Review in Control Flow Attestation (C-FLAT) Recent Improvement and Research Focus

Zan Xie
University of California, Los Angeles, USA
zxie425.ucla.edu

***Abstract*—In the rapidly increasing attention to embedded systems security, Control-Flow Attestation (C-FLAT) has emerged as an important technique for ensuring the integrity of software against runtime attacks. This review explores the core foundation and variations of C-FLAT, highlighting the transition from purely software-based schemes to innovative hardware-assisted and interrupt-safe methodologies, underlining the critical balance between ensuring rigorous security and maintaining system performance. We examine four pivotal works: C-FLAT, LO-FAT, ISC-FLAT, and MGC-FA through fully analyzing the motivation behind these methodologies, the solution to the problem, evaluation in case studies, and contribution to the system security field in a way that addresses the limitation of its predecessors.**

***Index Terms*—Control-Flow Attestation, Embedded Platform Security.**

I. Introduction

NOWADAYS, an increasing number of embedded devices have adopted to daily routine leading to increasing attention on device security. Remote Attestation which can verify the integrity and authenticity of a device's software and hardware configuration from a remote location is a competitive candidate for ensuring device security. However, such methodology only ensures the integrity of binaries and not of their execution. In particular, it does not capture the attack towards program control flow [1]. An advancement methodology, Control-Flow Attestation is invented to provide that additional layer of security on top of Remote Attestation. An overview of C-FLAT is illustrated in Figure1. The verifier pre-processes the program and includes all possible paths in a control flow graph. The hash measurement of individual paths is calculated and stored before sending the challenge to the prover. The prover receives the challenge and computes a hash corresponding to its execution status. By browsing through the pre-computed hash and finding the matching hash measure, the verifier can determine which path the device is executing and prevent malicious execution that does not belong to the program.

As C-FLAT sets the benchmark for software integrity verification, limitations like performance overhead and uninterruptible verification process persist. Recent research on C-FLAT branches out to variations including LO-FAT, a hardware-based solution that mitigates performance overhead [2]; MGC-FA, which introduces mutable granularity in attestation through a probabilistic model [3]; and ISC-FLAT, tailored for real-time operations by enabling secure, interactive execution [4].

This paper summarizes each paper, comments on their creative approach to addressing the problem, and synthesizes interconnections among these C-FLAT advancements, providing a comprehensive outlook on the future trajectory of embedded systems security. The following section II provides a detailed summary of those papers through their motivation, solutions, and evaluation methodology. Section III distinguishes the novelty and contribution to system security. Section IV evaluates the performance and comments on its persuasiveness. Section V discusses the strengths and weaknesses of each paper. Section VI concludes this review.

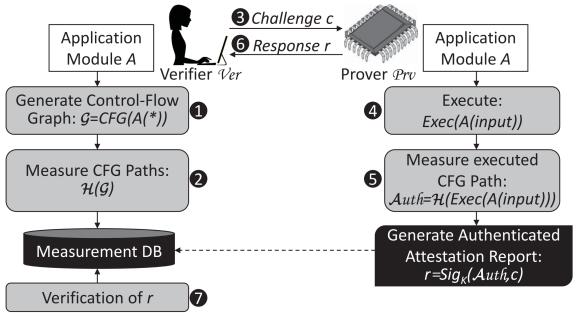

Fig. 1. Implementation of C-FLAT

II. Summary

A. C-FLAT: Control-Flow Attestation for Embedded Systems Software

The most recent remote attestation approaches are static in nature. This paper developed a Control-Flow Attestation to provide a more robust attestation mechanism capable of detecting and mitigating runtime attacks that static remote attestation methods fail to catch. C-FLAT employs a cumulative hash-based scheme to track and verify the control-flow path of an application during its execution. This method involves instrumenting the application to monitor its control-flow events and generate an attestation response that can be verified remotely. The solution effectively counters various runtime attacks, including control-flow and data-oriented exploits, by ensuring any deviation from the legitimate control-flow path is detected and reported.

The paper evaluates C-FLAT's efficacy and efficiency through a case study involving a cyber-physical system, specifically a syringe pump controlled by embedded software. Demonstrating C-FLAT's ability to detect runtime attacks that alter the application's control-flow path. Highlighting the ease of implementation and the feasibility of deploying C-FLAT in real-world embedded systems settings. Analyzing the overhead introduced by C-FLAT in terms of execution time and resource usage shows that the system incurs a manageable performance penalty.

