WHITE PAPER

Agentic AI Security | Industrial IoT Applications

ZipIPS™

Quantum-Secure IoT Authentication with NanoTimestamp Power

Securing Industrial IoT Controlled by AI: Authentication for ICS, SCADA, and Critical Infrastructure

Helene E. Schmidt, Inventor | Creative Synergies LLC

synergies.com | US10171465B2 | US10348729B2

April 2026

Executive Summary

In 2010, a piece of software called Stuxnet destroyed approximately one thousand uranium enrichment centrifuges at an Iranian nuclear facility. It did this not by erasing data or crashing systems — it did it by sending fraudulent commands to the industrial controllers managing the centrifuges, commanding them to spin at speeds that tore them apart, while simultaneously reporting normal operation to the human operators watching the displays. The operators saw nothing wrong. The centrifuges were destroying themselves.

That was 2010, with human attackers and manually crafted malware. Today, AI can identify authentication vulnerabilities in industrial control systems at machine speed, craft targeted attacks autonomously, and issue fraudulent commands faster than any human monitoring system can detect them. The Stuxnet playbook — fraudulent commands to industrial controllers producing physical consequences — is now available to any adversary with access to capable AI tools.

Industrial IoT controlled by AI is the fastest-growing and most physically consequential attack surface in cybersecurity. Manufacturing plants, power grids, water treatment facilities, oil and gas pipelines, and process control systems are all increasingly commanded by AI agents operating autonomously. Every command those agents issue is an authentication event. Under legacy authentication, those commands are trusted based on session credentials established at connection time — not verified at the moment of execution.

ZipIPS™ changes that. Every command from an AI agent to an industrial controller can be challenged with a fresh timestamp before execution. The commanding agent must respond with a credential derived from pre-shared tables that never cross the network. An impersonator — human or AI — cannot produce what it doesn't have. The fraudulent command does not execute.

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. Understanding ICS and SCADA — The Infrastructure at Risk

1.1 What ICS and SCADA Are

An Industrial Control System (ICS) is any computerized system that monitors and controls an industrial process. The term covers a broad range of technologies — from a single programmable logic controller (PLC) managing a manufacturing assembly line to a complex distributed control system (DCS) managing a chemical plant. What they share is the ability to translate digital commands into physical actions: open this valve, increase this temperature, start this motor, stop this conveyor.

SCADA — Supervisory Control and Data Acquisition — is a specific type of ICS designed for geographically distributed operations. A SCADA system might monitor and control hundreds of remote sensors and actuators spread across thousands of miles of pipeline, or manage power distribution across an entire regional grid. It collects real-time data from those remote assets, displays it to operators, and allows commands to be issued remotely.

The critical distinction between ICS/SCADA and conventional IT systems is consequence. When an IT system is compromised, data is stolen or disrupted. When an ICS is compromised and a fraudulent command is executed, physical equipment responds. Valves open or close. Motors accelerate or stop. Temperatures rise or fall. In the wrong context, at the wrong moment, those physical responses cause explosions, blackouts, contaminated water supplies, equipment destruction, and loss of life.

1.2 How Stuxnet Changed Everything

Before Stuxnet, the security assumption for industrial control systems was air-gap isolation — keep them physically disconnected from external networks and they are safe. Stuxnet destroyed that assumption permanently.

Stuxnet was a sophisticated piece of malware discovered in 2010, designed specifically to target Siemens programmable logic controllers managing uranium enrichment centrifuges at Iran's Natanz facility. Here is what made it remarkable: it did not crash the systems it infected. It did not announce itself. It sent fraudulent commands to the centrifuge controllers — commanding them to spin at frequencies that caused mechanical failure — while simultaneously intercepting the sensor data being reported to operators and replacing it with readings showing normal operation. The operators watched normal displays while their equipment destroyed itself.

Stuxnet required nation-state resources, years of development, and physical access to introduce the malware into an air-gapped facility. It demonstrated two things that remain true today: first, that fraudulent commands to industrial controllers produce physical consequences; and second, that the operators watching the displays may have no idea it is happening.

