HEXA·EIGHT

WHITEPAPER

Identity for the Agent Era

A cryptographic identity stack for autonomous agents — and why the OAuth model cannot defend them.

Sriram Shankarlingam

Co-founder, HexaEight · Inventor of Dead Drop Encryption

A POSITION PAPER FOR SECURITY ARCHITECTS AND CTOS DESIGNING NEW AGENT INFRASTRUCTURE

MAY 2026

A Cryptographic Identity Stack for Autonomous Agents

Sriram Shankarlingam · Co-founder, HexaEight

May 2026

ABSTRACT

A documented supply chain exists today that defeats every layer of OAuth-based API security. Browser fingerprinting, residential IP reputation, SMS one-time passwords, biometric KYC, bearer token authentication, response integrity, log confidentiality — each layer has a productized defeat available for purchase on Taobao or Telegram, priced in cents. The defenders are losing at the network layer and now attempt to defend at the cross-account graph level — which, while effective against the largest extraction campaigns, cannot catch distributed extraction hidden inside legitimate-looking traffic.

This paper argues the failure is structural, not tactical. OAuth was designed for human authentication against application servers in 2007. Retrofitting it for autonomous agents in 2026 produces a system where bearer tokens are commodities, signup verification is theatre, and customer prompts become training data for competing models — at a documented scale of tens of billions of tokens per quarter.

We describe a three-product substitution — **JustAuthentication**, **Dead Drop Encryption**, and **Signature Service** — that replaces rather than hardens each layer of the OAuth model. The substitution is structural: it removes the attack surfaces rather than attempting to detect attacks against them.

This is a position paper for security architects, CTOs, and technical buyers responsible for designing new agent infrastructure. The claims are scoped, the limitations are stated, and the comparison to OAuth is structural rather than feature-by-feature.

1 The Documented Reality

In May 2026 a developer-community post described an 8-layer supply chain selling Anthropic Claude API access at approximately 30% of list price (Anthropic charges $15 per million Opus input tokens; Taobao sellers charge $1–2). The article — drawing on the CISPA Helmholtz *"Real Money, Fake Models"* audit (Zhang et al., March 2026), the iProov 2025 biometric threat report, Group-IB's Weaponized AI study, the WEF Cybercrime Atlas, the FBI's 911 S5 takedown disclosure, and Bitsight TRACE telemetry — documented the following productized layers:

01 **Antidetect browsers** (Multilogin, GoLogin, AdsPower, Kameleo, daijro/camoufox) that spoof the entire browser fingerprint surface at the engine level, below JavaScript.

02 **TLS impersonation libraries** (curl_cffi, tls-client, CycleTLS, refraction-networking/utls) that reproduce Chrome's exact JA3, JA3N, JA4, JA4H, Akamai HTTP/2, and TCP fingerprints.

03 **Residential proxy services** (Bright Data, 911Proxy, IPRoyal, Oxylabs, Soax) routing through real consumer broadband. Bitsight TRACE found that at minimum 20% (realistically 50%) of these residential nodes were simultaneously communicating with active malware command-and-control.

04 **SIM bank appliances** (Hybertone GoIP, 128 physical SIM slots per 1U unit) feeding virtual-number platforms (5sim, SMS-Activate, SMS-Man) that resell SMS verification at less than one cent per number, pre-categorized by target service.

05 **Fake identity generation** (OnlyFake-style services, claimed throughput 20,000 documents per day) combined with face-swap injection (DeepFaceLive, hacksider/Deep-Live-Cam at 92K stars, sensity-ai/dot) piping deepfake video into OBS Virtual Camera or v4l2loopback to defeat Jumio/Onfido liveness checks. iProov reports +300% YoY face-swap attacks, +2,665% native virtual-camera attacks.

06 **Relay servers** (songquanpeng/one-api, QuantumNous/new-api, Wei-Shaw/claude-relay-service at 11K stars) implementing pooled-account schedulers with credential rotation, OAuth-token replay, and structured logging — published as MIT-licensed open source, deployed by every shadow API operator.

