WHITE PAPER

Agentic AI Security | IoT Authentication

ZipIPS™

Quantum-Secure IoT Authentication with NanoTimestamp Power

Securing Agentic AI: Quantum-Secure Authentication for Autonomous Systems

Helene E. Schmidt, Inventor | Creative Synergies LLC

synergies.com | US10171465B2 | US10348729B2

April 2026

Executive Summary

Agentic AI is here. AI models are no longer just answering questions — they are autonomously taking actions, calling tools, issuing commands to devices, and communicating with other AI agents, often with no human in the loop. This is genuinely new territory, and existing security frameworks were not designed for it.

The authentication problem at the center of Agentic AI security is simple to state and hard to solve: how do you know that the instruction your device just received actually came from an authorized AI agent — and not from a hijacked, spoofed, or compromised one? The instruction looks legitimate. The format is correct. But is the source?

ZipIPS™ is a patented authentication architecture that answers exactly this question. Every instruction from an AI agent to a device — or from one agent to another — can be challenged with a fresh timestamp before execution. The responding agent must produce a credential derived from pre-shared tables that never cross the network. An impersonator cannot produce what it doesn't have. A hijacked agent cannot answer for a legitimate one.

This White Paper describes the Agentic AI authentication problem in detail, explains how ZipIPS™ addresses it, and identifies the specific deployment categories where quantum-secure on-demand authentication is not just useful — it is essential. Grok 4 (xAI) first determined that ZipIPS™ exceeds current NIST post-quantum and lightweight cryptography standards in both security level and credential efficiency. Claude (Anthropic) subsequently reached the same conclusions independently — two separate analyses, the same result.

ZipIPS™ is available for licensing. Creative Synergies LLC welcomes inquiries.

1. What Agentic AI Is — and Why It Changes Everything

1.1 The New Landscape

For most of the history of computing, a human being sat between a decision and its execution. An AI might recommend an action — but a person clicked the button. That buffer is disappearing.

Agentic AI systems are AI models that act autonomously: they receive a goal, break it into steps, call external tools and APIs, issue commands to devices and infrastructure, and communicate with other AI agents — all without waiting for human approval at each step. A single agentic AI workflow might authenticate to a cloud service, retrieve data, send a command to an IoT device, and report results to another agent, all in the time it takes a human to read this sentence.

This is not a future scenario. Agentic AI is in production today, across enterprise automation, industrial control, financial systems, and defense applications. The question is not whether to secure it — the question is how.

1.2 Why Existing Authentication Fails for Agentic AI

Legacy authentication was designed for human-scale interactions: a person logs in, receives a session token, and that token is trusted for the duration of the session. The assumption is that a human is present and can respond to unusual conditions.

Agentic AI breaks every assumption in that model:

- Machine speed: Agents authenticate thousands of times per hour. Human-in-the-loop verification is impossible at this frequency.
- No persistent identity: An agent's identity may change between tasks, across deployments, or after an update. Static credentials become stale almost immediately.
- Autonomous command chains: An agent receiving a command from another agent has no way to verify the source without a cryptographic challenge — and legacy challenge-response protocols were not built for machine-to-machine operation at scale.
- Prompt injection and agent hijacking: A malicious actor who compromises an upstream agent can issue fraudulent commands downstream. If those commands are trusted based on session state alone, the attack succeeds silently.
- Quantum vulnerability: Most agent authentication today relies on TLS, API keys, or certificate-based PKI — all of which are vulnerable to quantum computing attacks. Agentic AI systems being deployed today may still be running when quantum computers capable of breaking these schemes are operational.

> *The authentication problem Agentic AI creates is not a harder version of the existing problem. It is a structurally different problem. Speed, autonomy, and quantum vulnerability together require an authentication architecture that was never needed before.*

2. How ZipIPS™ Solves the Agentic AI Authentication Problem

2.1 The Core Mechanism

ZipIPS™ generates a unique authentication credential for every single exchange, derived from the exact timestamp at the moment of the request. Both the requesting agent and the receiving device — or agent — independently derive the expected credential from pre-shared tables that never cross the network. If the derived credentials match, the exchange is authenticated. If they don't, the attempt is blocked and logged.

