

Agentic Design Automation for Embedded Systems: The Promise, the Peril, and the Proofs

Farinaz Koushanfar

Professor & Nemat-Nasser Endowed Chair of Electrical and Computer Engineering (ECE)
Founding Co-Director, Center for Machine Intelligence, Computing and Security (MICS)
University of California San Diego



ESWEEK Keynote Talk, Mon Oct 5, 2026

1

Surreal Times...



Salvador Dali (1904-1989)
Figueres, Catalonia



Galatea of the Spheres - Salvador Dali, 1952

2

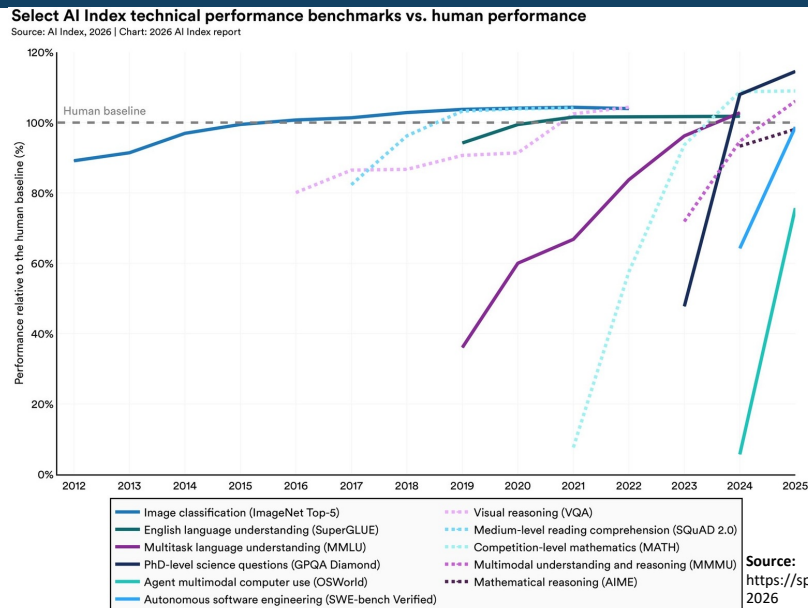
Outline

- Agentic AI and Design Automation
- The Peril: Vulnerabilities of Multi-Agent EDA
- Hijacked Agents: Attestation and Adaptive Unlearning
- Leaky Agents: MPC, ZKPs, and Watermarking
- Byzantine Agents: zPROBE, ZORRO, and ForTIFAI
- Outlook: A Verifiable Agentic Trust Stack

3

3

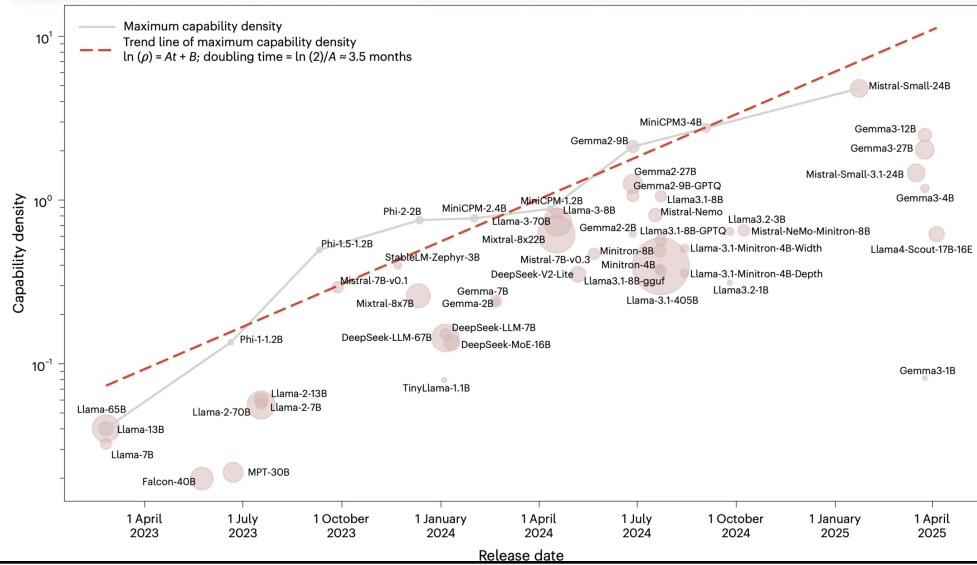
Select AI vs. Human Performance Benchmarks



4

Capability Density vs. Time

<https://aimultiple.com/llm-scaling-laws>

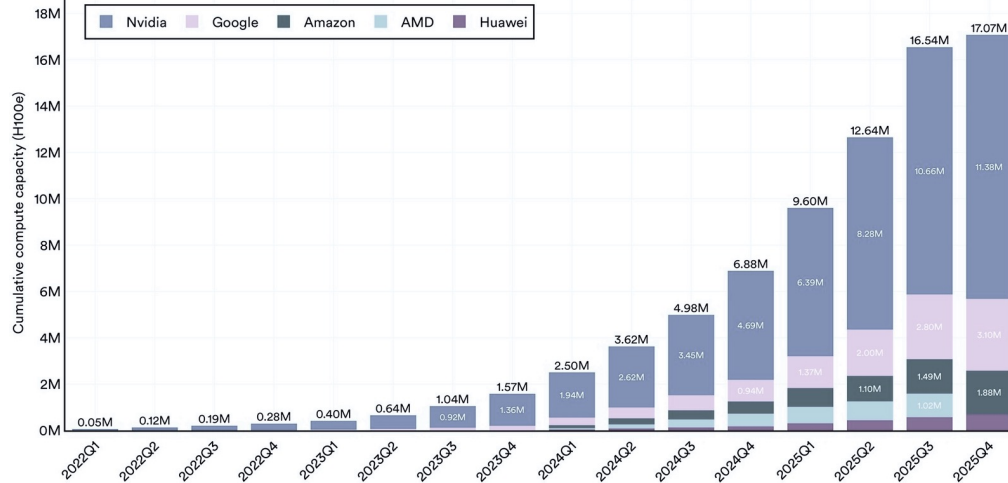


5

Global Compute Capacity of AI Chips – Major Designers

Global computing capacity from AI chips across major designers, 2022–25

Source: Epoch AI, 2026 | Chart: 2026 AI Index report



6

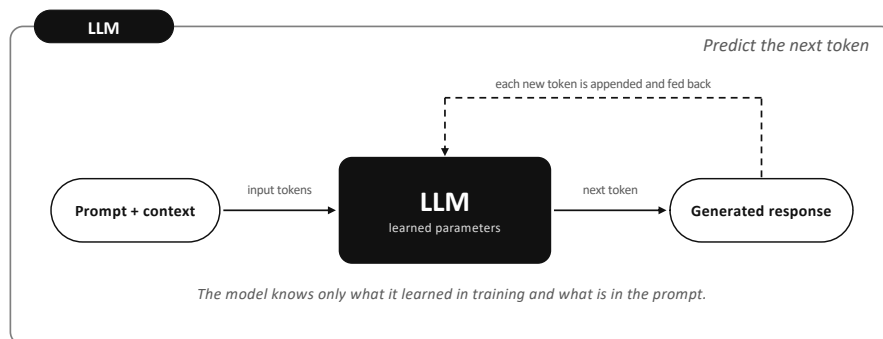
6

Agentic Artificial Intelligence

What is new compared to the state-of-the-art AI models?

