Journal of Advanced Research in Embedded System
Volume 12, Issue 1&2 - 2025, Pg. No. 7-12
Peer Reviewed Journal

Review Article

A Survey on Embedded System Security: Cryptographic Techniques and Vulnerability Mitigation

Sandeep Kulkarni

B Tech Student, Department of Computer Engineering, Ujjain Engineering College, Ujjain, India

I N F O

E-mail Id:
sandeepkulkarni@gmail.com
Orcid Id:
http://orcid.org/0009-0009-6240-3802
How to cite this article:
Kulkarni S. A Survey on Embedded System Security: Cryptographic Techniques and Vulnerability Mitigation. *J Adv Res Embed Sys* 2025; 12(1&2): 7-12.

Date of Submission: 2025-01-13
Date of Acceptance: 2025-02-25

A B S T R A C T

Embedded systems are increasingly deployed in critical applications, including IoT devices, industrial automation, healthcare, and automotive systems. However, their growing connectivity and resource constraints make them vulnerable to a range of cyber threats. This paper provides a comprehensive survey of embedded system security, focusing on cryptographic techniques and vulnerability mitigation strategies. We discuss hardware and software-based cryptographic implementations, lightweight encryption schemes, and secure key management. Additionally, we explore common security vulnerabilities in embedded systems, including side-channel attacks, firmware tampering, and physical attacks, along with countermeasures such as secure boot, intrusion detection, and trusted execution environments. The paper concludes with insights into future research directions and emerging trends in embedded security.

Keywords: Industrial Automation, Automotive Systems, Embedded System Security, Physical Attacks

Introduction

Embedded systems are integral to modern technology, operating in diverse environments from consumer electronics to mission-critical applications such as automotive control units, industrial automation, and healthcare devices. These systems are increasingly interconnected through the Internet of Things (IoT), enabling remote monitoring, real-time data processing, and autonomous decision-making. However, this connectivity also exposes embedded devices to various cybersecurity threats, including malware attacks, unauthorized access, side-channel attacks, and cryptographic key leakage.[1]

Unlike traditional computing systems, embedded devices often operate under stringent power, memory, and processing constraints, limiting the feasibility of conventional security solutions. Many embedded applications require real-time processing with minimal energy consumption, making it challenging to integrate complex cryptographic algorithms and security frameworks. Additionally, embedded systems often have long lifecycles, and outdated security measures may render them vulnerable to newly discovered threats.

The primary security concerns in embedded systems include ensuring data confidentiality, integrity, and authentication, especially in applications involving sensitive information such as financial transactions, medical records, and military communications. Security breaches in embedded systems can lead to severe consequences, including data theft, operational failures, financial losses, and threats to human safety in critical infrastructure.[2]

Addressing these challenges requires the development of specialized cryptographic techniques, lightweight encryption algorithms, secure boot mechanisms, and hardware-based

security features. Moreover, robust vulnerability mitigation strategies such as intrusion detection systems, firmware integrity verification, and runtime monitoring are essential to strengthen embedded security against evolving cyber threats.

This paper surveys the latest cryptographic techniques and vulnerability mitigation strategies designed for embedded systems. We highlight key security challenges, examine various attack vectors, and discuss countermeasures that enhance the resilience of embedded platforms. Additionally, we explore the integration of artificial intelligence (AI) and machine learning in embedded security, as well as emerging trends such as quantum-resistant cryptography and secure hardware enclaves. By providing a comprehensive overview of embedded system security, this survey aims to guide researchers and practitioners in designing and implementing more secure embedded architectures.[3]

Cryptographic Techniques for Embedded Systems

Cryptographic techniques play a fundamental role in securing embedded systems against data breaches, unauthorized access, and cyber threats. Due to the resource-constrained nature of embedded devices, selecting an appropriate cryptographic method is crucial for balancing security, performance, and energy efficiency. Embedded cryptographic solutions are typically classified into hardware-based, software-based, and lightweight cryptographic approaches.

