Research Statement - Mulong Luo

Overview. Computer system components from different levels of abstraction—including circuits, architecture, operating systems, and applications—work coherently to drive the information world forward. Securing modern computer systems against an ever-evolving threat landscape is major challenge that requires new approaches. First, modern general-purpose computer systems have become very complex and they are error-prone and laborious to inspect manually for security. Formal methods are typically constrained by scalability, and empirical testing methods can be time consuming. AI-enabled attackers also increase the offensive capabilities targeting these computer systems. Second, novel applications, especially AI applications such as compound AI [1] and embodied AI [2], are making computer systems increasingly heterogeneous. Specialized hardware, such as hardware accelerators and in/near-memory processing architectures, adds new attack surfaces. The interaction between AI algorithms and underlying systems further complicates efforts to secure these AI systems.

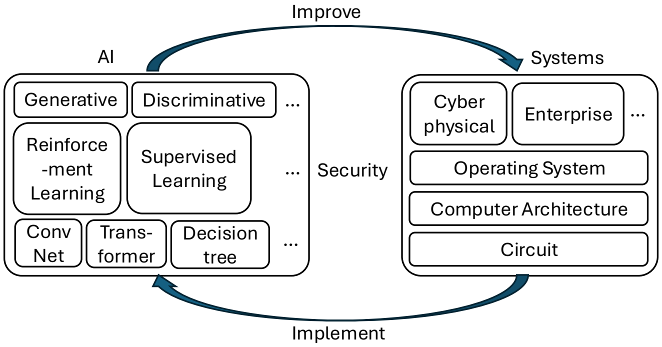

Figure 1: **The interaction of AI with systems in the security context. AI methods can be used to improve system security, and systems that implement AI have unique security concerns.**

To mitigate the security risks of modern computer systems, we need to address the security issues of both general-purpose systems and AI systems. As Figure 1 shows, AI methods can be used to improve system security, and the security of systems used to implement AI needs to be evaluated. The two main thrusts of my research are:

- Using AI methods for security of general-purpose computer systems;
- Finding vulnerabilities in AI systems.

First, emerging AI methods such as large language models, reinforcement learning, deep neural networks, and traditional supervised learning have demonstrated exceptional performance in various tasks, achieving superhuman levels in some areas. A major aspect of my graduate research is using AI methods to evaluate and improve system security. Specifically, one direction of my past work focused on general-purpose computer architecture security problems using reinforcement learning.

Second, AI-integrated computer systems—such as compound AI systems, embodied AI systems, cloud AI systems, and hardware neural networks have unique interfaces, architectures, and hardware specializations, which make their vulnerabilities different from general-purpose systems. Thus, another main thrust of my research is finding vulnerabilities in AI systems.

In the rest of the statement, I first describe my research on using AI methods for general-purpose system security, then I describe the second thrust on finding vulnerabilities in AI systems. Finally, I will outline future directions.

AI Methods for General-Purpose System Security

In this thrust, I use AI methods, specifically reinforcement learning (RL), to tackle CPU microarchitecture security challenges, focusing on the attack and detection of side-channel attacks in microprocessors. The aggressive performance optimizations in modern microprocessors can result in security vulnerabilities like cache timing attacks and speculative execution attacks. However, the current study of microarchitectural vulnerabilities relies heavily on heuristics and human expertise, which is inefficient and laborious.

In contrast, RL offers a promising approach for exploration of microarchitectural attacks. RL employs an agent controlled by a model to interact with an environment. The agent performs one action at each iteration, and the environment then provides feedback in the form of the corresponding state and reward. This state and reward are used to update the model within the agent. RL has been demonstrated to be useful to achieve superhuman performance in Go [3], video games [4], chip design [5], algorithm design [6], [7], and biology [8]. However, applying RL to a microarchitecture security problem faces several challenges in the formulation, generalization, and adaptivity. My research work AutoCAT [9], MACTA [10], AlphaEvict [11], SpecRL [12] addresses these challenges and successfully demonstrate the capability of RL in solving a wide range of RL security problems.