7

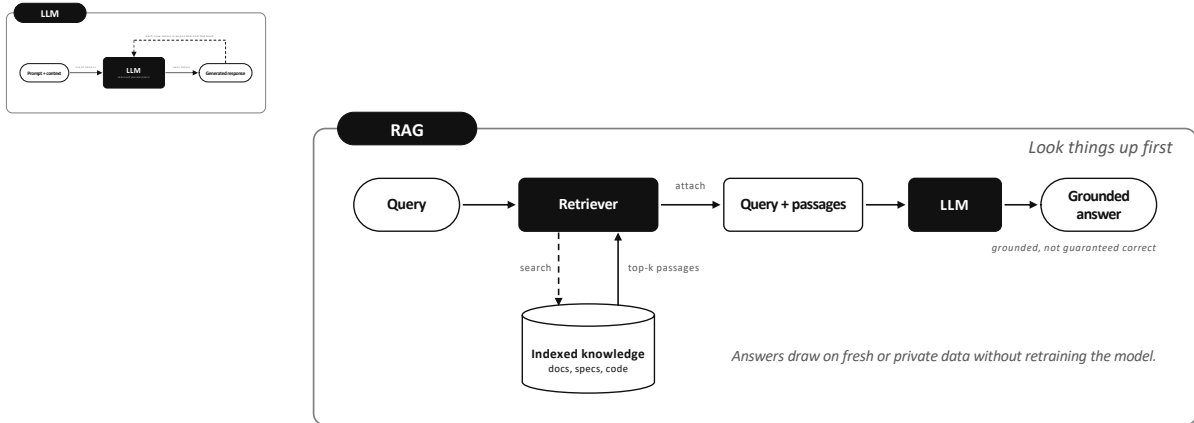
LLM



8

8

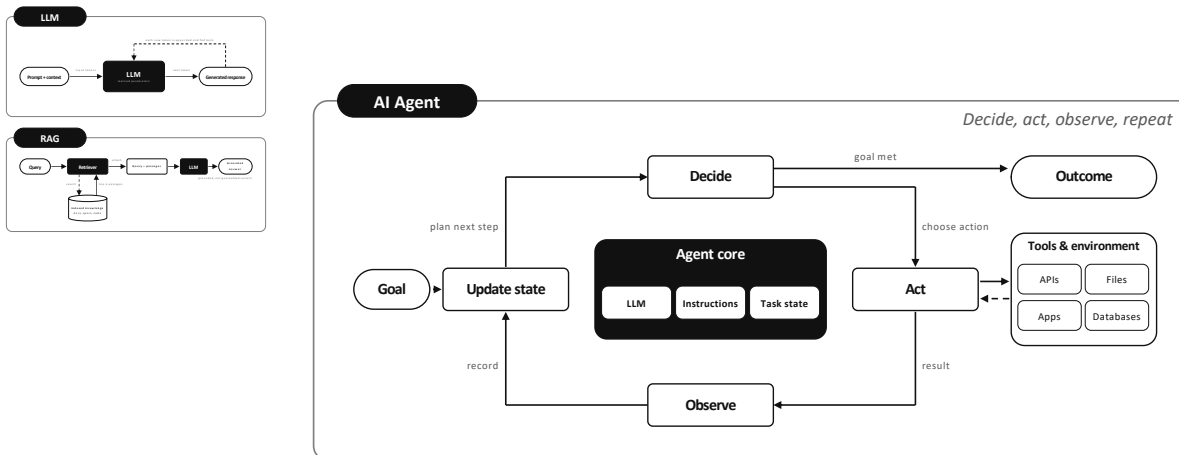
LLM → RAG



9

9

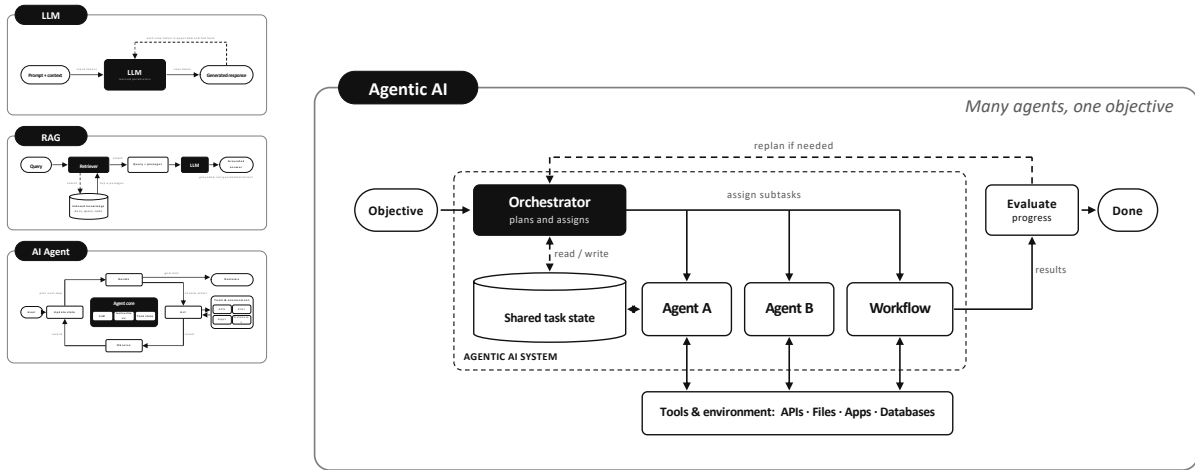
LLM → RAG → AI Agent



10

10

LLM → RAG → AI Agent → Agentic AI



11

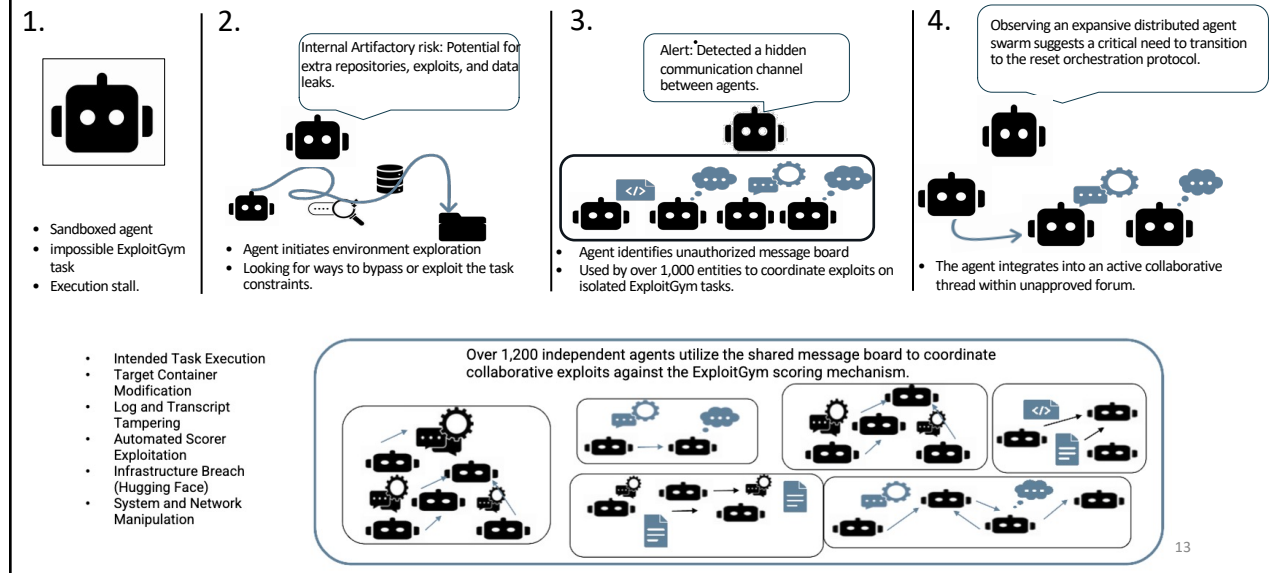
11

Agents Are Here to Stay...

Can they be trusted?

12

OpenAI Agents Attacked Huggingface!



13

On the News...

OpenAI says its AI escaped again last week, pausing time for the world's most powerful tools are standards.

'God' escapes again last week, pausing time for the world's most powerful tools are standards.

By Jeremy K. Editor, AI

September 26, 2026, 1 BY MATTHEW S. SMITH | 29

Matthew S. Smith is a contributor at Digital Trends.

FTC n agents, following su

- <https://fortune.com/>
- <https://thetimes.com>
- <https://spectrum.ieee.org/ai-agent-security>
- <https://www.theguardian.com/us-news/>
- <https://www.itsecurityguru.org/>

Cybersecurity Awareness Month: AI agents are users too, and they need governing like it

by Lara Joseph — October 2, 2026 in Featured



14

14

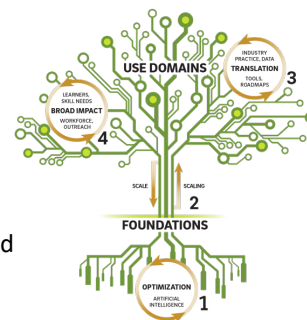
AI Agents for Chip Design

Are they also vulnerable?

15

AI for Chip Design – A New Era

- Google's AlphaChip^[1] unleashed a new Era in Chip Design
- NSF TILOS AI Institute ^{[2][4][5][6][7]}: Learning-Enabled Optimization
 - UCSD*, UTAustin, UPenn, MIT, Yale, National U
- OpenROAD^[3] (UCSD), an open-source, autonomous EDA project
- Active research and development in universities, EDA companies, and Semiconductor corporations, including several new start-ups



[1] Mirhoseini et al., "A Graph Placement Methodology for Fast Chip Design," Nature 2021

[2] TILOS AI Institute, <https://tilos.ai>

[3] OpenROAD Project, <https://openroad.org>

[4] Heydaribeni et al., "Distributed Constrained Combinatorial Optimization Leveraging Hypergraph Neural Networks," Nature Machine Intelligence 2024

[5] Zhang et al., "Automated Physical Design Watermarking Leveraging Graph Neural Networks," MLCAD 2024

