Journal of Multidisciplinary Healthcare

Dovepress
Taylor & Francis Group

Open Access Full Text Article

REVIEW

Decoding the Solution for Man-at-the-End Attacks and Reverse Engineering on IoMT Devices: An Experimental Review of Techniques and Defences

Agila Harshini T, Harini Sriraman

Department of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India

Correspondence: Harini Sriraman, Vellore Institute of Technology, Chennai Campus, Vandalur- Kelambakkam Road, Chennai, TN, 600127, India, Email harini.s@vit.ac.in

Abstract: The Internet of Medical Things (IoMT), which revolutionizes modern healthcare, is expanding to provide remote support and treatment options. The device is handled by the patient or the healthcare provider to monitor the data continuously. When the IoMT is not handled with care by the authorized person, the sensitive data can be compromised by Man-At-The-End attacks (MATE). The gain of physical access allows the attacker to insert malicious code and tamper with the normal functionality, leading to wrong diagnosis and treatment. The altered performance of the medical device causes direct harm to patients. The existing software solutions do not provide complete security when it comes to attacks that gain physical access to devices. Physically Unclonable Functions (PUF) and Field Programmable Gate Arrays (FPGA) based security offers protection by establishing robust authentication through customizable logic, real-time data processing, and obfuscation. Identifying MATE attacks that are resistant to PUF/FPGA protection is crucial to offering a robust and resilient solution. The classification of critical MATE attacks specific to IoMT devices, Experimental validation using hardware platforms like Arduino Uno, Raspberry Pi, and ESP32 for reverse engineering is performed. Existing security measures available to protect the healthcare device, and how FPGA and PUF can provide better security against MATE attacks are explained. This paper uniquely reviews a thorough analysis of the vulnerability of PUF-enabled IoMT devices in the face of MATE attacks is presented. Possible futuristic solutions such as tamper-proof hardware, evolvable PUF, and AI-assisted anomaly detection for the identified problems, with experimental findings and security assessments, offer practical insights.

Keywords: cyber threat, FPGA, IoMT vulnerabilities, PUF security

Introduction

Cyber threats are rapidly increasing, with attackers frequently developing modern strategies to hack the IoMT device and compromise sensitive patient data. Though the technologies are advancing to protect medical devices from attackers, it becomes challenging when physical access is gained. The attacks are broadly classified into network-level attacks (Man-in-the-Middle like Sniffing, DoS, Session hijacking), Software based (such as malware injection, Zero day attacks), and cloud-level attacks namely account hijacking are major threats to IoMT devices. This work prioritizes Man-At-The-End attacks (Reverse Engineering, Aging PUF) that has high risk and are hard to detect. Existing security mechanisms often fail to provide a lightweight solution with high security demands for resource-constrained devices. The attacker's motive is to demand money or sell the data on the darknet without a direct claim. One such advanced attacking technique is Man-at-the-End. Traditional cyberattacks occur at the start or during a transaction, but MATE attack targets the final stages. Existing studies mainly focus on software or network-level security rather than attacks posed by physical access. In the man-at-the-end attack, the attacker gains access to the device physically and spies on the activities that happen on the device. This makes the attacker bypass the initial security measures that prevent access to the IoMT device. The attacker takes control over the device component, allowing them to alter and manipulate the final output just before the actions are finalized, which makes it complicated to encounter such attacks. The attacker may change the dosage of

Received: 5 June 2025
Accepted: 27 August 2025
Published: 8 October 2025

medicine given to the patient through an insulin pump, or change the data of heart rate at critical times, which leads to a wrong diagnosis. The security and privacy of IoMT devices are vital for eminent healthcare. Providing security using Field Programmable Gate Arrays (FPGA) and Physically Unclonable Functions (PUF) can prevent Man-At-The-End attacks from an extinct. There are other factors to be considered, such as vulnerabilities through the network, cloud, and software implemented on the healthcare device. FPGA and PUF provide features to secure hardware-based attacks, but more advanced attacks need to be addressed. Because of the limited storage of IoMT, complex implementation will not be feasible. From the experimental analysis, the attacks that can be prevented using PUF-based security and the attacks that break the security despite PUF are detailed. A robust security solution that is resilient to the evolving threat landscape is vital for IoMT devices.

The Major Contributions of This Paper are

- A comprehensive categorization of Man-At-The-End attacks on IoMT devices that include reverse engineering, key extraction is supported with experimental analysis, gives practical insights
- A detailed review of PUF and FPGA hardware-based security evaluation is presented. Evaluating their strengths, limitations, and specific attack vectors to protect against MATE attacks
- Tailored techniques combining emerging hardware and AI techniques, a futuristic solution to enhance resilience in IoMT devices, including evolvable PUFs, role-based access controls, advanced debugging mechanisms, and AI-based anomaly detection, are discussed. To provide a guide for future security implementations, a tabular mapping of MATE attack types and suitable security features serves as a roadmap to researchers
- The role of regulatory frameworks (e.g., FDA, EMA) in IoMT device real-world deployment, enforcing security policies, highlights the need for compliance and surveillance of the IoMT lifecycle

Significance of IoMT in Healthcare

IoMT connects medical devices via the internet, allowing patients to receive healthcare remotely. This improves patient outcomes and lowers costs by allowing data interchange between healthcare professionals and patients. The major contribution of IoMT in healthcare is remote patient monitoring, such as wearables and implant devices, for bedridden patients, which reduces frequent hospital visits. Wearables monitor a range of health parameters and provide real-time diagnoses. IoMT integrated with telehealth has leveraged remote monitoring through virtual consultations. Predictive analysis by history of data helps identify the risk of patients developing complications and gives proactive treatment. Effective data collection, storage, and processing are critical for IoMT. IoMT devices, such as smart tablets and insulin dispensers, ensure that patients take their medications on time and track their results. Another milestone in IoMT is robotic surgery, which can provide surgical care to remote patients. IoMT by addressing its challenges can fully realize its potential. The different types of IoMT devices are mentioned in Figure 1.

Vulnerabilities in IoMT

The Internet of Medical Things is a network of linked healthcare devices that collect, process, and transmit data. Despite the fact that IoMT devices improve patient outcomes, software, hardware, and protocol vulnerabilities limit trust in them. These vulnerabilities pose unique risks, necessitating different mitigation measures.

Software Vulnerabilities

Software vulnerabilities are frequently created by poor authentication, a lack of encryption, and the usage of out-of-date software. When the software is not updated, the device is vulnerable to existing threats. Weak hardcoded passwords allow unwanted access. Because of the lack of encryption in the device during ideal and transmission, the attacker modifies the data. Most IoMT devices come with default credentials that are not modified by the user, allowing the attacker to easily exploit the device. For example, when an insulin pump is operating with obsolete firmware or weak authentication, the hacker can install malicious code and damage the data.

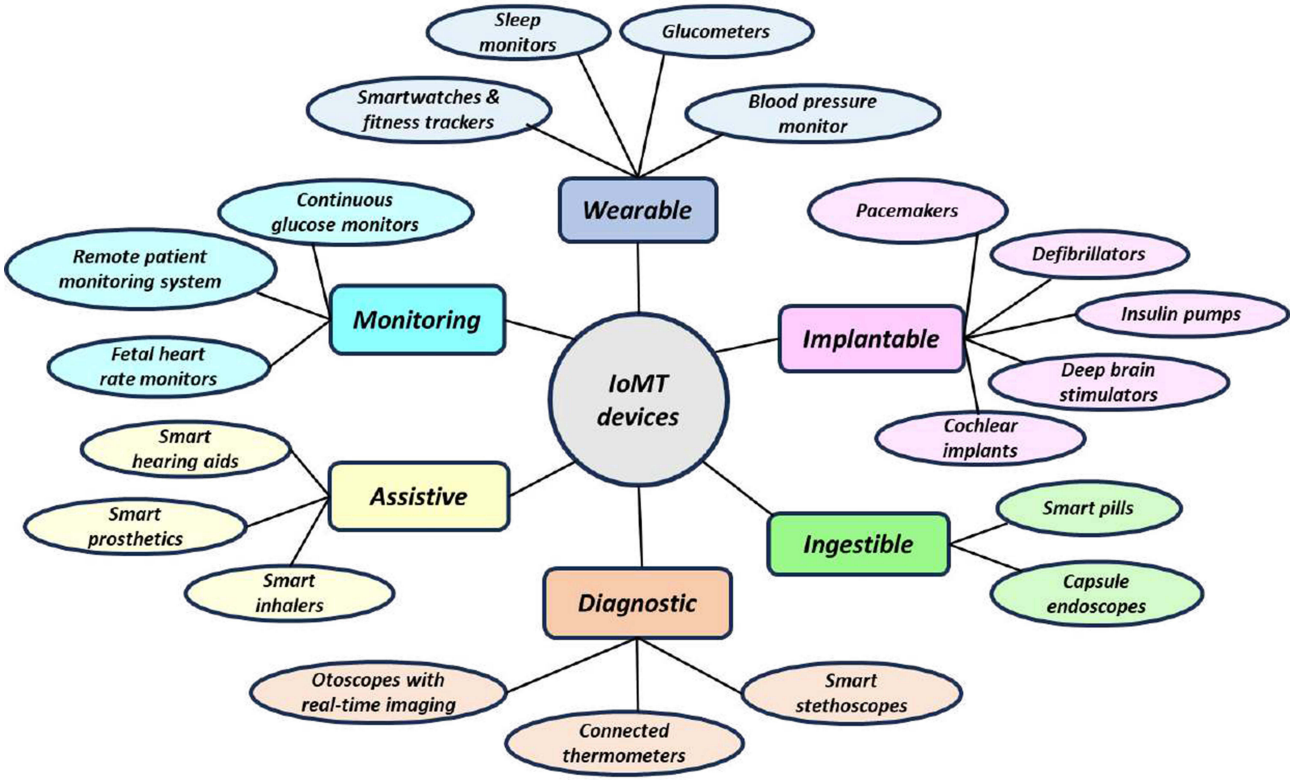

Figure 1 Types of IoMT devices.

Hardware Vulnerabilities

An IoMT device is vulnerable to side-channel attacks due to insecure hardware design, physical tampering, and a lack of protection resources. When an attacker has physical access to a device, the attacker can insert malware, tamper with the hardware, and change the functionality. The absence of a secure boot makes the IoMT vulnerable to assaults. Because healthcare equipment has limited resources, it is difficult to install complicated encryption mechanisms for solid security. Physical channels, such as power consumption patterns or electromagnetic emissions, through which an attacker can extract cryptographic keys, pose an additional danger. For example, installing a keylogger on medical equipment will record all of the patient's sensitive data. Addressing these concerns is critical.

