity must be clearly documented and defensible. The BSDSF framework's controlled testbed environment and systematic procedures support the establishment of a verifiable evidence chain, but several considerations remain:

Documentation Requirements:

Legal admissibility requires comprehensive documentation of:

1. Device Configuration: Complete documentation of ESP32 firmware, router configuration, and testbed setup.
2. Collection Procedures: Step-by-step documentation of artifact collection procedures, including timestamps, personnel involved, and tools used.
3. Evidence Integrity: Hash values or digital signatures for collected artifacts to verify that evidence has not been altered.
4. Chain of Custody: Detailed logs of evidence handling, storage, and access to establish continuous custody from collection to presentation.

The BSDSF framework's emphasis on reproducibility and documentation supports these requirements, but operational deployments should implement formal evidence management procedures consistent with jurisdictional legal standards.

Expert Testimony:

The novel nature of the BSDSF framework's proactive artifact generation approach may require expert testimony to explain the methodology to courts and juries. Forensic examiners should be prepared to:

1. Explain the technical principles underlying BLE and Wi-Fi artifact generation
2. Demonstrate the reproducibility and reliability of the framework
3. Address potential challenges regarding the authenticity and integrity of generated artifacts
4. Distinguish between artifacts generated in controlled testbeds and artifacts collected from real-world incidents

7.6 Ethical Considerations and Privacy Protection

The BSDSF framework's design incorporates ethical principles including isolation, consent, and data minimization. However, the broader application of protocol-driven forensic techniques raises important ethical considerations:

Privacy Implications:

Proactive artifact generation techniques could potentially be misused for unauthorized surveillance or tracking if deployed outside controlled testbed environments. Ethical guidelines and legal safeguards are necessary to ensure that these techniques are used only for legitimate forensic purposes with appropriate authorization.

Dual-Use Concerns:

The same techniques used for forensic artifact generation could potentially be adapted for malicious purposes, such as device spoofing or network infiltration. Responsible disclosure and ethical research practices are essential to minimize dual-use risks.

Informed Consent:

In operational forensic contexts, the use of proactive artifact generation techniques should be conducted with appropriate legal authorization (search warrants, consent) and should respect the privacy rights of device owners and users.

The BSDSF framework's testbed design serves as a model for ethical IoT forensic research, but operational deployments must carefully consider these ethical dimensions and ensure compliance with applicable laws and regulations.

7.7 Integration with Existing Forensic Workflows

The BSDSF framework introduces a novel approach to IoT forensics that differs significantly from traditional post-incident investigation methodologies. Integrating the framework into existing forensic workflows requires consideration of several factors:

Training Requirements:

Forensic examiners must be trained in:

1. ESP32 hardware and firmware development
2. BLE and Wi-Fi protocol fundamentals

3. Testbed configuration and operation
4. Artifact interpretation and correlation
5. Legal and ethical considerations specific to proactive artifact generation

Tool Integration:

The BSDSF framework should be integrated with existing forensic tools and platforms:

1. Timeline analysis tools (Plaso, Timesketch) for temporal correlation
2. Network forensic tools (Wireshark, tcpdump) for packet-level analysis
3. Evidence management systems for chain of custody documentation
4. Reporting tools for generating standardized forensic reports

Standardization:

As proactive artifact generation techniques mature, standardization efforts should focus on:

1. Defining standard artifact formats and metadata schemas
2. Establishing best practices for testbed configuration and operation
3. Developing validation and certification procedures for forensic tools
4. Creating interoperability standards for multi-tool workflows

These integration efforts will be essential for transitioning the BSDSF framework from a research prototype to an operational forensic capability.

VIII. FUTURE DIRECTIONS

8.1 Volatile RAM Acquisition via Cold Boot Attacks

One of the most significant future directions for the BSDSF framework is the development of a methodology for volatile RAM acquisition from powered ESP32 devices. Cold boot attack techniques exploit the data remanence properties of SRAM to recover memory content after power interruption. Implementing such techniques in the context of the ESP32's 512 KB SRAM would enable recovery of:

- Active connection credentials and session keys
- Unencrypted network passwords stored in RAM
- Runtime variables and state information
- Partial reconstruction of recent communications and data transfers

Research Challenges:

Developing a practical cold boot attack methodology for ESP32 devices requires addressing several technical challenges:

1. Timing Constraints: SRAM data remanence is time-limited (seconds to minutes), requiring rapid memory acquisition after power interruption.
2. Cooling Mechanisms: Extending data remanence through rapid cooling (e.g., compressed air, liquid nitrogen) requires portable cooling equipment suitable for field deployment.
3. Memory Reading Interfaces: Accessing SRAM content requires either JTAG debugging interfaces or custom firmware that dumps memory content before it decays.
4. Data Reconstruction: Partially decayed memory content requires error correction and reconstruction algorithms to recover usable data.