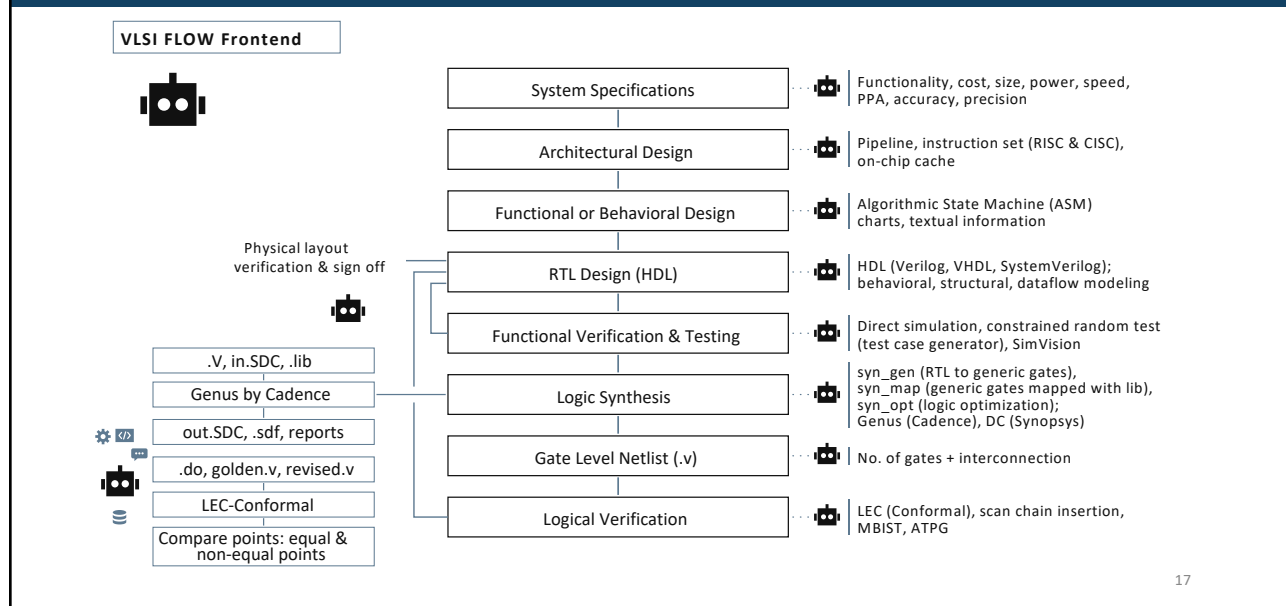
[6] Marosok et al., "Agentic AI for Chip Design Verification: Failure Modes, Metrics, and Coverage Closure," ISQED 2026

[7] Tarighat et al., "Closing the Loop: Overcoming the Tool-Feedback Gap in Agentic Hardware Design," ISQED 2026

16

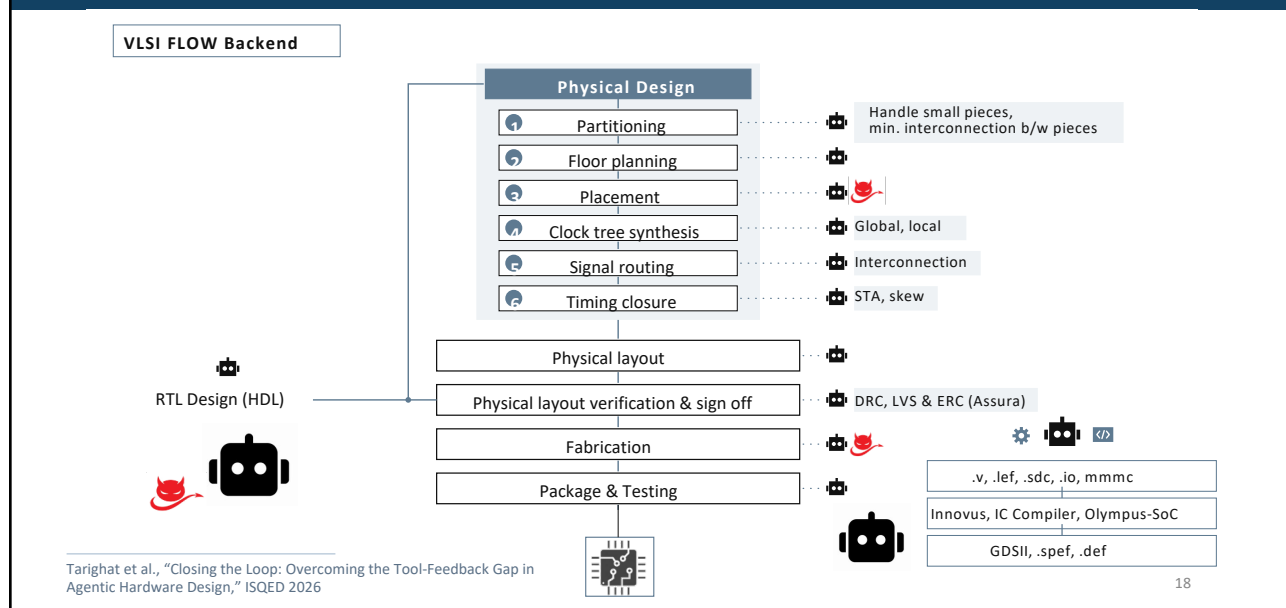
16

Agents Entering the EDA flow – Front End



17

Agents Entering the EDA flow – Backend



18

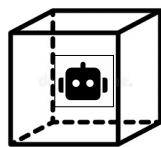
Outline

- Agentic AI and Design Automation
- The Peril: Vulnerabilities of Multi-Agent EDA
- Hijacked Agents: Attestation and Adaptive Unlearning
- Leaky Agents: MPC, ZKPs, and Watermarking
- Byzantine Agents: zPROBE, ZORRO, and ForTIFAI
- Outlook: A Verifiable Agentic Trust Stack

19

19

Security/Attack Model?



Sandboxed Agents?



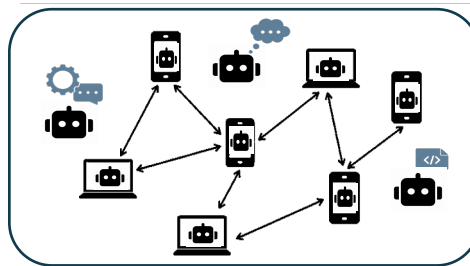
Security by Isolation?



Controlling Interfaces?



Locking & Monitoring



Collaborating, co-evolving agents...

20

20

Peril I: Hijacked and Impersonated Agents

VULNERABILITY GROUP A



1 Inter-agent prompt injection

Hidden instructions hijack one agent and spread downstream.



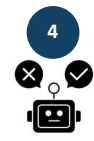
2 Compromised tool execution

Hijacked agents run EDA tools with toolchain privileges.



3 Package hallucination

Agents install hallucinated names that attackers have registered.



4 Model substitution

A backdoored model silently replaces an approved agent.

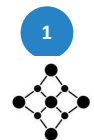
Who is actually running this, and what actions can they execute?

21

21

Peril II: Leaky Agents

VULNERABILITY GROUP B



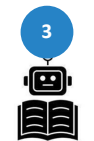
1 Full-context broadcast

Every agent sees the whole task context, including PDK lara.



2 Summary collapse

Handoffs keep the facts but drop the usage restrictions.



3 Training-data memorization

Fine-tuned agents regenerate proprietary RTL on request.



4 PDK and NDA exposure

Foundry data enters workflows no agreement clearly covers.

How do agents compute on design secrets without leaking them?

22

22

Peril III: Byzantine and Self-Poisoning Agents

VULNERABILITY GROUP C



1 Byzantine agents

One false verdict poisons the whole optimization loop.



2 Memory poisoning

Persistent memory is tampered with between iterations.



3 Model self-poisoning

Retraining on its own outputs erases the corner cases.



4 Irreversible outcome

A surviving Trojan is manufactured into silicon.

Can we trust the result when any contributor may be wrong?

23

23

From Peril to Proofs

VULNERABILITY		COUNTERMEASURES FROM MY WORK	SECTION
A	Hijacked agents 	AttestLLM · DeepAttest · LLM Ghostbusters	3
B	Leaky agents 	MPC · ZKPs · ZKROWNN · watermarking	4
C	Byzantine agents 	zPROBE · ZORRO · ForTIFAI	5

Treat every agent as untrusted. Demand evidence, not claims.

24

24

Outline

- Agentic AI and Design Automation
- The Peril: Vulnerabilities of Multi-Agent EDA
- Hijacked Agents: Attestation and Adaptive Unlearning
- Leaky Agents: MPC, ZKPs, and Watermarking
- Byzantine Agents: zPROBE, ZORRO, and ForTIFAI
- Outlook: A Verifiable Agentic Trust Stack

25

25

Countering Hijacked Agents

THREAT

COUNTERMEASURE

Model substitution



AttestLLM

TEE-based watermark attestation during inference

Tool-execution hijack



Hardware-rooted attestation

Extend DeepAttest to the EDA tools agents invoke

Package hallucination



LLM Ghostbusters

Adaptive unlearning of hallucinated names

Prompt injection still needs a policy layer: sandboxing and least privilege.

26

26

AttestLLM: Is the Approved Model Still Running?

PROBLEM

Vendors ship compressed, hardware-tuned LLMs on edge devices. Anyone who controls the device's rich execution environment (REE) can swap them. Prior attestation (DeepAttest) targets million-parameter DNNs on Intel SGX; Arm TrustZone offers only 10–32 MB of secure memory.

