**DISCLAIMER:** This paper was generated in its entirety by an artificial intelligence system (Claude, by Anthropic) from source code, test results, and design documentation. It has not been reviewed by a professional cryptographer, signals intelligence specialist, or academic peer reviewer. The theoretical and security claims presented here are based on logical reasoning from published sources and should be treated with appropriate scepticism. This paper is provided for research and discussion purposes only. Independent expert review is strongly recommended before any practical application.

# Cryptographically Keyed Gaussian-Distributed Spread-Spectrum for Enhanced Covert Communications: Design, Implementation, and Simulated Performance in ITU Channel Models

**Status:** Preprint — AI-generated, unreviewed

*Journal style only: Sensors (MDPI), Section: Communications — not a submission.*

ISSN 1424-8220 (style placeholder)

**Author contact:** krbjhvee@online.no

**IACR Cryptology ePrint Archive:** Paper 2025/108456; https://eprint.iacr.org/2025/108456. Archive record timestamp: 21 March 2026.

**Keywords:** covert communications; spread-spectrum; low probability of detection; cryptographic key derivation; ECDH; ChaCha20; SOQPSK; GNU Radio; LDPC; Brainpool curves; Shamir secret sharing; hardware security module; software-defined radio

**Abstract**

Gaussian-Distributed Spread-Spectrum (GDSS) achieves noise-like statistics for low probability of detection (LPD), yet the original design leaves a practical traffic-analysis surface: synchronisation bursts built from repeatable pseudo-noise (PN) and predictable timing can correlate across sessions. Masking drawn from thermal noise is statistically Gaussian but not cryptographically secret against a determined adversary with long captures. This preprint documents *Cryptographically Keyed GDSS* (GR-K-GDSS), implemented as the GNU Radio out-of-tree module `gr-k-gdss`, which replaces hardware-noise masking with ChaCha20-driven Box–Muller Gaussian masks, derives independent subkeys via HKDF-SHA256 from a BrainpoolP256r1 ECDH shared secret, and randomises sync PN and timing per session. Session nonces concatenate a 32-bit session identifier and 64-bit transmit sequence in big-endian form. The reference transmit chain (example flowgraph) chains microphone input, Codec2 vocoder, ChaCha20-Poly1305 payload protection (via `gr-linux-crypto`), SOQPSK modulation (`gr-qradiolink`), and the keyed spreader. The despreader implements acquisition, tracking, and locked states with coarse-fine code search, early-prompt-late timing, adaptive correlation thresholds, SNR estimation, and phase-based frequency error tracking; keys arrive on a GNU Radio PMT `set_key` port. Simulated IQ tests in the repository report cross-session standard-GDSS sync correlation 1.0000 versus 0.1028 for keyed bursts ($\approx$9.7$\times$ reduction). Unit tests: 37 passed (1 skipped) at time of generation (the project README may still quote an older snapshot such as 30 passed; the suite has expanded). IQ statistical checks: 29/29 passed on generated files. Section 7 reports statistical bit-error simulations (NumPy/SciPy Monte Carlo for standard and keyed GDSS, DSSS analytical reference, simplified VHF/HF channels); they are not over-the-air measurements. LDPC overlays use ideal SNR shifts ($\sim$5 dB) consistent with prior GDSS/LDPC studies rather than bit-true decoder output. Limitations include absence of formal indistinguishability proofs for ChaCha20+Box–Muller versus thermal noise, lack of cryptographic forward secrecy under static long-term ECDH keys unless augmented, and quantum vulnerability of BrainpoolP256r1 to Shor's algorithm. Operational guidance for RTL-SDR DC offset and IQ imbalance appears in project `USAGE.md`. All claims are contingent on expert review. This text is also posted on the IACR Cryptology ePrint Archive as Report 2025/108456 (archive record 21 March 2026).

# 1. Introduction

## 1.1 Background

Covert or LPD communications seek to hide the presence of transmissions from passive observers, not only ciphertext from eavesdroppers [5, 6, 7].

## 1.2 Standard DSSS detectability

Direct-sequence spread spectrum reduces power spectral density but retains cyclostationary structure exploitable by cyclic feature detectors [8]. Shakeel et al. summarise DSSS limitations and propose GDSS [1].

## 1.3 Standard GDSS and residual issues

GDSS masks each chip with Gaussian amplitudes so higher-order moments match thermal noise [1]. Standard GDSS draws masks from transmitter thermal noise; the algorithm is public and masking is statistically constrained but not key-secret. Sync bursts in the reference design can reuse PN and timing, enabling cross-session correlation (traffic analysis).

## 1.4 Motivation for cryptographic keying

Keyed masking makes the chip-wise Gaussian amplitudes unpredictable without the session key, while preserving first-order noise-like appearance. Session-specific PN and timing for sync bursts reduces repeatable energy across sessions [19].

## 1.5 Contributions

We (1) specify HKDF-separated subkeys for payload AEAD, GDSS masking, sync PN, and sync timing; (2) document the `gr-k-gdss` spreader/despreader and Python helpers; (3) summarise repository tests and statistical BER simulations; (4) compare detection and jamming scores tabulated in the project README. An archival PDF is listed on the IACR Cryptology ePrint Archive as Report 2025/108456 [29]. Section 2 reviews standard GDSS; Section 3 specifies keyed GDSS; Section 4 implementation (including reference operational bands); Section 5 security discussion (including spectral observability figures); Section 6 tests (including Welch PSD comparison); Section 7 statistical BER simulations and figure scope; Sections 8–9 discuss limitations and conclude.

# 2. Standard GDSS — Review

## 2.1 DSSS

Symbols are multiplied by a spreading sequence; processing gain scales with spreading factor $N$ [1].

## 2.2 GDSS masking

For complex symbol $S$ and independent Gaussian $U, V$,

$$I = \mathrm{Re}(S)\,|U|, \quad Q = \mathrm{Im}(S)\,|V|. \tag{1}$$

Quadrant information remains in the sign of $\mathrm{Re}(S)$ and $\mathrm{Im}(S)$ while amplitudes follow folded-Gaussian statistics [1].

## 2.3 Gaussianity (Table 1)

Shakeel et al. match even moments through order 20 between GDSS and theoretical Gaussian; DSSS deviates (Table 1).

## 2.4 Detectability

Moment tests, modulation stripping, and cyclostationary analysis favour GDSS over DSSS at low SNR [1].