RL for Exploration of Cache Timing and Speculative Execution Attacks. One of the main challenges in exploring cache timing attacks as an RL problem is the formulation. Cache timing attacks leverage cache access timing to infer the memory access pattern of the victim program. It can be represented by a sequence of assembly instructions that cause information leakage. However, gathering sufficient amount of such sequences and determining whether a sequence is a useful attack sequence that causes timing leakage is non-trivial. To address this challenge, my research work, AutoCAT [9], formulates the cache-timing attack as a guessing game between an attack program and a victim program holding a secret. The agent will be rewarded for guess the secret correctly. By playing this game many times using modern deep RL techniques, the agent can thus gather sufficient data. AutoCAT can explore attacks across various cache configurations without requiring knowledge of design details and can operate under different attack and victim program configurations. AutoCAT can also discover attacks that bypass certain detection and defense mechanisms, as well as a new attack StealthyStreamline that beats the then state-of-the-art attacks. AutoCAT is the first tool of its kind to use RL for crafting microarchitectural timing-channel attack sequences, enabling accelerated exploration of cache timing channels for secure microprocessor designs. Similarly, in SpecRL [12], we leverage RL to explore more advanced speculative attacks on real processors, identifying Spectre-V1 attacks in more efficiently compared to existing state-of-the-art processor fuzzing tools.

Multi-agent RL for Detection of Adaptive Cache Timing Attacks. Reinforcement learning can also be used for defense purposes, such as training a cache-timing attack detector with high accuracy. We can collect microarchitectural traces of attack and benign programs and train a classifier to distinguish between them. However, detectors trained using supervised learning or baseline single-agent RL lack adaptivity, meaning they cannot effectively counter an adaptive attacker whose behavior evolves to bypass detection. To build a robust detector against adaptive attackers, in my research work MACTA [10], we use a multi-agent approach in which one agent represents the adaptive attacker, and the other agent represents the detector. Multi-agent RL trains a highly capable attacker that can evade many existing detectors, as well as a powerful detector that achieves a higher detection rate against adaptive attackers. The experiment results suggest that MACTA is an effective solution without any manual input from security experts. MACTA detectors generalize well to heuristic attacks not encountered during training, achieving a higher detection rate and reducing the attack bandwidth of RL-based attackers. Meanwhile, MACTA attackers are qualitatively more effective than other studied attacks, with an average evasion rate of up to 99% against an unseen state-of-the-art detector.

Meta RL for Eviction Set-Finding. Baseline RL backs adaptivity to different environment, making it unfit for problems like eviction set-finding. An eviction set refers to a set of addresses that map to the same cache set. Finding an eviction set in the last-level cache is crucial for enabling cross-core cache-timing attacks. Cache slicing, address translation, and cache randomization make each processor instance unique, thus it is non-trivial to identify eviction sets. Traditional methods like single-holdout and group elimination [13] rely on human expertise. However, baseline RL needs to be trained in a specific environment and generally does not adapt well to different environments. Thus, an eviction set-finding agent generated by baseline RL would not be able to find eviction sets on a different processor. To address this challenge, my research [11] uses an advanced meta RL technique [14] to train a super-agent capable of adapting to various instances with different address-to-cache set mappings. Specifically, we formulate the eviction set-finding algorithm as an RL problem, optimizing the number of memory accesses the agent needs to deterministically evict the target address. We then train this RL agent on multiple cache instances with different address randomizations. The trained agent embodies an algorithm that can find eviction sets on any randomized cache, demonstrating the effectiveness of RL-based algorithm development.

Finding Vulnerabilities in AI Systems

AI systems are specialized towards AI workloads, which make the attack surface different from a general-purpose system. One thrust of my research focuses on discovering vulnerabilities in these AI systems, including compound AI systems for enterprises, embodied AI systems for autonomous and robotic vehicles, and hardware neural networks.

Vulnerabilities in Compound AI systems. Compound AI systems integrate computation, storage, and interface components on top of traditional ML models, making them more useful for real-world applications. My survey paper [15] systematizes the security risks of compound AI systems. In addition to systematization effort, I also studied and found new vulnerabilities in a concrete compound AI system, i.e., Retrieval-augmented generation (RAG). RAG is a system where a large language model is integrated with a database to ground responses with factual information. It is vulnerable to prompt engineering attacks that target the LLM as well as attacks on the retrieval process from the database. Despite this, the security risks of RAG vulnerabilities have not been systematically studied. I introduce ConfusedPilot [16], a class of security vulnerabilities in RAG systems that confuses Copilot, leading to integrity and confidentiality violations in its responses. First, I investigate a vulnerability that embeds malicious text in the modified prompt within RAG, corrupting the responses generated by the LLM. Second, I demonstrate a vulnerability that leverages the caching mechanism during retrieval to leak secret data. Third, we explore how both vulnerabilities can be exploited to propagate misinformation within an enterprise, ultimately impacting operations such as sales and manufacturing. I also discuss the root causes of these attacks by examining the architecture of a RAG-based system. This study highlights the security vulnerabilities in today's RAG-based systems and proposes design guidelines to secure future RAG-based systems, which is important for the practical adoption of RAG in large enterprises.