Hardware-Based Cryptographic Implementations

Hardware-based cryptographic modules provide dedicated security mechanisms that enhance performance and resistance to attacks. Unlike software-based solutions, hardware implementations reduce the risk of key extraction and side-channel attacks by isolating sensitive cryptographic operations within specialized hardware components.

- **Hardware Security Modules (HSMs):** These are tamper-resistant hardware devices that perform cryptographic operations such as key generation, encryption, and digital signatures. HSMs are commonly used in automotive, industrial automation, and banking systems where high security is required.
- **Trusted Platform Modules (TPMs):** TPMs are dedicated chips that store cryptographic keys, provide hardware-based authentication, and protect system integrity. They are widely used in embedded security frameworks for secure boot and firmware validation.
- **Physically Unclonable Functions (PUFs):** PUFs generate unique cryptographic keys based on the inherent physical variations of silicon chips, eliminating the need for permanent key storage. This technology enhances security against key extraction and cloning attacks.
- **Secure Elements (SEs):** SEs are embedded chips designed for secure transactions, particularly in payment systems, smart cards, and IoT devices. They provide tamper resistance and secure execution environments for cryptographic operations.[4]

Software-Based Cryptographic Approaches

Software-based cryptographic solutions offer flexibility and ease of deployment, making them a popular choice for embedded systems with general-purpose processors. These implementations rely on optimized cryptographic libraries to secure data transmission, storage, and authentication processes. However, software-based cryptography may be vulnerable to timing attacks, power analysis attacks, and code exploitation.

Key cryptographic algorithms commonly implemented in embedded systems include:

- **AES (Advanced Encryption Standard):** AES is a symmetric encryption algorithm widely used in embedded communications, securing data with key lengths of 128, 192, or 256 bits. It provides strong encryption while being computationally efficient.
- **ECC (Elliptic Curve Cryptography):** ECC is a public-key cryptographic method that offers high security with smaller key sizes compared to RSA, making it ideal for resource-constrained embedded systems. ECC is commonly used in secure IoT communication and digital signatures.
- **RSA (Rivest-Shamir-Adleman):** RSA is a widely used public-key encryption algorithm that ensures secure key exchange. However, it requires larger key sizes for strong security, which can be computationally expensive for embedded devices.
- **SHA (Secure Hash Algorithm):** The SHA family (SHA-2, SHA-3) is used for cryptographic hash functions that ensure data integrity, digital signatures, and authentication. Embedded systems use SHA for securing firmware updates and message integrity.
- **ChaCha20 and Poly1305:** A stream cipher and authentication algorithm combination that provides high-speed encryption with reduced susceptibility to side-channel attacks.[5]

Lightweight Cryptography for Embedded Systems

Embedded systems, particularly those in IoT, sensor networks, and low-power applications, require cryptographic solutions optimized for minimal computational overhead and power consumption. Lightweight cryptographic algorithms are designed to provide adequate security while maintaining efficiency in constrained environments.

- **Present, Simon, and Speck Ciphers:** These lightweight block ciphers are designed for low-power and high-speed encryption in resource-limited devices. Present

is widely used in RFID authentication, while Simon and Speck offer flexible performance trade-offs.

- **TinyCrypt:** An open-source cryptographic library designed specifically for IoT and constrained embedded systems. It includes lightweight implementations of AES, ECC, SHA, and HMAC (Hash-based Message Authentication Code).
- **LEA (Lightweight Encryption Algorithm):** LEA is optimized for real-time embedded applications, providing high-speed encryption with low power consumption. It is commonly used in smart grids and secure automotive communication.
- **Grain and Trivium Ciphers:** These stream ciphers offer lightweight encryption for wireless sensor networks and embedded authentication mechanisms.
- **SPONGENT and PHOTON Hash Functions:** Designed for lightweight hashing operations in embedded systems, providing efficient security mechanisms for message authentication and integrity verification.[6]

By integrating hardware-accelerated cryptographic functions, optimized software implementations, and lightweight encryption schemes, embedded systems can achieve strong security while maintaining performance efficiency. The choice of cryptographic technique depends on factors such as processing power, memory availability, power constraints, and required security levels.