Communication Protocol Vulnerabilities

Protocol flaws include insecure communication protocols, replay attacks, and man-in-the-middle attacks. When IoMT devices use obsolete and insecure HTTP or HTTPS protocols, an attacker can modify the data transmitted. The proprietary protocols are generally poorly described and include unknown vulnerabilities. This causes compatibility concerns. MitM attacks allow the attacker to intercept traffic between the wearable health gadget and the hospital's central monitoring system. Modifying communication and retransmitting incorrect data that appears to be legitimate poses a major hazard to the patient through replay assaults. Mutual authentication and time stamp solutions can help mitigate these attacks.

Different Ways IoMT Devices are Attacked by MATE

Widespread Physical Attacks

The device is physically attacked by tampering or side-cannel attacks. These include adding additional parts, and soldering components, to disable the security and change the functionality of the device. The power consumption patterns, acoustic signals, or other emissions from the device are tracked and the operational states, cryptographic keys, etc can be stolen without hardware or software interaction. Some of the physical attacks are as follows.

Tampering

Attacking the IoMT by physically altering the hardware components or configuration, like modifying sensors to manipulate the working of the device, producing wrong data.

Data Tampering/Manipulation Attacks

The configurations and the patient data are altered to give the wrong treatment plan that poses a risk to the safety of the patient.

Clone and Emulation

To study the behavior and functionality of the target device, the device operations are emulated under a controlled environment that can reveal the logic of operation, protocols used for communication, etc. These features are used to create a replica of the legitimate device. Cloning is done to mimic the physical characteristics.

Device Cloning

Using reverse engineering, a clone of the authorized device is duplicated. This device is used to perform any illegal access.

Device Emulation

To mimic the behavior with other devices attackers use a software emulator. It is used to manipulate data, disrupt functions, and infiltrate networks.

Protocol Emulation

Injection of false data, unauthorized access, and sensitive data of the healthcare device are captured by emulating the communication protocols.

Malicious Device introduction

A cloned device is introduced into the network of healthcare devices to spread malware.

Key Probing

Probing to intercept data bus or read memory directly, and micro probing to interact with the internal circuitry, extracts sensitive data.

Firmware Attacks

This acts as a backdoor for other future attacks. The attacker uploads malicious firmware when directly connected to the device. The attack may not be executed immediately, the attacker can monitor all functionalities and corrupt the function of healthcare devices during critical situations.

Firmware Manipulation

The attacker modifies the low-level software that controls the hardware and injects malicious code to change the device's functions. The data are extracted without the knowledge of the users. The attacker can also replace the legitimate software with a compromised version by reflashing the firmware. Device controls are gained even if the basic security measures are available like,

Firmware Replacement

The device control is taken by displacing the malignant firmware in place of legitimate firmware.

Firmware Injection

In the existing firmware, vicious codes are injected to create backdoors and introduce vulnerabilities. Techniques like firmware updates are exploited to disable security features.

Firmware Downgrade Attacks

By exploiting the weakness in the firmware and forcing the device to revert to an older version of the software. The new updates are restricted by this process which has the latest security patch.

Bootloader Attacks

For loading the device during startup bootloader is necessary. The attacker compromises the bootloader and runs unauthorized code to bypass the security checks of genuine firmware.

Side-Channel Attacks and Battery Drain Attacks

The device is made inoperable by draining the battery. For a resource-constrained IoMT device battery stability is important. To infer sensitive information, these attacks exploit the power consumption pattern, radiation, temperature, etc of the device. With physical access necessary set up to capture these emissions is done by the attacker.

Reverse Engineering

This attack can be done on the software and hardware of the target device. In software reverse engineering, the attacker infers the internal working by decompiling the software on the device and identifies vulnerabilities such as insecure configuration that can be exploited. Through hardware reverse engineering, the physical components like integrated circuits are deliberately studied to reveal the stored encryption keys.

Firmware Reverse Engineering

The framework is analyzed by the attacker using techniques like firmware extraction by physical access using UART interfaces or updates and finds the vulnerabilities to exploit the device.

Hardware Reverse Engineering

Analyzing the electronic components by dismantling the devices physically makes the attacker understand the operations of the device.

Protocol Reverse Engineering

The attackers capture and analyze the communication protocols used by the IoMT and intercept them.

Credential and Key Extraction

Power analysis, timing attacks and electromagnetic attacks are used to infer cryptographic keys and bypass security controls. Keyloggers are used to capture the inputs on the device, attacker extract the encryption keys. All the work done on the device input is captured, and the patient's personal data is exploited.

Malware Insertion

To intercept device memory during transmission, malware is inserted.

Interception of Communications and Data Extraction

The USB port or wireless interfaces, when gained physical access, allow the attacker to inject malicious commands and intercept the communication channels. The attacker can take all the data from the memory of the device through physical access. The hospital's logs, patient personal information, and medical summaries are stolen.

Eavesdropping/Interception Attacks

The attacker with unauthorized physical access installs code to sniff the transmitted data, device controls, and more. A man-in-the-middle attack is also performed by intercepting communication between the healthcare provider and the patient without their knowledge.

Replay Attacks

The attacker uses the captured power consumption patterns or any transmissions and replays them to persuade the device.

Injection Attacks

To take control over the medical device, SQL codes are injected by the attacker into the database through vulnerable software.

Software Reverse Engineering

To uncover the security vulnerabilities in algorithms, the software is analyzed. Methods like decompiling code and analyzing binaries are used to exploit the medical device. Debugging and Instrumentation by debugging interfaces like UART or SPI allows the attacker to control the processor execution, read, or write in the memory. By injecting malicious code through these interfaces, the security is bypassed and changes the behavior of the software.

JTAG and UART Interface Exploitation

Attackers use interfaces like JTAG (Joint Test Action Group) and UART (Universal Asynchronous Receiver/Transmitter) to debug and test electronic devices. This admits access to the low-level internal working of devices such as read-write memory.

Software Debugger Exploitation

The attacker inspects and modifies the memory and code execution of the running process on IoMT using debugging tools.

Instrumentation Framework

Real-time manipulation of device functionality is possible by using Frida, Xposed tools. This allows attackers to introduce vulnerabilities during a running operation.

Firmware and Software Analysis

Custom scripts are created, or existing tools can be used to automate the analysis. By analyzing and discovering software vulnerabilities, weak points are identified that can be exploited.

Memory Dumping

To extract the keys stored in poorly secured regions, the dumping of contents in device memory is done.

Network Traffic Interception

Without proper encryption in the protocol used by the IoMT, analyzing and capturing the network becomes easier. This attack extracts the cryptographic keys.

Denial of Service (DoS) Attacks

In a critical situation, the attacker disrupts the operation of the IoMT by damaging some components or disconnecting from the internet. This leads to denial of service and harms the patient.

Distributed Denial of Service (DDoS) Attacks

The network to which the IoMT is connected is given excessive traffic and makes the legitimate application unavailable to the authorized patient.

Ransomware

Completely blocks the healthcare device and demands something to restore the data. This attack severely affects the whole healthcare setup.

Role of FPGA and PUF in Securing IoT Devices

The resource-constrained nature of the IoMT devices is susceptible to MATE attacks. FPGA is a versatile hardware platform with customizable features. The unique digital signature of the PUF secures the device. FPGA and PUF consume more power, which is critical to be considered for IoMT devices with a battery. It is complex to implement a complex PUF or FPGA into IoT in healthcare.

FPGA in Securing IoMT

Field Programmable Gate Arrays (FPGAs) are programmable integrated circuits. FPGA can be used in various applications due to its flexibility in reprogramming. The logic blocks consist of a lookup table, flip-flop, and multiplexers, which are the basic logic blocks of an FPGA. These are connected by programmable pathways called interconnects. The entry and exit of the signals are managed by input/output blocks. Clock management is done to ensure synchronized operations. Modern

FPGAs consist of embedded blocks, like memory blocks, to enhance capabilities for specific tasks. FPGA has various benefits, such as customizable logic implementation by cryptographic algorithms directly on hardware to prevent malware. Hardware blocks are allotted for key storage to secure the device from unauthorized access. Using cryptographic signatures FPGA ensures that only verified firmware is executed on the device. Different functional blocks can be logically partitioned to limit the spread of security breaches to other functionalities. Sandboxing reduces cross-contamination with various applications. FPGAs can be programmed in a way that obfuscates critical operations and makes it strenuous to reverse-engineer the device. The nature of the reprogramming ability makes it unpredictable Physical tampering can be detected using built-in sensors. When a change is detected, such as temperature, voltage, and frequency, the sensor triggers alarms and initiates preventive measures. FPGAs can disable valuable functions or even erase the data automatically by detecting unauthorized access. Reconfigurable FPGAs mitigate developing vulnerabilities in response to new threats. Custom precautions are implemented to meet the specific needs of the IoMT device. Integrating advanced security features such as trusted execution environments and multi-factor authentication provides comprehensive protection for medical data. Biometric data can be used to verify identities, adding an extra layer of protection.

PUF in Securing IoMT

Tailored to the unique needs of IoMT devices, PUF offers hardware-based security. The variations in PUF are inherent in manufacturing and are difficult to replicate. The manufacturing variations are uncontrollable, which makes PUFs resistant to tampering. PUFs generate responses uniquely with specific input. In securing IoMT, PUFs are most advantageous. Different kinds of PUF can be used, according to the security needs of the device. PUFs can be used in on-demand cryptographic key generation with intrinsic properties of the device, which eliminates the need for keys to be stored in memory. Since keys are not stored permanently, it reduces the risk of attacks. This transient nature gives protection against side-channel attacks. To ensure that only trusted firmware is loaded, PUF supports secure boot and authenticates the device with a unique identifier. To secure the data in transmission, dynamic encryption-decryption is done, which makes attackers difficult to spoof. The authenticity of the communicant can be verified by PUF using challenge-response protocols based on PUF keys. PUFs offer lightweight, low computational overhead, and cost-effectiveness.

Tamper-Proof Hardware in Securing IoMT

To secure IoMT devices, PUF and FPGA alone cannot be efficient. FPGA/PUF without tamper-proof hardware has cryptographic functions, dynamic reconfiguration, unique key generation, and device identification. With tamper-proof hardware like a Trusted Platform Module (TPM), Hardware Security Modules (HSM) that resist physical attacks, an FPGA, and PUF, give adequate protection. Tamper-proof hardware is built to provide multilayer layer security, tamper detection, secure boot, and physical resistance.