Vulnerabilities in Embodied AI Systems. Embodied AI integrates machine learning, sensors, and actuators with computer systems. Typical examples of embodied AI systems include humanoid robots and autonomous vehicles. Because embodied AI systems interact with the physical world, the security and privacy risks extend to the physical realm as well. My research focuses on demonstrating the security risks of these embodied AI systems, including autonomous and robotic vehicles. In [17], I show that the location privacy of an autonomous vehicle may be compromised by software side-channel attacks if localization software shares a hardware platform with an attack program. Specifically, we demonstrate that a cache side-channel attack can be used to infer the route or location of a vehicle running the adaptive Monte Carlo localization (AMCL) algorithm. This work flags the massive privacy risks of autonomous vehicles. In [18], I show the physical integrity of robotic vehicles protected by a trusted execution environment (TEE) against interrupt attacks, demonstrating that even with strong protections from TEE, the perception of robotic vehicles can be significantly altered, leading to path deviation and crashes. My research shows that embodied AI systems are vulnerable to these security vulnerabilities originating from system-level threats. Additionally on the defense side, my research [19] proposes methods for securing embodied AI systems via information flow control. To ensure responsiveness, I propose an algorithm-hardware co-optimization approach [20] to accelerate path planning in a high-dimensional search space by leveraging content-addressable memory. My research demonstrates risks and provides guidelines for secure embodied AI system.

Vulnerabilities of Hardware Neural Network. Deep neural network inference is computationally intensive, and general-purpose processors are not efficient for this task. Instead, specialized hardware targeting neural networks has proven to be more efficient. Furthermore, the approximate nature of neural network applications allows them to tolerate slightly imprecise results caused by process, temperature, and noise [21]–[23]. Slight data corruptions [24] in data centers can also negatively impact these AI workloads. One of my project [25], [26] is to study how these variations impact the quality of results produced by neural networks and how to optimize neural network weights to account for these variations. We demonstrate that process variations can significantly affect the accuracy of hardware neural networks. However, through retraining, we can modify the neural network weights to adapt to these performance variations.

Future Research

My current work focuses on demonstrating and evaluating the vulnerabilities of existing systems. Inspired by AI methods, I aim to develop new design methodologies, hardware, and algorithms that are free from these vulnerabilities moving forward.

AI for Cross-level System Security. My current work leverages AI to explore vulnerabilities at the microarchitecture level. However, recent attacks exploit vulnerabilities across the system stack [27], making it insufficient to focus solely on vulnerabilities at the microarchitecture level. Reinforcement learning remains a potential tool for exploring these cross-level vulnerabilities. To this end, I plan to model the attacker as an RL agent, with actions ranging from measuring performance counters, executing assembly programs, and invoking system calls to inspecting network packets. With this agent trained via RL, it can generate end-to-end attacks that exploit vulnerabilities at different levels. My long-term goal is to train a single monolithic agent capable of finding vulnerabilities at any level on any system without human intervention. Achieving this will require creative approaches to incorporate the vast observation and action spaces that account for all levels and system configurations, as well as innovations in learning algorithms and substantial computational power levering high-performance computing and distributed systems. The ultimate goal is to create the **AI doctor** for vulnerability discovery.

AI for Security Verification. While we have demonstrated RL's capabilities in finding vulnerabilities such as microarchitectural side channels, proving that a system is free of vulnerabilities remains challenging. Traditional formal verification methods like model checking require enumerating the entire state space. Theorem proving using proof assistant such as Coq [28], on the other hand, uses axioms to prove security properties without enumerating the entire state space. Traditionally, these proofs are difficult to write and come with a steep learning curve. However, recent advances have demonstrated the use of LLMs as proof copilots [29]. In the short term, I plan to evaluate these LLM-based copilots for proving microarchitecture security properties such as non-interference. In the long term, the goal is to build a security verification agent with fully automated, scalable theorem proving capability using LLMs with minimal human intervention. Achieving this would require breakthroughs in the reasoning capabilities of LLMs. With this **AI verifier**, I envision any practical-size systems can be verified automatically.