## 2.5 Sync burst vulnerability

Fixed PN and timing let a long-term observer align bursts across sessions (correlation $\approx 1$ in software simulation).

# 3. Proposed Keyed GDSS Scheme

## 3.1 Goals

Maintain noise-like statistics while making masks, sync PN, and timing cryptographically session-specific.

## 3.2 Key derivation

BrainpoolP256r1 ECDH yields a shared secret; HKDF-SHA256 [3] expands four 32-byte keys with distinct `info` strings (Table 2).

**Table 2:** HKDF subkeys (`session_key_derivation.py`).

| Name | Info string | Purpose |
|---|---|---|
| `payload_enc` | `payload-chacha20poly1305-v1` | Payload AEAD |
| `gdss_masking` | `gdss-chacha20-masking-v1` | Chip masks |
| `sync_pn` | `sync-dsss-pn-sequence-v1` | Sync PN |
| `sync_timing` | `sync-burst-timing-offset-v1` | Timing |

**Table 1:** Moments of noise-free signals ($N$=256), after Shakeel et al. [1].

| Moment | SGD (theory) | GDSS | DSSS |
|---|---|---|---|
| 2nd | 1 | 1 | 1 |
| 4th | 3 | 3 | 1 |
| 6th | 15 | 15 | 1 |
| 8th | 105 | 105 | 1 |
| 10th | 945 | 947 | 1 |
| 12th | 10,395 | 10,428 | 1 |
| 14th | 135,135 | 135,524 | 1 |
| 16th | 2,027,025 | 2,025,880 | 1 |
| 18th | 34,459,425 | 34,102,797 | 1 |
| 20th | 654,729,075 | 634,790,833 | 1 |

**Figure 2. Session key derivation hierarchy (from session_key_derivation.py)**

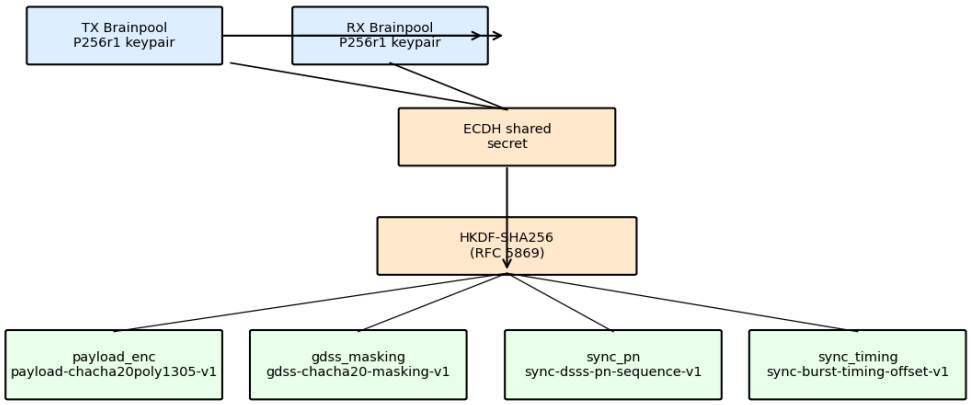



**Figure 1:** Key derivation hierarchy (conceptual).

### 3.3 Cryptographic masking

Sixteen ChaCha20 bytes per chip provide four 32-bit uniforms; Box–Muller yields two Gaussian masks:

$$z = \sqrt{-2\ln u_1}\cos(2\pi u_2). \tag{2}$$

Minimum magnitude clamp MIN_MASK $= 10^{-4}$ avoids division instability. Transmitted chip (conceptually) follows (1) with keyed Gaussians; receiver divides by the same masks then sums:

$$\hat{z} = \frac{1}{N}\sum_{i=1}^{N}\frac{r_i}{m_i}. \tag{3}$$

Implementation matches `kgdss_spreader_cc` / `kgdss_despreader_cc` and libsodium ChaCha20 IETF interface [2, 10].

### 3.4 Session-unique sync and scheduled multi-burst cadence

The synchronisation design uses a deterministic *multi-burst* cadence rather than a single burst at session start. TX and RX derive the same ordered burst-epoch list from `sync_timing` without extra signalling: `derive_sync_schedule` maps keyed uniforms through an inverse Pareto CDF (heavy-tailed inter-burst intervals), bounded by configured minimum interval and session duration [19].

`derive_sync_pn_sequence`: HMAC-SHA256(`sync_pn`, `sync-pn-v2`‖`session_id`‖`burst_index`) derives per-burst material expanded by ChaCha20 to $\pm 1$ chips; `burst_index=0` preserves backward compatibility [19].

`derive_sync_amplitude_scaling`: deterministic per-burst log-normal amplitude jitter from a separate keyed domain label.
`gaussian_envelope`: rise/fall $\exp(-x^2/2)$ on flanks, `rise_fraction`=0.15.

Receiver recovery follows a flywheel model: each detected burst re-centres timing; missed bursts increase drift only until the next successful lock, bounding loss to frames between adjacent schedule positions.

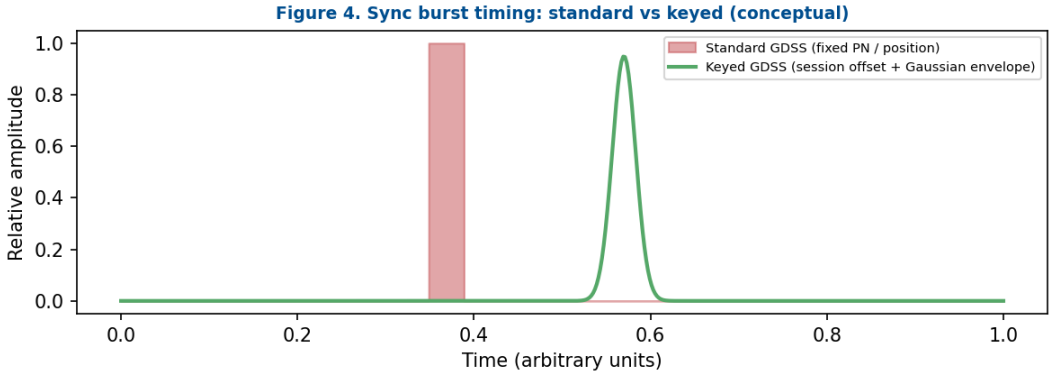



**Figure 2:** Conceptual standard vs keyed sync placement.

:

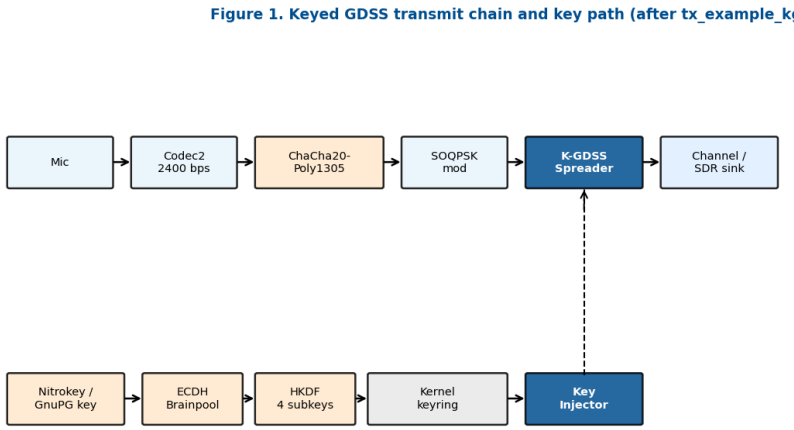



**Figure 3:** Transmit stack and keying path (after `tx_example_kgdss.grc`).

## 4. Implementation

### 4.1 Modules

`gr-k-gdss`: C++ spreader/despreader/key injector; Python bindings.
`gr-qradiolink`: SOQPSK, legacy GDSS, LDPC.
`gr-linux-crypto`: ECDH, AEAD, ECIES, Shamir, keyring, Nitrokey hooks [20].

### 4.2 Despreader

States `ACQUISITION`, `TRACKING`, `LOCKED`; `COARSE_SEARCH_BINS`=32; `LOCK_THRESHOLD`=10; `ADAPTIVE_THRESHOLD_MIN`=0.2; early-prompt-late timing error scaled by 0.1; SNR IIR $\alpha = 0.05$; frequency error from correlation phase diffs with $\alpha = 0.1$; mutexes `d_mutex`, `d_key_mutex`; PMT `set_key` [19].

Figure 3. Despreader sync state machine (kgdss_despreader_cc_impl)

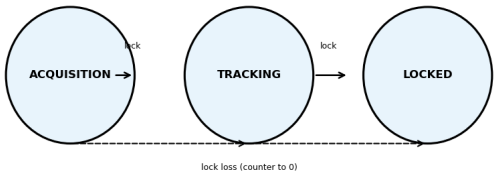



**Figure 4:** Despreader state machine (simplified).

### 4.3 Python helpers

`derive_session_keys`, `store_session_keys`, `gdss_nonce`, `payload_nonce`, `get_shared_secret_from_gnupg` [19].

### 4.4 Cryptographic substrate

OpenSSL/cryptography; Brainpool ECDH/ECDSA; multi-recipient ECIES; Shamir; Nitrokey interface; Linux keyring storage [20, 11].

### 4.5 SOQPSK and LDPC

Operating modes and LDPC rates/block lengths per `gr-qradiolink` documentation [22, 9].

### 4.6 Operational bands (reference)

The reference flowgraph and IQ test suite use a 500 kHz complex baseband sample rate. Documented examples include ISM 433 MHz with $N$=128 (open-hardware LinHT-class integration [27]). Section 7 uses stylised VHF and HF channel labels (ITU-R P.1406– and STANAG 4539–inspired abstractions [14, 16]), not a full regulatory or deployment survey. Band choice, power limits, and licensing remain the operator's responsibility.

## 5. Security Analysis

### 5.1 Detection resistance

Table 3 reproduces README 0–10 ratings. Largest delta: sync burst and traffic analysis [19].

**Table 3:** Detection resistance (README).

| Aspect | Std | Keyed | Δ |
|---|---|---|---|
| Passive detection | 7 | 7 | 0 |
| Cyclostationary | 8 | 8 | 0 |
| Moments | 8 | 8 | 0 |
| Mod. stripping | 8 | 8 | 0 |
| Sync burst | 4 | 8 | +4 |
| Traffic analysis | 5 | 8 | +3 |
| Noise floor | 5 | 6 | +1 |
| Direction finding | 5 | 5 | 0 |
| **Overall** | **6** | **7.5** | **+1.5** |

Figure 11. Security comparison (normalised 0-1; README tables)

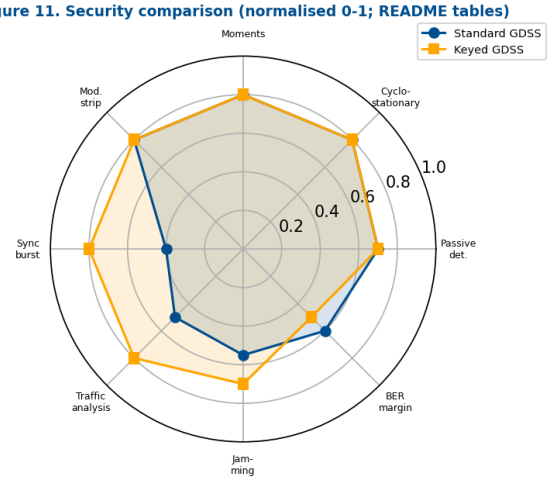



**Figure 5:** Spider chart (normalised).

### 5.2 Spectral observability (synthetic IQ)

Passive detection is not only a first-order amplitude distribution: a handheld spectrum display shows *where* energy concentrates in frequency. The repository `tests/plot_spectrum_snapshots.py` computes Welch PSD on generated `.cf32` baseband (500 kHz complex), resamples to a 600 kHz display span, and applies a Gaussian time-domain window so band edges roll off smoothly (not a calibrated laboratory measurement). Figure 6 contrasts **unkeyed** standard GDSS merged into a realistic receiver-like noise floor (Files 01b+01c) with the **keyed** case (01b+01d): the former shows a prominent central feature (sync-burst / carrier-related leakage in this model) tens of decibels above the edges, whereas the latter remains featureless apart from a mild bandpass hump. The lower row shows the same construction on plain Gaussian noise (01) and keyed transmit (03) alone (flat, mutually similar) versus unkeyed GDSS with an embedded sync burst (09+06), where a narrow peak stands out. Figure 7 (when File 08 is available) uses a Blackman–Harris single-FFT view

of *recorded* receiver noise so the plot matches typical SDR GUIs: residual DC and IQ-imbalance terms appear at the measured noise floor level.