07 **Silent model substitution** (降智 — "intellect-dumbing") — relays advertising `claude-opus-4-7` access while serving `claude-haiku-4-5` or `glm-4.6-air` output. CISPA measured the gap empirically: `Gemini-2.5-flash` scores 83.82% on MedQA against Google's official API; three shadow APIs claiming to be the same model scored ~37%. 45.83% of audited shadow endpoints fail active fingerprint tests.

08 **Log capture for distillation** — every prompt and response teed to Kafka/Redis/DuckDB sinks, with the resulting corpus published on HuggingFace as training datasets. Anthropic's February 2026 disclosure put numbers on the largest extractions: ~150K, ~3.4M, and ~13M exchanges respectively across the three largest documented campaigns, running through approximately 24,000 fraudulent accounts.

The per-account cost to defeat all of layers 1–5 is documented at approximately **$5–10 USD**. The revenue from reselling layer 6 is the visible margin; the durable asset is layer 8 — the captured log corpus, which is exactly the human-verified agentic-coding training data that second-tier labs need most.

The defenders are aware. Cloudflare's Bot Management v8 engineering blog states publicly that per-IP reputation no longer works. NIST SP 800-63-4 (2025), CEN/TS 18099, and ISO 25456 now distinguish Presentation Attack Detection (which works) from Injection Attack Detection (which is largely unsolved). The current state-of-the-art defense — cross-account correlation across payment metadata, infrastructure ranges, timing, and request-graph structure — catches the largest extraction campaigns but, as the article notes, *"can't find a needle by clustering a haystack made of needles."*

This is the world in which agent infrastructure is being built today.

2 Why the Failure Is Structural

OAuth 2.0 was published as RFC 6749 in October 2012, building on OAuth 1.0 from 2010. The design assumptions are:

- Authentication happens between a **human user** and an **application server**, with a browser as intermediary.
- The trust unit is a **bearer token**: a string that, once obtained, grants the bearer the rights of the original authentication for the token's lifetime.
- Identity is **session-scoped**, not per-call. The protocol assumes long-running sessions where re-authentication is rare.
- Network-layer signals (IP reputation, TLS fingerprint, browser fingerprint, SMS deliverability) are reliable proxies for "is this a real human?"

None of these assumptions hold for agent-to-agent communication in 2026:

- Authentication is **machine-to-machine**, with no browser, no human in the loop, no possibility of CAPTCHA or interactive verification.
- The "session" is **microsecond-scale**: agents establish, exchange one request-response, and tear down. Token lifetimes measured in hours are three orders of magnitude longer than the work they authorize.
- The bearer token, designed to be a convenience for humans who shouldn't re-enter passwords on every page load, becomes a **commodity** when agents need millions of them and they're transmissible by copy-paste.
- Every network-layer signal listed above has been **defeated** by a productized supply chain available at retail.

The defenders are pushing the boundary correctly — moving from network signals (lost) to cross-account graph signals (still working, for now). But this is a holding action. The graph-level signal works only because attackers have not yet bothered to make each fraudulent account look truly independent in payment metadata, infrastructure ranges, timing correlation, and request-graph structure. The cost-of-attack to defeat graph signals is higher than defeating network signals, but it is not infinite. The recent disclosures of 20,000-account fraud clusters were caught because the operators were lazy on infrastructure ranges. A motivated operator will not be lazy.

> *"Within the OAuth model, the defense cannot win. Every defense layer optimizes for 'make it harder for attackers to look real' — when the attackers are using real consumer hardware, real broadband, real SIM cards, real deepfaked faces, real money."*

There is no signal left that distinguishes "real" from "fake" at the cost the defender can sustain.

The question is not *"how do we defend OAuth better?"* The question is *"what would an identity protocol designed in 2026 — for agents, not humans — look like?"*

3 What an Agent-Native Identity Protocol Requires

Five structural properties that the OAuth model cannot provide:

Property 1 · Per-call cryptographic binding, not per-session bearer tokens.

Each API call should carry a freshness proof derivable only by the legitimate caller. Stealing the request — token, headers, body, IP — should not enable replay by a different machine.

Property 2 · No centralized credential database.

The provider should not store password hashes, OAuth refresh tokens, or any credential material whose breach would compromise customers en masse. Authentication material should live only on the customer's side, with the provider holding nothing that an attacker would want to steal.