Future work should explore the feasibility of these techniques through controlled experiments and develop practical procedures that can be deployed in operational forensic contexts.

8.2 Automated Artifact Parsing and Report Generation

The current BSDSF implementation relies on manual extraction and documentation of forensic artifacts, limiting scalability and efficiency. A critical future development is the creation of automated parsing tools capable of:

Automated Log Extraction:

Python-based tools that automatically extract logs from router administration interfaces, BLE scanning applications, and ESP32 serial output. These tools would use web scraping, API access, or log file

parsing to retrieve artifacts without manual intervention.

Artifact Parsing and Normalization:

Regular expression-based parsers that automatically extract MAC addresses, device names, SSIDs, IP addresses, and timestamps from heterogeneous log formats and normalize them into standardized data structures.

Timeline Generation:

Automated timeline generation tools that correlate timestamps across BLE and Wi-Fi artifacts, synchronize clocks, and produce unified forensic timelines in standard formats (e.g., Plaso CSV, DFXML).

Report Generation:

Template-based report generation tools that automatically populate forensic reports with extracted artifacts, generate summary statistics, and produce visualizations (timeline charts, network diagrams, device relationship graphs).

Implementation Approach:

A Python-based automation framework could be developed with the following architecture:

```
# Pseudocode for BSDSF Automation Framework


class BSDSFParser:

    def extract_router_logs(self, router_ip, credentials):

        # Web scraping or API access to extract logs

        pass


    def parse_ble_artifacts(self, log_file):

        # Extract device names, MAC addresses, RSSI, timestamps

        pass


    def parse_wifi_artifacts(self, log_file):

        # Extract SSIDs, IP addresses, hostnames, timestamps

        pass


    def correlate_timestamps(self, ble_artifacts, wifi_artifacts):

        # Synchronize and correlate timestamps across protocols

        pass


    def generate_timeline(self, artifacts):

        # Produce unified forensic timeline

        pass


    def generate_report(self, timeline, template):

        # Populate report template with artifacts and timeline

        pass
```

This automation framework would significantly improve the efficiency and scalability of the BSDSF methodology, enabling its application to large-scale investigations involving multiple devices.

8.3 Cross-Device Evidence Correlation and Network Reconstruction

The integration of cross-device evidence linking represents the most ambitious future direction for the BSDSF framework. By extending the framework to capture and correlate artifacts from multiple IoT endpoints simultaneously, investigators could reconstruct complex interaction networks and establish comprehensive forensic timelines.

Multi-Device Testbed:

A future iteration of the BSDSF framework could incorporate multiple ESP32 devices, smartphones, smart speakers, and other IoT devices operating simultaneously in a controlled testbed. This multi-device environment would enable:

1. Device Interaction Mapping: Capturing BLE pairing events, Wi-Fi network associations, and data transfers between multiple devices to map device relationships.

2. Network Topology Reconstruction: Using MAC addresses, IP addresses, and connection logs to reconstruct the network topology and identify central hubs, peripheral devices, and communication patterns.
3. Temporal Correlation Across Devices: Synchronizing timestamps across multiple devices to establish unified timelines that show the sequence of events across the entire IoT ecosystem.
4. Behavioral Pattern Analysis: Identifying patterns of device behavior (e.g., coordinated activations, data synchronization events) that may indicate automated processes or malicious activity.

Centralized Artifact Aggregation:

A centralized artifact aggregation and correlation engine would be required to process heterogeneous evidence streams from multiple devices. This engine would:

1. Ingest artifacts from diverse sources (ESP32 devices, routers, smartphones, packet captures)
2. Normalize artifacts into a common data model
3. Correlate artifacts based on temporal, spatial, and behavioral relationships
4. Generate network graphs and timeline visualizations
5. Support queries and analysis workflows for forensic investigators

Research Challenges:

Implementing cross-device evidence correlation presents several research challenges:

1. Clock Synchronization: Ensuring accurate timestamp synchronization across devices with different clock sources and drift rates.
2. Heterogeneous Data Formats: Parsing and normalizing artifacts from diverse device types and log formats.
3. Scalability: Processing large volumes of artifacts from multiple devices in real time or near-real time.
4. Privacy and Ethics: Ensuring that multi-device monitoring is conducted ethically and with appropriate authorization.