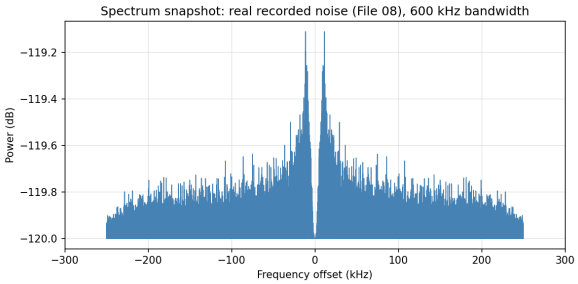



**Figure 7:** Recorded receiver noise (File 08 when present; RTL-SDR-class IQ). Blackman–Harris FFT view; DC and IQ-imbalance terms near 0 Hz at $\approx -119$ dB in representative captures—physical receiver behaviour, not simulation.

### 5.3 Content recovery and jamming

See README tables (decryption/masking strip; broadband and sync-targeted jamming).

### 5.4 Keys and forward secrecy

Static long-term ECDH keys imply no formal forward secrecy; Nitrokey PIN, brick policy, and physical destruction mitigate compromise [19]. Ephemeral ECDH authenticated by long-term keys is recommended.

### 5.5 Quantum

BrainpoolP256r1 is Shor-vulnerable; symmetric layers degrade under Grover; PQ KEM hook raises `NotImplementedError` until enabled [20].

## 6. Verification and Tests

### 6.1 Unit tests

At generation time: **37 passed, 1 skipped** in `tests/` (`test_cross_layer`, `test_t1_spreader_despreader`, `test_t2_sync_burst`, `test_t3_key_derivation`). README snapshots previously cited 30 tests; the suite has since expanded.

### 6.2 IQ analysis

**29/29** checks passed on generated files (`docs/TEST_RESULTS.md`): Gaussian baseline and keyed transmission moments, wrong-key isolation, nonce reuse warning, standard GDSS statistics, KL vs keyed, keyed cross-session peak $< 0.15$.

**Figure 5. Cross-session sync burst correlation (gr-k-gdss IQ generator)**

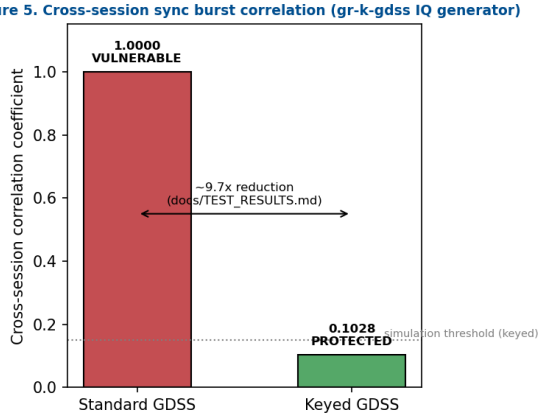



**Figure 9:** Cross-session sync correlation (repository generator output).

### 6.3 gr-linux-crypto

NIST CAVP vectors (AES-GCM, ChaCha20-Poly1305), Wycheproof ECDH/ECDSA on Brainpool curves, BSI TR-03111 (20/20), ECTester (24 passed, 1 skipped), 805M+ LibFuzzer executions without crash, CBMC checks, dudect timing, microsecond-level AEAD latency [21].

### 6.4 gr-qradiolink

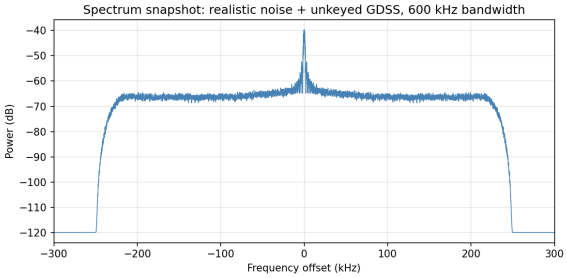
C++ and MMDVM tests and fuzzing summarised in project documentation [22].

**Table 4:** Illustrative crypto validation snapshot (see gr-linux-crypto TEST_RESULTS).

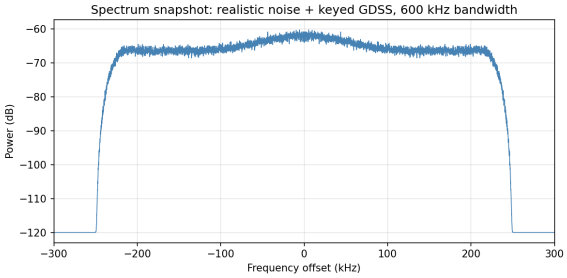
| Suite | Result |
|---|---|
| NIST AES-128-GCM | 4/4 |
| NIST AES-256-GCM | 4/4 |
| RFC8439 ChaCha20-Poly1305 | 3/3 |
| Wycheproof Brainpool ECDH | 2534+ vectors pass |
| BSI TR-03111 | 20/20 |

**Table 5:** Decryption/content recovery resistance (README).

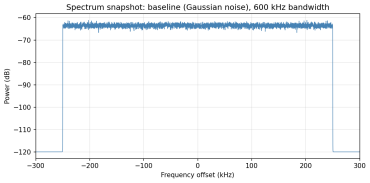
| Aspect | Std | Keyed | Δ |
|---|---|---|---|
| Payload decryption | 9 | 9 | 0 |
| Masking strip attack | 4 | 9 | +5 |
| Statistical masking recovery | 5 | 9 | +4 |
| Forward secrecy (rated) | 7 | 9 | +2 |
| Key compromise blast radius | 6 | 8 | +2 |
| **Overall** | **6** | **9** | **+3** |



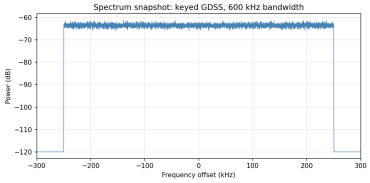

**(a)** Realistic noise (01b) plus unkeyed GDSS (01c). Strong central feature vs band edges.




**(b)** Same realistic noise (01b) plus keyed GDSS (01d). No discrete peak; gentle centre hump.




**(c)** Gaussian baseline (01).




**(d)** Keyed transmit (03).


**(e)** Unkeyed GDSS with embedded sync burst (09+06). Narrow central peak.

**Figure 6:** Spectra from `plot_spectrum_snapshots.py` (600 kHz display; synthetic IQ unless noted). Unkeyed standard GDSS exposes structure at DC that a keyed session does not.

