Billions of IoT Devices Need Authentication Without Keys

by Nick Clark | Published March 27, 2026 | PDF

The IoT industry deploys billions of devices with authentication models designed for servers in data centers. Certificates require rotation infrastructure that most devices cannot support. Pre-shared keys require secure provisioning that does not scale. Hardware security modules add cost that commodity sensors cannot absorb. The result is that most IoT devices either operate with weak authentication or with credentials that are never rotated. Keyless identity offers an alternative built for the constraints that IoT devices actually face.

Why key management fails at IoT scale

A certificate-based identity system requires each device to store a private key securely, communicate with a certificate authority for issuance and renewal, and rotate certificates on a schedule. For a server in a data center with dedicated operations staff, this is manageable. For a temperature sensor on a factory

floor, a soil moisture probe in a field, or a pressure gauge on a pipeline, it is not.

The constraints are physical and economic. Constrained devices lack tamper-resistant storage for private keys. They lack reliable network connectivity for certificate renewal. They lack the compute resources for complex TLS handshakes. And at commodity price points, adding a hardware security module to every device is economically unviable.

The result is a security gap that grows with every deployment. Manufacturers ship devices with pre-shared keys that are identical across production batches. Certificates are issued at manufacture and never rotated. Devices authenticate with credentials that were provisioned years ago and have never been validated since. The authentication model was not designed for devices that ship in millions and operate unattended for a decade.

Why lightweight alternatives still depend on keys

The IoT security community has developed lighter-weight protocols: DTLS for constrained transports, EDHOC for efficient key exchange, OSCORE for object-level security. These reduce the computational cost of cryptographic operations, but they do not eliminate the fundamental dependency on stored key material.

Every lightweight protocol still requires the device to possess and protect a secret. A lighter handshake is still a handshake that proves possession of a stored credential. A more efficient key exchange still produces a key that must be stored securely. The operational problems of key provisioning, rotation, and revocation at billion-device scale remain.

How keyless identity works for IoT devices

Keyless identity replaces stored credentials with behavioral continuity. Instead of proving possession of a pre-provisioned secret, a device authenticates by demonstrating continuity with its own operational history. Each authentication event generates a hash from locally available entropy: sensor readings, timing jitter, environmental conditions, internal state. The hash is unique to that device at that moment.

Over time, these hashes form a trust slope: a trajectory of authenticated interactions that becomes progressively harder to forge as it deepens. A device that has been generating consistent trust slope entries for months carries an identity that is computationally expensive to impersonate, because forging it requires reproducing the exact sequence of local entropy across the entire history.

For constrained devices, this approach has concrete advantages. No secure key storage is required because there is no persistent key. No certificate authority communication is needed because there are no certificates to issue or rotate. No pre-shared key provisioning is necessary because the device builds its identity from its own operational behavior after deployment.

A device that goes offline and reconnects can re-establish its identity by demonstrating continuity with its previous trust slope. The gap in connectivity does not invalidate the identity. It simply means the slope has a dormancy period that is evaluated as part of the continuity assessment.

What deployment looks like

An IoT deployment using keyless identity ships devices without pre-provisioned credentials. Each device begins building its trust slope from the moment it is powered on and starts interacting with its environment. Initial trust is low, and the device operates in a restricted scope. As the trust slope deepens through consistent authenticated interactions, the device earns broader access.

For fleet operators managing hundreds of thousands of devices, this eliminates the provisioning bottleneck entirely. No credential database to maintain. No rotation schedule to enforce. No bulk key revocation when a production batch is compromised. A compromised device's trust slope diverges from its established trajectory, and the divergence is detectable without consulting a central authority.

For critical infrastructure operators, keyless identity provides authentication that is resistant to both quantum computing threats and operational key management failures, because there are no keys to quantum-attack and no keys to mismanage. The identity is the device's behavior, not a stored secret that can be exfiltrated.