B. LO-FAT: Low-Overhead Control Flow Attestation in Hardware

This paper addresses the limitations of software-based control-flow attestation by eliminating the need for software instrumentation and significantly reducing performance overhead. LO-FAT's architecture includes key components like the branch filter and loop monitor to efficiently track and verify control-flow paths in hardware. It captures control-flow events parallel to the main processor operation, hashing them into a cumulative authenticator without causing processor stalls. This method ensures accurate, real-time attestation of the program's execution path, enhancing security against runtime exploits. By leveraging existing processor features and commonly used IP blocks, LO-FAT achieves control-flow attestation without requiring software

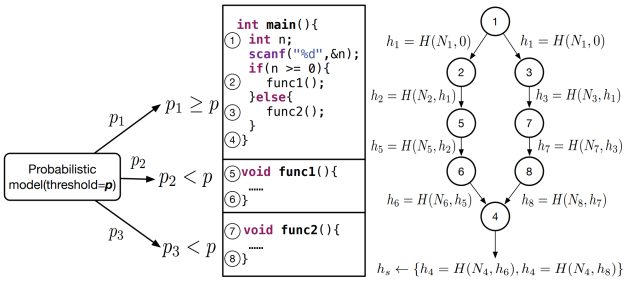

Fig. 2. The prediction and hash computation procedure

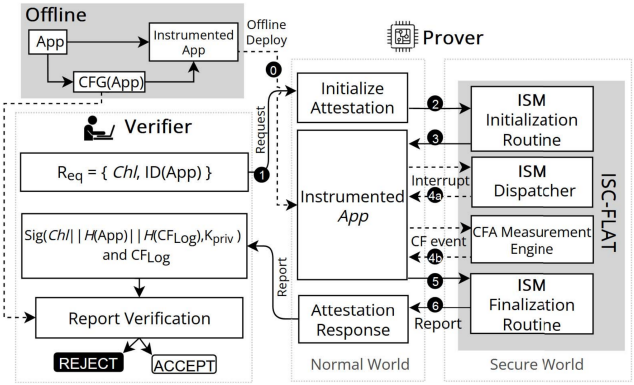

Fig. 3. Implementation of ISC-FLAT protocol

instrumentation, ensuring compatibility with legacy software and eliminating performance penalties associated with software-based solutions.

The paper presents a proof-of-concept implementation of LO-FAT on a RISC-V SoC, evaluating its functionality, performance, security, and hardware resource utilization. Simulations confirm its ability to capture and compress control flow accurately without performance overhead. The evaluation demonstrates LO-FAT's efficiency and practicality for embedded systems, with minimal logic overhead and reasonable memory requirements.

C. A Probability Prediction Based Mutable Control-Flow Attestation Scheme on Embedded Platforms

Mutable Control-Flow Attestation (MGC-FA) provides a scheme that balances runtime efficiency with security guarantees by selectively applying fine-grained attestation only to vulnerable parts of the program. Figure2 depicts the prediction logic and hashing procedure. MGC-FA employs a machine learning model to predict the vulnerability probability of each function in a program, denoted as P. Functions examined vulnerability probability P_i that is greater than a predefined probability threshold are subjected to fine-grained attestation, while others are checked more lightly at a coarse-grained level. The hashing procedure returns one hash for one control flow graph instead of multiple hashs corresponding to individual paths in traditional C-FLAT. This approach not only enhances security by focusing on critical areas of the software but also improves efficiency by reducing the overall attestation burden on non-vulnerable parts.

The paper evaluates MGC-FA through implementation on Raspberry Pi devices, focusing on its ability to detect control-flow attacks, including both control-data and non-control-data attacks. The evaluation considers the scheme's security guarantees by introducing vulnerabilities into test programs and assessing MGC-FA's effectiveness in detecting these vulnerabilities under different attestation granularities. Furthermore, the impact on runtime efficiency is measured by comparing the performance overhead of MGC-FA against traditional fine-grained and coarse-grained attestation schemes.