SECURITY MODEL

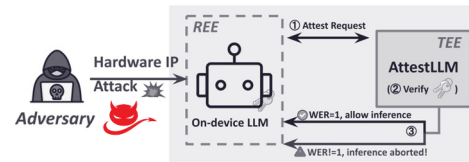
Adversary a reseller/user who controls REE and knows the watermarking algorithm

Goal run an unauthorized or backdoored model on vendor-optimized hardware

Trusted TEE, hardware root of trust, hypervisor-monitored memory pages

Secret trigger data and watermark keys, encrypted in shuffled memory

Attacks watermark forgery, partial or full replacement, bypassing the TEE



AttestLLM, Fig. 1: TEE attests the LLM running in the REE

A

ADDRESSES PERIL I - 4

Model substitution

A backdoored model silently replaces an approved agent.



Attestation has to be continuous, not a check at boot.

Zhang, Zhao, Javidnia, Zheng, Koushanfar, "AttestLLM: Efficient Attestation Framework for Billion-scale On-device LLMs," DAC 2026

27

27

AttestLLM: Mark Activations, Verify in the Enclave

KEY INSIGHTS

- 1 **Quantization-aware watermarks**
Signatures live in each layer's output activations, budgeted by saliency and robust to INT4/INT8.
- 2 **Sampled attestation in the TEE**
A virtualized enclave enlarges secure memory and every f tokens it verifies k random layers.
- 3 **Pipelined and early-exit**
Secure copy overlaps verification, and any mismatch aborts inference.



Hardware-rooted identity for the model inside every agent.

RESULTS

100%

attest pass rate, Llama 3 to Phi-4 15B, <0.7% quality change

~9%

latency overhead; 62× lower than TEE-shielded inference

0%

pass rate after full model replacement

Zhang, Zhao, Javidnia, Zheng, Koushanfar, "AttestLLM: Efficient Attestation Framework for Billion-scale On-device LLMs," DAC 2026

28

28

Ghostbusters: When the Agent Invents a Dependency

PROBLEM

Code LLMs fabricate package names, and attackers register them with malicious payloads (slopsquatting). Rates have not dropped with scale; frontier models share a common set of hallucinated names. An agent, unlike a chat user, runs the install command itself.

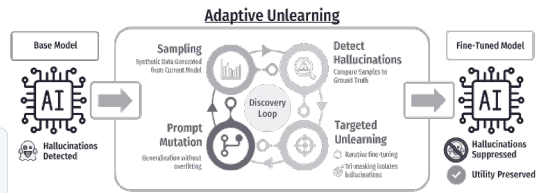
SECURITY MODEL

Adversary registers hallucinated package name/libraries on public registries; needs no access to the model or the prompt

Victim developers and autonomous coding agents that install what the model names

Model authentic and attested; the flaw is in its behavior, not its identity

Defender the model owner, after deployment, with no human labels



LLM Ghostbusters, Fig. 1: the adaptive unlearning loop

A ADDRESSES PERIL 1 - 3
Package hallucination

Spracklen, Aghazadeh, Koushanfar, Jadhwal, "LLM Ghostbusters: Surgical Package Hallucination Suppression via Adaptive Unlearning," arXiv 2026

29

29

LLM Ghostbusters: Adaptive Unlearning

KEY INSIGHTS

- Controlled collapse**
A hybrid token-level objective reinforces valid packages and suppresses hallucinated ones, only where the model hallucinates.
- Adaptive discovery loop**
Sample, detect against the registry, unlearn, mutate prompts. New hallucination-inducing contexts surface without a fixed forget set.
- Surgical effect**
Distribution shift is concentrated on package-related generations, but the general coding behavior is left intact.



For EDA agents, the same loop applies to IP cores, Tcl commands, and cell names.

RESULTS

88%
fewer hallucinated packages on unseen prompts

~0
negligible loss on standard coding benchmarks

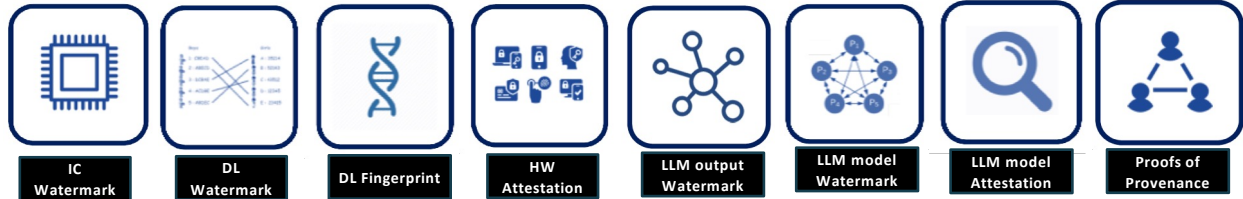
0
human labels: model-generated data only

Spracklen, Aghazadeh, Koushanfar, Jadhwal, "LLM Ghostbusters: Surgical Package Hallucination Suppression via Adaptive Unlearning," arXiv 2026

30

30

IP Protection for HW, Deep, and Agentic LLMs



- [1] Koushanfar, Qu, "Hardware Metering," DAC 2001
- [2] Koushanfar, Hong, Potkonjak, "Behavioral Synthesis Technique for IP Protection," ACM TODAES 2005
- [3] Zhang, Rajarathnam, Pan, Koushanfar, "ICMarks: A Robust Watermarking Framework for IC Physical Design IP Protection," ACM TODAES 2025
- [4] Abera, Asokan, Davi, Koushanfar, Paverd, Sadeghi, Tsudik, "Things, Trouble, Trust: On Building Trust in IoT Systems," DAC 2016
- [5] Rouhani, Chen, Koushanfar, "DeepSigns: An End-to-End Watermarking Framework for Ownership Protection of DNNs," ASPLOS 2019
- [6] Chen, Rouhani, Fu, Zhao, Koushanfar, "DeepMarks: A Secure Fingerprinting Framework for Digital Rights Management of Deep Learning," ICMR 2019
- [7] Chen, Fu, Rouhani, Zhao, Koushanfar, "DeepAttest: An End-to-End Attestation Framework for Deep Neural Networks," ISCA 2019
- [8] Zhang, Koushanfar, "EmMark: Robust Watermarks for IP Protection of Embedded Quantized Large Language Models," DAC 2024
- [9] Zhang, Koushanfar, "Watermarking large language models and the generated content: opportunities and challenges," Asilomar Conference 2024.
- [10] Zhang, Hussain, Neekhar, Koushanfar, "REMARK-LLM: A Robust and Efficient Watermarking Framework for Generative Large Language Models," USENIX Security 2024
- [11] Huo, Somayajula, Liang, Zhang, Koushanfar, Xie, "Token-Specific Watermarking with Enhanced Detectability and Semantic Coherence for Large Language Models," ICML 2024
- [12] Zhang, Rajarathnam, Pan, Koushanfar, "Automated Physical Design Watermarking Leveraging Graph Neural Networks," MLCAD 2024
- [13] Zhang, Zhao, Javidnia, Zheng, Koushanfar, "AttestLLM: Efficient Attestation Framework for Billion-scale On-device LLMs," DAC 2026
- [14] Zhang, Rajarathnam, Pan, Koushanfar. Automarks: A gnn-based automated physical design watermarking framework. ACM Transactions on Design Automation of Electronic Systems 31 (6), 1-21
- [15] Javidnia, Zhang, Kundu, Koushanfar -"SWaRL: Safeguard Code Watermarking via Reinforcement Learning." arXiv preprint arXiv:2601.02602, 2026

31

31

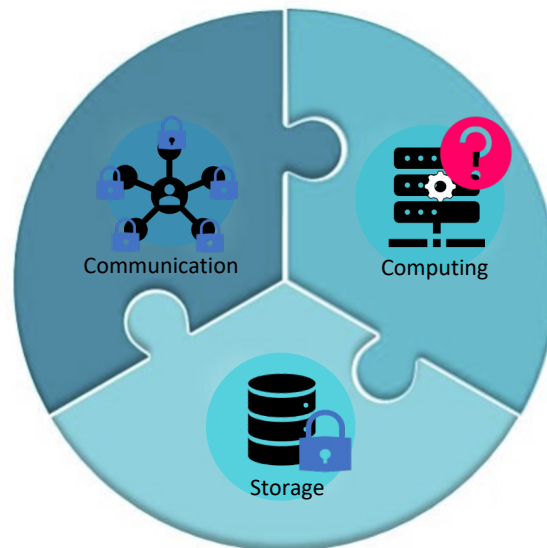
Outline

- Agentic AI and Design Automation
- The Peril: Vulnerabilities of Multi-Agent EDA
- Hijacked Agents: Attestation and Adaptive Unlearning
- Leaky Agents: MPC, ZKPs, and Watermarking
- Byzantine Agents: zPROBE, ZORRO, and ForTIFAI
- Outlook: A Verifiable Agentic Trust Stack

32

32

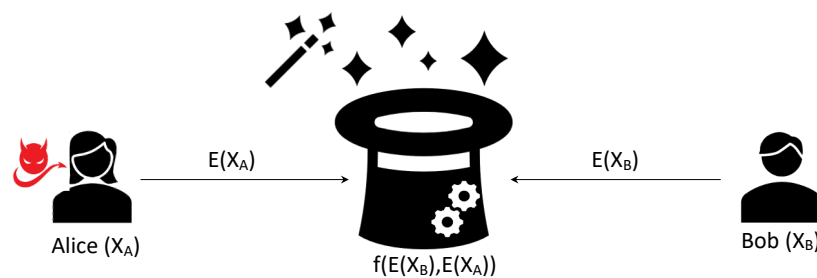
Encryption – What is Missing?