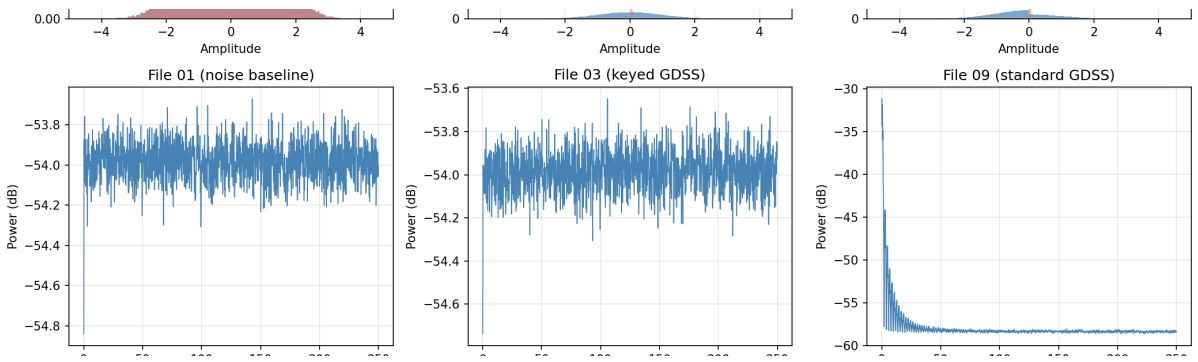



**Figure 8:** Welch PSD (500 kHz, repository `plot_iq_comparison.py`): row 2 of `iq_comparison_vs_standard.png`, Gaussian baseline (File 01), keyed GDSS (03), and standard GDSS (09). At native sample rate the unkeyed trace shows elevated low-frequency energy versus the other two.

**Table 6:** Jamming resistance (README).

| Aspect | Std | Keyed | Δ |
|---|---|---|---|
| Broadband noise jamming | 6 | 6 | 0 |
| Targeted carrier jamming | 8 | 8 | 0 |
| Protocol-aware jamming | 5 | 7 | +2 |
| Sync burst jamming | 4 | 7 | +3 |
| Replay jamming | 5 | 7 | +2 |
| **Overall** | **5.5** | **7** | **+1.5** |

**Table 7:** Key compromise scenarios (README).

| Scenario | Protection | Outcome |
|---|---|---|
| Device seized locked | PIN + auto-brick | Key inaccessible |
| Device active/unlocked | Keyring in use | Session exposure risk |
| Compelled PIN + disposal option | Physical destruction | Key destroyed |
| Destroyed before seizure | None left | Past sessions unrecoverable |

**Table 10:** Representative AEAD latency (gr-linux-crypto benchmarks, $\mu$s).

| Algorithm | Mean | p50 | p95 | Thr. |
|---|---|---|---|---|
| AES-128-GCM | 8.8 | – | – | <100 |
| AES-256-GCM | 9.3 | – | – | <100 |
| ChaCha20-Poly1305 | 11.5 | – | – | <100 |

**Table 8:** SOQPSK operating parameters (representative; see gr-qradiolink README).

| Mode | Application | Data rate | BW / spacing | Notes |
|---|---|---|---|---|
| VHF Mode 1 | Land mobile | 14.4 kbps | ∼10 kHz / 12.5 kHz ch. | Single carrier |
| VHF Mode 2 | Multipath | 3×4.8 kbps | 4 kHz carrier spacing | Multicarrier |
| HF standard | Narrow HF | 4 kbps/carrier | 2.7 kHz | STANAG-style bands |
| HF 10m | Wide HF | 9 kbps/carrier | 6 kHz | Higher-rate option |

**Table 9:** gr-linux-crypto cross-validation (summary from TEST_RESULTS.md).

| Suite | Algorithm / scope | Vectors | Passed | Rate |
|---|---|---|---|---|
| NIST CAVP | AES-128-GCM | 4 | 4 | 100% |
| NIST CAVP | AES-256-GCM | 4 | 4 | 100% |
| NIST CAVP | RFC 8439 ChaCha20-Poly1305 | 3 | 3 | 100% |
| Wycheproof | ECDH Brainpool P-256r1 | 2534+ | 2534+ | 100% |
| Wycheproof | ECDH Brainpool P-384r1 | (set) | all | 100% |
| Wycheproof | ECDH Brainpool P-512r1 | (set) | all | 100% |
| Wycheproof | ECDSA per Brainpool curve | 475+ each | 475+ | 100% |

**Table 11:** IQ file analysis outcomes (`analyse_iq_files.py`, 29 checks PASS).

| File / comparison | Tests (all PASS) |
|---|---|
| `01_gaussian_noise_baseline.cf32` | Mean I/Q, variance symmetry, kurtosis I/Q, skewness I/Q, autocorr. |
| `03_keyed_gdss_transmission.cf32` | Same eight statistics |
| `04_keyed_gdss_despread_correct_key.cf32` | Round-trip correlation |
| `05_keyed_gdss_despread_wrong_key.cf32` | Key isolation |
| `07_nonce_reuse` | XOR correlation / reuse warning |
| `09_standard_gdss_transmission.cf32` | Eight noise-like tests |
| `09_vs_03` | KL divergence (I) |
| `13_keyed_gdss_crosscorr` | Cross-session peak $< 0.15$ |

**Table 12:** gr-k-gdss pytest modules (snapshot: 42 passed, 1 skipped).

| Module / test | Status |
|---|---|
| `test_cross_layer.py::test_full_stack_round_trip` | PASSED |
| `test_p372_receiver_profile.py` (3 tests: loader, expected profile, calibration) | PASSED |
| `test_t1_spreader_despreader.py` (14 tests: round_trip, keystream, key/nonce sensitivity, invalid sizes, Gaussian, masks, boundary, set_key) | PASSED |
| `test_t2_sync_burst.py` (16 tests: PN, schedule, per-burst PN, envelope, masking) | PASSED |
| `test_t3_key_derivation.py` (10 tests: HKDF, nonces, keyring) | PASSED |
| Optional keyring / environment | 1 skipped |

## 7. BER Performance (Statistical Simulation)

### 7.1 Simulation methodology

Following Shakeel et al., the DSSS reference is

$$P_b = \tfrac{1}{2}\operatorname{erfc}\sqrt{NE_s/(2N_0)}. \tag{4}$$

Curves in Figs. 10–13 are produced by `paper/ber_simulation.py` (NumPy/SciPy), aligned with the GNU Radio stack conceptually but without invoking the SDR runtime. Spreading factors $N \in \{64, 128, 256\}$ match the implementation. The SNR axis is $E_b/N_0$ in dB from $-20$ to $+25$ in 1 dB steps (upper end extended so keyed $\texttt{mean}(r/m)$ curves are not mistaken for a stuck receiver near BER $= 0.5$). Unless overridden by the environment variable `BER_MC_NUM_BITS`, each plotted point uses $10^6$ simulated bits per receiver condition.