D. ISC-FLAT: On the Conflict Between Control Flow Attestation and Real-Time Operations

The paper identifies a fundamental limitation in current Control Flow Attestation (CFA) methods—their inability to handle interrupts during software attestation—which restricts their practicality in real-world applications where interrupts are essential for operation. It introduces an Interrupt-Safe Control Flow Attestation (ISC-FLAT) method that uniquely enables the handling of interrupts without compromising the authenticity of CFA reports. Figure3 depicts the attestation flow. The procedure is similar to traditional C-FLAT, except that ISC-FLAT is equipped with an Interrupt Safe Module which keeps track of the execution state and can trace back to the last execution state. The methodology ensured the interruption did not interfere with the control flow path of the software being attested, making the real-time interaction possible. It utilizes the TEE for secure context switching between the attested application and interrupt service routines, thereby preserving the authenticity of the control flow attestation process.

The evaluation involves implementing ISC-FLAT on a real-world MCU and measuring its impact on runtime overhead and security. The case study focuses on a syringe pump system, demonstrating ISC-FLAT's efficacy in detecting control flow attacks and its compatibility with real-time operations.

III. Novelty and Contribution

C-FLAT contributes a novel scheme for attesting the execution path of applications on embedded devices, capturing runtime behavior beyond what static attestation can achieve. The implementation of C-FLAT on Raspberry Pi using ARM TrustZone hardware security extensions demonstrates practical applicability.

Unlike C-FLAT, which requires software instrumentation and incurs significant performance overhead, LO-FAT is designed as a hardware-based control-flow attestation architecture that provides similar security guarantees but without associated overhead. LO-FAT's hardware-centric design offers a more efficient and broadly applicable solution, particularly suited to environments where preserving performance and legacy software compatibility is crucial.

MGC-FA extends the foundational ideas of control-flow attestation by introducing a mutable granularity approach. While C-FLAT focuses on fine-grained control-flow attestation, MGC-FA advances this concept by selectively applying fine-grained or coarse-grained attestation based on the predicted vulnerability of program parts. This evolution addresses one of the key limitations of C-FLAT—its impact on runtime efficiency—by offering a more flexible and performance-aware solution and introducing the usage of machine learning to tackle attestation limitations. MGC-FA builds upon the premise of C-FLAT by offering a more nuanced approach to attestation that considers the varying security needs of different program parts, thus extending C-FLAT's applicability with efficiency improvements.

ISC-FLAT directly addresses and resolves a significant limitation of the foundational C-FLAT model by enabling interrupt-safe control flow attestation. While C-FLAT laid the groundwork for attesting to the control-flow integrity of applications on embedded devices, its practical application was limited by the requirement to disable interrupts. ISC-FLAT extends the utility of C-FLAT to real-time embedded applications by ensuring that interrupts can be safely processed without compromising the security objectives of control

flow attestation. Furthermore, ISC-FLAT can be implemented without necessitating hardware changes, making it compatible with off-the-shelf MCUs. This represents a significant step forward in the domain of embedded systems security, broadening the applicability of CFA to a wider range of use cases where real-time responsiveness is crucial.

IV. Performance Evaluation

C-FLAT conducted two case studies on real applications, a Syringe Pump and a Soldering Iron Temperature Controller. The syringe pump is designed to dispense precise quantities of fluid. The case study involved uploading an application from Arduino to a Raspberry Pi platform, instrumenting the binary for C-FLAT, and then performing control-flow attestation during the application's execution. Different inputs were provided to the system to simulate real-world usage and to demonstrate C-FLAT's ability to detect unauthorized control-flow paths. As a secondary case study, C-FLAT was implemented on a larger and more complex temperature controller. This application controls the temperature of a soldering iron through a proportional-integral-derivative (PID) system, making it representative of a broad class of temperature control systems. The evaluation highlighted C-FLAT's scalability and its minimal performance overhead, 237 us on average 100 runs with a standard deviation of 1 us on more expensive applications.

The syringe pump, which served as a representation of the cyber-physic system that is small but requires high precision was proved to be secure with C-FLAT methodology. Although there is difficulty implementing C-FLAT on complex applications, the authors moved on to evaluate C-FLAT performance on the temperature controller and demonstrated an acceptable overhead introduced to the program. These two case studies clearly showed the capability of C-FLAT and demonstrated the authors' consideration. We found the evaluation convincing.