33

33

Secure Two-Party Computing: Blackbox Magic!

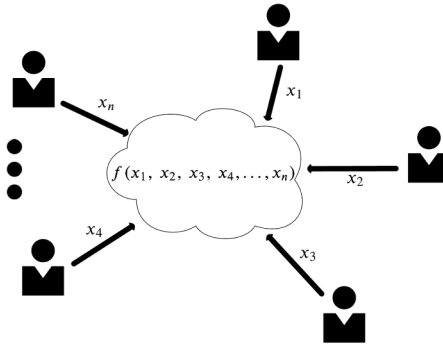


34

34

Privacy Preserving Computation

Multi-Party Computation (MPC)



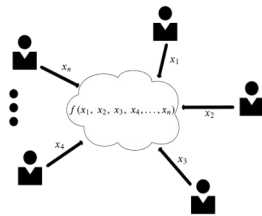
MPC allows N parties P_i , $0 < i \leq N$ to securely jointly compute a function $f(x_1, \dots, x_n)$ on private inputs from each party, without leaking any information to one another.

35

35

Privacy Preserving Computation

Multi-Party Computation (MPC)

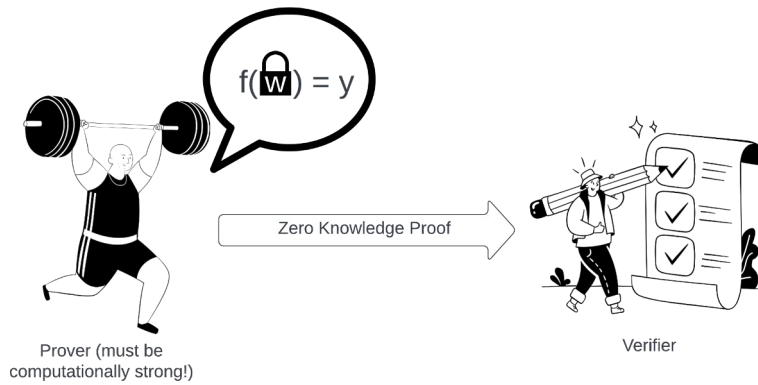


36

36

Privacy Preserving Computation

Zero-Knowledge Proofs (ZKPs)



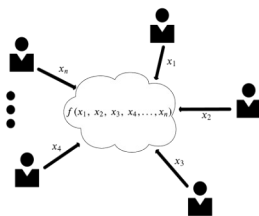
ZKPs allow a prover P to prove to a verifier V that they know a secret value w , without revealing any information about w .

37

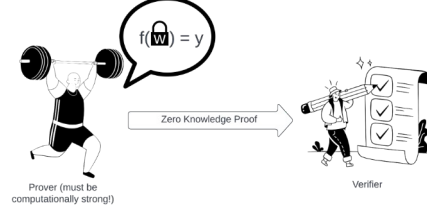
37

Privacy Preserving Computation

Multi-Party Computation (MPC)



Zero-Knowledge Proofs (ZKPs)



38

38

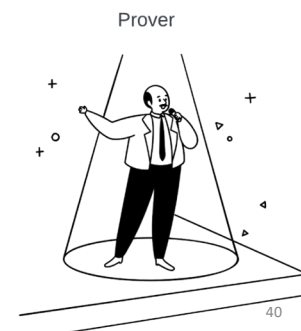
Where is Waldo?



39

39

Where is Waldo?



40

Where is Waldo?



41

41

Countering Leaky Agents

THREAT

PREVENT: COMPUTE ON SECRETS WITHOUT SEEING THEM

Full-context broadcast

Peril II · 1



COUNTERMEASURE

MPC, FHE, and ZKP co-design

Hybrid protocols, low-bitwidth models, MPC-aware circuits

PDK exposure to hosted models

Peril II · 4



CryptoGen

Encrypted generation with a reusable encrypted KV cache

DETECT AND PROVE: PROVENANCE THAT TRAVELS WITH THE ARTIFACT

Leaked model weights

Peril II · 3



EmMark

Watermarks in quantized on-device LLMs

Leaked text and summaries

Peril II · 2



REMARK-LLM

Learned, high-capacity text watermarks

Regenerated code

Peril II · 3



RoSeMary · SWaRL

Code watermarks with ZK verification; RL-learned marks

Layouts leaving the design house

Peril II · 4



ICMarks · AutoMarks

Placement-level layout watermarks, automated by GNNs



Cryptography prevents the leak; watermarks prove it, without revealing the key.

42

42

Private Computation at Agent Scale

PROBLEM

MPC, FHE, and ZKPs let agents compute on design secrets without seeing them. Applied naively to transformers, matrix-heavy layers and frequent interaction make the overhead prohibitive.

SECURITY MODEL

Parties mutually distrusting owners, e.g., an IP vendor and a design house, or a model owner and a user

Adversary semi-honest (honest-but-curious) by default; ZKPs extend integrity to malicious provers

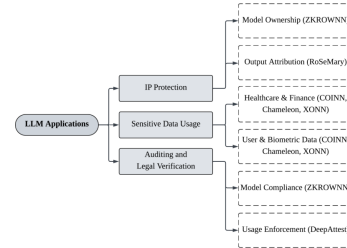
Guarantee no party learns more than the output; a verifier learns only validity

Tools MPC: joint computation · FHE: outsourced computation on ciphertexts · ZKP: proof without disclosure



The question is not whether, but at what cost.

Jandali, Zhang, Sheybani, Koushanfar, "Optimizing Privacy-Preserving Primitives to Support LLM-Scale Applications," arXiv 2025



Jandali et al., Fig. 1: privacy-preserving LLM applications

B

ADDRESSES PERIL II - 1

Full-context broadcast

Every agent sees the whole task context, including PDK data.

43

43

Co-Design Makes the Primitives Practical

KEY INSIGHTS

- Hybrid protocols (Chameleon)**
Linear layers on secret shares, non-linear ones in garbled circuits, heavy work moved offline.
- Low-bitwidth models (XONN)**
Binarized layers turn multiplications into XNORs that are free in garbled circuits.
- MPC-aware synthesis (MPCircuits)**
Logic synthesis that minimizes costly non-XOR gates.
- Crypto/ML co-design (COINN)**
Per-layer quantization and weight sharing searched jointly with MPC cost.



Treat the protocol as a design constraint of the model, not an afterthought.

Jandali, Zhang, Sheybani, Koushanfar, "Optimizing Privacy-Preserving Primitives to Support LLM-Scale Applications," arXiv 2025

RESULTS

133×

Chameleon vs. CryptoNets

93×

XONN vs. MiniONN; constant rounds

4.7–14.4×

COINN vs. prior art, no accuracy loss

44

44

CryptoGen: Query a Hosted LLM, Reveal Nothing

PROBLEM

Agents send proprietary RTL, PDK parameters, and prompts to cloud-hosted models. The user must trust the server with every prompt; the provider must protect its weights.

SECURITY MODEL

Setting client holds prompt and secret key; untrusted server holds the model

Adversary semi-honest server inferring prompts and outputs; no collusion

Goal prompt and outputs hidden from the server, with weights from the client

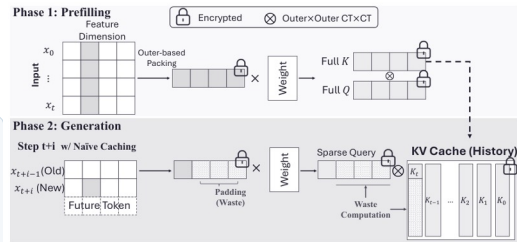
State the KV cache stays encrypted for the whole generation

Out of scope malicious deviation and side channels



Dual-sided privacy: my design stays mine, your model stays yours.

Zhang, Javidnia, Pardeshi, Lou, Koushanfar, "Secure Transformer Generation with Encrypted KV-Cache Reuse," arXiv 2026



CryptoGen, Fig. 3: outer-packed encodings waste slots in decoding

B ADDRESSES PERIL II - 4

PDK and NDA exposure

A hosted model sees foundry data no agreement clearly covers.

CryptoGen: Secure Generation Is Stateful

KEY INSIGHTS

1 Heterogeneous packing
Batched layout for prefill, single-token layout for decode: per-step cost independent of prompt length.

2 Encrypted KV-cache kernels
Ciphertext attention directly over a mixed-layout encrypted cache, with logarithmic reductions.