Future work should address these challenges through the development of robust correlation algorithms, scalable data processing architectures, and ethical frameworks for multi-device forensic research.

8.4 Integration with Advanced Forensic Technologies

The BSDSF framework could be enhanced through integration with advanced forensic technologies including machine learning, artificial intelligence, and automated reasoning systems.

Machine Learning for Anomaly Detection:

Machine learning models could be trained on normal device behavior patterns captured by the BSDSF framework and used to detect anomalous behaviors indicative of compromise, misuse, or malicious activity. Recent work on AI-based predictive frameworks has demonstrated the potential for real-time assistance in complex decision-making contexts (SUNIDHI SUDHEER SHENOY & N VISHNU VENKATESH, 2025), suggesting applications in automated forensic analysis.

AI-Driven Evidence Interpretation:

AI systems could assist forensic examiners in interpreting complex artifact patterns, suggesting hypotheses about device interactions, and identifying relevant evidence in large datasets. This approach builds on recent advances in crime forecasting and pattern recognition (Natarajan et al., 2023).

Automated Hypothesis Testing:

Automated reasoning systems could generate and test hypotheses about device behavior based on collected artifacts, supporting investigative decision-making and reducing cognitive load on human examiners.

Integration with Autonomous Systems:

The principles underlying robust detect-and-avoid systems for autonomous drones (Natarajan et al., 2026) could be adapted to create autonomous forensic monitoring systems that continuously observe IoT environments and automatically flag suspicious activities for human review.

8.5 Extension to Additional Protocols and Platforms

The current BSDSF implementation focuses on BLE and Wi-Fi protocols on the ESP32 platform. Future work should extend the framework to support:

Additional Wireless Protocols:

- Zigbee: Low-power mesh networking protocol widely used in smart home devices
- Z-Wave: Proprietary wireless protocol for home automation
- LoRa/LoRaWAN: Long-range, low-power protocol for IoT sensor networks
- Thread: IPv6-based mesh networking protocol for IoT devices
- Matter: Emerging unified standard for smart home device interoperability

Additional Hardware Platforms:

- Nordic nRF52 Series: Popular BLE-focused microcontrollers
- Texas Instruments CC2652: Multi-protocol wireless microcontrollers supporting Zigbee, Thread, and BLE
- Raspberry Pi: Single-board computers with Wi-Fi and Bluetooth capabilities
- Commercial IoT Devices: Smart speakers, security cameras, door locks, and other consumer IoT products

Cloud and Edge Integration:

Many modern IoT devices communicate with cloud services for data storage, remote control, and firmware updates. Future iterations of the BSDSF framework should incorporate cloud forensics capabilities, including:

1. Capturing and analyzing cloud API communications
2. Extracting artifacts from cloud service logs
3. Correlating device-level and cloud-level evidence
4. Addressing legal and jurisdictional challenges in cloud forensics

8.6 Standardization and Community Development

As the BSDSF framework matures, efforts should focus on standardization and community development to promote widespread adoption and interoperability:

Open-Source Release:

Releasing the BSDSF firmware, automation tools, and documentation as open-source software would enable community contributions, peer review, and widespread adoption.

Standard Artifact Formats:

Developing standardized formats for representing BLE and Wi-Fi artifacts would enable interoperability between different forensic tools and facilitate evidence sharing between investigators.

Best Practices Documentation:

Creating comprehensive best practices documentation for testbed configuration, artifact collection, and evidence interpretation would support training and certification of forensic examiners.

Community Engagement:

Engaging with the digital forensics community through conferences, workshops, and publications would promote awareness of the BSDSF framework and gather feedback for continuous improvement.

Validation and Certification:

Pursuing validation and certification of the BSDSF framework through recognized forensic science organizations would enhance its credibility and legal admissibility.

IX. CONCLUSION

This paper has presented the BLE Smart Devices Security Framework (BSDSF)—a Protocol-Driven Plug-in Module for IoT forensic analysis deployed on the ESP32 microcontroller as a dual-channel BLE and Wi-Fi artifact generation framework. The experimental results demonstrate that the framework successfully produces a comprehensive catalog of forensically verifiable artifacts, including Bluetooth device identities, pairing request timelines, Wi-Fi SSID registrations, device fingerprints, and session timestamps, within a controlled, ethically compliant testbed environment.

9.1 Summary of Contributions

The BSDSF framework makes several significant contributions to the field of IoT forensics:

Methodological Innovation: The framework introduces a proactive, protocol-driven approach to forensic artifact generation, shifting from passive post-incident recovery to active evidence creation in controlled testbeds. This approach addresses fundamental limitations of traditional forensic methodologies that rely on residual data recovery.