Existing Security Measures and Best Practices

A comprehensive study on "Man-at-the-End attacks is conducted by Akhunzada et al,[1] analyzing attacks and real-world scenario implications, emphasizing resilient security. A defensive mechanism by Basile et al[2] proposes a decision-making in software protection. To leverage PUF and hash-based signatures for low-end devices, Roman et al[3] demonstrate minimal overhead with security. Ghubaish et al,[4] did a survey to secure IoMT devices by identifying data integrity and secure communication. Existing solutions do not provide enough security to resource-constrained device. A security assessment framework introduced by Alsubabei et al[5] categorises threats and provides a comprehensive assessment of vulnerabilities in IoMT. MATE attacks" A security assessment framework based on the ontological scenario. It helps the stakeholders choose the best security. If security cannot be measured, then planning, monitoring, and controlling security cannot be done, as reported by Alsubaei et al.[5] A framework with an insulin pump focuses on the security aspects of hardware. To find whether the device is directly or indirectly attacked, Nomikos et al,[6] a side-channel attack is performed. Uses advanced encryption standards mainly in image processing to resist already-known attacks. An authentication-based reconfigurable PUF for medical images is proposed. To prove the attack resistance, Chhabra et al,[7] did hardware attacks like side-channel, key-based, and reverse engineering. A post-quantum method for privacy preservation in cloud-based IoMT devices. The paper by Chen et al[8] uses hash functions to create a multi-factor

authentication and meets the necessary security requirements. This paper is especially for remote patient monitoring smart devices. A lightweight biometric-based authentication with minimal overhead is tested against different existing attacks. The computational cost is also minimal, said by Ali et al.[9] The healthcare devices are not very reconfigurable, and the security is very complex. IoMT device security by a blockchain-based system is proposed. It combines elephant head optimization and grey wolf optimization and uses a dual fitness function. Many attacks are tested to prove the performance said by Kanneboina et al.[10]

5G wireless communication has developed the telecare medical information system. The sensors are also resource-constrained, and the data is transmitted over public channels. The paper presents a PUF security for privacy preservation in the telecare medical system. The analysis is performed on a Raspberry PI by Yu et al.[11] Hardware-assisted blockchain for security is proposed for the first time. PUFchain 2.0, which combines PUF and blockchain, is given. Instead of storing the PUF keys on the IoMT device, they are obtained from the edge device. An arbiter PUF is implemented in the testbed and 200 keys are obtained out of which 75% are reliable said by Bathalapalli et al.[12] Artix-7 FPGA is used for implementation. AEAD cipher, which is lightweight, is used for mutual authentication. Using the ASCON cipher, the encryption is done by an agreed key. The proposed protocol's cost of communication and storage is less than the available protocols proposed by Raj et al.[13] To store and retrace VI, an SSI cryptographic wallet is used. SRAM PUF is used and temporary keys are generated. ATMEGA 328 microcontroller is used and as the IoT node, ESP32 is kept. It is implemented using limited resources said by Barbareschi et al.[14] A list of generators and how in FPGA random numbers are generated is given through a statistical comparison Bakiri et al.[15] A ring-oscillator PUF-based key generator is implemented in FPGA. Key reliability is guaranteed by a low-overhead BCH decoder. The presented key is deployable in an embedded system implemented by Maes et al.[16]

Elliptical curve-based keys are used. By inversion model analysis of 251-bit binary. Implementation in FPGA is done for computing elliptical curve points. A suitable w-coordinate-based differential addition is used. It is analyzed with a power analysis attack. Using the ECMQV protocol, this considered the first FPGA design analysed by Anandakumar et al.[17] The advantage of DRAM PUF is reconfigurability. A D-PUF is proposed by DRAM refreshing pausing intervals. It authenticates the device with low overhead. Altera Stratix IV GX FPGA board is used for implementation. It results in authentication time reduction. Robustness is shown by the rate of aging in controlled temperatures stated by Sutar et al.[18] An IFRD-based system is used for authentication between readers and tags. By combining the output of PUFs, a novel board ID is generated. Using a PCB prototype, radio-frequency communication has been verified by Yang et al.[19] The study focuses on CMOS-based PUF. The PCM devices and PUF based on PCM exploit the programming variability, according to Noor and Silva.[20] RO-PUF, that is, programmable based on the switch matrix of the FPGA, is proposed. It changes the structure of PUF by programming. Implementation is done in an Xilinx Spartan 6 FPGA, which has good uniqueness and reliability stated by Cui et al.[21]

Security measures in various classifications of DRAM-based PUF are detailed. In the survey, DRAM proves to be better for commercial IoT. To address the current security issues, DRM can be included said by Anagnostopoulos et al.[22] The paper proposes hardware security to protect the software. Hardware-entangled software protection protects the software from malicious code injection during runtime. The physical properties of the device and DRAM PUF timings are combined to create the HESP Xiong et al.[23] A real-time data analysis is replayed by presenting an architecture for an attack vector in PCIe based on a MitM attack. An emulator-based proof of concept with a Stratix 5 FPGA is given by Khelif et al.[24] A lightweight protocol based on cryptographic security is implemented by dynamic partial reconfiguration capabilities. Using PUF, a solution is implemented to eliminate attacks with the availability of DPR at IoT nodes stated by Johnson et al.[25] The public cloud is vulnerable to various attacks, and a solution using an FPGA is proposed. It gives temporary access to data to a client on a private cloud by establishing session key authentication. To support the publish mode operation, proxy re-encryption is used. The proposed method is 6 times faster in ciphertext transformation proposed by Al-Asli et al.[26]

A lightweight attestation during runtime to detect malicious changes in hardware is proposed. The integrity of the data path and hardware state can be verified. The delay paths are verified by attestation of the finite state machine. The response finds the anomalies if present. By supporting a reconfigurable platform, the proposed system has the capability of remote attestation schemed by Usama et al.[27] An isolation and protection mechanism in hardware is done. It operates

in between network and devices. The central server analyses network traffic beyond one IoT node and finds malicious activities. It has a 98.68% attack detection rate against DoS attacks proved by Hategekimana et al.[28] A filter to secure the IoT gateway for messages is given. Packet inspection is done by the Wu Manber algorithm in FPGA implemented by Stanciu et al.[29] A very efficient secret unknown ciphers are used in hardware to make the device clone-resistant. It is low-cost and can be used in non-volatile, FPGA-based devices said by Hamadaqa et al.[30] A non-invertible transformation is introduced to ECG, and fuzzy logic changes the dynamic range of signals. This is based on a cancellable biometric system. An XOR operation encryption adds more security. The user gives access to the network only when requested implemented by El-Banby et al.[31]

This work implements a 256*256 watermark image for the patient's sensitive data. It is then given to a DICOM image and encrypted using a Tent map. Stratix FPGA is used to carry the image in the form of a bitstream. This also overcomes false positive problems stated from Arumugham et al.[32] A lightweight cipher is implemented on an FPGA. For permutation of these ciphers, Substitution box is applied. The proposed ASCON architecture has better throughput for medical and normal images, as implemented by Raj et al.[33] An ECG machine for energy is proposed, and it is implemented on an Artix-7 FPGA, said Singh et al.[34] A secure IGLU is proposed for an insulin-delivering pump that is non-invasive. For the security of the device, hardware-assisted security using PUF is used. The simulation results prove to be safe for IoMT devices, said Joshi et al.[35] Post-quantum cryptography enhances security in IoMT. To balance flexibility, scalability, and security, the proposed work uses docker and Kubernetes. Integrating post-quantum computing with Kubernetes shows better benefits, explained Mohamed et al.[36] A review paper explaining the IoMT security of smart care homes. The paper by Khan et al[37] shows the future guidelines to address secure IoMT and existing attacks to be addressed. A new approach combining PUF with IOTA tangle is proposed. PUF keys are stored in the body area network of the patient. Masked authentication messaging is used for communication among the patient, stakeholders, and healthcare. Bathalapalli et al[38] said that his method provides decentralized security with minimal energy needs.

NIST selects the ASCON cipher as the standard for IoT devices. The proposed paper is implemented on different FPGA boards to show the increase in throughput. Koppuravuri et al[39] explained that, among them, Virtex-7 is used for real-time medical imaging. The implementation of round-based, unrolled, and serialized strategies for achieving high throughput is presented. This helps develop highly efficient hardware, said Khan et al.[40] A lightweight PRESENT cipher is proposed by Damodharan et al,[41] to accelerate hardware because most ciphers are unsuitable for IoT due to resource constraints. It is implemented in the XILINX FPGA board and has a better throughput. An increase in machine learning attacks necessitates strong security like PUF. This thesis addresses the feasibility, time, and challenge-response pair of PUF. Prakash et al[42] stated that introducing poison data to PUF and learning the listening model achieves 99.63% accuracy in detecting new challenges. Existing security against machine learning attacks is not very reliable. The proposed configurable poisoning PUF by Lin et al,[43] with learning parity noise, performs best when compared to other adversarial-based solutions. Protecting implantable medical devices from attacks that modify data. Shield uses radio design as a jammer and receiver; it is a personal base station that is a physical layer of security. This proposed method by Gollakota et al[44] will jam the unauthorized commands. The attacks are sneaky when performed at a low level where no threat identification units exist. Misuse of out-of-order execution in a processor, like Meltdown and Spectre attacks, is difficult to uncover. This paper depicts reverse engineering firmware and gaining access to the router. Through a backdoor, several MitM attacks are made on the device to steal the data, Adithyan et al.[45] A detailed review of IoT firmware vulnerability is presented under various categories and detect threats. Existing tools, analysis, and how vulnerabilities are detected and addressed for unknown attacks. Different auditing tools are mapped to vulnerabilities and their vectors. The paper by Bakhshi et al[46] also discusses the research gaps in securing the IoT firmware. The first manual reverse engineering for protocol discovery is proposed and tested on different IoT devices, and it is implemented for embedded Linux filesystems. The system by Liu et al[47] is analyzed statically and dynamically to extract firmware from Flash and reverse engineer. This disclosed the flaws of the design that attackers can take control of. Alsubaei F et al,[5] proposed a security framework that adopts an ontological approach. It helps the stakeholders choose optimal security. Nomikos K et al[6] proposed a system that identifies direct and indirect attacks through side-channel analysis. An obfuscated AES cryptosystem was propeosed by Chhabra S et al[7] to secure IoMT edge devices. The resistance against known attacks like reverse engineering and key extraction is improved.