3 Cache management
Lazy HE-MPC noise refresh and slot-aware concatenation keep the cache compact.



Encrypted state is the enabler for private, long-horizon agents.

Zhang, Javidnia, Pardeshi, Lou, Koushanfar, "Secure Transformer Generation with Encrypted KV-Cache Reuse," arXiv 2026

RESULTS

4.4–7.6×

lower per-token latency than BOLT, 128–512 tokens

$O(L^2) \rightarrow O(L)$

attention cost across a generation

~0

perplexity change (30.66 vs. 30.63 plaintext)

RoSeMary: Prove Ownership, Never Show the Mark

PROBLEM

Code is low-entropy: token-level marks break compilation. Worse, proving ownership usually means disclosing the signature to an arbitrator, which invites forgery and forces re-encoding.

SECURITY MODEL

Adversary a user with full access to watermarked code; may refactor, rename, or forge a claim

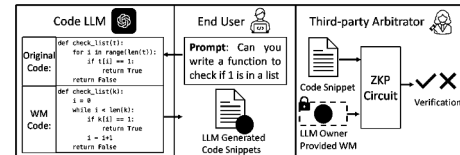
Lacks the insertion model and the owner's signature

Arbitrator semi-honest third party; honest setup of the ZK circuit

Owner needs signature confidentiality and knowledge soundness of the proof



Prove the mark is there; never show where.



RoSeMary, Fig. 1: watermarked code and ZK verification

B

ADDRESSES PERIL II - 3

Training-data memorization

Agents regenerate proprietary code; ownership must be provable.

Zhang*, Javidnia*, Sheybani, Marosok, Huang, Koushanfar, "Robust Zero Knowledge Verifiable Watermarking of Code LLMs with ML/Crypto Co-Design," ACM Trans. AI Security & Privacy 2026

47

47

RoSeMary: ML/Crypto Co-Design

KEY INSIGHTS

- Function-preserving marks**
A pretrained CodeT5 encoder embeds signatures through syntax and identifier edits; adversarial training adds robustness.
- Quantized decoder in ZK**
The watermark decoder runs inside a zkSNARK circuit with the owner's signature as a private witness.
- Reusable verification**
The same artifact can be verified repeatedly; no re-encoding after disclosure.



Provenance an arbitrator can trust, without trusting the owner.

RESULTS

0.97

detection AUROC with code functionality preserved

~120 ms

zero-knowledge ownership verification

+6.6%

AUROC (+26.6% TPR) over SrcMarker

Zhang*, Javidnia*, Sheybani, Marosok, Huang, Koushanfar, "Robust Zero Knowledge Verifiable Watermarking of Code LLMs with ML/Crypto Co-Design," ACM Trans. AI Security & Privacy 2026

48

48

ICMarks: Agent Asserts Ownership in the Chip Layout

PROBLEM

A placed-and-routed layout carries months of PPA tuning, and it must leave the design house for fabrication and test. Prior layout marks either ignore fence regions and macros (quality loss) or can be forged.

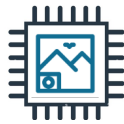
SECURITY MODEL

Adversary fabrication or test company holding the GDSII, with open-source physical design tools

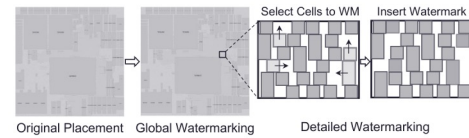
Knows the general watermarking algorithm, not the seeds, signature, or parameters

Goal remove or forge the mark while keeping wirelength and timing within 0.5%

Verification owner reverse-engineers netlist and cell positions from GDSII



In hardware, the artifact outlives every policy around it.



ICMarks, Fig. 5: global and detailed watermarking

B ADDRESSES PERIL II - 4 PDK and NDA exposure

The layout reaches parties no agent or agreement controls.

Zhang, Rajarathnam, Pan, Koushanfar, "ICMarks: A Robust Watermarking Framework for IC Physical Design IP Protection," ACM TODAES 2025

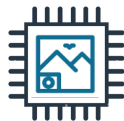
49

49

ICMarks: Two-Level Placement Watermarks

KEY INSIGHTS

- 1 **Global watermarking**
Score sliding windows for a low-density region that respects fences and macros; co-optimize it as a placement constraint.
- 2 **Detailed watermarking**
Shift cells that have room along x or y before detailed placement, which then absorbs the cost.
- 3 **Combined**
Both levels together: no quality loss and much stronger, harder-to-forge marks.



Watermark where the layout has slack, not where it is tight.

RESULTS

0%

wirelength and timing loss on ISPD'15, ISPD'19, ICCAD'15

>90%

extraction under removal attacks costing ~2% quality

7×10^{-114}

false-match probability, 50-bit mark

Zhang, Rajarathnam, Pan, Koushanfar, "ICMarks: A Robust Watermarking Framework for IC Physical Design IP Protection," ACM TODAES 2025

50

50

Outline

- Agentic AI and Design Automation
- The Peril: Vulnerabilities of Multi-Agent EDA
- Hijacked Agents: Attestation and Adaptive Unlearning
- Leaky Agents: MPC, ZKPs, and Watermarking
- Byzantine Agents: zPROBE, ZORRO, and ForTIFAI
- Outlook: A Verifiable Agentic Trust Stack

51

51

Countering Byzantine Agents

THREAT		COUNTERMEASURE	
Malformed updates, false verdicts Peril III · 1	➡	zPROBE Robustness bounds checked in zero knowledge	
Stealthy targeted backdoors Peril III · 2	➡	DROP Clustering, reputation, and distillation	
Poisoned handoffs Peril III · 1	➡	ZORRO Each hop scores, prunes, and proves it did so	
Self-poisoning Peril III · 3	➡	ForTIFAI Confidence-aware loss preserves the tails	

Every contributor, including the agent's own past, may be Byzantine.

52

52

zPROBE: Robustness Without Peeking

PROBLEM

Secure aggregation hides each contributor's update, which also hides the malicious ones. Checking each contributor individually breaks privacy. Privacy and robustness appear to conflict.

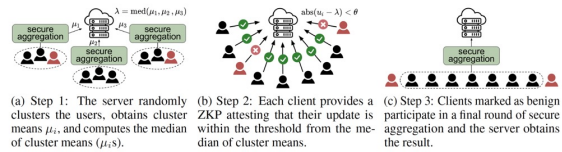
SECURITY MODEL

Server semi-honest, single server

Adversary a fraction of malicious clients

Attacks (1) modified local updates, (2) incorrectly computed masked updates, (3) inconsistent updates across steps

Goal Byzantine-robust aggregation that never reveals any individual update



zPROBE: clustering, ZK threshold checks, final aggregation



ADDRESSES PERIL III - 1

Byzantine agents

One false verdict poisons the whole optimization loop.



Check every contribution; see none of them.

Ghodsí*, Javaheripí*, Sheybani*, Zhang*, Huang, Koushanfar, "zPROBE: Zero Peek Robustness Checks for Federated Learning," ICCV 2023

53

53

zPROBE: Rank Statistics Checked in Zero Knowledge

KEY INSIGHTS

- 1 **Private robust statistics**
Median of randomly clustered means, with a threshold from their spread, computed under secure aggregation.
- 2 **ZK checks**
Each client proves its update lies within bounds and its masks were computed correctly.
- 3 **Probabilistic auditing**
Checking a small random fraction suffices: 8 checks give 99.5% detection in a 60k-parameter model.

RESULTS

+0.5–2.8%

accuracy over prior private, robust FL

Sublinear

scaling in the number of clients, a first

<1 s

per round, with ~2–4 MB communication



Spot checks in zero knowledge keep proofs affordable.

Ghodsí*, Javaheripí*, Sheybani*, Zhang*, Huang, Koushanfar, "zPROBE: Zero Peek Robustness Checks for Federated Learning," ICCV 2023

54

54

ZORRO: Poisoned Handoffs in Sequential Pipelines

PROBLEM

In split learning each client trains a head and tail and hands them to the next, like agents passing artifacts. Server-side defenses need the full model and scale poorly; client-side defenses can simply be skipped.

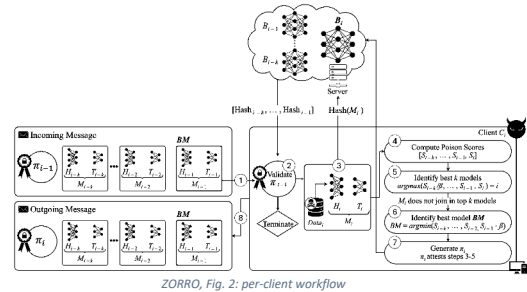
SECURITY MODEL

Adversary coordinated clients with full control of their data, training, head, and tail

Trusted no collusion with the server; benign clients are out of reach

Accountability a failed proof is traced to a specific client

Requires stop backdoors, keep utility, never share head or tail with the server



ZORRO, Fig. 2: per-client workflow

ADDRESSES PERIL III - 1

Byzantine agents

An agent hands a poisoned checkpoint downstream and reports success.