Explainable AI for System Security. While my current work successfully applies AI methods to system security problems, a key limitation is the lack of explainability in AI-generated results. In my previous work, AutoCAT

[9], although it is capable of generating novel attacks, AI-generated attacks differ from human-designed attacks in their lack of explainability. AI-generated attacks can include multiple steps with varying contributions to the success of the attack. Without explainability, it is not only challenging to understand AI-generated solutions but also difficult to generalize these solutions to similar yet different scenarios. A short-term plan is to incorporate explainable AI (xAI) techniques to analyze AI-generated solutions in microarchitecture attacks, similar to [30]. This approach can pinpoint critical steps in AI-generated solutions, helping humans design and generalize similar solutions more efficiently. Additionally, xAI can be combined with large language models to automatically generate natural language-based explanations. I envision this **AI explainer** can eventually find vulnerabilities and generate paper-quality reports that offer insights to fellow researchers.

Securing AI Algorithms with Information Flow Control. From an algorithmic perspective, many vulnerabilities in AI systems originate purely at the algorithm level. As my paper [16] demonstrates, the lack of distinction between control-plane and data-plane information causes confused deputy risks in RAG-based AI systems. Information Flow Control (IFC), which labels and tracks the provenance of each piece of information, is a useful tool for managing these access control risks. My previous work [19] applied IFC in robotics; I plan to adopt IFC for emerging generative AI systems, including LLMs and RAGs, to enhance their security [31]. One challenge preventing the mass adoption of IFC is the high cost of labeling, which is typically performed by humans. I plan to explore automated labeling techniques with the help of LLMs to make IFC in generative AI more user-friendly. With strict enforcement of access control through IFC, I believe future LLMs and RAGs can be free of security violations. In the long term, I believe programming language and architecture support is required to embed IFC in compound AI systems automatically.

Hardware Support for Secure Compound AI. At the hardware level, secure hardware, such as trusted execution environments (TEEs), provides basic separation and security assurances. Commercial TEEs like Intel SGX, TDX, and AMD SEV have been relatively successful in general-purpose computing. However, as my previous work [18] shows, these TEEs provide insufficient protection, leaving embodied AI systems vulnerable. To offer stronger security guarantees for these AI systems, I plan to explore new TEE architectures that offer customizability for performance and real-time awareness, to compound AI systems.

References

[1] M. Zaharia, O. Khattab, L. Chen, J. Q. Davis, H. Miller, C. Potts, J. Zou, M. Carbin, J. Frankle, N. Rao, *et al.*, “The shift from models to compound ai systems”, *Berkeley Artificial Intelligence Research Lab. Available online at: https://bair. berkeley. edu/blog/2024/02/18/compound-ai-systems/(accessed February 27, 2024)*, 2024.

[2] J. Duan, S. Yu, H. L. Tan, H. Zhu, and C. Tan, “A survey of embodied ai: From simulators to research tasks”, *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 2, pp. 230–244, 2022.

[3] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, “Mastering the game of go with deep neural networks and tree search”, *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[4] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, *et al.*, “Grandmaster level in starcraft ii using multi-agent reinforcement learning”, *nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[5] A. Mirhoseini, A. Goldie, M. Yazgan, J. W. Jiang, E. Songhori, S. Wang, Y.-J. Lee, E. Johnson, O. Pathak, A. Nazi, *et al.*, “A graph placement methodology for fast chip design”, *Nature*, vol. 594, no. 7862, pp. 207–212, 2021.

[6] D. J. Mankowitz, A. Michi, A. Zhernov, M. Gelmi, M. Selvi, C. Paduraru, E. Leurent, S. Iqbal, J.-B. Lespiau, A. Ahern, *et al.*, “Faster sorting algorithms discovered using deep reinforcement learning”, *Nature*, vol. 618, no. 7964, pp. 257–263, 2023.

[7] A. Fawzi, M. Balog, B. Romera-Paredes, D. Hassabis, and P. Kohli, *Discovering novel algorithms with alphatensor*, 2022.

[8] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, *et al.*, “Highly accurate protein structure prediction with alphafold”, *nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[9] **M. Luo**, W. Xiong, G. Lee, Y. Li, X. Yang, A. Zhang, Y. Tian, H.-H. S. Lee, and G. E. Suh, “Autocat: Reinforcement learning for automated exploration of cache-timing attacks”, in *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, IEEE, 2023, pp. 317–332.

[10] J. Cui, X. Yang, **M. Luo**, G. Lee, P. Stone, H.-H. S. Lee, B. Lee, G. E. Suh, W. Xiong, and Y. Tian, "Macta: A multi-agent reinforcement learning approach for cache timing attacks and detection", in *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

[11] **M. Luo** and M. Tiwari, "Towards reinforcement learning for eviction-set finding for randomized caches", in *SRC Technical Conference*, 2024.

[12] E. Lai, **M. Luo**, and M. Tiwari, "Specrl: Using reinforcement learning to detect speculative vulnerabilities", 2024.