Table 1 explains the existing security mechanisms that offer security to different MATE attacks, mainly focused on PUF-based security measures.

Experimental Setup for Reverse Engineering

Figure 2 shows the experimental setup of Arduino Uno, Raspberry Pi and ESP32 for Reverse Engineering. The code for the experiment used is available on GitHub. https://github.com/agilaharshini/MATE-attacks-and-Reverse-engineering.

Arduino Uno

To expose the vulnerability of Arduino Uno, a setup is created to monitor the heartbeat using a pulse sensor and Arduino. Figure 3 represents the flowchart of the original code. To check the possibility of reverse engineering a device using Arduino, the code is converted into an .elf, which is an executable file on the Linux platform. This file is disassembled and analyzed. Figure 4 represents the flowchart of the disassembled code. Some values in the code are altered and dumped into the device. The modified code gives abnormal values. Reverse engineering, a piece of code in the medical device, creates deadly consequences for patients through wrong diagnoses and treatment by healthcare professionals.

Raspberry Pi

An ultrasonic sensor is connected to the Raspberry Pi to detect objects. Figure 5 represents the flowchart of the original code. The Python file is converted to an executable using PyInstaller. The file generated by the PyInstaller gets encrypted when we copy and paste it to other systems. The executable file will be saved in the dist directory as an .elf, and it is disassembled into assembly code. Figure 6 represents the flowchart of the disassembled code. When altered the function of the device using assembly code and install it in the sensor, it leads to incorrect detection.

ESP32

The ESP32 has built-in security. The setup is created to monitor the heartbeat using a pulse sensor. ESP32 generates only a .bin file, which is not readable. The code cannot be reversed. In Arduino and RPi, security features like secure boot and flash encryption features are absent, whereas ESP32 outperforms them.

The security features of the Arduino Uno, Raspberry Pi, and ESP32 are compared in Table 2.

Security Features of Arduino Uno

To protect code from unauthorized reading/writing on flash memory, lock bits can be set that are configured by AVR fuses. There is no built-in hardware encryption or secure boot in Arduino Uno. Table 3 explains the structural similarity of the original code and the reverse-engineered code of Arduino Uno.

Security Features of Raspberry Pi

RPi provides secure boot only for the bootloader, not for the operating system, starting with RPi 5; the previous versions lack this feature. This can use only an external Trusted Platform Module (TPM) since there are no built-in hardware security features. RPi supports iptables, an uncomplicated firewall for network security. Table 4 details the structural similarity of the original code and the reverse-engineered code of the Raspberry Pi.

Security Features of ESP32

Built-in security makes ESP32 more secure than RPi and Arduino Uno. Secure boot ensures the device runs only a signed firmware, which prevents the execution of malicious firmware. ESP32 provides flash encryption using AES-256. Timing attacks are prevented because ESP32 provides AES, RSA, SHA, and HMAC hardware to perform cryptographic operations at constant time. eFuse (key storage) stores data securely. eFuse disables JTAG debugging permanently, which restricts debugging access. Non-volatile storage is used to encrypt configuration settings.

Table 5 explains the Code obfuscation strength measurement of Arduino Uno, Raspberry Pi, and ESP32.

Table 1 MATE Attacks and Possible Combinations of Solutions

Solutions / Attacks	Tamper-Resistant Hardware	Encryption/ Decryption	Digital Signature/ Watermark	Intrusion Detection System	Virtual Private Network	Firewall	Segmentation	Obfuscation and Minification	Anti Debugger/ Emulator	Opaque Predicates	Behavioral Analysis	Key Rotation/ Expiry
Tampering	✓	✓	✓	✓								
Data interception and manipulation		✓			✓	✓	✓					
Reverse Engineering	✓	✓	✓					✓				
Firmware Manipulation	✓	✓	✓									
Debugging and Instrumentation		✓							✓	✓		
Cloning and Emulation			✓					✓			✓	
Credential and Key Extraction	✓	✓										✓

Note: The table indicates which security solutions can mitigate specific attack types targeting IoMT device.

https://doi.org/10.2147/JMDH.S538176

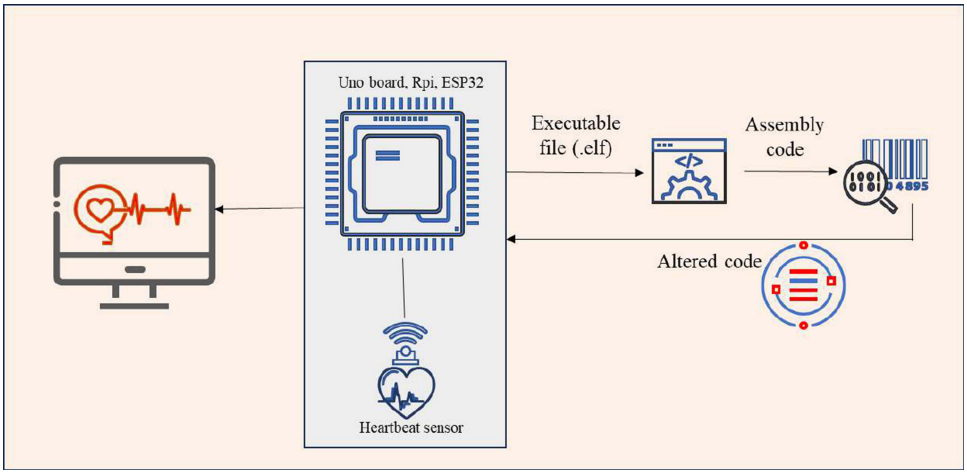

Figure 2 Reverse Engineering by code flow alteration in an embedded health monitoring system.

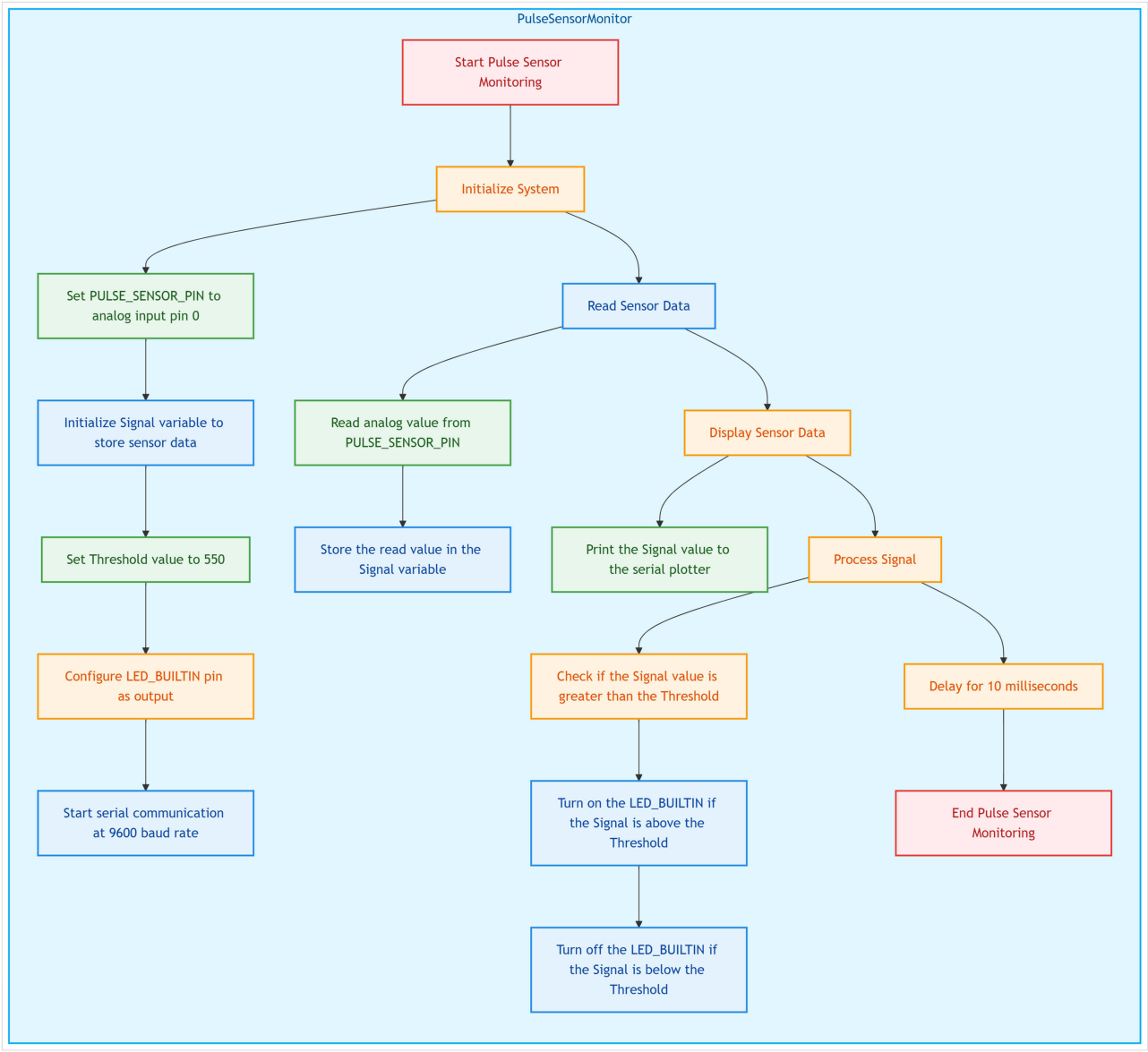

Figure 3 Flowchart of the original code (Arduino Uno) (generated by Codeflow.com).