LO-FAT was evaluated by integrating it with Pulpino, an open-source RISC-V-based micro-controller SoC, and performing cycle-accurate functional simulation while Pulpino executed code segments from Open Syringe Pump. The evaluation ensured that LO-FAT accurately captures and compresses the control flow of an uninstrumented application while incurring no performance overhead compared to existing software-based solutions like C-FLAT. The paper further accesses LO-FAT's ability to provide accurate, complete, authentic, and refreshed attestation against adversary models where attackers were assumed to have full control over the data memory, by recording the full control flow graph in the verifier's end.

Implementing LO-FAT on open-source RISC-V-based SoC (Pulpino) and providing clear metrics on hardware utilization (registers, LUTs, BRAMs) on a Virtex-7 XC7Z020 FPGA making the evaluation convincing. The demonstration of a real-world application syringe pump and the challenge in memory requirement added credibility to the evaluation. However, it would be beneficial to see comparative analyses with existing software-based C-FLAT in terms of security and performance. Additionally, the evaluation on a broader range of real-world applications could further solidify LO-FAT's capability.

MGC-FA was implemented on top of Raspbian on Raspberry Pi with TrustZone extension. The methodology was evaluated on a syringe pump and successfully attested control flow activities with the assistance of a self-developed vulnerability predictor. The paper neither compared the efficiency with traditional C-FLAT nor examined the performance on a broader range of applications, lacking connections to related works in C-FLAT. However, the idea of combining machine learning and remote attestation is appealing while proving to be successful. On top of that, the provided figures and tables clearly illustrated the training process, convincing the capability of the methodology.

ISC-FLAT was implemented on three real-world applications, a syringe pump, an Ultrasonic Ranger, and a fire sensor. The measurement of runtime overhead and energy consumption were collected in figure4. The table included the performance of the baseline CFA Engine with and without Interrupt Safe Module (ISM) in all three cases. Through comparison, ISC-FLAT introduced minimal overhead at low interrupt frequencies (less than 0.1% at up to 1kHz), with overhead becoming noticeable but still manageable at higher frequencies (up to 6.4% at 70kHz for runtime). Energy consumption overhead mirrors this trend but at roughly half the rate of runtime overhead. Additionally, two attacks were deployed to address the security concerns, a return address attack and ISC-FLAT configuration deactivation. The methodology was able to detect illegal access and generate fault exceptions.

Three real-world applications that can represent a wide range of embedded devices were evaluated, as well as including tons of analysis on overhead measurement of individual components of ISC-FLAT, energy consumption, Trusted Computing Base sizing, and security tests. The methodology is particularly convincing and innovative.

The threat model for Control-Flow Attestation (C-FLAT) assumed attackers can perform runtime attacks that alter the control-flow of applications but does not account for hardware-based attacks or attacks exploiting vulnerabilities outside the scope of control-flow integrity, such as side-channel attacks which was assumed to be covered by Data Execution Prevention Unit (DEP). LO-FLAT is primarily concerned with software-based attacks that manipulate control flow, similar to C-FLAT, but with an emphasis on mitigating the performance impact of attestation. MGC-FA focused on runtime attacks, especially on control-flow integrity, and assumed that attackers might exploit specific software vulnerabilities to execute control-flow hijacking attacks. ISC-FLAT expanded on the models of previous works by considering attacks that exploit interrupt handling mechanisms to bypass control-flow attestation protections.

V. Discussion

The first paper that introduced C-FLAT [1] is structured logically, with distinct sections for the introduction, problem setting, system model, C-FLAT design, implementation, evaluation, security considerations, and discussion. The C-FLAT design is described in detail, with a high-level description and a discussion of challenges and solutions, which demonstrates the authors' deep understanding of the subject matter.

The paper about LO-FLAT [2] effectively introduces the need for control-flow attestation in embedded systems and identifies the limitations of existing software-based solutions, regarding their performance overhead. Particularly, the security analysis which addresses potential concerns about the hardware approach's ability to withstand various attack vectors makes the solution credible.