[13] M. K. Qureshi, "New attacks and defense for encrypted-address cache", in *Proceedings of the 46th International Symposium on Computer Architecture*, 2019, pp. 360–371.

[14] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel, "Rlˆ2: Fast reinforcement learning via slow reinforcement learning", *arXiv preprint arXiv:1611.02779*, 2016.

[15] S. Banerjee, P. Sahu, **M. Luo**, A. Vahldiek-Oberwagner, N. J. Yadwadkar, and M. Tiwari, "Sok: A systems perspective on compound ai threats and countermeasures", *arXiv preprint arXiv:2411.13459*, 2024.

[16] A. RoyChowdhury, **M. Luo**, P. Sahu, S. Banerjee, and M. Tiwari, "Confusedpilot: Compromising enterprise information integrity and confidentiality with copilot for microsoft 365", *arXiv preprint arXiv:2408.04870*, 2024.

[17] **M. Luo**, A. Myers, and G. Suh, "Stealthy tracking of autonomous vehicles with cache side channels", in *29th USENIX Security Symposium (USENIX Security 20)*, 2020.

[18] **M. Luo** and G. E. Suh, "Wip: Interrupt attack on tee-protected robotic vehicles", in *Workshop on Automotive and Autonomous Vehicle Security (AutoSec)*, 2022.

[19] J. Liu, J. Corbett-Davies, A. Ferraiuolo, A. Ivanov, **M. Luo**, G. E. Suh, A. C. Myers, and M. Campbell, "Secure autonomous cyber-physical systems through verifiable information flow control", in *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, 2018, pp. 48–59.

[20] **M. Luo** and G. E. Suh, "Accelerating path planning for autonomous driving with hardware-assisted memorization", in *2022 IEEE 33rd International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, IEEE, 2022, pp. 126–130.

[21] **M. Luo**, R. Wang, J. Wang, S. Guo, J. Zou, and R. Huang, "Compact modeling of random telegraph noise in nanoscale mosfets and impacts on digital circuits", in *Proceedings of Technical Program-2014 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA)*, IEEE, 2014, pp. 1–2.

[22] **M. Luo**, R. Wang, S. Guo, J. Wang, J. Zou, and R. Huang, "Impacts of random telegraph noise (rtn) on digital circuits", *IEEE Transactions on Electron Devices*, vol. 62, no. 6, pp. 1725–1732, 2014.

[23] J. Zou, R. Wang, **M. Luo**, R. Huang, N. Xu, P. Ren, C. Liu, W. Xiong, J. Wang, J. Liu, *et al.*, "Deep understanding of ac rtn in mugfets through new characterization method and impacts on logic circuits", in *2013 Symposium on VLSI Technology*, IEEE, 2013, T186–T187.

[24] P. W. Deutsch, V. Sridharan, V. Q. Ulitzsch, J. S. Emer, S. Gurumurthi, and M. Yan, "Delayavf: Calculating architectural vulnerability factors for delay faults",

[25] X. Jiao, **M. Luo**, J.-H. Lin, and R. K. Gupta, "An assessment of vulnerability of hardware neural networks to dynamic voltage and temperature variations", in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, IEEE, 2017, pp. 945–950.

[26] J.-H. Lin, X. Jiao, **M. Luo**, Z. Tu, and R. K. Gupta, "Vulnerability of hardware neural networks to dynamic operation point variations", *IEEE Design & Test*, vol. 37, no. 5, pp. 75–84, 2020.

[27] S. Li, X. Wang, M. Xue, H. Zhu, Z. Zhang, Y. Gao, W. Wu, and X. S. Shen, "Yes, one-bit-flip matters! universal dnn model inference depletion with runtime code fault injection", in *Proceedings of the 33th USENIX Security Symposium*, 2024.

[28] A. Chlipala, *Certified programming with dependent types: a pragmatic introduction to the Coq proof assistant.* MIT Press, 2013.

[29] K. Yang, A. Swope, A. Gu, R. Chalamala, P. Song, S. Yu, S. Godil, R. J. Prenger, and A. Anandkumar, "Leandojo: Theorem proving with retrieval-augmented language models", *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[30] J. Yu, W. Guo, Q. Qin, G. Wang, T. Wang, and X. Xing, "{Airs}: Explanation for deep reinforcement learning based security applications", in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 7375–7392.

[31] F. Wu, E. Cecchetti, and C. Xiao, "System-level defense against indirect prompt injection attacks: An information flow control perspective", *arXiv preprint arXiv:2409.19091*, 2024.