SARX Feature Overview

Luke Miller

November 15, 2025

SARX Feature Overview

- **Substrate-Inspired ARX Stream Cipher (SARX):**
  - 256-bit ARX-256 keystream in CTR mode: 32-byte blocks, 8-round column+diagonal ARX core with feedforward, implemented as a stateless keystream generator.
  - Per-vault keying: the encryption key `base_key32` is derived as BLAKE3(password ‖ timestamp_ns), binding each keystream to a specific password + wall-clock timestamp so streams are never reused across vaults.
  - ARX core selected by search: the round structure and constants were chosen via an ARX parameter search tuned for near-ideal avalanche and near-isometry in the bit metric, rather than via any Cantorian or lattice construction.
  - Unicode-aware keying: accepts any UTF-8 password (any language, emoji, symbols); policy enforces 30–100 Unicode codepoints with no normalization or downcasting, so no entropy is silently discarded.
  - Pure ARX keystream: the legacy 2D postmix layer is disabled in v1.1.0; the shipping cipher is an "honest" ARX-256 stream with a well-defined, auditable core.
  - Full AEAD-style vaults: encryption is stream XOR with the ARX keystream, and every vault carries a keyed BLAKE3 MAC over the header (as associated data) and ciphertext; any wrong password or modification is detected before decryption.
  - Parallelizable: the keystream API is byte-seekable via offsets; the CLI slices files into multi-MiB chunks and encrypts/decrypts them in parallel with Rayon, reaching multi-GB/s throughput on desktop CPUs.
  - Cross-platform: Rust core plus C mirror; builds on Linux, macOS, and Windows, and the ARX core is suitable for ARM, x86_64, RISC-V, MIPS, PowerPC, and WebAssembly.
- **Security and Randomness:**
  - Statistical quality: the ARX-256 stream passes NIST SP800-22, Dieharder, and PractRand batteries in long-run testing; output is empirically indistinguishable from random under standard suites.
  - Strict avalanche behavior: single-bit flips in the key or plaintext propagate through the ARX core to flip $\approx 50\%$ of output bits (about 32 of 64 per word) in experiments, matching the ideal avalanche target.
  - Balanced Hamming weight: keystream words exhibit stable Hamming weight distribution over long runs; there is no observable bias drift or positional structure in the bit metric.

- **Password, KDF, and Key Management:**
  - Primary KDF: Argon2id with parameters $t = 3$, $m = 2^{17}\,\text{KiB}$ (128 MiB), $lanes = 1$ derives a variable-length stream key k_{stream} plus a fixed 256-bit MAC key k_{mac} from the UTF-8 password and a 256-bit salt.
  - Thermodynamic hardening (`-thermo`): an optional mode that runs a large ARX-based random walk over a 512 MiB scratch buffer, then folds a BLAKE3 digest of the walk back into the KDF output to raise the per-guess energy and memory cost of brute force.
  - Physically-informed parameters: KDF settings and thermo-hardening are calibrated against Landauer and Bekenstein bounds via dedicated bit-energy and brute-force cost experiments, tying the effective attack cost to a concrete bit-energy floor.
  - Header-bound KDF: each vault header records the salt, Argon2 parameters, and a KDF identifier (Argon2id vs. Argon2id+thermo), so keys are reproducible for decryption but distinct across vaults.
  - Native support for USB "sigilbook" tokens for password storage and retrieval, with atomically scoped export: passwords never hit disk outside the hardware token.
  - CLI and GUI workflows for encrypting, decrypting, verifying, and managing vaults, with enforced 30–100 Unicode codepoint password policy.
- **Performance and Edge/IoT Integration:**
  - Desktop performance: benchmarks of the ARX-256 stream show keystream generation above 1 GB/s and end-to-end encryption in the hundreds of MB/s to multi-GB/s range on commodity multi-core CPUs, depending on thread count and KDF settings.
  - Streaming-friendly: the ARX-256 keystream is seekable via a byte offset and supports long-lived streams without re-keying, ideal for pipe-style encryption and large files.
  - Minimal runtime state: aside from KDF memory, the cipher core keeps only a 256-bit key and counter; memory overhead for encrypt/decrypt is essentially the chunk buffer.
  - Live cam/mic encryption: supports real-time audio/video vaulting on commodity hardware, suitable for privacy-preserving surveillance and recording; mobile and SBC performance is being re-benchmarked against the new core.
- **Usability and Developer Features:**
  - Modern desktop GUI (Rust, `egui`): file manager, drag-and-drop vault creation, batch tools, and sigilbook USB integration.
  - CLI dispatcher with subcommands for encryption, decryption, verification, benchmarking, and optional `-thermo` mode to test hardened KDF settings.
  - Structured, human-readable vault format: versioned headers encoding salt, timestamp, nonce, Argon2 parameters, KDF ID, plus a 32-byte keyed BLAKE3 MAC tag per vault.
  - Modular, extensible core: the ARX keystream, KDF, and MAC layers are cleanly separated, making it easy to embed SARX into new frontends and hardware targets.
  - UX niceties: when the sigilbook USB is present, vaults can be auto-decrypted and re-encrypted on double-click without exposing passwords or derived keys to disk.