Privatera Zero-Trust Security Whitepaper

Multi-Layer Cryptographic Architecture, Fail-Closed Kill Switch, Ephemeral Relay & Anti-MitM Protocol

SECURITY PARADIGM	CRYPTOGRAPHIC SUITE	VERIFICATION (SAS)	FAIL-CLOSED PROTECTION
Zero-Trust / Fail-Closed	**Ed25519 · X25519 · AES-256-GCM**	**Out-of-Band Single-Attempt**	**Confidentiality Over Availability**

1. Executive Summary

Privatera is an accountless, end-to-end encrypted (E2EE) 1:1 communication architecture built on a zero-trust model. Rather than relying on administrative policy alone, Privatera enforces confidentiality and authentication through client-side cryptography in the browser Web Crypto API.

There are no user accounts, central contact directories, phone numbers, or email addresses. Identity is an Ed25519 keypair generated on-device. Its private material is stored in a password-encrypted envelope and imported as a non-extractable runtime handle while unlocked. Text, signalling and Room Call audio are encrypted between the two devices. Direct and Relayed WebRTC calls use DTLS-SRTP and require additional AES-256-GCM media encryption. Unsupported WebRTC call modes are refused; Room Call remains a separate encrypted option.

2. Cryptographic Architecture

Privatera's defense is structured into five cryptographic layers: Layers 1–4 govern device-bound identity, ephemeral key agreement, key derivation, and authenticated message encryption; Layer 5 (§5) governs real-time media streams:

Layer 1 — Device-bound Identity
`Ed25519 (RFC 8032)` keys generated on-device via SubtleCrypto and stored in password-encrypted envelopes. Runtime private-key handles are non-extractable; this is software protection, not hardware backing. Authenticates peer handshakes.

Layer 2 — Ephemeral Key Agreement (PFS)
`X25519 (ECDH)` ephemeral key agreement generates fresh key material for each session, providing forward secrecy for past sessions without relying on static encryption keys.

Layer 3 — Key Derivation
`HKDF-SHA-256 (RFC 5869)` derives independent symmetric keys for session encryption and local storage protection from the agreed key material.

Layer 4 — Authenticated Channel Sealing
`AES-256-GCM` authenticated symmetric payload encryption. Detects ciphertext tampering, unauthorized injection, and replayed frames.

3. Identity Verification & Anti-MitM Protocol: 9-Digit Safety Code

To detect Man-in-the-Middle (MitM) attacks during initial pairing, Privatera produces an out-of-band Short Authentication String (SAS) after the authenticated handshake:

Short Authentication String (9-Digit Safety Code): A verification code derived from both parties' authenticated public keys and session parameters, rendered on both endpoints as a nine-digit code grouped into readable triplets (e.g., `123 456 789`) for out-of-band voice or visual comparison.

The Safety Code is verified out-of-band; a network attacker gets a single attempt per pairing with a success probability of **1 in 10^9 (one in one billion)**. On mismatch, do not proceed.

4. Ephemeral Relay & Network Architecture

Privatera's network architecture separates packet transport from message content access:

- **No Access to Message Content:** The edge relay bridges encrypted frames in volatile memory with no access to message content, no payload inspection, and no traffic logging.
- **No Stored Message Content:** Routing state is released on disconnect, with no stored message content on the relay (encrypted deletion commands are retained for offline peers until execution is acknowledged).
- **Bilateral Deletion:** A deletion request is sent to the peer and honored by compliant clients; it cannot be enforced against modified clients or copies already made.

5. Layer 5 — Voice Modalities & Real-Time Media Security

Direct P2P Voice (WebRTC DTLS-SRTP)
Direct and Relayed WebRTC media require DTLS-SRTP and additional AES-256-GCM frame encryption. These call modes are refused if the required transforms cannot be installed. Call signalling travels through the authenticated encrypted room session.

Encrypted Room Voice (E2EE Relay)
Opus audio frames encoded client-side, sealed with AES-256-GCM over the encrypted session, and routed through the relay with no server access to audio content.