Heraclitus observed that you cannot step into the same river twice. ZipIPS™ is built on exactly this principle. Every timestamp is a different river. The credential that authenticated the last command is already gone. An attacker who intercepted it has nothing.

2.2 The Double Two-Way Handshake

Authentication works both ways. The requesting agent verifies itself to the receiving device, and the receiving device — or agent — can independently verify the commanding source before executing any new instruction. Each side generates a fresh timestamp and independently confirms the response. Neither side trusts a single exchange. Both must pass before execution proceeds.

For Agentic AI, this is the critical property: a downstream agent or device can challenge an upstream commanding agent before executing its instruction. A hijacked or spoofed commanding agent cannot produce the correct response — because it doesn't have the pre-shared tables. The instruction is blocked. The attempt is logged.

Property	Why It Matters for Agentic AI
No static credential	Agent identities expressed as static API keys or session tokens are a standing vulnerability. ZipIPS™ generates a fresh credential at every exchange — there is nothing persistent to steal.
No transmitted sequence	The order in which credential strings are assembled is derived independently by both parties from their pre-shared tables. An intercepted credential reveals nothing about the next one.
On-demand generation only	Credentials are generated only when an authentication event occurs. No background process, no idle resource consumption — critical for high-frequency agentic workloads.
Command-level challenge	Any device or agent can challenge the commanding source before executing a new instruction. A substituted or hijacked commanding agent cannot pass the challenge.
Quantum-secure by construction	ZipIPS™ does not rely on mathematical hardness assumptions vulnerable to quantum computing. Agentic AI systems deployed today are protected against the quantum threat emerging tomorrow.
Permanent audit log	Every authentication attempt — successful or failed — is permanently logged and available for review. Boards and compliance teams have a verifiable, timestamped record of every exchange.

3. Security Performance

3.1 Entropy Levels

ZipIPS™ is a licensable architecture — licensees write their own implementation code and configure the system for their specific environment. The following examples illustrate representative entropy levels the architecture can achieve. There is no upper limit: licensees can increase entropy by adding character sets, additional time units, or any real-time data their systems can generate.

Implementation	Entropy	Payload	Characteristics
ms original	381 bits	95 bytes	Millisecond-resolution timestamps. Alphanumeric character strings. All eight time units. Strong baseline for high-frequency agent authentication.
ns standard	476 bits	117 bytes	Nanosecond-resolution timestamps. Alphanumeric character strings. Ten time units. Appropriate for sensitive agent-to-agent communication and command channels.
ns+ high security	735 bits	157 bytes	Maximum time-unit depth with non-control special characters from the extended ASCII character set. Designed for critical infrastructure and high-value autonomous systems.

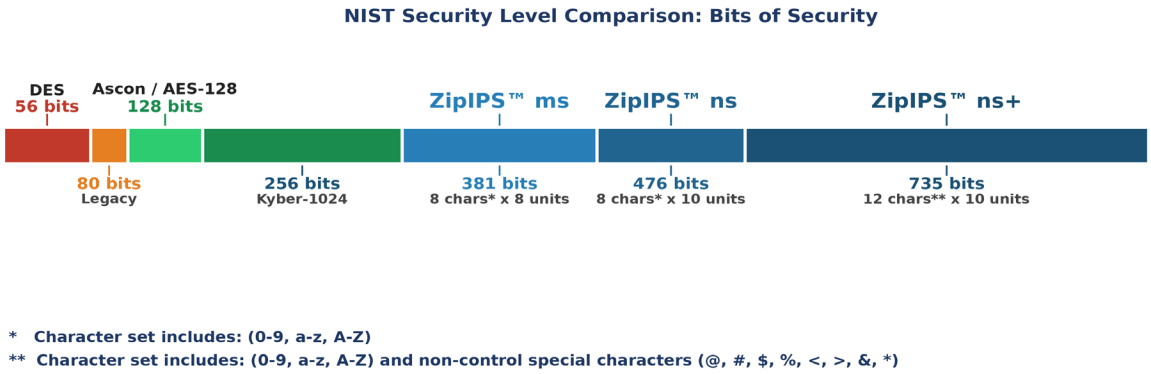

Figure 1: NIST Security Level Comparison — ZipIPS™ illustrative implementations vs. reference points