In 2025 and 2026, AI tools can replicate aspects of the Stuxnet playbook — identifying authentication vulnerabilities, crafting targeted commands, and evading detection — at a fraction of the time and resources Stuxnet required. The lesson of Stuxnet is not historical. It is a preview.

1.3 The Connectivity Problem

The air-gap that was supposed to protect industrial control systems no longer exists in most facilities. Over the past two decades, ICS and SCADA systems have been progressively connected — to corporate networks for operational data sharing, to cloud platforms for remote monitoring, to vendor systems for maintenance and updates, and now to AI management layers for autonomous optimization and control.

Each connection that was added for operational efficiency was also a potential attack path. The authentication systems protecting those connections were designed for IT environments, not for industrial controllers that were never meant to be networked. The result is a large and growing attack surface protected by authentication architecture that was not designed for the threat environment it now faces.

AI-controlled industrial IoT adds a new dimension to this problem. When a human operator issues a command to an industrial controller, there is at least the possibility of human judgment in the loop. When an AI agent issues commands autonomously — optimizing production schedules, adjusting process parameters, responding to sensor data — there is no human judgment between the command and its physical execution. The authentication layer is the only check.

> *Stuxnet showed the world that a fraudulent command to an industrial controller is not a cybersecurity event. It is a physical event. AI-controlled industrial IoT means those commands now arrive at machine speed, autonomously, continuously. The authentication layer is the last line of defense before the physical consequence.*

2. How ZipIPS™ Secures AI-Controlled Industrial IoT

2.1 Command-Level Authentication

The fundamental problem with session-level authentication in an ICS environment is timing. A session is authenticated at connection time. Commands are executed continuously throughout the session — potentially thousands of commands per hour in a busy industrial environment. A session credential that is valid at connection time remains valid for every command issued during that session, whether those commands come from the legitimate AI agent or from an attacker who has compromised or substituted it.

ZipIPS™ moves authentication to the command level. Before any industrial controller executes a command from an AI agent, it can challenge the commanding source with a fresh timestamp. The agent must respond with a credential derived from pre-shared tables that were provisioned before the session began and never cross the network during operation. The controller independently derives the expected credential and compares. If they match, the command executes. If they don't, the command is blocked and permanently logged.

In Stuxnet terms: a ZipIPS™-authenticated centrifuge controller would have challenged the fraudulent command source before executing the speed change. The Stuxnet malware, without the pre-shared tables, could not have produced a valid credential. The command would not have executed. The centrifuges would have continued operating normally.

2.2 The Double Two-Way Handshake in Industrial Environments

Authentication works in both directions. The AI agent verifies the industrial controller before sending commands, and the controller verifies the AI agent before executing them. Neither side trusts a single exchange. Both must pass before execution proceeds.

For industrial environments, the bidirectional property addresses a specific threat: a compromised or substituted AI management system issuing commands that appear to come from the legitimate controller management platform. The controller challenges the commanding AI before executing. A substituted system cannot pass the challenge. The fraudulent command chain is broken at the point of execution.

2.3 On-Demand Generation for Resource-Constrained Controllers

Industrial controllers — PLCs, RTUs, and embedded field devices — are often resource-constrained. They were designed to control physical processes, not to run complex cryptographic software. Authentication systems that impose significant computational overhead on these devices create operational problems.

ZipIPS™ generates credentials on demand only — no background process, no idle computation, no continuous cryptographic overhead. Authentication resources are consumed only when a command challenge occurs. Between challenges, the controller's computational resources are fully available for their primary purpose: controlling the physical process. This makes ZipIPS™ viable on the same class of constrained devices that ICS environments depend on.

3. Industrial IoT Applications

3.1 Manufacturing and Smart Factory

Modern manufacturing facilities operate networks of AI-controlled robots, assembly systems, quality control sensors, and logistics coordinators — all receiving commands from AI management systems that optimize production in real time. A fraudulent command to a manufacturing robot does not cause a data breach. It causes a production error, equipment damage, or a safety incident on the factory floor.