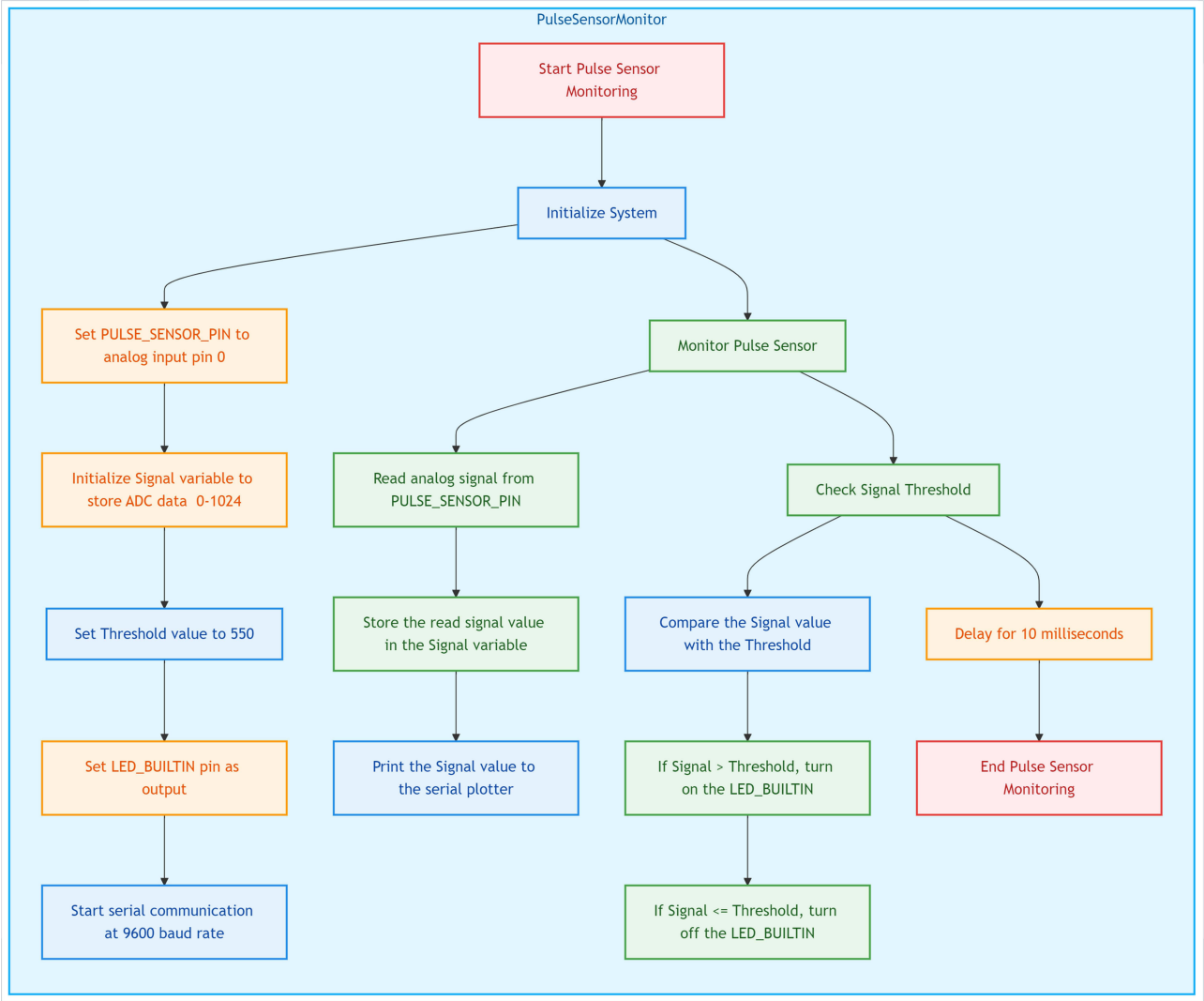

Figure 4 Flowchart of the disassembled code (Arduino Uno) (generated by Codeflow.com).

Findings and Discussions

Preventable Mate Attacks by PUF/FPGA Protection

Several types of Man-at-the-End (MATE) attacks can be effectively restrained with tamper-proof hardware. Figure 7 shows the MATE attacks and the possible solutions to prevent the device from them. The attacks for which tamper-proof hardware servers as a solution are as follows.

Hardware Tampering and Manipulation Attacks

Healthcare devices can be tampered with physically by an attacker, which involves opening the device circuitry or inserting malicious code. The features of tamper-proof hardware include tamper-evident seals, intrusion detection sensors, and self-destructive mechanisms to prevent the device from unauthorized access. A hardware security module (HSM) stores cryptographic keys safely and makes them unreadable. Even if the attacker gains physical access, the data remains safe because of encryption.

Data Interception and Alteration

Data exchanged between medical devices and healthcare systems is intercepted and altered by the attacker, which leads to wrong output data by the device. These modified data are misinterpreted by healthcare professionals, causing serious threats to patients. Establishing secure communication channels using Transport Layer Security and Secure Sockets

https://doi.org/10.2147/JMDH.S538176

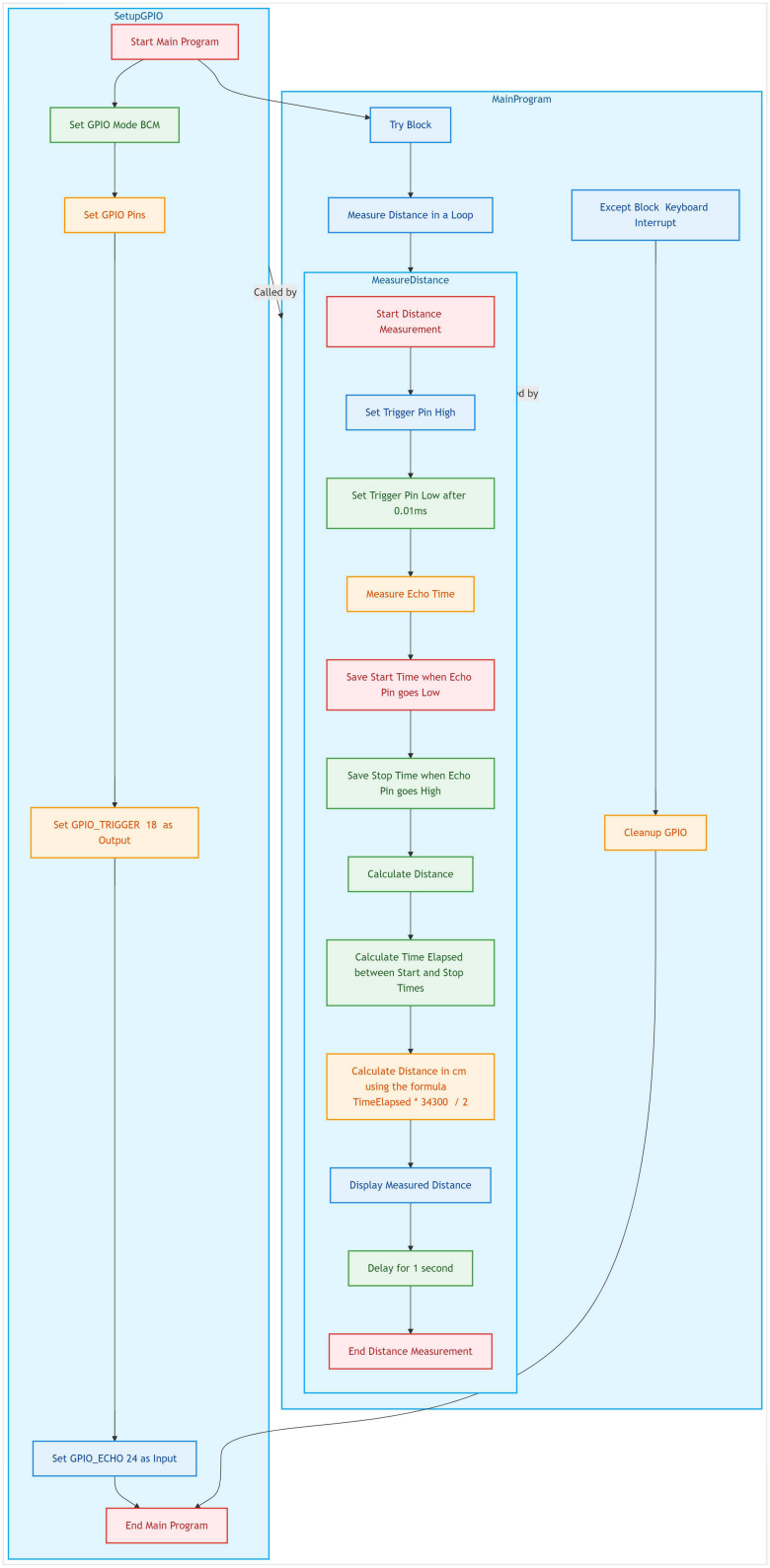

Figure 5 Flowchart of the original code (Raspberry Pi) (generated by Codeflow.com).

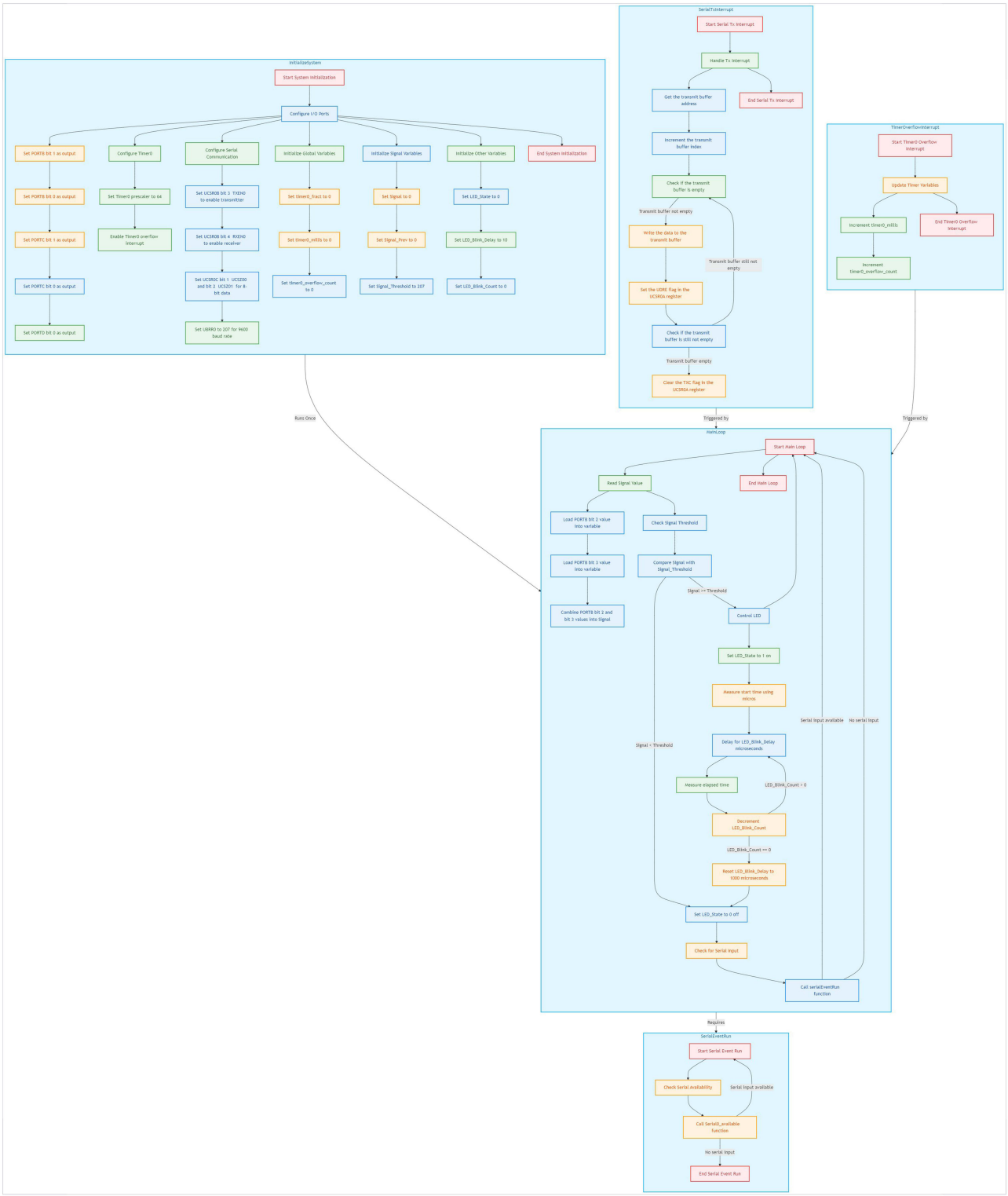

Figure 6 Flowchart of the disassembled code (Raspberry Pi) (generated by Codeflow.com).

