

Literature Review Table

Dhruv Sunil Bhatia

24CSB0A19

CSE - A

Q119 - Secure OTA Update Compiler

Comparative Literature Review

Paper	Main Focus / Contribution	Approach / Technique	Strengths
Comparative Analysis of Firmware Security: A Proactive Paradigm (BBAD)	Behavior-Based Anomaly Detection (BBAD) for firmware security	Machine learning + real-time behavioral monitoring	Proactive threat detection; adaptable to evolving threats
Static Analysis for Security (McGraw)	Security testing through static source-code analysis	Rule-based static pattern detection	Early vulnerability detection; scalable; automated
Secure Firmware Updates for Constrained IoT Devices (Reality Check)	Practical secure OTA implementation using open standards (e.g., SUIT)	Cryptographic signatures; standards-compliant prototype	Feasible on constrained devices (<32kB RAM)
Control-Flow Integrity (CFI)	Control-flow safety to prevent code hijacking	CFG-based runtime enforcement via binary rewriting	Strong formal guarantees; prevents exploit classes
Secure Compilation to Protected Module Architectures	Fully abstract compilation preserving source-level security	Compiler targeting protected module architectures	Formal proof of security preservation; minimal runtime overhead
Robustly Safe Compilation (RSC)	Safety-property preservation against adversarial contexts	Alternative to fully abstract compilation	More efficient than FAC; security-focused
Verified Secure Compilation for Mixed-Sensitivity Concurrent Programs	Verified compiler preserving noninterference	Formal verification in Isabelle/HOL	Strong confidentiality guarantees at assembly level

Secure and Lightweight FOTA Mechanism (Electronics 2025)	Lightweight FOTA resistant to MITM attacks	Dual-XOR encryption + DEFLATE compression	Reduced latency; lower memory usage; multi-channel transmission
Secure Firmware OTA Updates for IoT: Survey (IoT Journal 2022)	Comprehensive OTA mechanisms and threat survey	Classification of SoCs + trust chain analysis	Identifies OTA challenges; trust chain importance
Firmware Over-the-Air Programming Techniques for IoT Networks – A Survey	OTA programming mechanisms and limitations	Delta updates; compression; bootloaders	Highlights resource constraints and update stages

Gap Statement

Dhruv Sunil Bhatia

24CSB0A19

CSE - A

Q119 - Secure OTA Update Compiler

While secure OTA mechanisms and secure boot frameworks improve firmware safety at runtime, they do not enforce update security requirements at compile time. As a result, insecure firmware update logic can still compile and be deployed, exposing devices to attacks.