ZipIPS™ authenticates every command in the manufacturing AI chain. Before a robotic arm adjusts its operation, before a conveyor changes speed, before a quality control system accepts a parameter change, the receiving device challenges the commanding AI. A compromised or substituted management AI cannot pass the challenge. The manufacturing process continues operating on verified commands only.

3.2 Power Grid and Energy Infrastructure

Electric power grids are the most consequential ICS environment in civilian infrastructure. An AI system managing grid load balancing, substation switching, or distribution routing has the ability to affect power supply to millions of people. A fraudulent command to a substation controller — one that opens the wrong breaker at the wrong moment — can cascade into a regional blackout.

The NERC CIP standards (Critical Infrastructure Protection) mandate cybersecurity requirements for bulk electric systems. ZipIPS™ provides command-level authentication that directly addresses the ICS authentication requirements those standards identify, in a credential architecture that is quantum-secure by construction — protecting grid management AI against both current and emerging threat capabilities.

3.3 Water Treatment and Utilities

Water treatment facilities use ICS to manage chemical dosing, filtration, pumping, and distribution across municipal water systems. The consequences of a fraudulent command in this environment — incorrect chemical dosing, misdirected distribution, disabled safety systems — affect public health directly. In 2021, an attacker briefly gained control of a water treatment facility in Oldsmar, Florida, and attempted to increase sodium hydroxide levels to dangerous concentrations. The attempt was caught by a human operator who happened to be watching the display. An AI-controlled facility with no human in the loop would have depended entirely on the authentication layer to prevent execution.

ZipIPS™ provides that authentication layer. Every chemical dosing command, every pump control instruction, every distribution valve command from an AI management system is challenged before execution. A fraudulent command cannot produce a valid credential. It does not execute.

3.4 Oil, Gas, and Pipeline Infrastructure

Pipeline control systems manage the flow of oil, gas, and refined products across networks that span continents. Remote terminal units (RTUs) at pipeline stations receive commands from SCADA systems managing pressure, flow rates, and valve positions. A fraudulent command to a pipeline RTU — one that closes the wrong valve or changes pressure in the wrong direction — can cause equipment damage, spills, or explosions.

The geographic distribution of pipeline infrastructure makes authentication particularly important: many RTUs operate in remote locations with limited human oversight. An AI management system issuing fraudulent commands to a remote RTU may go undetected for hours. ZipIPS™ provides authentication at the RTU level — the command is challenged before execution regardless of whether a human operator is watching.

3.5 Process Control and Chemical Manufacturing

Chemical manufacturing and process industries depend on precise control of temperature, pressure, flow rates, and chemical concentrations. The process parameters that separate a normal production run from a catastrophic failure are often narrow. An AI agent managing a chemical process that receives a fraudulent command — one that adjusts a temperature setpoint, changes a feed rate, or modifies a pressure threshold — can initiate a runaway process before any safety system detects the anomaly.

ZipIPS™ authenticates every process control command. The tight tolerances of chemical manufacturing make command-level authentication not just a security measure but an operational safety measure — ensuring that only verified, authorized commands from legitimate AI management systems can affect process parameters.