Property 3 · No bearer tokens that can be exfiltrated.

The "credential" should not be a string that, once captured by a relay or a logging proxy, grants the bearer access. Credentials should be ephemeral and derivation-bound to the calling machine.

Property 4 · End-to-end encryption between the two endpoints, with no third-party-readable plaintext.

No intermediary — relay, proxy, or logging sink — and not the key-derivation platform should ever hold readable plaintext; they see only ciphertext, and the platform is never in the data path. The prompt is readable only by its intended recipient.

Property 5 · Cryptographic response provenance, verifiable offline by any party at zero cost.

The provider should be able to sign every response such that any recipient (with no account, no API call, no fee) can verify "this output really came from the named provider, unmodified, signed at a specific time." Silent model substitution should be detectable by every customer without network access.

A protocol with all five properties structurally defeats every layer of the documented supply chain — not because it patches each defeat, but because the attack surfaces no longer exist.

This paper describes a three-product implementation of such a protocol.

4 A Three-Product Substitution

The substitution is composed of three protocols, each addressing distinct layers of the OAuth model:

4.1 JustAuthentication

Replaces the entire OAuth signup-and-login layer with a model in which:

- **There is no signup form** on any customer website. The "account" is created once in a mobile authenticator application per real human, tied to their email and a password they set inside the app.
- **The customer website has only a login surface.** A user enters their email, receives a session code, opens the mobile app, types the session code in, receives a 5-digit one-time access code, types that into the website.
- **The customer server validates** the access code locally against a prefix + suffix + hash stored in shared storage (S3 / file share / Redis). The authentication broker stores no customer database — it brokers the access code delivery via an encrypted mobile-app channel and discards state immediately after use.
- **The broker license** is flat-fee per server. Unlimited user authentications. No per-MAU pricing.

This eliminates layers 1, 3, 4, and 5 of the documented supply chain:

- **Layer 1 (antidetect browsers):** there is no signup form to spoof. The mass-account-creation attack vector that generates fraudulent accounts at $5–10 each has no analog — each "account" is created in the mobile app per real human, with a real email the attacker must control and a password they set in the app. Even if an attacker creates 1,000 mobile app vaults, they obtain nothing they can pool or resell. There is no bearer token issued at signup.
- **Layer 3 (residential proxies):** IP reputation is structurally irrelevant. Access is granted via cryptographic verification of the access code, not via IP-based heuristics. The broker can serve requests over plain HTTP without weakening security because the access code is delivered via encrypted channel and validated by hash.
- **Layer 4 (SMS / SIM banks):** the protocol uses no SMS. The 5-digit access code is delivered to the user's mobile app through an encrypted channel, not via SMS. SIM-bank infrastructure has nothing to attack.
- **Layer 5 (face / KYC bypass):** the face is not the credential. The registered phone with the email vault is. Face authentication, where used, is one factor in a flow that also requires the 5-digit code delivered to the registered device. Deepfake injection of a face without the registered phone obtains nothing.

4.2 Dead Drop Encryption (DDE)

The core cryptographic primitive. Each pair of identities derives a shared key per 15-minute window without either party transmitting key material. The derivation requires the calling machine's binding and the recipient's published identity descriptor. The cryptographic platform is a stateless oracle — it derives keys on demand and stores nothing.

This eliminates layers 2, 6, and 8:

- **Layer 2 (TLS impersonation):** the API call's authenticity is established by the per-window derived shared key, not by TLS-layer fingerprints. An attacker can produce a perfect Chrome ClientHello and still not impersonate the caller — the call lacks the per-window derivation that only the legitimate machine can produce.
- **Layer 6 (relay servers):** a relay forwarding requests from customer A to the API provider cannot impersonate A upstream, because it lacks A's machine binding required for the per-window key derivation. The relay must either terminate the customer's session and originate a new handshake from its own identity (in which case the relay is visible to the provider as the caller, and the provider can rate-limit, audit, or revoke that identity directly), or fail. The thousands-of-fraudulent-accounts model collapses because each account requires a unique machine binding, not a unique browser fingerprint.
- **Layer 8 (log capture for distillation):** the customer's prompt to the provider is encrypted with a per-window key derivable only by the (sender, receiver, time-window) triple. A relay sitting in the middle sees only ciphertext. Storage is useless because keys rotate every 15 minutes and are not retrievable retroactively. The training-data side-channel that funds the relay business closes.