The MGC-FA paper's proposition [3] of using mutable granularity for CFA based on a probability prediction model is highly innovative. The combined use of CFA methodology with the machine learning model is a breakthrough in the field. Therefore, the focus shifted toward educating the training procedure, including collecting suitable data and finding a proper threshold value for a program was quite a smart move for emphasizing its innovation.

The ISC-FLAT paper [4] filled in a large gap of incompatibility with real-time operation in CFA methodology. While its work is innovative, the paper evaluated the methodology scientifically by analyzing the performance of individual components of the Interrupt

OVERHEAD GENERATED BY ISC-FLAT MODULES OVER THE BASELINE.

	E1 – Syringe Pump				E2 – UltrasonicRanger				E3 – Fire Sensor			
	CFA Engine		CFA Engine + ISM		CFA Engine		CFA Engine + ISM		CFA Engine		CFA Engine + ISM	
Interrupt Frequency	Runtime	Energy	Runtime	Energy	Runtime	Energy	Runtime	Energy	Runtime	Energy	Runtime	Energy
100Hz	19.3%	9.1%	19.3%	9.1%	4.5%	2.4%	4.5%	2.4%	10.2%	4.9%	10.2%	4.9%
1kHz	19.3%	9.1%	19.3%	9.1%	4.5%	2.4%	4.5%	2.4%	10.2%	4.9%	10.3%	4.9%
10kHz	19.5%	9.2%	20.9%	10.1%	4.6%	2.4%	5.7%	2.6%	10.5%	4.9%	10.9%	5.1%
30kHz	21.3%	9.7%	25.2%	12.9%	6.5%	3.6%	8.4%	3.7%	11.7%	5.4%	13.1%	6.7%
50kHz	24.8%	11.5%	29.6%	14.7%	10.3%	5.1%	16.1%	6.4%	13.0%	6.1%	18.8%	8.6%
70kHz	28.7%	14.0%	35.1%	17.5%	13.7%	6.3%	18.8%	9.0%	15.8%	7.3%	21.6%	9.9%

Fig. 4. ISC-FLAT overhead evaluation

Safe Module (ISM), providing detailed experiment results in real-world application, and conducting rigorous security tests against various attack vectors.

VI. CONCLUSION

We walked through the original C-FLAT methodology which laid the groundwork for control-flow attestation and advancements of LO-FLAT, introducing a hardware–base approach to minimize performance overhead; MGC-FA, presenting a mutable granularity approach, allowing for dynamic adjustment of attestation based on the predicted vulnerability of different program parts; and ISC-FLAT, further advancing the field by enabling interrupt-safe control-flow attestation.

A clear trend towards optimizing the efficiency and minimizing the performance impact of control-flow attestation methods is evident across these developments. This trend is crucial for their practical applicability in real-time systems and IoT devices, where performance cannot be significantly compromised for security. While focused on control-flow integrity, the methodologies implicitly acknowledge the broader security landscape by addressing scenarios like interrupt handling (ISC-FLAT) and mutable attestation granularity (MGC-FA). These approaches suggest an evolving perspective that considers a wider array of threats and operational contexts. The progression from C-FLAT to more refined attestation schemes showcases the dynamic nature of research in embedded systems security. Future directions will likely involve the integration of hardware and software attestation techniques, exploration of machine learning models for vulnerability prediction and attestation adaptability, and comprehensive security models that encompass not just control-flow integrity but also data-flow integrity and side-channel resistance.

REFERENCES

[1] T. Abera, N. Asokan, L. Davi, J.-E. Ekberg, T. Nyman, A. Paverd, A.-R. Sadeghi, and G. Tsudik, "C-flat: control-flow attestation for embedded systems software," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 743–754.

[2] G. Dessouky, S. Zeitouni, T. Nyman, A. Paverd, L. Davi, P. Koeberl, N. Asokan, and A.-R. Sadeghi, "Lo-fat: Low-overhead control flow attestation in hardware," in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2017, pp. 1–6.

[3] J. Hu, D. Huo, M. Wang, Y. Wang, Y. Zhang, and Y. Li, "A probability prediction based mutable control-flow attestation scheme on embedded platforms," in *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2019, pp. 530–537.

[4] A. J. Neto and I. D. O. Nunes, "Isc-flat: On the conflict between control flow attestation and real-time operations," in *2023 IEEE 29th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2023, pp. 133–146.