Security Vulnerabilities and Mitigation Strategies

Embedded systems are increasingly targeted by cyber threats due to their growing connectivity and critical applications in industries such as automotive, healthcare, and industrial automation. Security vulnerabilities in embedded systems can lead to unauthorized access, data breaches, and system failures. Implementing robust mitigation strategies is essential to ensure the integrity, confidentiality, and availability of embedded devices.

Common Vulnerabilities in Embedded Systems

Embedded systems face a wide range of security threats that exploit hardware, software, and communication protocols. Some of the most prevalent vulnerabilities include:

- **Side-Channel Attacks (SCA):** These attacks exploit unintentional information leakage from a system, such as power consumption patterns, electromagnetic emissions, or execution timing variations. Common types include:
- **Power Analysis Attacks:** Use variations in power consumption to extract cryptographic keys.
- **Timing Attacks:** Measure the time taken for cryptographic operations to infer secret information.
- **Electromagnetic (EM) Attacks:** Analyze emitted signals to recover sensitive data.
- **Firmware Tampering:** Attackers modify embedded system firmware to introduce malicious code, alter system behavior, or bypass security mechanisms. This can lead to unauthorized access, backdoors, and device malfunction.
- **Man-in-the-Middle (MitM) Attacks:** These attacks intercept and manipulate communications between embedded devices. MitM attacks are particularly dangerous in IoT and wireless embedded systems, where encrypted data transmission is crucial for security.
- **Buffer Overflow Exploits:** Poor memory management in embedded firmware can lead to buffer overflow vulnerabilities, allowing attackers to execute arbitrary code or cause system crashes.
- **Hardware Trojans and Malicious Modifications:** Attackers may insert malicious hardware modifications during manufacturing or supply chain attacks, leading to hidden backdoors or unauthorized data extraction.
- **Physical Attacks:** Direct access to embedded devices allows attackers to extract sensitive data, reprogram memory, or manipulate system behavior. These attacks include:
- **Microprobing:** Physically tapping into internal circuits to read stored data.
- **Fault Injection Attacks:** Use voltage, clock, or laser-based disturbances to alter chip behavior and extract cryptographic keys.[7,8]

Mitigation Techniques

To counteract these security threats, various hardware and software-based mitigation techniques can be implemented:

- **Secure Boot:** Ensures that only authenticated and digitally signed firmware is executed at startup. This prevents unauthorized firmware modifications and protects against rootkit infections. Secure boot mechanisms use cryptographic signatures to verify software integrity before execution.
- **Code Obfuscation and Anti-Reverse Engineering Techniques:** Transform software code into an unintelligible format to prevent attackers from analyzing and modifying it. Common methods include:
- **Control Flow Obfuscation:** Alters execution paths to make reverse engineering difficult.
- **Instruction-Level Encryption:** Encrypts code to prevent static analysis.
- **Intrusion Detection Systems (IDS) for Embedded Devices:** IDS monitors real-time activities of embedded systems to detect anomalies, unauthorized access, or abnormal behavior. These systems employ machine learning and behavioral analysis to identify potential threats.
- **Tamper-Resistant Hardware:** Embedded devices can incorporate physical security measures to detect and

respond to physical attacks. These include:

- **Sensor-Based Protection:** Detects unauthorized physical access and triggers an alert or self-destruct mechanism.
- **Protective Coatings:** Shield circuitry from microprobing and invasive attacks.
- **End-to-End Encryption:** Encrypts data from source to destination, ensuring secure communication across networks. Strong encryption protocols such as AES-256 and ECC are commonly used in embedded systems to mitigate MitM attacks.
- **Trusted Execution Environments (TEE):** TEEs provide an isolated and secure environment for processing sensitive operations. They protect cryptographic keys, authentication mechanisms, and confidential data from being compromised by malware or unauthorized applications.