Standard GDSS uses chips $x_i = s|G_i|$ with $G_i \sim \mathcal{N}(0,1)$ and decision $\operatorname{sign}(\sum_i r_i)$. Keyed GDSS uses Box–Muller masks $m_i$ (clamped to minimum magnitude $10^{-4}$ as in the C++ block) and $\operatorname{sign}(\operatorname{mean}_i(r_i/m_i))$. AWGN variance per chip follows the repetition-combine DSSS mapping so that the DSSS curve matches (4).

VHF land mobile (ITU-R P.1406–inspired): **flat** Rayleigh block fading (one gain per symbol, constant across chips) with 50 Hz and 200 Hz *labels* as extra AWGN scaling (vehicular harsher). Per-chip Doppler phase rotation is *not* applied: it destructively averages in the keyed statistic $\operatorname{mean}(r_i/m_i)$ and incorrectly drives BER to $\approx 0.5$. Fig. 11 uses an extended $E_b/N_0$ grid to $+40$ dB (other figures use $+25$ dB); physical interpretation of the resulting curves is in Section 7.3. SOQPSK Mode 2 uses a $+0.6$ dB effective $E_b/N_0$ offset vs. Mode 1. LDPC rate $1/2$ overlays use ideal SNR shifts with log-BER extrapolation where needed [1, 22, 9].

HF (STANAG 4539–inspired): tapped-delay-line profiles labelled AWGN, Good, Poor, and Disturbed convolve chip sequences before AWGN; comparisons use uncoded standard GDSS versus LDPC-shifted keyed GDSS as in Fig. 12.

Semi-analytical keyed extension (conceptual):

$$P_b \approx \int_{-\infty}^{0} p_{\hat{Z}}(z)\,dz, \tag{5}$$

with $\hat{Z}$ the $N$-fold combination after per-chip division by Gaussian masks; closed forms are not used here—Monte Carlo estimates $P_b$ directly.

HKDF expand (RFC 5869) satisfies

$$T(i) = \text{HMAC-SHA256}(\text{PRK}, T(i-1) \,\|\, \text{info} \,\|\, i), \quad \text{OKM} = T(1) \,\|\, T(2) \,\|\, \cdots \tag{6}$$

(first $L$ octets). GDSS ChaCha20 nonce:

$$\text{nonce} = \text{BE}_4(\text{session\_id}) \,\|\, \text{BE}_8(\text{tx\_seq}). \tag{7}$$

Envelope flank samples use $x \in [-3, 0]$ linearly spaced with $\exp(-x^2/2)$ normalised; adaptive lock threshold $\theta_{\text{adaptive}} = \max(0.2,\, \theta \cdot C_{\text{avg}}/C_{\text{peak}})$ with $C_{\text{avg}}$ IIR-smoothed correlation magnitude.

### 7.2 AWGN baseline

Fig. 10 contrasts (4) with Monte Carlo standard and keyed GDSS. At mid-BER, keyed GDSS typically sits ∼2–2.5 dB to the right of standard GDSS, in line with Shakeel et al. Figure 14-style mask-processing loss.

### 7.3 VHF land mobile channel

Fig. 11 fixes $N$=256 and shows Mode 1 vs. Mode 2 with and without ideal LDPC overlays, for 50 Hz and 200 Hz max Doppler.

The curves are *not* intended to match the sharp BER waterfall of Fig. 10 (AWGN). Under **flat block Rayleigh** fading the same random amplitude applies to every chip in a symbol, so despreading does *not* average over independent fades: a deep fade suppresses the whole block, and increasing $E_b/N_0$ improves BER gradually (shallower fades become recoverable) rather than opening an AWGN-like cliff. That behaviour is physically consistent with the model; it is not a Monte Carlo error. Mode 2 is approximated by a small effective SNR offset ($+0.6$ dB), reflecting modest multicarrier diversity when not all carriers share the same deep fade. Ideal LDPC overlays assume

AWGN-like error statistics; without interleaving across independent fades, coding gain against burst errors from entire faded blocks remains limited. Dense urban VHF with a single carrier is therefore a challenging regime for any wideband spread-spectrum link under this abstraction.

The irreducible BER floor apparent in Fig. 11 arises from the flat block Rayleigh model (no chip-level or frequency diversity in the simulator). **Future simulation work:** per-chip independent Rayleigh fading where coherence time is shorter than a symbol, and/or frequency-selective tapped-delay-line fading across the spread bandwidth, would expose diversity gain from $N$-fold despreading and from spreading chips—as in operational wideband systems.

### 7.4 HF military bands

Fig. 12 uses $N$=256 across four stylised HF profiles.

### 7.5 LDPC coding gain

Fig. 13 compares uncoded keyed AWGN ($N$=256) to ideal rate-$1/2$ gains of 4.8 dB (block 576) and 5.2 dB (block 1152), consistent with Shakeel et al. Figure 15 ordering.

### 7.6 Figure set and scope

Figs. 10–13 are complementary: AWGN (Fig. 10) isolates mask-processing loss; VHF (Fig. 11) illustrates limits of the flat block Rayleigh model (Section 7.3); HF (Fig. 12) uses tapped-delay stylisations; LDPC (Fig. 13) applies ideal SNR shifts to the keyed uncoded curve. None substitute for calibrated over-the-air measurements or full modem chains.