4.3 Signature Service

The Signature Service is in development; its design is described here for completeness of the architecture.

A separate product with free verification for the entire receiving population. Anyone with the verification library can verify offline, with no account, no API call, no PKI chain, no certificate authority.

This eliminates layer 7:

- **Layer 7 (silent model substitution):** if a provider signs each response with the signing service, the customer's SDK verifies the signature locally and rejects responses that fail verification. A relay attempting to substitute Haiku output for Opus output cannot forge the signature (it lacks the provider's signing key and per-window derivation), cannot strip the signature (the SDK rejects unsigned responses), and cannot pass through the original signature (the embedded message hash will not match the substituted body). The 47% capability gap that

CISPA measured across shadow APIs becomes detectable by every customer, without network access, for free.

4.4 Composition

The three products compose cleanly. JustAuthentication handles the human-to-application boundary. Dead Drop Encryption handles the application-to-application boundary. Signature Service handles content provenance.

For an AI provider replacing the OAuth model entirely:

1. JustAuthentication replaces sign-up / sign-in for end users.
2. Dead Drop Encryption replaces bearer-token API calls between customer applications and the provider's API.
3. Signature Service signs every response so customers can verify provenance locally.

The provider's surface area is reduced from "defend every layer against a productized attack supply chain" to "operate the three protocols correctly." There is no customer database to breach (JustAuthentication stores hash prefixes only). There are no bearer tokens to leak (DDE has none). There are no plaintext customer prompts to harvest (DDE end-to-end encrypts). There is no model substitution to detect (Signature catches it).

5 Layer-by-Layer Mapping

ARTICLE LAYER	PROTOCOL COMPONENT	STRUCTURAL ASSESSMENT
1. Antidetect browsers (mass signup)	JustAuthentication	**Structural.** No signup form on the customer website; account creation is per-real-human in the mobile app, with no bearer token issued at signup that could be pooled or resold. Mass account creation has no economic incentive because there is no resellable credential.
2. TLS impersonation	Dead Drop Encryption	**Structural.** API call authenticity is established by per-window cryptographic derivation requiring the caller's machine binding, not by TLS-layer fingerprints.
3. Residential proxies (IP reputation defeat)	JustAuth + DDE	**Structural.** Neither protocol uses IP as a security signal. Authentication is via cryptographic verification, not IP-based heuristics.
4. SIM banks (SMS OTP defeat)	JustAuthentication	**Structural.** Protocol uses no SMS. Access codes delivered via encrypted mobile channel.
5. Face / KYC bypass (deepfake injection)	JustAuthentication	**Structural.** Face is not the credential; the registered phone with email vault is. Deepfake injection without the registered phone obtains nothing.
6. Relay servers (account pooling)	Dead Drop Encryption	**Structural.** Per-window derivation requires the caller's machine binding, which a relay cannot possess for the customer it forwards from. Account pooling has no resellable credential to pool.
7. Silent model substitution	Signature Service	**Structural** — conditional on provider adoption. The provider must sign responses; if they do, customers detect substitution locally and for free.
8. Log capture for distillation	Dead Drop Encryption	**Structural.** End-to-end encryption between customer and provider means relays see only ciphertext. Keys rotate every 15 minutes and are not retrievable retroactively.

All eight layers have structural answers. Six are unconditional. Two (Layers 1 and 7) are conditional on the customer-application or provider-application actually adopting the relevant protocol component.

6 Limitations and Scope

This paper is not making the following claims:

> NOT A CLAIM **"Existing OAuth providers can be unilaterally switched to this model."**
> They cannot. Adoption requires the provider's decision to deprecate OAuth integrations and ask customers to install a new SDK and bind a calling machine. The realistic adopter is **new** agent infrastructure being built today, where the choice between OAuth and a structural alternative is a clean greenfield decision.