Table 2 Comparison of Security Features

Security Features	Secure Boot	Encryption	Hardware Crypto Engine	Firmware Protection	Tamper Proof	Key Storage	JTAG Debugging
Arduino Uno	Absent	Absent	Absent	Absent	Absent	Not Secure	Enabled
Raspberry Pi	Absent	Software-only	Absent	Absent	Absent	Not Secure	Enabled
ESP32	Advanced	Hardware	Present	Enabled	Built-in	Secure	Controlled

Note: Comparison of security features across Arduino Uno, Raspberry Pi, and ESP32 platforms.

Table 3 The Structural Similarity of the Original Code and Reverse-Engineered Code of Arduino Uno

Code	Sematic Similarity
Step 1	60%
Step 2	92%
Step 3	36%
Step 4	68%
Step 5	49%
Step 6	69%
Step 7	64%
Step 8	58%
Step 9	25%
Step 10	12%
Step 11	59%
Average	54%

Table 4 The Structural Similarity of the Original Code and the Reverse-Engineered Code of the Raspberry Pi

Code	Sematic Similarity
Step 1	60%
Step 2	35%
Step 3	92%
Step 4	34%
Step 5	31%
Step 6	37%
Step 7	45%

(*Continued*)

Table 4 (Continued).

Code	Sematic Similarity
Step 8	36%
Step 9	27%
Step 10	30%
Average	42%

Table 5 Code Obfuscation Strength Measurement

Features	Technique	Arduino	RPi	ESP
Structural Similarity	Jaccard similarity with flowchart	46.2%	42%	0%
Sematic Similarity	Sentence Transformers	54%	58%	0%
Binary Difference (Binary level metric)	Hex files -byte by byte	< 90%	82%	0%
Obfuscation strength	Using the above metrics	Weak	Moderate	Strong – Not reverse engineered

Layer, tamper-proof hardware secures the device from unauthorized interception. Before executing the firmware, the digital signature verifies the integrity of the user.

Firmware Tampering

Unauthorized modification of the firmware introduces vulnerabilities in the device and alters the data. It is done by intercepting firmware patch updates or direct device access. With tamper-proof hardware, a chain of trust is created, and each step is verified for valid authorization before proceeding to the next. Encrypting firmware protects it from being read by an attacker. Run-time integrity checks and behavior monitoring can detect anomalous activities.

Credential and Key Access Through Side-Channel Attacks

Exploit the physical characteristics of a device, such as power consumption or electromagnetic emissions, through physical access. This information reveals the encryption keys. Dynamic execution and modifying the timing make the function unpredictable to an attacker. The level of noise introduced can detect any malicious activity. Obfuscating critical operations protects from side-channel attacks. Reverse engineering is possible.

Replay Attacks

Attackers capture legitimate data transmissions and replay them to gain access. Electromagnetic shielding can prevent the leak of signals that are captured and intercepted. Time-stamping makes every communication unique and makes sure it is not reused. To obscure specific patterns, noise can be added. The intrusion detection system triggers alerts and shuts down the system if it is compromised. Masking techniques hide the actual value processed. Changing device parameters frequently by dynamic reconfiguration prevents any patterns that could be exploited.

Advanced Techniques Attackers Use to Break the Security Despite PUF

While tamper-proof hardware is an effective defense against many physical and some digital attacks, it has limitations and cannot prevent all types of man-in-the-middle (MITM) attacks. Figure 8, portrays the attacks that are vulnerable even if solutions are available. Here are some types of MATE attacks where tamper-proof hardware may not offer sufficient protection to some attacks, such as,

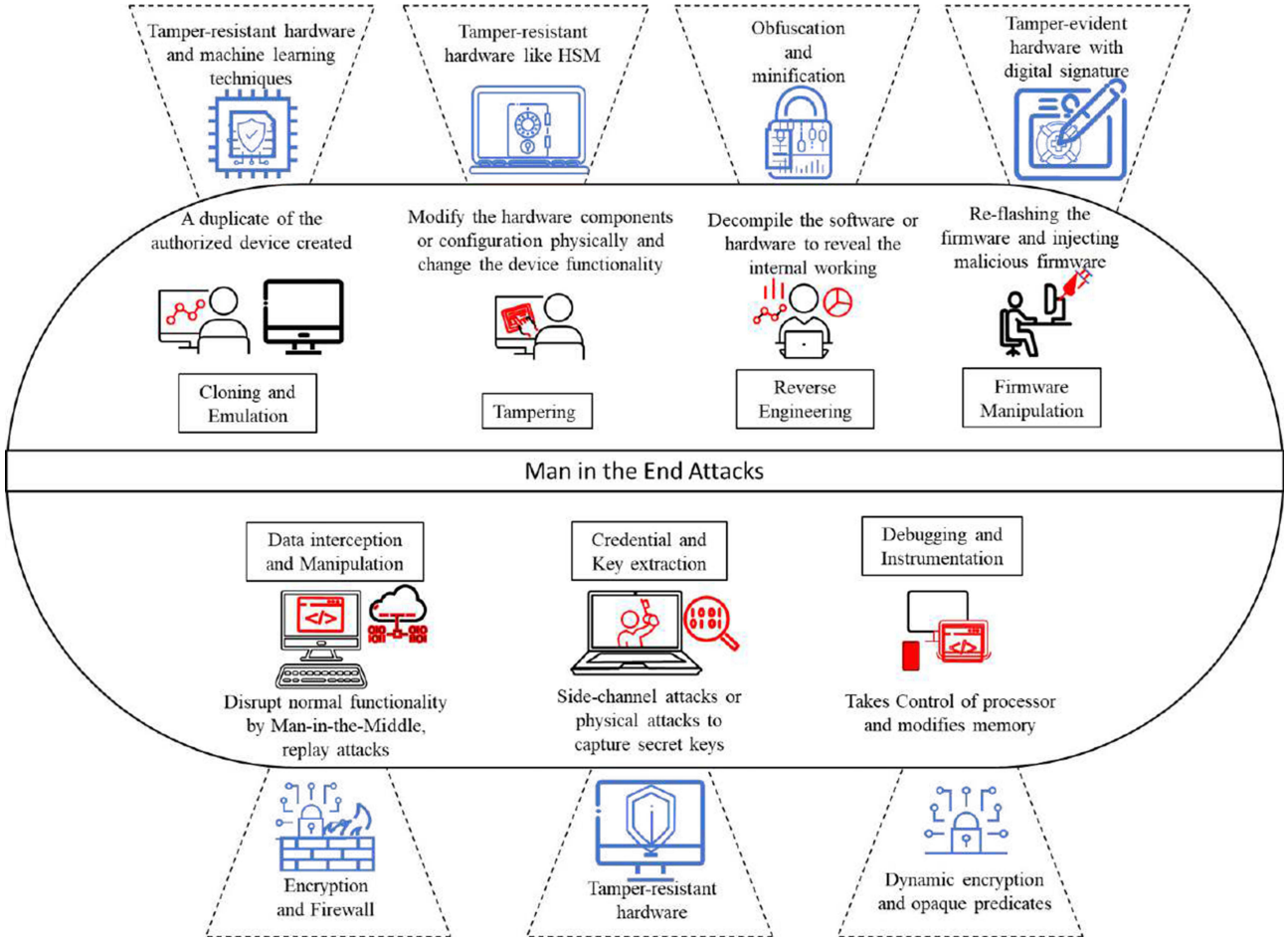

Figure 7 Types of MATE attacks and protection provided.

Aging PUF

The wear and tear of the device increase over time. The physical coating and sensors become less sensitive. Side-channel attacks collect information about electromagnetic radiation, power analysis, and external radiation, which are not protected by tamper-evident hardware. IoMT devices are used in general places where everyone is physically closer to a medical device. The attacker might physically tamper with an external slot, insert a pen drive with malicious code, and break any critical components. Supply chain attacks in which the device is jeopardized during manufacturing or while shipping. The attacker remains ideal, monitoring the data and attacks when needed. Design flaws of the device can make it vulnerable and be exploited the device.

Inability to Provide Complex PUF Security for Simple/Smaller Devices

Tamper-proof hardware is complex and expensive to implement in a resource-constrained IoMT device, which increases the risk of design issues. Weak encryption techniques used in devices act as a gateway for attackers, even if tamper-proof hardware is provided. The tamper-evident detects any modification only in the physical components of the IoMT device. By probing, the attacker creates a clone or can emulate the function of the device.

External Variation-Based Reverse Engineering

The physical signals, like acoustic, temperature, power, etc, are used to analyze and attack a device that is difficult to mask. The indirect external information leaks are undetectable by tamper-proof hardware. Side-channel attacks do not require internal hardware access; it is non-invasive and depends on external variations.