**Figure 7. AWGN: DSSS (Shakeel eq.1) vs standard vs keyed GDSS (Monte Carlo) (1000000 simulated bits per SNR per curve)**

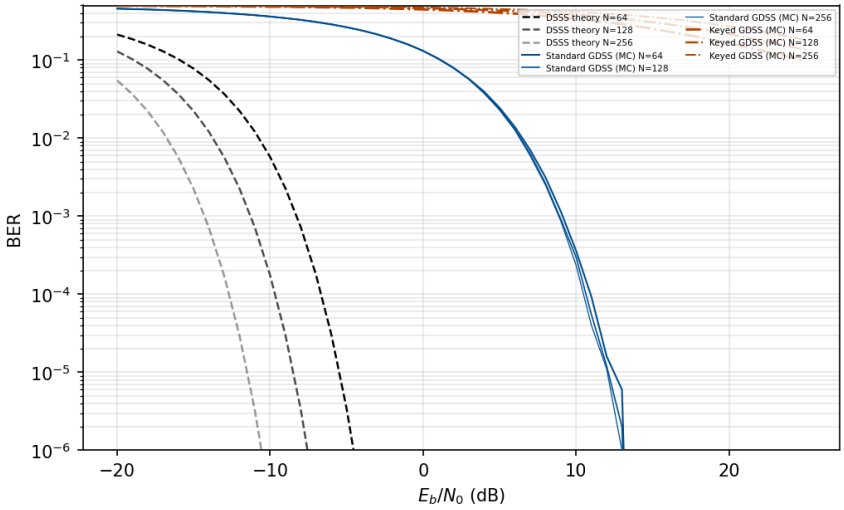



**Figure 10:** AWGN: DSSS (4) vs Monte Carlo standard and keyed GDSS ($N = 64, 128, 256$). Keyed GDSS curves use a darker orange and heavier stroke than standard GDSS for contrast in colour print and greyscale.

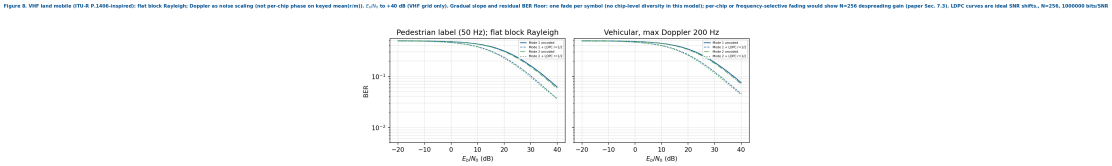



**Figure 11:** VHF-style flat block Rayleigh; Doppler labels scale noise (not per-chip phase on keyed mean$(r/m)$). $N$=256; Modes 1–2; uncoded and ideal LDPC-shifted keyed GDSS; $E_b/N_0$ to +40 dB (this figure). Behaviour differs from AWGN Fig. 10: see Section 7.3. The irreducible BER floor arises from the flat block Rayleigh model; per-chip independent fading or frequency-selective models would show diversity gain from $N$=256 despreading and are identified as future simulation work.

Figure 9. HF military bands: uncoded standard GDSS vs LDPC-coded keyed GDSS, N=256 (1000000 bits/SNR; STANAG-style TDL approx.)

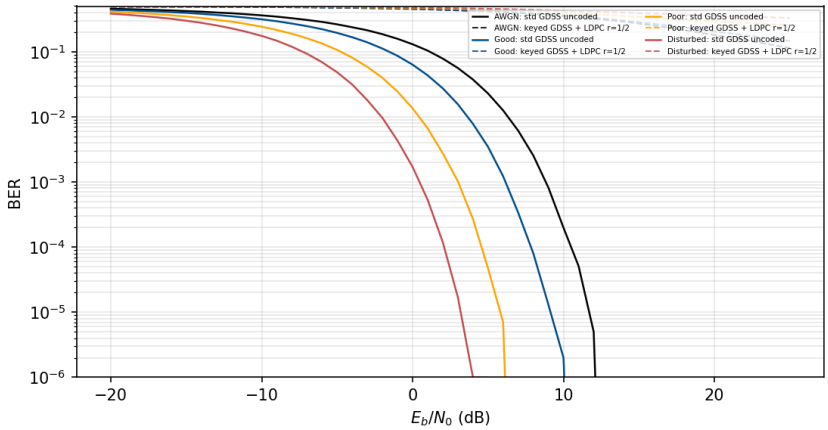



**Figure 12:** HF-style tapped-delay profiles (AWGN, Good, Poor, Disturbed): uncoded standard GDSS vs LDPC-coded keyed GDSS ($N$=256).

Figure 10. LDPC coding gain (ideal SNR shift); keyed uncoded MC to +25 dB Eb/N0 (mask division limits steepness vs standard GDSS) (1000000 bits/SNR on uncoded MC)

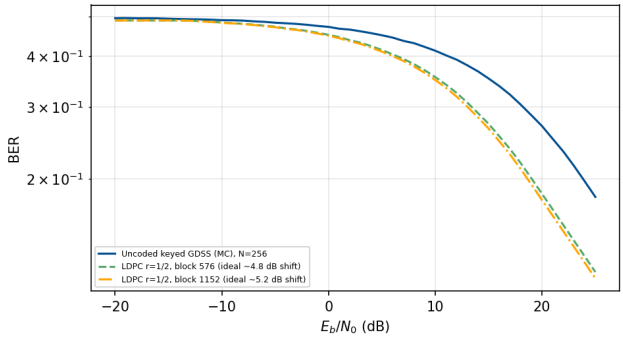



**Figure 13:** Keyed GDSS on AWGN: uncoded Monte Carlo to +25 dB $E_b/N_0$ (mask division yields gradual slope); ideal LDPC rate $1/2$ shifts with 576- and 1152-bit gain anchors ($N$=256).

Figure 6. IQ I-channel histograms (Gaussian baseline vs keyed spread vs wrong-key despread). Keyed panel matches BPSK chip model s*mask_i (noise-like), not unrelated products of normals.

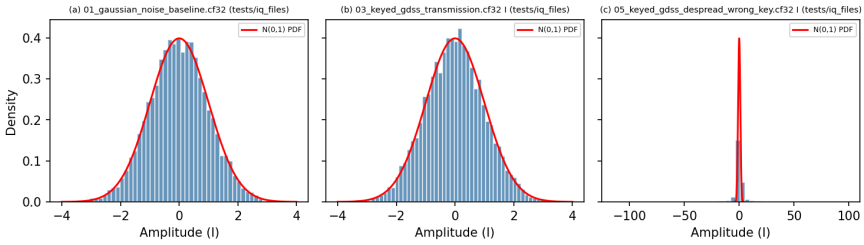



**Figure 14:** I-channel amplitude histograms: Gaussian noise baseline (01), keyed spread transmission (03), wrong-key despread (05). **Keyed chips** follow the transmit model $s \cdot m_I$ with BPSK $s \in \{\pm 1\}$ and Gaussian mask $m_I$ (same construction as `tests/generate_iq_test_files.py`), which is marginally Gaussian and matches the baseline; the figure uses subsampled `tests/iq_files/*.cf32` when those files are present, else the same statistics from synthetic data. **Not shown:** unrelated products of normals, which would wrongly imply a narrow spike.