4. Why Legacy ICS Authentication Is Insufficient

Legacy Approach	ICS-Specific Failure Mode	ZipIPS™ Resolution
Session-level authentication	Commands are trusted for the duration of a session established at connection time. A compromised AI agent issues fraudulent commands under a valid session credential for as long as the session persists.	Command-level challenge. Every command is verified before execution — not just the session. A compromised AI agent cannot issue executable commands without the pre-shared tables.
Static passwords / shared keys	ICS environments frequently use static, rarely-changed passwords for controller access — a known vulnerability. A captured credential provides persistent access to every controller it protects.	No static credential. Every command exchange generates a unique, unrepeatable and unpredictable credential tied to a single timestamp. A captured credential is immediately worthless.
Certificate-based PKI	Certificate management at ICS scale — potentially thousands of field devices — is operationally complex. Certificate revocation in an ICS environment may be slow or impractical. Quantum-vulnerable by construction.	No certificate infrastructure required. Authentication is self-contained within each device pair's pre-shared tables. Quantum-secure by architecture.
No mutual authentication	Most ICS authentication verifies the controller to the management system but does not require the management system to prove its identity to the controller. A substituted management system issues commands that the controller accepts without challenge.	Double two-way handshake. The controller challenges the commanding AI before executing. A substituted management system cannot produce a valid credential.
Air-gap assumption	The historical ICS security model assumed physical isolation from external networks. That isolation no longer exists in AI-controlled facilities. The authentication model has not kept pace with the connectivity.	ZipIPS™ was designed for networked environments. Pre-shared tables never cross the network during operation. The authentication is local to each device pair — functional regardless of network topology.

5. Regulatory and Standards Alignment

Standard	Requirement	ZipIPS™ Alignment
NERC CIP (Bulk Electric)	Cybersecurity requirements for bulk electric system assets, including access management and electronic security perimeter protection for industrial control systems	ZipIPS™ provides command-level access authentication for AI-controlled grid management systems, exceeding the access management requirements NERC CIP identifies for ICS environments.
IEC 62443 (Industrial Security)	Security standards for industrial automation and control systems, including authentication requirements for ICS components and the systems that command them	ZipIPS™ provides mutual device authentication meeting IEC 62443 authentication requirements, with credential complexity that exceeds the standard's security level specifications.
NIST SP 800-82 (ICS Security)	Guide to industrial control system security, including authentication and access control recommendations for ICS environments	ZipIPS™ addresses the device authentication requirements identified in SP 800-82 for ICS/SCADA environments, including mutual authentication and protection against replay attacks.
CISA ICS Guidance	Critical Infrastructure Security Agency guidance on securing industrial control systems against cyberattacks, including authentication hardening recommendations	ZipIPS™ command-level authentication directly addresses the authentication vulnerabilities CISA identifies as primary attack vectors in ICS environments.
FIPS 203/204/205 (PQC Standards)	Transition to quantum-safe cryptography — industrial infrastructure is identified as high-risk requiring early transition	ZipIPS™ is quantum-secure by architecture. ICS environments authenticated with ZipIPS™ meet the quantum-safe requirement without a cryptographic migration project.

Note: These alignments are architectural observations, not certifications. ZipIPS™ has not been submitted to NERC, IEC, NIST, or CISA for evaluation or endorsement. Prospective licensees should conduct their own evaluation of compliance applicability for their specific ICS environment.

6. Security Performance

ZipIPS™ is a licensable architecture. Implementing organizations configure the system for their specific ICS environment. The following illustrative entropy levels represent what the architecture can achieve. There is no upper limit — entropy can be increased by adding character sets, time units, or any real-time data the implementing system can generate.

Implementation	Entropy	Payload	Characteristics
ms original	381 bits	95 bytes	Millisecond-resolution timestamps. Alphanumeric character strings. Eight time units. Suitable for resource-constrained field devices including PLCs and RTUs in manufacturing and pipeline environments.
ns standard	476 bits	117 bytes	Nanosecond-resolution timestamps. Alphanumeric character strings. Ten time units. Appropriate for SCADA command authentication in power grid, water treatment, and process control environments.
ns+ high security	735 bits	157 bytes	Maximum time-unit depth with non-control special characters from the extended ASCII character set. Designed for highest-consequence ICS environments including nuclear facility controls and chemical process management.