Multi-Protocol Integration: By simultaneously capturing BLE and Wi-Fi artifacts from a single device, the framework addresses a critical gap in cross-protocol evidence correlation. The experimental results demonstrate successful temporal, spatial, and behavioral correlation of artifacts across both wireless channels.

Reproducibility and Accessibility: The use of widely available hardware (ESP32) and open-source development tools (Arduino IDE) ensures that the methodology is accessible and reproducible by forensic practitioners and researchers worldwide. The high consistency of results across multiple experimental runs validates the reliability of the framework.

Ethical Framework: The study establishes clear ethical guidelines for IoT forensic research, including isolation, consent, and data minimization principles that ensure legal defensibility and respect for privacy rights.

Foundation for Future Work: The framework provides a foundation for advanced capabilities including volatile RAM acquisition, automated artifact parsing, cross-device evidence linking, and integration with machine learning and AI technologies.

9.2 Addressing Research Gaps

The BSDSF framework directly addresses three critical research gaps identified in the literature review:

1. Proactive Artifact Generation: The framework demonstrates that protocol-level forensic artifacts can be systematically generated through controlled device behaviors, providing an alternative to passive post-incident data recovery.
2. Multi-Protocol Correlation: The framework establishes the feasibility of correlating BLE and Wi-Fi evidence from a single device source, enabling more comprehensive forensic timelines and device attribution.
3. Volatile Data Challenges: While the framework does not yet provide volatile RAM acquisition capabilities, it demonstrates that valuable forensic evidence can be captured at the protocol level before volatile data is lost, partially mitigating the volatile memory challenge.

9.3 Practical Implications

The BSDSF framework has several practical implications for IoT forensic investigations:

Training and Education: The framework provides a controlled environment for training forensic examiners in IoT device analysis, BLE and Wi-Fi protocol fundamentals, and multi-protocol evidence correlation.

Tool Development: The framework serves as a reference implementation for developing automated forensic tools, including log parsers, timeline generators, and report generation systems.

Legal Admissibility: The framework's emphasis on reproducibility, documentation, and chain of custody supports the legal admissibility of IoT forensic evidence in judicial proceedings.

Research Platform: The framework provides a platform for conducting controlled experiments on IoT security, protocol vulnerabilities, and forensic methodologies.

9.4 Broader Context and Related Work

The BSDSF framework contributes to a broader research agenda on IoT security and forensics. Recent work on endpoint security for smart homes has emphasized the importance of vantage point recreation and novel monitoring approaches (VENKATESH et al., 2026), principles that are directly embodied in the BSDSF framework's proactive artifact generation methodology. Similarly, research on IoT-driven solutions for vehicular networks has highlighted the need for systematic approaches to trustworthiness, misconduct detection, and position security (Shukla et al., 2024)—challenges that are analogous to those addressed by the BSDSF framework in the context of smart home devices.

The integration of AI and machine learning into forensic workflows, as demonstrated by recent work on predictive frameworks for real-time assistance (SUNIDHI SUDHEER SHENOY & N VISHNU

VENKATESH, 2025), suggests promising future directions for automating artifact analysis and evidence interpretation. The systematic data collection and analysis methodologies developed for crime forecasting (Natarajan et al., 2023) provide valuable insights for structuring forensic evidence and supporting investigative decision-making.

Furthermore, the principles underlying robust detect-and-avoid systems for autonomous drones (Natarajan et al., 2026) demonstrate the feasibility of implementing sophisticated monitoring and decision-making capabilities in resource-constrained environments—a key consideration for IoT forensic systems that must operate on embedded devices with limited computational resources.

9.5 Limitations and Future Work

While the BSDSF framework demonstrates significant advances in IoT forensic methodology, several limitations remain:

Volatile RAM Acquisition: The framework does not currently provide a mechanism for acquiring volatile RAM content from powered ESP32 devices. Future work should explore cold boot attack techniques and other methods for volatile memory acquisition.

Scalability: The current implementation relies on manual artifact extraction and documentation, limiting scalability in large-scale investigations. Future work should prioritize the development of automated parsing and analysis tools.

Protocol Coverage: The framework currently addresses only BLE and Wi-Fi protocols. Extension to additional protocols (Zigbee, Z-Wave, LoRa) would enhance its applicability to diverse IoT ecosystems.

Cross-Device Correlation: The framework currently focuses on single-device artifact generation. Future work should extend the framework to support multi-device testbeds and cross-device evidence correlation.