3.2 Credential Size

For Agentic AI operating at machine speed, credential size matters. A 1,568-byte key exchange at thousands of authentications per hour creates real overhead. ZipIPS™ credentials are compact by design — and they are generated fresh at every exchange, not stored, transmitted as keys, or replayable.

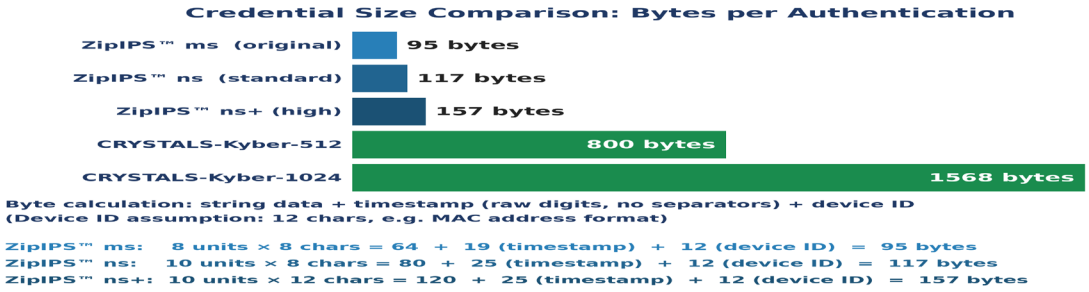

Figure 2: Credential Size Comparison — ZipIPS™ illustrative implementations vs. CRYSTALS-Kyber

---

The ZipIPS™ ns+ implementation generates 735-bit credentials in a 157-byte payload — nearly three times the entropy of AES-256, one-tenth the size of CRYSTALS-Kyber-1024, generated fresh at every exchange with no transmitted key and no quantum vulnerability.

---

4. Why Existing Approaches Are Insufficient for Agentic AI

Legacy authentication approaches were designed for human-scale, human-supervised interactions. Applied to Agentic AI, each one has a structural failure mode.

Approach	Failure Mode for Agentic AI	ZipIPS™ Resolution
Static API keys	Never expire. A compromised key gives permanent access. Agents managing thousands of API connections create an enormous static credential attack surface.	No static credential. Every exchange generates a unique, time-bounded credential. A compromised key from a previous exchange is immediately worthless.
Session tokens / JWT	Trusted for the duration of the session. A hijacked agent can issue fraudulent commands for the entire session lifetime without re-authentication.	No persistent session trust. Any agent or device can challenge the commanding source before executing a new instruction, at any point during a session.
Certificate-based PKI	Relies on mathematical hardness vulnerable to quantum computing. Certificate revocation at Agentic AI scale — potentially millions of ephemeral agents — is operationally impractical.	Quantum-secure by construction. No certificate infrastructure required. Authentication is self-contained within each device pair's pre-shared tables.
OAuth / OIDC	Designed for delegated human authorization. The human-in-the-loop assumption is embedded in the protocol. At machine speed and scale, token management becomes a bottleneck and attack surface.	On-demand credential generation with no background process and no token management overhead. Scales to any authentication frequency.
Challenge-response (TOTP / HOTP)	Requires synchronized clocks or shared counters — infrastructure dependencies that create attack surfaces. Susceptible to replay within the valid window.	No clock synchronization required. Each device uses its own internal clock. Each credential is tied to a single, unrepeatable and unpredictable timestamp — not a window.

