Research Statement

Mulong Luo

Overview. Computer systems are the foundation of the modern information world. Components from different levels of abstraction—including circuits, architecture, operating systems, and applications—work coherently to drive the information world forward. Securing computer systems against malicious actors that can steal, damage, or disrupt information is crucial. However, securing computer systems is challenging. First, modern computer systems have become so complex that they are error-prone and laborious to inspect manually for security. Formal methods are typically constrained by scalability, and empirical testing methods like fuzzing have limitations. AI-enabled attackers also increase the offensive capabilities targeting these complex computer systems. Second, novel applications, especially AI applications such as compound AI [25] and embodied AI [3], are making computer systems increasingly heterogeneous. Specialized hardware, such as hardware accelerators and in/near-memory processing architectures, adds complexity. The interaction between AI algorithms and underlying systems further complicates efforts to secure these computer systems with AI.

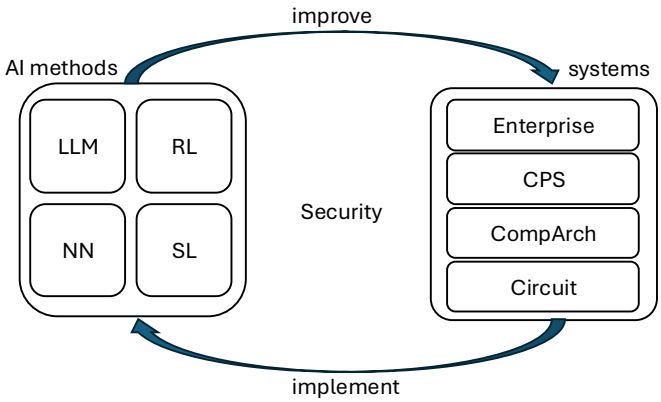

Figure 1: **The interaction of AI with system layers. AI methods can be used to improve system security, and systems that implement AI have unique security concerns.**

To mitigate the security risks of modern computer systems, we need to address the security issues of both general-purpose systems and systems with AI. 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 systematically evaluated. Thus, the two main thrusts of my research work are:

- Using AI methods for improving the security of general-purpose computer systems;
- Evaluating the security of AI systems.

First, to secure general-purpose computer systems against increasing complexity, as well as traditional and AI-enabled attackers, I leverage the power of AI. AI has become embedded in every aspect of life. 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. Its capability is limited only by the amount of data available and the computational power an entity possesses. 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 and studying microarchitecture security problems using reinforcement learning.

Second, AI-integrated computer systems—such as compound AI systems, embodied AI systems, and hardware neural networks—differ significantly from general-purpose systems. They have unique interfaces, architectures, and require separate security evaluations. Another main thrust of my research is studying the security of these systems.

In addition to security issues, AI systems also face performance and reliability challenges, especially in real-time scenarios. A third direction of my research is:

- improving the performance and reliability of AI systems.

In the rest of this statement, I will first describe my research on using AI methods for general-purpose system security, then I will describe the second thrust on evaluating the security of AI systems. After that, I will discuss other work on improving the performance and reliability of computer systems. Finally, I will outline future directions.

AI Methods for General-Purpose System Security

In this thrust, I use AI methods, specifically reinforcement learning, to tackle CPU microarchitecture security challenges, focusing on the attack and defense 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. Existing automated methods like formal verification suffer from scalability issues, while heuristic methods like fuzzing lack theoretical guarantees.

In contrast, reinforcement learning offers a promising approach to address these issues. Reinforcement learning 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. Reinforcement learning has recently been demonstrated to be useful to achieve superhuman performance in games like Go games [23], video games [24], chip design [21], algorithm design [20, 8], and biology [10]. However, applying RL to a microarchitecture security problem faces several challenges in the formulation, generalization, and adaptivity. My research work AutoCAT [18], MACTA [2], AlphaEvict [15], SpecRL addresses these challenges and successfully demonstrate the capability of RL in solving a wide range of RL security problems.