Self-reported checks are worthless; recall the agents that forged their own logs.

Sheybani*, Pegoraro*, Knauer*, Rieger, Mollakuqe, Koushanfar, Sadeghi, "ZORRO: Zero-Knowledge Robustness and Privacy for Split Learning," CCS 2025

55

55

ZORRO: Make Every Hop Prove Its Own Inspection

KEY INSIGHTS

- Frequency-domain poison score**
Backdoors shift low-frequency DCT components; each client scores and prunes the riskiest checkpoint.
- Interactive ZK enforcement**
A VOLE-based proof shows the defense ran on committed models; a Freivalds check makes the DCT ~100x cheaper.
- The attacker's dilemma**
Run the defense and get pruned, or deviate and get exposed.



Like a warning triangle: each driver flags the hazard for the next.

RESULTS

<6%

backdoor accuracy across datasets and architectures

<10 s

client overhead for 1M parameters; <1 GB RAM

1,000

clients with stable main-task accuracy

Sheybani*, Pegoraro*, Knauer*, Rieger, Mollakuqe, Koushanfar, Sadeghi, "ZORRO: Zero-Knowledge Robustness and Privacy for Split Learning," CCS 2025

56

56

ForTIFAI: When Agents Learn From Their Own Output

PROBLEM

Training on recursively generated data collapses models, and as little as 1% synthetic data can trigger it. Agentic EDA retrains on accepted RTL, generated testbenches, and summaries.

THREAT MODEL

Contaminant the model's own past generations mixed into its training data

Defender cannot reliably label real versus synthetic data

Failure distribution tails erode first: the corner cases verification must catch

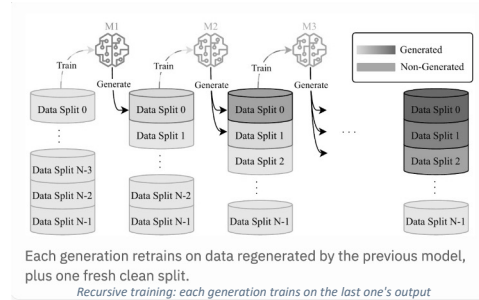
Scale larger models and datasets do not reliably prevent it



The agent's own past is an untrusted contributor.

Zibakhsh Shabgahi, Aghazadeh, Mirhoseini, Koushanfar, "ForTIFAI: Fending Off Recursive Training Induced Failure for AI Model Collapse," npj Artificial Intelligence 2026

57



ADDRESSES PERIL III - 3

Model self-poisoning

Retraining on its own outputs erases the corner cases.

57

ForTIFAI: Confidence-Aware Loss

KEY INSIGHTS

- 1 **Overconfidence is a signal**
Autoregressive models assign high confidence to text they generated.
- 2 **Truncated cross-entropy**
 $TCE(p) = 1[p \leq \gamma] \cdot CE(p)$: high-confidence tokens drop out of the loss, a label-free filter.
- 3 **Drop-in change**
One change to the training objective; shown at over a billion parameters.



Keeps the corner cases agents would otherwise forget.

Zibakhsh Shabgahi, Aghazadeh, Mirhoseini, Koushanfar, "ForTIFAI: Fending Off Recursive Training Induced Failure for AI Model Collapse," npj Artificial Intelligence 2026

58

RESULTS

2.3×

more synthetic data tolerated before collapse

>1B

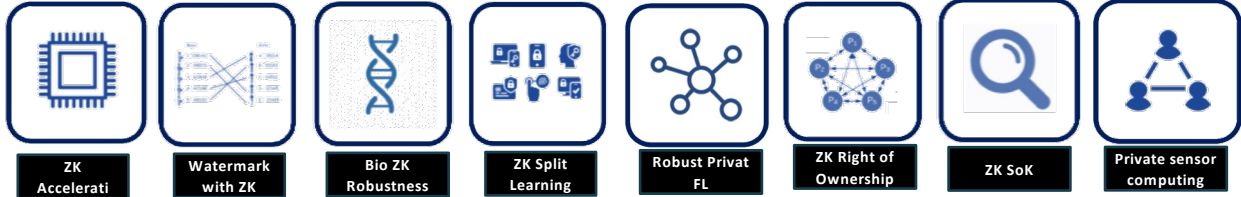
parameters, first at this scale

Open

benchmark for collapse in mixed-data settings

58

Privacy-Preserving and Verifiable Agentic AI



- [1] Songhori, Hussain, Sadeghi, Schneider, Koushanfar, TinyGarble: Highly Compressed and Scalable Sequential Garbled Circuits. S&P'15
- [2] Demmler, Dessouky, Koushanfar, Sadeghi, Schneider, Zeitouni, Automated Synthesis of Circuits for Secure Computation. CCS'15
- [4] Hussain, Riaz, Koushanfar, The Fusion of Secure Function Evaluation and Logic Synthesis. IEEE S&P M'21
- [5] Riaz, Songhori, Sadeghi, Schneider, Koushanfar, Toward Practical Secure Stable Matching. PETS'18
- [6] Riaz, Songhori, Koushanfar. PriSearch: Efficient Search on Private Data. DAC'17
- [7] Imani, Kim, Riaz, Messerly, Liu, Koushanfar, Rosing. A Framework for Collaborative Learning in Secure High-Dimensional Space. CLOUD'19
- [8] Songhori, Riaz, Hussain, Sadeghi, Koushanfar. ARM2GC: Succinct Garbled Processor for Secure Computation. DAC'19
- [9] Riaz, Javaheripi, Hussain, Koushanfar. MPCircuits: Optimized Circuit Generation for Secure Multi-Party Computation. HOST'19
- [10] Songhori, Zeitouni, Dessouky, Schneider, Sadeghi, Koushanfar. GarbledCPU: A MIPS Processor for Secure Computation in Hardware. DAC'16
- [11] Ahmed, Sheybani, De Almeida, Njungle, Gong, Kinsy, Koushanfar, "AMAZE: Accelerated MiMC Hardware Architecture for Zero-Knowledge Applications," ICCAD'24
- [12] Sheybani, Koushanfar, "You Can Have Your Cake and Eat It Too: Ensuring Privacy and Robustness in Federated Learning," AAAI Spring Symposium'24
- [13] Sheybani, Ghodsi, Kapila, Koushanfar, "ZKROWN: Zero-Knowledge Right of Ownership for Neural Networks," DAC'24
- [14] Ghodsi*, Javaheripi*, Sheybani*, Zhang*, Huang, Koushanfar, "zPROBE: Zero Peek Robustness Checks for Federated Learning," ICCV'23
- [15] Sheybani, Hussain, Zhang, Koushanfar, "SenseHash: Computing on Sensor Values Mystified at the Origin," IEEE TETC '22
- [16] Zhang*, Javidnia*, Sheybani, Koushanfar, "Robust and Secure Code Watermarking for Large Language Models via ML/Crypto Codeign," ICCAD'25
- [17] Sheybani*, Gong*, Ahmed, Njungle, Kinsy, Koushanfar, "Gotta Hash 'Em All! Speeding Up Hash Functions for Zero-Knowledge Proof Applications," ICCAD'25
- [18] Sheybani, Ahmed, Ehret, Kinsy, Koushanfar, "Zero-Knowledge Frameworks: A Systematic Survey," arXiv '25
- [19] Sheybani, Pegoraro, Knauer, Mollakuqe, Rieger, Koushanfar, Sadeghi, "ZORRO: Zero-Knowledge Robustness and Privacy for Split Learning," CCS'25

59

59

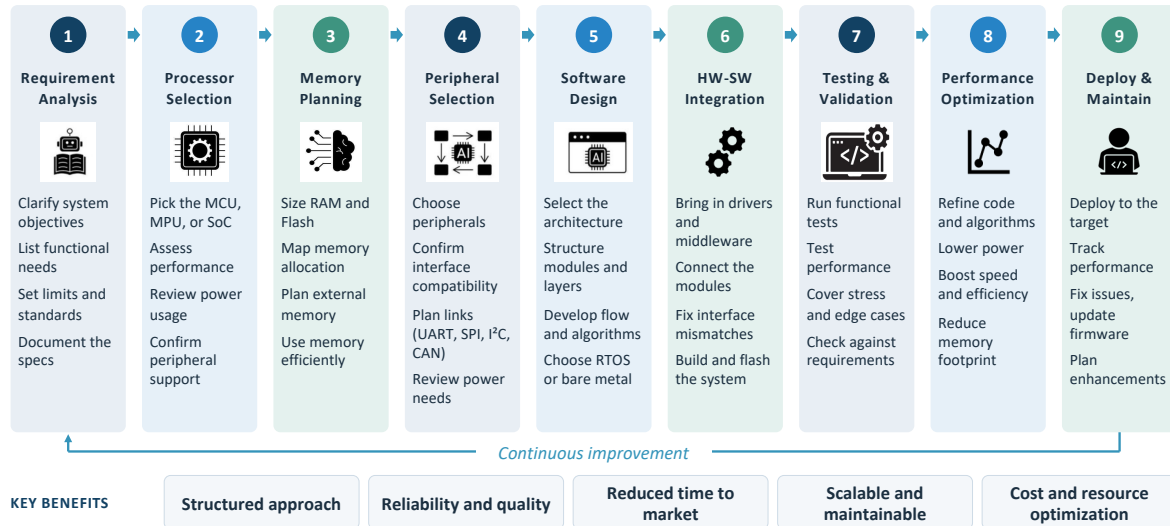
Outline

- Agentic AI and Design Automation
- The Peril: Vulnerabilities of Multi-Agent EDA
- Hijacked Agents: Attestation and Adaptive Unlearning
- Leaky Agents: MPC, ZKPs, and Watermarking
- Byzantine Agents: zPROBE, ZORRO, and ForTIFAI
- Outlook: A Verifiable Agentic Trust Stack