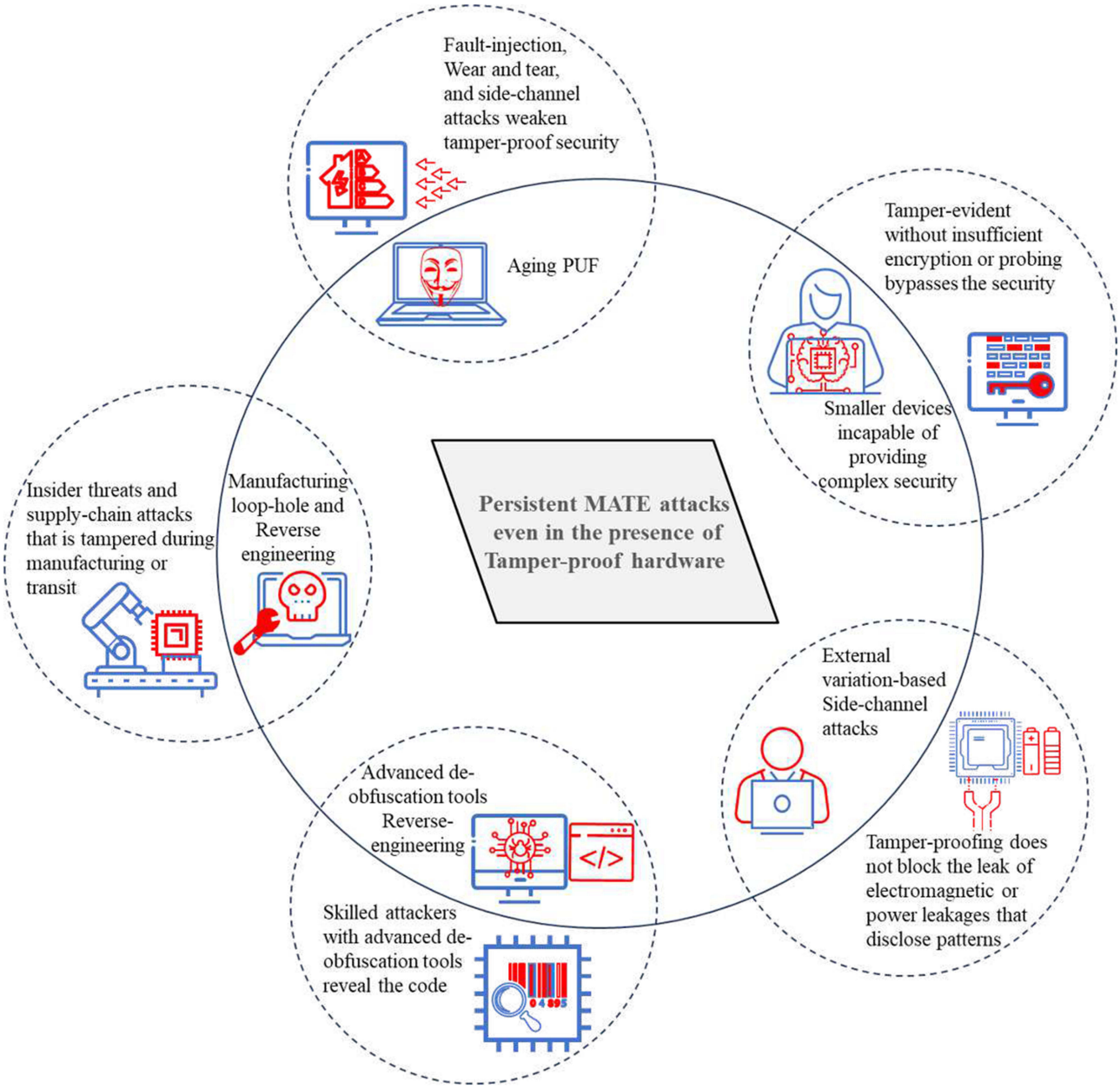

Figure 8 MATE attacks that break tamper-proof protection.

Advanced De-Obfuscation for Reverse Engineering

Code obfuscation is usually used to prevent reverse engineering. But with the rise of AI and improved computing capabilities, modern de-obfuscation tools enable a skilled attacker to find the implementation of the device. Using de-obfuscation tools, a skilled attacker can find the exact implementation and attack the device. The use of these tools removes the layer of security given by obfuscation that makes the pattern readable by the attacker. A partial understanding of the algorithm used or keys stored is enough to bypass the security. Though tamper-proof hardware with code obfuscation provides protection, the device is vulnerable to reverse engineering attacks.

Manufacturing Loop-Hole

Loopholes like inadequate security testing, lack of standardization, weak authentication mechanisms, and overlooked supply chain risks pose a significant threat to IoMT devices. The devices are often manufactured focusing on functionality, but not security. No standards are enforced universally on the manufacturers of medical devices.

Possible Futuristic Solutions

- The size of a device is a major constraint. In smaller devices, implementing a complex PUF is difficult. The complexity of the protection device will be obtained from the reconfigurability of PUF. Even here the PUF has to be of a certain size to offer reconfiguration. Evolvable reconfigurable hardware can be used for this purpose. Adaptive to the aging effects adjusting periodically or reconfiguring their internal parameters, evolvable PUFs maintain reliable responses even as the hardware ages. This adaptability is essential in environments where devices are expected to operate over extended periods under varying conditions. The ability to evolve and reconfigure makes it difficult for attackers to establish a similar model of the PUF's behavior. Evolvable PUFs integrate principles of adaptive systems and self-repair mechanisms.
- For external variation-based side-channel attacks, traditional PUFs seldom provide security. These cases avoid external variation-based PUF for IoMT devices, which are prone to Man-At-The-End physical attacks. Another solution for this problem could be role-based access control. The use of hardware security modules and trusted platforms can store keys securely. These are tamper-resistant. Secure enclaves like Intel SGX and ARM TrustZone protect by executing the code in isolated environments. For an additional layer of security, use of Key Encryption Key is used. Use static and dynamic analysis to identify and fix the issues during the development of software. This will minimize the vulnerabilities of key extraction. Regular key rotation limits the damage if the key is compromised. To ensure that old keys are no longer used, regular updates of the key expiration policy are implemented.
- Advanced de-obfuscation tools could be designed with latest generation AI methods. The design of the hardware chip or the software code is analyzed by the attacker to exploit the device. This can be averted by code obfuscation. Tools like .Net obfuscator, Skater, crypto obfuscator, etc are used to protect the device. It includes variable renaming, altering the control flow, and inserting misleading codes. To reduce the readability in websites by the attacker, minification is used. It removes whitespaces, shortens variable names, and reduces the code size. Checksum validation and self-check codes are used for integrity validation so that when the checksum mismatches the software refuses to respond. Encrypting the critical parts of the code and decrypting it only during runtime by code encryption to prevent data from being easily read. The image, audio, and configuration files are also encrypted to secure from analyzing the data. Anti-debugging to check the presence of debugging tools and anti-emulation to find if the software is run on an emulator are used to terminate the application if found vulnerable. To prevent the device from running with unauthorized software, hardware locking systems are used. This will bind the hardware to specific software. Digital watermarks on software and hardware with their uniqueness prevent reverse engineering.

Role of Regulatory Bodies in Securing IoMT

Unique legal regulatory issues are in IoMT because it has many stakeholders such as manufacturer, device provider, network provider, software, and end user. The ownership of the data generated is not clear. The data is shared with other parties. By developing and enforcing guidelines designed to ensure the effectiveness, safety, and security of Internet of Medical Things (IoMT) devices, governing bodies play a crucial part in guaranteeing their safety. To protect patient health and data, these organizations—such as the European Medicines Agency (EMA) in the EU and the Food and Drug Administration (FDA) in the US—set rigorous standards for the creation, manufacture, and utilization of IoMT devices. Before permitting these devices to hit the market, the officials perform comprehensive evaluations and approvals, and keep surveillance on them afterward to identify and eliminate any new hazards. Regulatory agencies additionally encourage cybersecurity best practices, which defend devices like these from hacking attempts that may threaten patient privacy and safety. Regulatory bodies use these exhaustive approaches to guarantee that IoMT devices make a good contribution.

Conclusion and Future Enhancement

Man-At-The-End attacks reduce the reliability of IoMT devices. MATE attacks that target hardware components can be combated by PUF and FPGA. An attacker with direct physical access poses more challenges to medical devices. The hardware-level security, design obfuscation, reconfigurability, and intrinsic uniqueness provided by PUF and FPGA are advantageous to secure devices. Though tamper-evident hardware is the best solution for safeguarding IoMT devices, it is not foolproof. While considering MATE attacks, PUF and FPGA alone do not provide complete security. The experimental analysis done proves that Arduino and Raspberry Pi can be tampered with, highlighting the pitfalls of traditional software-based defences, whereas ESP32 has secure encryption and boot is resilient. An evolvable PUF, with deep learning techniques, can reduce unauthorized and physical device access. The continuously evolving nature of evolvable threats to IoMT requires adaptation to new strategies for providing safe diagnoses and treatment. As the attacks evolve, robust and ingenious protection mechanisms are mandatory. The security measures should involve a complete defense addressing MATE attacks. Safeguarding IoMT devices with advanced PUFs like quantum and optical PUFs can create highly secure identifiers. Reverse engineering is quite impossible with Light-based photonic circuits. These include high implementation cost and increased power consumption to be implemented in a resource-constrained IoMT device. A tamper-proof hardware cannot provide complete security against advanced machine learning-based models. Self-healing and hardware randomization could spontaneously alter the device configuration or operations, repairing any damage caused by attacks. Hardware integrated with AI-based abnormality detection is used for real-time responses to attacks. A 3-dimensional IC makes probing difficult and inaccessible for an attacker without destroying the chip entirely. Camouflaged logic gates, which look identical and are implemented in an IoMT device, are nearly impossible to tamper with and reverse engineer the hardware. An adaptive multi-layered defence technique is the future of IoMT device security.

Disclosure

The authors declare that there are no conflicts of interest related to this manuscript. No financial, professional, or personal relationships with organizations or individuals have influenced the research, analysis, or conclusions presented in this work.