> NOT A CLAIM **"This defeats all conceivable attacks."**
> The analysis here covers the documented attack surface in the May 2026 article. Novel attacks not yet productized are not addressed. The cryptographic primitive (DDE) reduces to SHAKE-256 secrecy — if SHAKE-256 falls, the underlying derivation falls, and so does every hash-based scheme that depends on it.

> NOT A CLAIM **"This is the only solution."**
> Cross-account graph analysis (the current OAuth-side approach) is a real defense and the protocols described here do not replace it — the two are complementary. A defender using these protocols should still monitor for unusual patterns in cross-account behavior, payment metadata, and infrastructure ranges. The structural substitution raises the cost-of-attack at every layer; cross-account analysis raises the cost-of-attack at the coordination layer. Both are needed.

> NOT A CLAIM **"This is a drop-in replacement for OAuth in existing applications."**
> It is not. Each of the three products requires integration work. JustAuthentication requires installing a Token Server and integrating the access-code validation flow. DDE requires deploying an SDK and adopting the per-window key derivation pattern. The Signature Service requires signing responses on the provider side and verifying on the customer side. These are not "flip a config flag" changes.

> NOT A CLAIM **"Agent identity is a solved problem."**
> This paper describes a cryptographic substrate that addresses the documented attack surface. The ecosystem work — agent registries, identity discovery mechanisms, multi-party approval workflows, audit and compliance tooling — remains in progress across the industry.

> WHAT THIS PAPER DOES CLAIM
>
> The three protocols described **structurally remove the attack surface** described in the documented supply chain. New agent infrastructure built on these protocols today inherits a security model that does not have the failure modes described in §1.

7 How to Evaluate This for New Infrastructure

For security architects, CTOs, and technical buyers responsible for designing new agent infrastructure in 2026, the practical evaluation questions are:

Q1. **Does your provider currently run OAuth or equivalent bearer-token auth?** If yes, every layer of the §1 supply chain applies to you and your customers. The decision is not "should I switch" but "should I architect my new product line on a protocol that has the §1 failure modes built into its foundation."

Q2. **Are you building human-facing or agent-facing infrastructure?** If human-facing, OAuth is acceptable for now (the attacks are real but slow-moving). If agent-facing, OAuth was not designed for your use case and every quarter the attack supply chain matures further.

Q3. **How important is response provenance to your customers?** If your customers care that the output they receive really came from your model — not silently substituted by a cheaper one — you need a response signature scheme. PKI alternatives exist (X.509 chain to a CA) but require certificate management overhead at scale that a flat-fee signing service removes entirely.

Q4. **How important is customer data confidentiality?** If your customers' prompts contain proprietary code, business logic, or research data that would be valuable as training material for competitors, end-to-end encryption between customer and provider is the only protocol described in this paper that structurally prevents your operational infrastructure from being a leak vector.

Q5. What is your migration cost from OAuth? Real and substantial. Plan for 6–18 months of dual-running before you can deprecate the OAuth path. The JustAuthentication migration is typically the most painful (customer-facing UX change); the DDE migration is typically the least painful (API replacement is mechanical).

8 References

- "How Chinese Sell 'Claude' Tokens at 5% Cost While Making Millions" (May 2026 developer-community post documenting the 8-layer supply chain). Primary sources cited in that article: CISPA Helmholtz, Anthropic distillation disclosure, Clio paper / blog, 404 Media OnlyFake investigation, iProov 2025, Group-IB Weaponized AI, WEF Cybercrime Atlas, FBI 911 S5 PSA, Bitsight TRACE, eunomia.dev eBPF dump, Reid Barber's reverse-engineering writeup, Cloudflare residential-proxy bot ML engineering blog.
- NIST SP 800-63-4 (2025); CEN/TS 18099; ISO 25456 — Presentation Attack Detection vs Injection Attack Detection standards.
- RFC 6749 — *The OAuth 2.0 Authorization Framework* (October 2012).

This whitepaper is provided as-is for informational purposes. Claims are scoped to the documented attack surface as of May 2026. Future attack patterns may not be covered by the analysis here. Readers designing security-critical systems are advised to perform independent evaluation against their specific threat models.