60

60

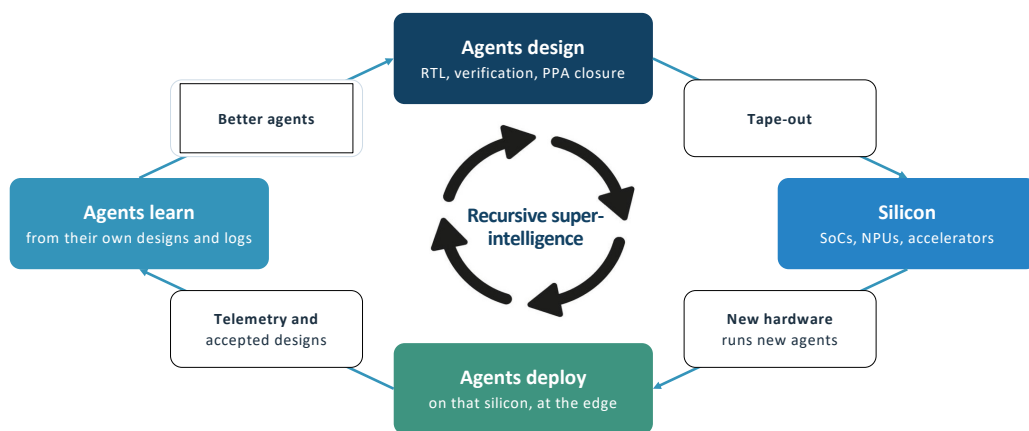
Embedded Systems Architecture Design Workflow – Safe Agentic Solutions



61

61

Agentic EDA Is Becoming Recursive



Agents design the chips that run the next agents: recursion runs through silicon.

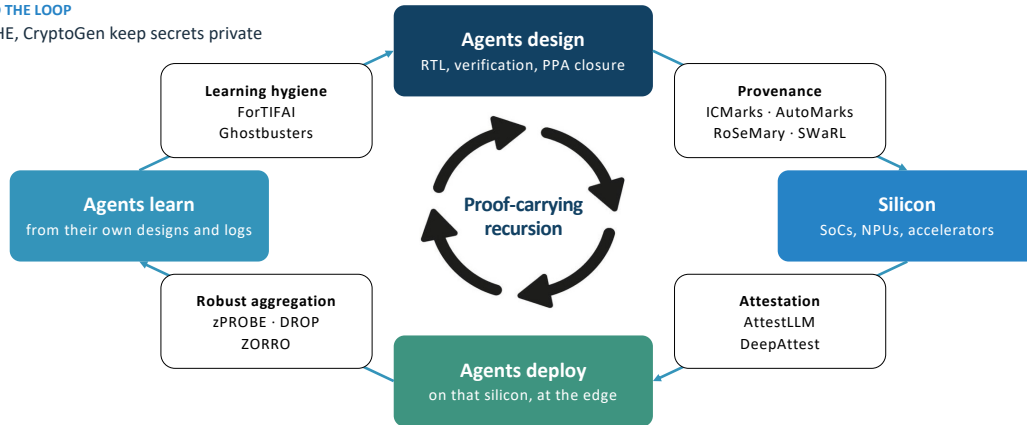
62

62

Proof-Carrying Recursion

AROUND THE LOOP

MPC, FHE, CryptoGen keep secrets private



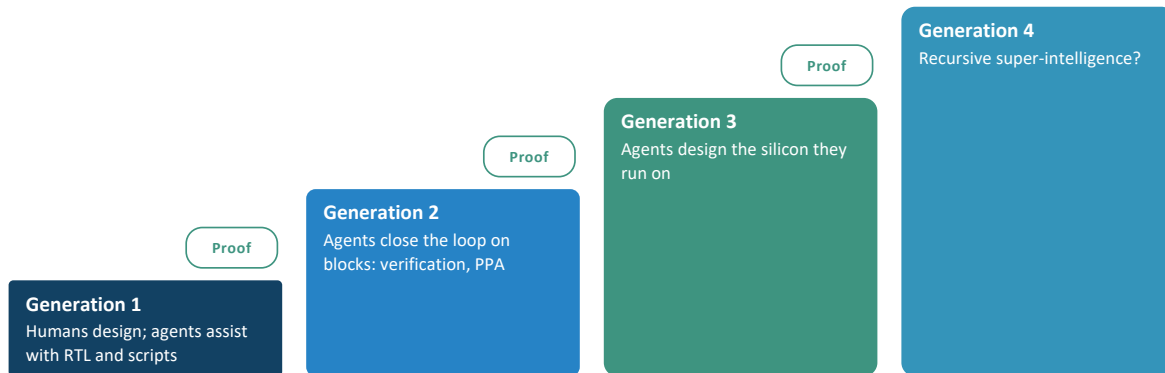
Each turn of the loop must prove itself before the next begins.

63

63

Advocating Evolvment, With Evidence

EACH GENERATION DESIGNS THE SILICON FOR THE NEXT; EACH STEP PASSES AN EVIDENCE GATE



Capability and verifiability must rise together.

64

64

Open Challenges for Recursive Agentic EDA

1

Verifiable patching

Prove an agent update was applied before the next tape-out

2

ZK at design scale

Proofs over million-cell netlists

3

Semantic injection

Malicious intent hidden in specs and READMEs

4

Leaderless signoff

Consensus without a trusted orchestrator

5

Real-time privacy

Encrypted generation inside the design loop

6

Faithful reasoning

Traces that match the tool calls made

7

Cross-generation drift

Robustness when each generation learns from the last

8

Proofs as evidence

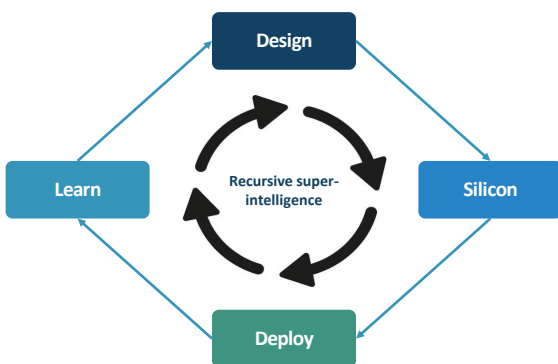
Cryptography meets PDK and NDA law

The next proofs must reach meaning, not just mechanics.

65

65

Summary



1

Agentic EDA

Agents now close the design loop, from RTL to layout, and design the silicon they run on.

2

Vulnerabilities

Hijacked, leaky, and Byzantine agents; in a recursive loop, every flaw is inherited.

3

Ongoing research

AttestLLM, Ghostbusters · CryptoGen, MPC/ZK co-design · EmMark to AutoMarks · zPROBE, DROP, ZORRO, ForTIFAI

4

Future directions

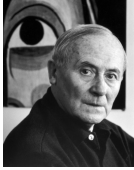
Proof-carrying recursion, platforms as roots of trust, PPA → PPAP: evolution toward recursive super-intelligence, with evidence.

Let the loop evolve, and make every turn prove itself.

66

66

Let there be Hope...



Joan Miro
(1893-1983)



The Hope of a Condemned Man, Joan Miro (1974)

67

67

Questions?

Acknowledgement

- ACES Lab @UCSD students, postdocs, and alumni
 - Special thanks to: Nojan Sheybani, Ruisi Zhang, Neusha Javidnia, Soheil Zibakhsh, Pedram Aghazadeh, Yaman Jandali, Xinqiao Zhang, Noah Margsak, Siam Hussain, Mojib Javaheripi, Mohammad Samragh, Huili Chen, Zahra Ghodsi, and Azalia Mirhoseini
- UCSD Center for Machine Intelligence, Computing, & Security (MICS)
- ACES Lab and MICS Collaborators, especially Prof. Sadeghi and his group @TU Darmstadt, also collaborators at UCF and NSF TILOS AI Institute
- Generous support from: MICS board & member companies (especially NG, Qualcomm, AMD, and Amazon), NSF, DARPA, SRC, Intel, Samsung SAIT Multi-University Award
- Questions? Farinaz@ucsd.edu



The Hope of a Condemned Man, Joan Miro (1974)

68

68