References

1. Akhunzada A, Sookhak M, Anuar NB, et al. Man-at-the-end attacks: analysis, taxonomy, human aspects, motivation and future directions. *J Network Comput Appl*. 2015;48:44–57. doi:10.1016/j.jnca.2014.10.009
2. Basile C, De Sutter B, Canavese D, Regano L, Coppens B. Design, implementation, and automation of a risk management approach for man-at-the-end software protection. *Comp Secur*. 2023;132:103321. doi:10.1016/j.cose.2023.103321
3. Roman R, Arjona R, Baturone I. A lightweight remote attestation using PUFs and hash-based signatures for low-end IoT devices. *Future Gener Comput Syst*. 2023;148:425–435. doi:10.1016/j.future.2023.06.008
4. Ghubaish A, Salman T, Zolanvari M, Unal D, Al-Ali A, Jain R. Recent advances in the internet-of-medical-things (IoMT) systems security. *IEEE Internet Things J*. 2021;8(11):8707–8718. doi:10.1109/JIOT.2020.3045653
5. Alsubaei F, Abuhussein A, Shandilya V, Shiva S. IoMT-SAF: internet of medical things security assessment framework. *Internet Things*. 2019;8:100123. doi:10.1016/j.iot.2019.100123
6. Nomikos K, Papadimitriou A, Stergiopoulos G, et al. On a security-oriented design framework for medical IoT devices: the hardware security perspective. In: 2020 23rd Euromicro Conference on Digital System Design (DSD); IEEE; 2020:301–308.
7. Chhabra S, Lata K. Obfuscated AES cryptosystem for secure medical imaging systems in IoMT edge devices. *Health Technol*. 2022;12(5):971–986.
8. Chen X, Wang B, Li H. A privacy-preserving multi-factor authentication scheme for cloud-assisted IoMT with post-quantum security. *J Inf Secur Appl*. 2024;81:103708. doi:10.1016/j.jisa.2024.103708
9. Ali Z, Mahmood S, Mansoor K, Daud A, Alharbey R, Bukhari A. A lightweight and secure authentication scheme for remote monitoring of patients in IoMT. *IEEE Access*. 2024;12:73004–20.
10. Kanneboina A, Sundaram G. Improving security performance of Internet of Medical Things using hybrid metaheuristic model. *Multimedia Tools Appl*. 2024;84(9):6403–2.
11. Sungjin Y, Park K. PUF-PSS: a physically secure privacy-preserving scheme using PUF for IoMT-enabled TMIS. *Electronics*. 2022;11(19):3081. doi:10.3390/electronics11193081
12. Bathalapalli VKVV, Mohanty SP, Kougianos E, Baniya BK, Rout B. PUFchain 2.0: hardware-assisted robust blockchain for sustainable simultaneous device and data security in smart healthcare. *SN Comp Sci*. 2022;3(5):344. doi:10.1007/s42979-022-01238-2
13. Raj K, Bodapati S, Chattopadhyay A. PUF-based lightweight mutual authentication protocol for internet of things (IoT) devices. In: 2024 *IEEE International Symposium on Circuits and Systems (ISCAS)*; IEEE; 2024:1–5.
14. Barbareschi M, Boi B, Cirillo F, De Santis M, Esposito C. Securing the internet of medical things using PUF-based SSI authentication. 2022.
15. Bakiri M, Guyeux C, Couchot J-F, Oudjida AK. Survey on hardware implementation of random number generators on FPGA: theory and experimental analyses. *Comput Sci Rev*. 2018;27:135–153. doi:10.1016/j.cosrev.2018.01.002

16. Maes R, Van Herrewege A, Verbauwhede I. PUFKY: a fully functional PUF-based cryptographic key generator. In: Cryptographic Hardware and Embedded Systems–CHES 2012: 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings 14; Springer Berlin Heidelberg, 2012.
17. Anandakumar NN, Das MPL, Sanadhya SK, Hashmi MS. Reconfigurable hardware architecture for authenticated key agreement protocol over binary edwards curve. *ACM Trans Reconfigurable Technol Syst*. 2018;11(2):1–19. doi:10.1145/3231743
18. Sutar S, Raha A, Kulkarni D, Shorey R, Tew J, Raghunathan V. D-PUF: an intrinsically reconfigurable DRAM PUF for device authentication and random number generation. *ACM Trans Embedded Comput Syst*. 2017;17(1):1–31. doi:10.1145/3105915
19. Kun Y, Forte D, Tehranipoor MM. Cdta: a comprehensive solution for counterfeit detection, traceability, and authentication in the iot supply chain. *ACM Trans Des Autom Electron Syst*. 2017;22(3):1–31.
20. Noor N, Silva H. Phase change memory for physical unclonable functions. *Appl Emerg Memory Technol*. 2020;59–91.
21. Cui Y, Chen Y, Wang C, Chongyan G, O'Neill M, Liu W. Programmable ring oscillator PUF based on switch matrix. In: 2020 *IEEE International Symposium on Circuits and Systems (ISCAS)*; IEEE; 2020:1–4.
22. Anagnostopoulos NA, Katzenbeisser S, Chandy J, Tehranipoor F. An overview of DRAM-based security primitives. *Cryptography*. 2018;2(2):7. doi:10.3390/cryptography2020007
23. Xiong W, Schaller A, Katzenbeisser S, Szefer J. Software protection using dynamic PUFs. *IEEE Trans Inf Forensics Secur*. 2019;15:2053–2068. doi:10.1109/TIFS.2019.2955788
24. Khelif MA, Lorandel J, Romain O, Regnery M, Baheux D, Barbu G. Toward a hardware man-in-the-middle attack on PCIe bus. *Microprocess Microsyst*. 2020;77:103198. doi:10.1016/j.micpro.2020.103198
25. Johnson AP, Chakraborty RS, Mukhopadhyay D. A PUF-enabled secure architecture for FPGA-based IoT applications. *IEEE Trans Multi-Scale Comp Syst*. 2015;1(2):110–122. doi:10.1109/TMSCS.2015.2494014
26. Mohammed A-A, Elrabaa MES, Abu-Amara M. FPGA-based symmetric re-encryption scheme to secure data processing for cloud-integrated internet of things. *IEEE Int Things J*. 2018;6(1):446–457.
27. Usama M, Aman MN, Sikdar B. Runtime self-attestation of FPGA-based IoT devices. *IEEE Int Things J*. 2024;11(20):33406–33417. doi:10.1109/JIOT.2024.3429109
28. Hategekimana F, Whitaker TJ, Pantho MJ, Bobda C. IoT device security through dynamic hardware isolation with cloud-based update. *Journal of Systems Architecture*. 2020;109:101827. doi:10.1016/j.sysarc.2020.101827
29. Stanciu A, Balan T-C, Gerigan C, Zamfir S. Securing the IoT gateway based on the hardware implementation of a multi pattern search algorithm. In: 2017 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP); IEEE; 2017:1001–1006.
30. Hamadaqa E, Adi W. Clone-resistant authentication for medical operating environment. In: 2020 Fourth World Conference on Smart Trends in Systems, *Security and Sustainability (WorldS4)*; IEEE; 2020:757–762..
31. El-Banby GM, Elazm LA, El-Shafai W, et al. Security enhancement of the access control scheme in IoMT applications based on fuzzy logic processing and lightweight encryption. *Complex Intelligent Syst*. 2024;10(1):435–454. doi:10.1007/s40747-023-01149-6
32. Arumugham S, Rajagopalan S, Rayappan JBB, Amirtharajan R. Tamper-resistant secure medical image carrier: an IWT–SVD–Chaos–FPGA combination. *Arab J Sci Eng*. 2019;44(11):9561–9580. doi:10.1007/s13369-019-03883-x
33. Raj K, Bodapati S. Fpga based light weight encryption of medical data for iomt devices using ascon cipher. In: 2022 IEEE International Symposium on Smart Electronic Systems (iSES); IEEE; 2022:196–201.
34. Pankaj S, Pandey B, Bhandari N, Bisht Bisht S, Bisht N, Budhani SK Design of Energy Efficient IoMT Electrocardiogram (ECG) Machine on 28 nm FPGA. In: Rishiwal V, Kumar P, Tomar A, Malarvizhi Kumar P, editors. Towards the Integration of IoT, Cloud and Big Data: Services, Applications and Standards. Singapore: Springer Nature Singapore; 2023pp. 43–55.
35. Joshi AM, Jain P, Mohanty SP. Secure-iGLU: a secure device for noninvasive glucose measurement and automatic insulin delivery in IoMT framework. In: 2020 *IEEE Computer Society annual symposium on VLSI (ISVLSI)*; IEEE; 2020:440–445.
36. Mohamed EI-H, Ankunda PV, Ung J, Hwu W-M. Securing the internet of medical things (IoMT) with K3S and hybrid cryptography: integrating post-quantum approaches for enhanced embedded system security. In: 2024 IEEE 17th Dallas Circuits and Systems Conference (DCAS); IEEE; 2024:1–6.
37. Khan L, Kabir F. In-depth analysis on secure and privacy-preserving smart care homes based on internet of medical things (IoMT). In: 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI); IEEE; 2024:1–6.
38. Bathalapalli VKVV, Mohanty SP, Kougianos E, Iyer V, Rout B. PUFchain 3.0: hardware-assisted distributed ledger for robust authentication in healthcare cyber-physical systems. *Sensors*. 2024;24(3):938. doi:10.3390/s24030938
39. Koppuravuri A, Pasupuleti H, Gvk S, Bapat J. A high throughput ASCON architecture for secure edge IoT devices. In: 2024 37th International Conference on VLSI Design and 2024 23rd International Conference on Embedded Systems (VLSID); IEEE; 2024:486–491.
40. Khan S, Lee W-K, Hwang SO. Scalable and efficient hardware architectures for authenticated encryption in IoT applications. *IEEE Int Things J*. 2021;8(14):11260–11275. doi:10.1109/JIOT.2021.3052184
41. Damodharan J, Michael ER, Shaikh-Husin N. High throughput present cipher hardware architecture for the medical iot applications. *Cryptography*. 2023;7(1):6. doi:10.3390/cryptography7010006
42. Prakash K. Building modeling resistant physically unclonable functions (PUFs) using adversarial machine learning. 2024.
43. Lin -C-C, Chen M-S. Enhancing reliability and security: a configurable poisoning puf against modeling attacks. *IEEE Trans Computer-Aided Design Integrated Circuits Syst*. 2022;41(11):4301–4312. doi:10.1109/TCAD.2022.3197529
44. Gollakota S, Hassanieh H, Ransford B, Katabi D, Kevin F. They can hear your heartbeats: non-invasive security for implantable medical devices. In: *Proceedings of the ACM SIGCOMM 2011 conference*; 2011:2–13.
45. Adithyan A, Nagendran K, Chethana R, Pandy G. Reverse engineering and backdooring router firmwares. In: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS); IEEE; 2020:189–193.
46. Bakhshi T, Ghita B, Kuzminykh I. A review of IoT firmware vulnerabilities and auditing techniques. *Sensors*. 2024;24(2):708. doi:10.3390/s24020708
47. Liu K, Yang M, Ling Z, et al. On manually reverse engineering communication protocols of Linux-based IoT systems. *IEEE Int Things J*. 2020;8(8).