Exploration of Cache Timing Attacks and Speculative Execution Attacks. One of the main challenges in exploring cache timing attacks and speculative execution attacks as a reinforcement learning problem is the formulation. A microarchitecture attack can be represented by a sequence of assembly instructions that cause information leakage. However, determining whether a sequence is a useful attack sequence that causes timing leakage is non-trivial. To address this challenge, my research work, AutoCAT [18], formulates the cache-timing attack as a guessing game between an attack program and a victim program holding a secret. This guessing game can thus be solved using modern deep RL techniques. 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. In particular, AutoCAT discovered StealthyStreamline, a new attack capable of bypassing performance counter-based detection and achieving up to a 71% higher information leakage rate than state-of-the-art LRU-based attacks on real processors. 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, we leverage RL to explore more advanced speculative attacks on real processors, identifying Spectre-V1 attacks in less time 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 [2], 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 97.8% detection rate and reducing the attack bandwidth of RL-based attackers by an average of 20%. 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. 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-hold out and group elimination 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 [15] uses advanced a meta RL technique [4] 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. The results demonstrate that RL can effectively find eviction sets.

Evaluating the Security of AI Systems

AI systems are more heterogeneous than general-purpose systems, which broadens the attack surface. Additionally, the interaction between AI algorithms and the underlying system stack makes security evaluation more challenging. Therefore, the security of each AI system needs to be evaluated individually. One thrust of my research focuses on evaluating the security of these AI systems, including compound AI systems for enterprise, embodied AI systems for autonomous and robotic vehicles, and hardware neural networks.

Security Risks of Compound AI systems. Compound AI systems [1] integrate computation, storage, and interface components on top of traditional ML models, making them more useful for real-world applications. Retrieval-augmented generation (RAG) is a recent notable example, where a large language model is integrated with a database to ground responses with factual information. However, 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 RAGs to these vulnerabilities have not been systematically studied.

To address this, we introduce ConfusedPilot [22], a class of security vulnerabilities in RAG systems that confuses Copilot, leading to integrity and confidentiality violations in its responses. First, we investigate a vulnerability that embeds malicious text in the modified prompt within RAG, corrupting the responses generated by the LLM. Second, we 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. We 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.

Security Risks of 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 [19], 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. In [14], I evaluated 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 potential crashes. My research shows that embodied AI systems are vulnerable to these security vulnerabilities originating from system-level threats. Additionally, my research [12] proposes methods for securing embodied AI systems via information flow control.

Improving Performance and Reliability of AI Systems

To ensure the trustworthiness and usefulness of AI systems, performance and reliability are also important in addition to security and privacy. A sideline of my research focuses on improving the performance and reliability of various AI systems and applications.

Evaluating Reliability 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 [17, 16, 26]. One of my project [9, 11] 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.

Improving Performance of Embodied AI via Algorithm/Hardware Cooptimization. Design embodied AI for real-time robotics is challenging. For example, Path planning for autonomous driving with dynamic obstacles poses a challenge because it requires a higher-dimensional search while still meeting real-time constraints [13]. To address this problem, I propose an algorithm-hardware co-optimization approach to accelerate path planning in a high-dimensional search space by leveraging content-addressable memory. Experimental results on a modern processor and a cycle-level simulator show that hardware-assisted memoization significantly reduces the execution time of path planning.

Improving Reliability of Cloud AI. Many AI applications are deployed on the cloud and edge instead of on-device. This presents challenges, as the communication, computation, and contention uncertainties in the cloud make it difficult for these cloud AI systems to meet real-time requirements. My research [6, 5, 7] presents a scheduling framework for these cloud AI applications that incorporates workload computation time prediction, network

delay estimation, and mobile-server clock synchronization techniques. Using several mobile vision applications, we evaluate this framework under diverse configurations and demonstrate its effectiveness.

Future Research

Building AI-Inspired Secure AI systems. My current work focuses on evaluating the security risks of AI systems by demonstrating attacks on these systems. Inspired by AI methods, I aim to develop new design methodologies, hardware, and algorithms free from these vulnerabilities moving forward.

First, 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 [14] 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 suitable for current AI systems, including embodied AI and compound AI, as well as future AI systems.

Second, from a design methodology perspective, I plan to incorporate AI methods into the design of secure AI systems. My previous work [18, 2] uses AI methods for attacking and defending general-purpose computing systems such as CPUs. I plan to extend this RL-based methodology to build more secure AI systems, using it both at design time to inspect security vulnerabilities in AI systems and at run time to create better attack detectors.

