<ephcall/>

Secure Ephemeral 1:1 Voice

Mission Statement, User Manual, and Security Overview

The AI Machine UG
D-66333 Voelklingen, Germany

February 11, 2026

Legal company name: **The AI Machine UG (haftungsbeschraenkt)**.
Short name used in this document: **The AI Machine UG**.
Application URL: https://ephcall.aimachine.io

Contents

1 Mission Statement

<ephcall/> provides browser-native, peer-to-peer voice calls with strong privacy defaults: pseudonymous handles, authenticated key exchange, ephemeral session state, and no server-side voice content storage.

2 Concise User Manual

2.1 Getting Started

1. Open <ephcall/> at https://ephcall.aimachine.ioin a modern browser.
2. Generate a random handle (recommended) or enter a custom handle.
3. Enter a shared secret and click `Connect + Authenticate` to establish an authenticated session.
4. Start call after authentication completes.

2.2 Security Modes

- **Default (Convenience):** Calls proceed with DTLS-SRTP transport encryption if insertable-stream E2EE is not negotiated.
- **Strict E2EE (Optional):** If enabled, calls are blocked unless insertable-stream E2EE is negotiated.
- **Shared-Secret Strength Enforcement (Optional):** If enabled, minimum secret policy is enforced before connection.

3 Security and Privacy Overview

3.1 Authenticated Session Establishment

Each peer link uses ECDH (P-256), HKDF-SHA-256 over `ECDH || shared_secret`, and HMAC-based challenge/proof/ack key confirmation before a session is marked authenticated.

3.2 Voice Protection Layers

- WebRTC DTLS-SRTP transport encryption.
- Insertable-stream audio E2EE when negotiated and supported by both peers.

When insertable-stream E2EE is not negotiated, the UI indicates `TRANSPORT ONLY` and logs a warning that full app-layer E2EE is unavailable.

3.3 Ephemerality

- No call history storage by the application.
- Session keys and runtime state cleared on hangup/reset/disconnect.
- Minimal session-only browser storage for handle convenience.

3.4 Operational Model

Server-side components are limited to static hosting and signaling/rendezvous. Voice content is not relayed or stored by the application server itself.

4 Recommended Best Practices

- Use strong, unique shared secrets per session; avoid reuse across unrelated calls.
- Exchange peer IDs and shared secrets over a trusted out-of-band channel.
- Prefer random generated handles over meaningful personal identifiers.
- Consider using a reputable VPN to reduce IP-level exposure to signaling/relay infrastructure.
- Verify you are using the official application URL before entering session data.
- Keep browser and operating system updated; endpoint compromise can bypass application protections.
- Use strict E2EE mode when both peers support insertable-stream E2EE and higher assurance is required.

5 Security Notes and Limits

- Endpoint compromise is out of scope (malware/OS compromise can bypass app protections).
- Network metadata (IP, timing, traffic patterns) may still be observable.
- Browser capability differences can affect insertable-stream availability.

6 References

- NIST FIPS 197 (AES): https://csrc.nist.gov/publications/detail/fips/197/final
- NIST SP 800-38D (GCM): https://csrc.nist.gov/publications/detail/sp/800-38d/final
- RFC 5869 (HKDF): https://www.rfc-editor.org/rfc/rfc5869
- Web Crypto API: https://developer.mozilla.org/en-US/docs/Web/API/Web_Crypto_API
- WebRTC: https://webrtc.org/

Document End