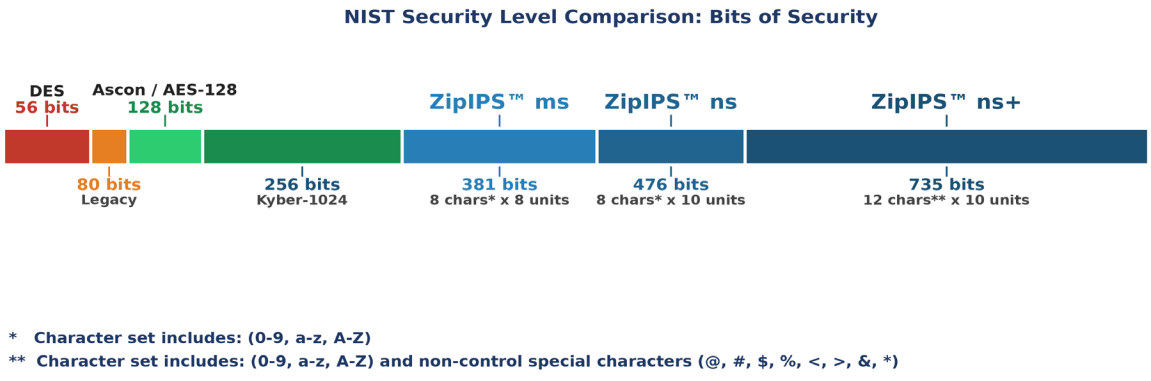

Figure 1: NIST Security Level Comparison — ZipIPS™ illustrative implementations vs. reference points

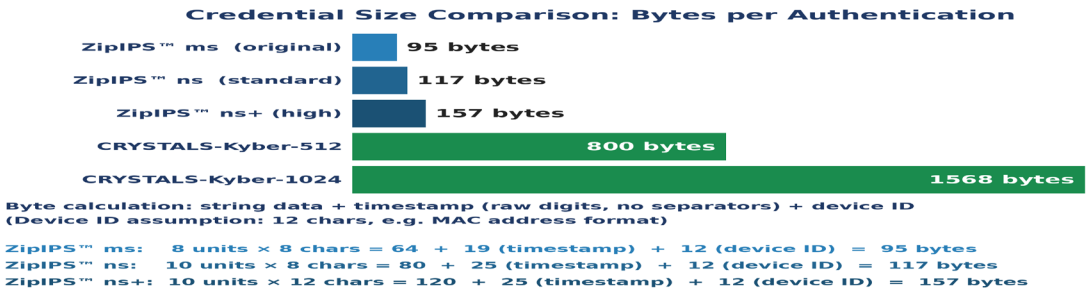

Figure 2: Credential Size Comparison — ZipIPS™ illustrative implementations vs. CRYSTALS-Kyber

ZipIPS™ ms original generates 381-bit credentials in a 95-byte payload — strong enough for the most resource-constrained field devices, generated on demand only, with no background computational overhead. A PLC authenticating AI commands with ZipIPS™ spends its computational resources controlling the physical process — not running continuous cryptographic processes.

7. Implementation Considerations

ZipIPS™ is a licensable architecture — not a product. Implementing organizations write their own code, configured for their specific ICS environment, device capabilities, and operational requirements. For industrial IoT specifically, three implementation considerations are important:

- Command-level integration: The most effective implementation builds the timestamp challenge into the controller's command-processing logic — every received command is challenged before execution. For ICS environments, this means the authentication check happens at the controller level, not at the network perimeter, ensuring that fraudulent commands are blocked at the point of physical execution.
- Resource-constrained device compatibility: ZipIPS™ on-demand generation is compatible with resource-constrained field devices. The implementing organization should evaluate the computational capabilities of their specific field device population and configure the implementation accordingly — ms original for the most constrained devices, ns standard or ns+ for controllers with greater computational headroom.
- Table provisioning and rotation: Pre-shared tables are loaded during device provisioning. In ICS environments where field devices may be physically inaccessible, table rotation policy requires careful operational planning. The implementing organization is responsible for defining and executing table rotation procedures appropriate to their specific operational environment.

Implementation requirements will vary by ICS environment, device population, and operational constraints. These are the licensee's responsibility to evaluate. Creative Synergies LLC makes no representation regarding specific integration requirements for any particular ICS installation.

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. Applicable to any networked device pair, including AI agent-to-ICS controller command channels.
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.

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