Real-World Validation: The framework has been validated in controlled testbed environments. Future work should include validation in real-world forensic scenarios to assess its practical utility and identify operational challenges.

9.6 Concluding Remarks

As the Internet of Things continues to expand, the need for robust, protocol-aware forensic frameworks will only intensify. The BSDSF framework represents a foundational step toward meeting this need by demonstrating that proactive, protocol-driven artifact generation is both feasible and valuable for IoT forensic investigations. By establishing a reproducible methodology, providing a platform for future research, and addressing critical gaps in current forensic practice, the BSDSF framework contributes to the ongoing evolution of digital forensics in the age of ubiquitous connected devices.

The framework's emphasis on ethical compliance, reproducibility, and legal defensibility ensures that it can serve as a model for responsible IoT forensic research and practice. As the framework continues to evolve through community contributions, automation enhancements, and integration with advanced technologies, it has the potential to become a standard tool in the digital forensics toolkit, supporting investigators in their efforts to understand, analyze, and attribute activities in increasingly complex IoT environments.

ACKNOWLEDGMENTS

The authors acknowledge the support of the Department of Forensic Science at JAIN (Deemed-to-be University) for providing the resources and infrastructure necessary for conducting this research. We also thank the reviewers for their valuable feedback and suggestions that improved the quality of this paper.

Author Contributions

Y. P. Rakshith: Conceptualization, methodology, software development, experimental validation, data analysis, writing—original draft.

N. Vishnu Venkatesh: Supervision, conceptualization, methodology, writing—review and editing, project administration.

Conflict of Interest Statement

The authors declare no conflicts of interest related to this research.

Data Availability Statement

The firmware source code, experimental data, and supplementary materials are available from the corresponding author upon reasonable request,

subject to ethical approval and institutional data sharing policies.

REFERENCES

[1] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, *17*(4), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095

[2] Goldsmith, A. (2005). *Wireless communications*. Cambridge University Press.

[3] Kent, S., & Atkinson, R. (1998). *Security architecture for the Internet Protocol* (RFC 2401). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc2401

[4] Lopes, A., Pereira, F., & Silva, J. (2021). Firmware extraction and analysis techniques for ESP32 devices. *IEEE Access*, *9*, 45678–45692.

[5] Natarajan, V. V., & Das, P., & Rajiv, A. (2026). A robust detect and avoid system for autonomous drone navigation. *NexusTech*, *1*, 2026004. https://doi.org/10.31893/tech.2026004

[6] Natarajan, V. V., Singhal, P., Pandey, D., Sharma, M., Rautdesai, R., Khubalkar, D., & Gupta, A. (2023). Crime forecasting using historical crime location using CNN-based images classification mechanism. In *Handbook of research on AI and ML for intelligent machines and systems* (pp. 245–268). IGI Global. https://doi.org/10.4018/978-1-6684-8618-4.ch013

[7] Nath, P., & Giri, S. (2020). Bluetooth and Wi-Fi evidence correlation in IoT forensics. *Journal of Network Security*, *12*(3), 156–172.

[8] Patil, R., & Singh, A. (2023). Challenges in IoT forensics for resource-constrained devices. *Forensic Science Review*, *35*(2), 89–104.

[9] Rahman, S., & Bhattacharya, R. (2022). IoT device forensics: A survey of methods and challenges. *International Journal of Digital Forensics*, *14*(1), 23–45.

[10] Shukla, M., Srivastav, V., Khare, M. D., & Venkatesh, N. V. (2024). IoT-driven solutions for VANET trustworthiness: Examining misconduct and position security challenges. *Multidisciplinary Reviews*, *6*, 2023ss059. https://doi.org/10.31893/multirev.2023ss059

[11] Stallings, W. (2013). *Network security essentials: Applications and standards* (5th ed.). Pearson.

[12] SUNIDHI SUDHEER SHENOY, & N VISHNU VENKATESH. (2025). A predictive framework for real-time courtroom assistance using AI-based mock legal advisor. *International Journal of Research and Analytical Reviews (IJRAR)*, *12*(2), 440–444. http://www.ijrar.org/IJRAR25B2617.pdf

[13] VENKATESH, M. N. V., Rajiv, D. A., Das, M. P., & Warrier, M. S. (2026). Vantage point recreation: A novel approach in endpoint security for smart homes. *International Journal of Innovative Research in Technology (IJIRT)*, *12*(8), 459–468. https://doi.org/10.64643/IJIRTV12I8-191180-459