Keyless Identity All 21 steps →

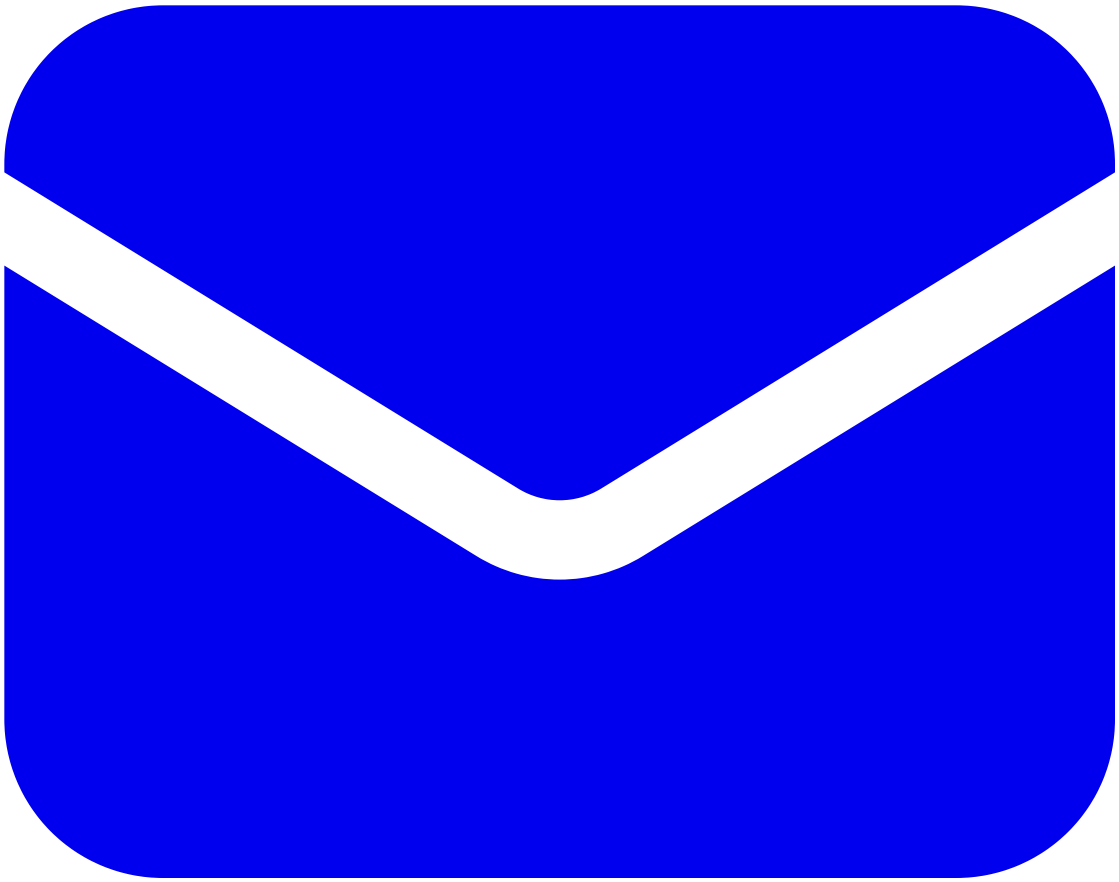
Identity from accumulated continuity. Post-quantum by construction.

Patent
US 19/388,580 · published
Primary Technical Disclosure
○ Stateless Device Pseudonymity and Secure Messaging in Cognition-Native Systems
Secondary Technical
○ Continuity-Based Biological Identity Using Trust-Slope Validation○ Trust Slope as Identity Primitive: Cumulative Hash Chains Replace Static Credentials○ Dual-Source Identity Derivation: Hardware Anchors and Local State Vectors Combined Per Epoch○ Stateless Symmetric Encryption: Session Keys Derived From Current Identity State○ Two-Stage Message Authentication: Transport Continuity Screening Before Semantic Validation○ Agent-Substrate Slope Entanglement: Binding Every Mutation Step to Its Execution Host○ Append-Only Mutation Lineage Log: Forward-Secure Identity Transition Chains○ Cumulative Slope Validation Across Substrates: Multi-Node Provenance Verification○ Quorum-Based Identity Recovery: Peer Attestation After Memory Loss○ Entropy Anchor Rotation: Proactive Identity Reseeding With Forward Links○ Biometric-Assisted Reseeding: Privacy-Preserving Fuzzy Extractors for Anchor Rotation○ Delayed Slope Validation: Bounded Proof Windows for Disconnected Environments○ Sparse Trust Slope Recovery: Compact Checkpoints for Resource-Constrained Devices○ Predictive Identity Validation: Drift Detection Before Full Discontinuity○ Legacy PKI Fallback: Session-Scoped Adapters With Strict Isolation Boundaries○ Post-Quantum Alignment: Hash-Based Security Without Vulnerable Hardness Assumptions
Applications (General)
○ Trust Slope Entanglement: Cryptographic Lineage for Semantic Agents○ Post-Quantum Enterprise Identity Migration● Billions of IoT Devices Need Authentication Without Keys○ Financial Identity Without Credential Databases○ Patient Identity Through Behavioral Continuity○ Supply Chain

Authentication Without PKI○ Smart Building Access Through Continuity○ Vehicle Operator Identity Binding○ Displaced Person Identity Without Documents
Applications (Specific)
○ Okta Centralized Enterprise Identity. The Keys That Prove It Are Still Stored Somewhere.○ Auth0 Made Developer Identity Easy. The Credential Model Underneath Did Not Change.○ YubiKey Made Hardware Authentication Practical. The Key Is Still the Vulnerability.○ CLEAR Made Airport Identity Fast. It Built a Biometric Database to Do It.○ Worldcoin Scans Irises to Prove Humanity. The Proof Depends on a Central Enrollment System.○ Jumio Automated ID Verification. The Verification Still Depends on Documents.○ Microsoft Entra Unified Cloud Identity. Identity Still Depends on Stored Credentials.○ Ping Identity Built Enterprise Federation. The Federation Depends on Shared Secrets.○ OneLogin Simplified Enterprise SSO. The SSO Token Is Still a Credential.○ Duo Security Made MFA Ubiquitous. The Second Factor Is Still a Credential.○ Thales HSMs Protect Key Material. The Keys Still Exist.○ Entrust Issues Digital Certificates. The Certificate Is a Stored Credential.○ DigiCert Secures the Web With TLS Certificates. The Certificate Model Has Structural Limits.○ Let's Encrypt Made TLS Free. The Certificate Model Is Still the Same.
Keyless Identity overview →
AQ
deterministic
autonomy

Legal

Last updated: 2026-03-03



-
- nick@qu3ry.net
- 72 28 14 36 01

Invented by Nick Clark | Founding Investors: Devin Wilkie