Hardware-Based Security Features:

- **Physically Unclonable Functions (PUFs):** Provide unique cryptographic keys derived from intrinsic chip variations, enhancing security against key extraction attacks.
- **Memory Protection Units (MPUs) and Trusted Platform Modules (TPMs):** Secure data storage and prevent unauthorized memory access.
- **Firmware Integrity Verification and Secure Updates:** Implementing cryptographically signed firmware updates ensures that only trusted software is installed. Techniques such as Over-The-Air (OTA) updates with secure boot verification prevent firmware rollback and unauthorized modifications.
- **AI-Based Anomaly Detection:** Machine learning models can be used to monitor embedded system behavior and detect deviations that may indicate security breaches, malware infections, or system tampering.[9,10]

By integrating these mitigation strategies, embedded systems can significantly enhance their security posture, ensuring resilience against cyber threats while maintaining performance and efficiency.

Emerging Trends and Future Directions

The field of embedded system security is rapidly evolving to counter increasingly sophisticated cyber threats. As embedded devices become more interconnected, security mechanisms must adapt to new challenges. Several emerging trends and future directions are shaping the security landscape of embedded systems.

AI-Powered Security Solutions

Artificial intelligence (AI) and machine learning (ML) are being leveraged to enhance embedded security by enabling real-time threat detection, anomaly identification, and predictive defense mechanisms. AI-driven security solutions provide:

- **Intrusion Detection and Prevention Systems (IDPS):** AI models analyze system behavior and detect deviations that may indicate malware infections, unauthorized access, or network intrusions.
- **Behavior-Based Authentication:** AI-driven behavioral analytics use device usage patterns to enhance authentication mechanisms, improving security beyond traditional passwords or PINs.
- **Automated Threat Intelligence:** Machine learning models can process vast amounts of security data to identify new attack patterns, allowing embedded systems to respond proactively.
- **Self-Healing Security Systems:** AI-powered embedded systems can automatically detect, isolate, and mitigate threats without requiring human intervention.[11]

Post-Quantum Cryptography (PQC)

The rise of quantum computing poses a significant threat to traditional cryptographic algorithms such as RSA and ECC, which rely on mathematical problems that quantum computers can solve efficiently. Post-quantum cryptography (PQC) aims to develop encryption algorithms resistant to quantum attacks. Key developments include:

- **Lattice-Based Cryptography:** Uses complex mathematical lattice structures to ensure security against quantum decryption techniques.
- **Code-Based Cryptography:** Employs error-correcting codes for cryptographic functions that remain secure even in a post-quantum era.
- **Multivariate Polynomial Cryptography:** Relies on solving complex polynomial equations, which are computationally hard even for quantum computers.
- **Hybrid Cryptographic Models:** Combines classical and post-quantum cryptographic techniques to ensure a smooth transition to quantum-resistant security frameworks.

PQC is becoming increasingly important for embedded systems used in industries such as defense, automotive, and healthcare, where long-term data security is crucial.[12]

Blockchain for Embedded Security

Blockchain technology is gaining traction in securing embedded and IoT ecosystems by providing decentralized, tamper-resistant security mechanisms. Key benefits include:

- **Decentralized Authentication:** Blockchain-based identity management prevents unauthorized access to embedded systems by eliminating centralized authentication vulnerabilities.
- **Immutable Data Storage:** Ensures that security logs, firmware updates, and critical data remain unaltered, reducing the risk of data tampering.
- **Smart Contracts for Automated Security Policies:** Blockchain-based smart contracts enable automated

enforcement of security policies, such as device revocation and access control.
- **Secure Firmware Updates:** Blockchain can be used to verify and authenticate firmware updates, preventing the installation of malicious software.

Adopting blockchain for embedded security enhances trust, transparency, and resilience against cyber threats in IoT, industrial automation, and critical infrastructure applications.[13]

Zero-Trust Architectures in Embedded Systems

The traditional security model assumes trust within a network perimeter, but modern cyber threats demand a "zero-trust" approach, where every access request is continuously verified. Key principles of zero-trust security in embedded systems include:

- **Continuous Authentication and Authorization:** Embedded devices must verify the identity and security posture of users, applications, and connected systems before granting access.
- **Micro-Segmentation:** Divides network resources into isolated segments to limit the impact of a security breach and prevent lateral movement of attackers.
- **Least Privilege Access Control:** Ensures that devices, applications, and users only have the minimum required permissions to perform their functions.
- **Secure Boot and Trusted Execution:** Embedded systems adopt secure boot mechanisms and Trusted Execution Environments (TEEs) to ensure that only verified and authorized software runs on devices.[14]

Hardware-Rooted Security and Secure Enclaves

Future embedded systems will integrate more advanced hardware security features to prevent unauthorized access and physical tampering. Innovations include:

- **Secure Enclaves:** Encrypted hardware environments that execute critical operations in isolation, preventing exposure to malware.
- **Secure Elements (SE):** Dedicated chips for cryptographic operations, commonly used in secure payments and identity verification.
- **Hardware Security Modules (HSMs):** Specialized devices designed to generate, store, and manage cryptographic keys securely.
- **Physical Layer Security:** Techniques such as dynamic encryption, spread spectrum modulation, and PUFs to enhance the resilience of embedded systems against physical and side-channel attacks.[15]

AI-Driven Adaptive Security Models

Future embedded security frameworks will integrate AI-driven adaptive security models that can dynamically adjust security policies based on real-time threat intelligence. These systems will:

- Detect evolving threats autonomously and adjust defenses accordingly.
- Utilize federated learning to improve security models without compromising data privacy.
- Enhance real-time risk assessment for embedded applications in automotive, medical devices, and industrial control systems.[16]

Next-Generation Lightweight Security Solutions

As embedded systems continue to be deployed in constrained environments, next-generation lightweight security solutions are emerging to provide robust protection without excessive computational overhead. These include:

- **Efficient Cryptographic Primitives:** Optimization of AES, ECC, and other cryptographic functions to reduce energy consumption.
- **Ultra-Low-Power Security Protocols:** Lightweight authentication and encryption protocols designed for IoT and battery-operated embedded systems.
- **Secure AI Models on Edge Devices:** Implementing privacy-preserving AI models with minimal computational requirements.[17]

Conclusion

As embedded systems become increasingly pervasive and interconnected across various domains—including IoT, industrial automation, automotive, healthcare, and smart infrastructure—securing them against evolving cyber threats is of paramount importance. Unlike traditional computing systems, embedded devices often operate under strict resource constraints, making conventional security solutions impractical. Consequently, tailored cryptographic techniques and advanced security frameworks are essential for ensuring data integrity, confidentiality, and system resilience.

Cryptographic techniques such as hardware-based encryption, lightweight ciphers, and efficient key management solutions form the backbone of embedded security. Hardware-based security implementations, including Trusted Platform Modules (TPMs) and Physically Unclonable Functions (PUFs), offer robust protection against unauthorized access and cryptographic key leakage. Similarly, software-based cryptographic methods, such as AES, ECC, and SHA, provide scalable and effective security solutions for embedded applications. The integration of lightweight cryptography, specifically designed for constrained devices, further enhances the feasibility of secure embedded systems.

Beyond encryption, vulnerability mitigation strategies play a crucial role in strengthening embedded security. Secure boot mechanisms ensure that only authenticated firmware is executed, mitigating risks associated with malicious code injection and firmware tampering. Trusted Execution

Environments (TEEs) isolate critical processes, preventing attackers from accessing sensitive data. Intrusion Detection Systems (IDS) monitor system activity in real time, detecting and responding to security anomalies before they cause significant damage. Furthermore, secure hardware design, tamper-resistant architectures, and end-to-end encryption protect embedded devices from a wide range of attacks, including side-channel attacks, man-in-the-middle (MitM) threats, and physical intrusions.

Looking ahead, the future of embedded security will be shaped by several emerging trends and advanced security mechanisms. Artificial intelligence (AI)-driven security solutions will enable real-time threat detection, adaptive security policies, and autonomous risk mitigation. Post-quantum cryptography (PQC) will address the potential threats posed by quantum computing, ensuring long-term data protection. Blockchain technology will introduce decentralized authentication and tamper-proof data storage, improving trust and transparency in embedded environments. Additionally, zero-trust architectures, micro-segmentation, and hardware-rooted security measures will further enhance the resilience of embedded platforms against cyberattacks.