6. Fail-Closed Architecture: Zero-Compromise Cryptographic Kill Switch

⚡ Cryptographic Kill Switch (Fail-Closed Zero-Trust)

Under Privatera's zero-trust model, security never degrades opportunistically, favoring confidentiality over availability. If any cryptographic verification fails — whether an authentication tag failure, invalid signature or protocol violation:

- **Instant Connection Severance:** The affected contact's sessions and call stop on authentication failure. Duplicate or old media packets are dropped without rendering.
- **Best-Effort Key Clearing:** Active session references are dropped and key material is cleared on a best-effort basis.
- **Strict Zero Fallback:** The protocol never falls back to plaintext, unauthenticated, or weakened modes.
- **Confidentiality Over Availability:** Repeated failures halt reconnection rather than risking an untrusted channel.

7. The Ten Foundational Principles

01 **Identity:** Device-bound keys stay on-device; private keys are non-exportable via the API (not hardware-backed).

02 **Invite:** Pairing uses one-time invitation links. No invite secrets or contact directories are stored in a central database.

03 **Handshake:** Ephemeral session secrets derive locally on endpoints via X25519 ECDH + HKDF rather than crossing the wire.

04 **Safety Code:** Verified out-of-band; an attacker gets a single attempt per pairing (1 in 10^9); on mismatch, do not proceed.

05 **Fail-Closed Kill Switch:** Any verification failure severs the connection with no fallback, favoring confidentiality over availability.

06 **The Server:** The relay routes encrypted frames in memory only, with no access to message content, no stored message content, and no application logging.

07 **Delete & Wipe:** A deletion request is sent to the peer and honored by compliant clients; it cannot be enforced against modified clients or copies already made.

08 **Session Lifecycle:** Sessions generate fresh ephemeral keys with regular rekeying to maintain forward secrecy for past sessions.

09 **Web Crypto:** Cryptographic operations execute via the browser Web Crypto API; key material is cleared on a best-effort basis.

10 **Accountless by Design:** No email, phone number or server account is required. A local device password protects stored private keys.

8. Security & Architectural Comparison Matrix

Security Vector	Legacy Cloud Platforms	Standard Encrypted Apps	Privatera Zero-Trust
Key Storage	Cloud HSM / Server Key Escrow	Client Keychain / Cloud Backup	**Device-bound SubtleCrypto (Non-exportable)**
User Identity	Email, Password, Mobile Phone	Mandatory Phone Number / PII	**Accountless Keypair**
Degraded Fallback	Silent Plaintext / HTTP Downgrade	Opportunistic Weak Modes	**Fail-Closed Kill Switch**
Out-of-Band SAS	None / In-Band SMS	Fingerprint Comparison	**Single-Attempt Out-of-Band Verification**
Message Archiving	Permanent Server Database	Cloud Backups (iCloud/G-Drive)	**No Stored Message Content on Relay**
Cipher Suite	TLS In-Transit (Server Decrypts)	Signal Protocol / Proprietary	**Ed25519 · X25519 · HKDF · AES-256-GCM**
Voice Calls	Central Media Server Mixing	WebRTC (Relayed or P2P)	**Direct DTLS-SRTP & Encrypted Relay**
Client Hardening	Third-Party Trackers	App Store Binary Packaging	**Strict CSP, Trusted Types, No Client Trackers**

9. Threat Model & Environmental Boundaries

Privatera protects message and voice content against passive network surveillance, ISP payload inspection, a relay that cannot read message content, and server-side message history subpoenas. As with any internet service, network-level metadata (IP, timing, size) is visible to network infrastructure; Privatera does not log it.

Boundary Note: Because the client code is delivered by the service itself, a compromised delivery path is outside the protection scope. Endpoint cryptography operates within the host device: key material is cleared on a best-effort basis, and Privatera cannot protect against a compromised OS, rootkits, keyloggers, malicious browser extensions, or a modified peer client.

Encryption does not hide traffic timing or packet sizes. Voice payloads are not padded to a fixed length. Key cleanup is best effort; JavaScript cannot guarantee memory erasure.