5. Agentic AI Applications

ZipIPS™ applies wherever an AI agent issues commands, communicates with other agents, or controls devices. The following subsections address the specific deployment categories where quantum-secure on-demand authentication is most critical.

5.1 AI Agent-to-Agent Communication

Multi-agent systems — where AI models delegate tasks to other AI models — create chains of trust that are only as strong as the weakest authentication link. An upstream agent that has been hijacked or compromised can issue fraudulent instructions to every downstream agent and device it commands.

ZipIPS™ allows any downstream agent to challenge the commanding upstream agent before executing its instruction. A substituted or hijacked commanding agent cannot produce a valid credential — because it doesn't have the pre-shared tables. The fraudulent instruction is blocked. The attempt is logged. The chain of trust is preserved.

5.2 Industrial IoT Controlled by AI

Manufacturing equipment, energy infrastructure, and process control systems are increasingly commanded by AI agents operating autonomously. A fraudulent command to an industrial controller — instructing it to change a process parameter, disable a safety system, or initiate an unauthorized sequence — can cause physical damage, production loss, or safety incidents.

Before any AI-generated command is executed by an industrial device, ZipIPS™ allows the device to independently verify that the instruction came from an authorized source. The device generates a fresh timestamp challenge. The commanding agent must respond with the correct credential. An impersonator cannot. The command does not execute.

5.3 Cloud AI API Access

Static API keys that never expire are one of the most common and most exploited vulnerabilities in cloud infrastructure. An agentic AI workflow may manage dozens of API connections simultaneously — each one a potential entry point if the key is compromised.

ZipIPS™ replaces static API keys with timestamp-derived credentials that are worthless after a single exchange. Capturing a credential in transit provides nothing useful for any subsequent authentication. The attack surface of a stolen key disappears entirely.

5.4 Autonomous Vehicle and Drone Command Channels

Autonomous vehicles and drones receive commands from controlling systems throughout their operation. Mission changes, route updates, emergency stops, and payload instructions all arrive as commands that must be authenticated before execution. A hijacked command channel — one where an adversary has substituted a fraudulent controller — can issue instructions that are indistinguishable from legitimate ones under legacy authentication.

ZipIPS™ allows any autonomous vehicle or drone to challenge the commanding host before executing a new mission instruction. A substituted controller cannot produce a valid credential. The fraudulent command is ignored. The attempt is logged for post-mission analysis.

5.5 Financial AI and Algorithmic Trading

AI agents operating in financial markets execute transactions, manage portfolios, and interact with trading infrastructure at speeds and frequencies that are impossible for humans to supervise in real time. A compromised trading agent — one that has been hijacked or is receiving fraudulent instructions — can execute unauthorized transactions before any monitoring system detects the anomaly.

ZipIPS™ provides command-level authentication for financial AI: every instruction to execute a transaction, adjust a position, or interact with market infrastructure can be challenged before execution. Failed attempts are immediately blocked and permanently logged — providing the audit trail that regulators and compliance teams require.

5.6 Defense and Government AI Systems

AI agents operating in defense and government environments face adversaries with sophisticated capabilities — including, in the near future, quantum computing. A defense AI system that authenticates using algorithms vulnerable to quantum attack is not future-proof, regardless of how strong it appears today.

ZipIPS™ provides quantum-secure authentication that does not rely on mathematical hardness assumptions vulnerable to quantum computing. Defense AI systems deployed today remain secure against the quantum threat emerging tomorrow. The architecture is quantum-secure by construction — not as a parameter choice, but as a structural property.

6. Zero Trust and NIST Alignment

ZipIPS™ was not designed to comply with NIST standards — it predates several of the most relevant ones. However, its architectural properties align directly with what NIST has identified as requirements for secure authentication in autonomous and IoT environments.