To ensure the security of next-generation embedded systems, industry collaboration, continuous research, and regulatory advancements will be necessary. Standards for embedded security must evolve alongside technological advancements to address new vulnerabilities and attack vectors. By integrating AI-powered security frameworks, post-quantum cryptographic models, and blockchain-based solutions, the next generation of embedded devices can achieve robust, scalable, and resilient security, ensuring their safe deployment in critical applications.

In conclusion, securing embedded systems is an ongoing challenge that requires a multi-layered approach, combining cryptographic protection, hardware and software security measures, and emerging technologies. As embedded systems continue to play a vital role in modern infrastructure, implementing advanced security mechanisms will be key to safeguarding them against sophisticated cyber threats. The future of embedded security lies in proactive defense strategies, continuous innovation, and the seamless integration of cutting-edge security technologies.

References

1. Jacquemart Q. Applied Cryptography.
2. Menezes AJ, Van Oorschot PC, Vanstone SA. Handbook of applied cryptography. CRC press; 2018 Dec 7.
3. Paar C, Pelzl J. A Textbook for Students and Practitioners. Understanding Cryptography, Springer. 2009.
4. Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM. 1978 Feb 1;21(2):120-6.
5. Kocher PC. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. InAdvances in Cryptology—CRYPTO'96: 16th Annual International Cryptology Conference Santa Barbara, California, USA August 18–22, 1996 Proceedings 16 1996 (pp. 104-113). Springer Berlin Heidelberg.
6. Kelsey J, Schneier B, Wagner D, Hall C. Side channel cryptanalysis of product ciphers. InComputer Security—ESORICS 98: 5th European Symposium on Research in Computer Security Louvain-la-Neuve, Belgium September 16–18, 1998 Proceedings 5 1998 (pp. 97-110). Springer Berlin Heidelberg.
7. Güneysu T, Paar C. Ultra high performance ECC over NIST primes on commercial FPGAs. InCryptographic Hardware and Embedded Systems–CHES 2008: 10th International Workshop, Washington, DC, USA, August 10-13, 2008. Proceedings 10 2008 (pp. 62-78). Springer Berlin Heidelberg.
8. Smart NP. Cryptography made simple. Springer; 2016.
9. Bhunia S, Tehranipoor MM. Hardware security: a hands-on learning approach. Morgan Kaufmann; 2018 Oct 30.
10. Huang Z, Wang Q, Chen Y, Jiang X. A survey on machine learning against hardware trojan attacks: Recent advances and challenges. IEEE Access. 2020 Jan 8;8:10796-826.
11. Lesi V, Jovanov I, Pajic M. Security-aware scheduling of embedded control tasks. ACM Transactions on Embedded Computing Systems (TECS). 2017 Sep 27;16(5s):1-21.
12. Suh GE, Devadas S. Physical unclonable functions for device authentication and secret key generation. InProceedings of the 44th annual design automation conference 2007 Jun 4 (pp. 9-14).
13. Wang Y, Wang J, Zhang W, Zhan Y, Guo S, Zheng Q, Wang X. A survey on deploying mobile deep learning applications: A systemic and technical perspective. Digital Communications and Networks. 2022 Feb 1;8(1):1-7.
14. Zhou Y, Feng D. Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing. Cryptology ePrint Archive. 2005.
15. Wallrabenstein JR. Practical and secure IoT device authentication using physical unclonable functions. In2016 IEEE 4th international conference on future internet of things and cloud (FiCloud) 2016 Aug 22 (pp. 99-106). IEEE.
16. Conti M, Dragoni N, Lesyk V. A survey of man in the middle attacks. IEEE communications surveys & tutorials. 2016 Mar 29;18(3):2027-51.
17. Rajendran J, Sam M, Sinanoglu O, Karri R. Security analysis of integrated circuit camouflaging. InProceedings of the 2013 ACM SIGSAC conference on Computer & communications security 2013 Nov 4 (pp. 709-720).