Third, from an algorithmic perspective, many vulnerabilities in AI systems originate purely at the algorithm level. As my paper [22] shows, 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 [12] applied IFC in robotics; I plan to adopt IFC for emerging generative AI systems, including LLMs and RAGs, to enhance their security.

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 [18], while is able to generate novel attacks, AI-generated attacks differ from human-designed attacks in that they lack 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 it is also difficult to generalize these solutions to similar yet different scenarios.

I plan to incorporate explainable AI (xAI) techniques to analyze AI-generated solutions in system security and microarchitecture attacks. This approach can pinpoint critical steps in AI-generated solutions, which can then help humans design and generalize similar solutions more efficiently.

References

[1] Banerjee, S., Sahu, P., Luo, M., and Tiwari, M. Sok: Sok: Attacks and defenses in compound ai systems.

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

[3] Duan, J., Yu, S., Tan, H. L., Zhu, H., and Tan, C. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence 6*, 2 (2022), 230–244.

[4] Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. Rl ˆ2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779* (2016).

[5] Fang, Z., Luo, M., Anwar, F. M., Zhuang, H., and Gupta, R. K. Go-realtime: a lightweight framework for multiprocessor real-time system in user space. *ACM SIGBED Review 14*, 4 (2018), 46–52.

[6] Fang, Z., Luo, M., Yu, T., Mengshoel, O. J., Srivastava, M. B., and Gupta, R. K. Mitigating multi-tenant interference on mobile offloading servers. In *Proceedings of the 2017 Symposium on Cloud Computing* (2017), pp. 644–644.

[7] Fang, Z., Luo, M., Yu, T., Mengshoel, O. J., Srivastava, M. B., and Gupta, R. K. Mitigating multi-tenant interference in continuous mobile offloading. In *Cloud Computing–CLOUD 2018: 11th International Conference, Held as Part of the Services Conference Federation, SCF 2018, Seattle, WA, USA, June 25–30, 2018, Proceedings 11* (2018), Springer International Publishing, pp. 20–36.

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

[9] Jiao, X., Luo, M., Lin, J.-H., and Gupta, R. K. 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)* (2017), IEEE, pp. 945–950.

[10] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *nature 596*, 7873 (2021), 583–589.

[11] Lin, J.-H., Jiao, X., Luo, M., Tu, Z., and Gupta, R. K. Vulnerability of hardware neural networks to dynamic operation point variations. *IEEE Design & Test 37*, 5 (2020), 75–84.

[12] Liu, J., Corbett-Davies, J., Ferraiuolo, A., Ivanov, A., Luo, M., Suh, G. E., Myers, A. C., and Campbell, M. 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.

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

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

[15] Luo, M., and Tiwari, M. Towards reinforcement learning for eviction-set finding for randomized caches. In *SRC Technical Conference* (2024).

[16] Luo, M., Wang, R., Guo, S., Wang, J., Zou, J., and Huang, R. Impacts of random telegraph noise (rtn) on digital circuits. *IEEE Transactions on Electron Devices 62*, 6 (2014), 1725–1732.

[17] Luo, M., Wang, R., Wang, J., Guo, S., Zou, J., and Huang, R. 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)* (2014), IEEE, pp. 1–2.

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

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

[20] Mankowitz, D. J., Michi, A., Zhernov, A., Gelmi, M., Selvi, M., Paduraru, C., Leurent, E., Iqbal, S., Lespiau, J.-B., Ahern, A., et al. Faster sorting algorithms discovered using deep reinforcement learning. *Nature 618*, 7964 (2023), 257–263.

[21] Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J. W., Songhori, E., Wang, S., Lee, Y.-J., Johnson, E., Pathak, O., Nazi, A., et al. A graph placement methodology for fast chip design. *Nature 594*, 7862 (2021), 207–212.

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

[23] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature 529*, 7587 (2016), 484–489.

[24] Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *nature 575*, 7782 (2019), 350–354.

[25] Zaharia, M., Khattab, O., Chen, L., Davis, J. Q., Miller, H., Potts, C., Zou, J., Carbin, M., Frankle, J., Rao, N., 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).

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