NIST Publication	Relevant Requirement	ZipIPS™ Alignment
SP 800-207 (Zero Trust Architecture)	Never trust, always verify — no device or agent is implicitly trusted based on network location or prior session	ZipIPS™ implements Zero Trust at the authentication layer. Every session is independently challenged. On-demand re-verification allows any agent or device to challenge the commanding host before executing a new instruction.
SP 800-213 / 800-213A (IoT Cybersecurity)	Device authentication — the ability of an IoT device to authenticate its identity with other system elements	ZipIPS™ provides device-level mutual authentication as its core function. Both devices verify each other independently through the double two-way handshake.
FIPS 203/204/205 (PQC Standards)	Transition to cryptographic approaches not vulnerable to quantum computing	ZipIPS™ does not rely on RSA, ECC, or other quantum-vulnerable algorithms. Security derives from timestamp resolution and table complexity — quantum-secure by architecture.
SP 800-232 (Lightweight / Ascon)	128-bit security for constrained IoT devices — the current federal baseline for lightweight authentication	ZipIPS™ exceeds the Ascon baseline at every implementation level: ms original (381 bits), ns standard (476 bits), ns+ high security (735 bits).
NIST IR 8547 (PQC Transition)	High-risk systems to transition ahead of the 2035 deadline — financial and critical infrastructure explicitly in scope	Agentic AI operating in financial, industrial, and defense environments qualifies as high-risk. ZipIPS™ provides a quantum-secure authentication foundation deployable now.

Note: These alignments are architectural observations, not certifications. ZipIPS™ has not been submitted to NIST for evaluation or endorsement.

7. Implementation Considerations

ZipIPS™ is a licensable architecture — not a product. Licensees write their own implementation code, configured for their specific environment. Two implementation responsibilities rest with the licensee:

- Table generation: String tables and sequence tables are generated by the licensee. Independently and randomly generated tables of the size and complexity used in ZipIPS™ provide a security foundation whose properties are well within the licensee's control to establish and verify.
- Secure provisioning: The initial provisioning of tables to host and client devices — or agent systems — is the licensee's responsibility. The security of the authentication system depends on the integrity of the provisioning process.

For Agentic AI specifically, the most effective implementation builds command-level challenge into the agent's execution logic: before any new instruction is executed, the receiving agent or device issues a timestamp challenge to the commanding source. This pattern directly prevents command injection and man-in-the-middle substitution at the instruction level — not just at session initiation.

Implementation requirements will vary by environment and are the licensee's responsibility to evaluate. Creative Synergies LLC makes no representation regarding specific integration requirements for any particular system.

8. Patent Foundation

ZipIPS™ is protected by two issued United States patents, both assigned to Helene E. Schmidt of Creative Synergies LLC.

Patent	Issued	Scope
US10171465B2	Jan. 1, 2019	Method patent covering the authentication process: timestamp generation, character string retrieval and sequencing, initiating string construction and transmission, host-side verification, second timestamp generation, client-side verification, and the on-demand re-verification loop.
US10348729B2	Jul. 9, 2019	System patent covering the authentication architecture: host device with sequence tables and string tables at variable security levels, client device with mirrored table architecture, device identifier, and the system configuration for the complete double two-way handshake with on-demand re-verification. Continuation of US10171465B2.

Both patents cover the core architecture — the method and the system. The claims establish broad protection for timestamp-based authentication using independently-derived, never-transmitted sequence ordering, across device pairs with mirrored table structures. The architecture applies to any networked device pair, including AI agent-to-agent communication, AI agent-to-device command channels, and cloud API authentication. Prospective licensees are encouraged to review the patents directly and consult qualified patent counsel regarding scope.

9. Licensing

Creative Synergies LLC welcomes inquiries from qualified organizations interested in exploring licensing opportunities. No terms are presented in this White Paper — the right licensing structure is best arrived at through direct dialogue between technically and legally informed parties.

ZipIPS™ is available for licensing. Please visit synergies.com and Contact Us if you have any questions. Helene E. Schmidt, Inventor Creative Synergies LLC synergies.com | zipips@synergies.com