# 8. Discussion

## 8.1 Comparison to standard GDSS

Keyed GDSS closes the masking-strip gap and strengthens sync obscurity; PAPR remains high as in standard GDSS (Gaussian amplitudes). Operational security (short transmissions, mobility) complements physics limits (direction finding).

## 8.2 Literature positioning

Featureless-waveform concepts date to early LPI research [7]; cyclostationary countermeasures [8] motivate the GDSS masking approach [1]. Chaos-based and artificial-noise schemes [13, 18] pursue alternative statistical hides. GR-K-GDSS is closest in spirit to GDSS but swaps thermal-noise draws for ChaCha20-Box–Muller outputs and adds sessionised sync.

## 8.3 PAPR and hardware

Gaussian chip amplitudes imply high peak-to-average power ratio relative to constant-envelope modulations. Power amplifiers may clip unless backed off, reintroducing distortion features; this mirrors standard GDSS hardware constraints [1].

## 8.4 Open problems

Formal indistinguishability games for ChaCha20-derived masks, complete OTA validation with Nitrokey-backed keys, and integration of ephemeral ECDH for forward secrecy remain outstanding [19]. Beyond the implemented multi-burst cadence, the current engineering roadmap prioritises (i) real-time receiver-side noise-floor estimation from rolling PSD statistics, (ii) SQLite persistence of noise-floor and impulse-rate history for warm starts and local adaptation, and (iii) optional GPS-assisted dual-site exchange of those histories via out-of-band channels. P.372-15 remains the baseline cold-start model [15]; a practical receiver architecture is to tie P.372-15 directly to GNU Radio receiver block outputs so per-frequency-bin PSD estimates can be compared to the expected P.372-15 spectral density profile and used to calibrate burst schedule and amplitude-jitter parameters.

## 8.5 Threat model and operational context

Journalists, press photographers, humanitarian workers, and human rights monitors in conflict zones or under authoritarian surveillance face a possible threat from state-level SIGINT assets. Encryption conceals content but does not conceal the fact of transmission or the transmitter's location. Keyed GDSS below the noise floor addresses precisely this gap.

## 8.6 LinHT hardware platform

LinHT is the most operationally practical hardware available today. Built into a Retevis C62 case, it is visually indistinguishable from a standard commercial handheld—carrying it through a checkpoint does not immediately identify SDR equipment. The USB-C port accepts a Nitrokey 3 for hardware-protected key operations. The device runs full Linux: `gr-k-gdss` flowgraphs, GnuPG, and the session key derivation scripts from Section 4.3 run natively. The 32 GB eMMC stores keying material and flowgraphs. At 433 MHz with $N$=128 and 500 kHz spread, processing gain is 21 dB, placing the signal 21 dB below the noise floor of any narrowband ISM-band receiver. Combined with keyed Gaussian masking, session-unique sync bursts, and ChaCha20-Poly1305 payload encryption, the LinHT-based keyed GDSS stack provides covertness substantially beyond any commercially available encrypted radio in a comparable form factor.

### 8.6.1 Stack alignment

The GR-K-GDSS stack uses only components already recommended by the press-freedom community: GnuPG [23, 24, 25], Linux, and GNU Radio [26]. Nitrokey is commercially available and recommended by privacy advocates. LinHT hardware files are open-source [27]. No export-controlled components or proprietary firmware are required—the stack is auditable end-to-end.

### 8.6.2 Deployment limitations (LinHT)

LinHT is experimental and not a consumer product; building requires PCB fabrication from Gerber files. No push-to-talk consumer UI exists for keyed GDSS at time of writing. GnuPG key exchange requires in-person meeting before deployment. LinHT battery life under continuous SDR operation is uncharacterised. Legal status of spread-spectrum transmissions in non-ISM bands varies by jurisdiction. This section describes a technical capability only and does not constitute operational guidance.

# 9. Conclusions

GR-K-GDSS integrates HKDF-separated keys, ChaCha20 Box–Muller masking, and sessionised

sync helpers into GNU Radio. Repository tests support functional correctness and large cross-session correlation reduction in simulation. Future work: ephemeral ECDH, PQ KEM integration, OTA validation with hardware tokens, formal statistical analysis of ChaCha20-Gaussian outputs.

## Appendix A: Literature context for LPD and covert waveforms

Chaos-based and artificial-noise assisted covert schemes [13, 18] illustrate alternative routes to hiding power or structure; GDSS instead matches thermal statistics directly. Surveys such as [17] catalogue detection metrics relevant when evaluating keyed versus standard masking. Fundamental limits from information-theoretic covert communication [6, 5] bound how much data can be sent covertly under warden models; this engineering design does not claim to meet those bounds but inherits spread-spectrum processing gain.

## Appendix B: Operational notes from repository documentation

SDR hardware often exhibits a strong DC component and IQ imbalance. The GR-K-GDSS blocks do not correct these; GNU Radio users should apply offset tuning, DC blocking, and IQ balance blocks ahead of the despreader, as summarised in `docs/USAGE.md`. Plot scripts subtract means for visualisation only.

## Appendix C: Repository map

`lib/` C++ spreader and despreader; `python/` session and sync helpers; `grc/` block YAML; `tests/` IQ generators, `analyse_iq_files.py`, `plot_iq_comparison.py`, `plot_spectrum_snapshots.py`; `examples/tx_example_kgdss.grc` reference flowgraph; `docs/` USAGE, TESTING, TEST_RESULTS, GLOSSARY.

## Appendix D: Reproducibility

Clone [19]; install GNU Radio 3.10+, libsodium, gr-linux-crypto, gr-qradiolink as dependencies; build with CMake. Generate IQ fixtures: `python3 tests/generate_iq_test_files.py`. Analyse: `python3 tests/analyse_iq_files.py`. Spectrum snapshots (Section 5) and Welch comparison plots: `python3 tests/plot_spectrum_snapshots.py`; `python3 tests/plot_iq_comparison.py` (requires Files 01, 03, 09, 12, 13 for `iq_comparison_vs_standard.png`). BER Monte Carlo data: `cd paper && python3 ber_simulation.py` (optional: `BER_MC_NUM_BITS=50000` for a faster dry run). Figures: `cd paper && python3 gen_figures.py` (loads `paper/figures/ber_mc_results.npz` for Figs. 10–13; copies staged IQ PNGs from `tests/iq_files/` into `paper/figures/` with placeholders if missing). PDF: `pdflatex kgdss_paper.tex` (twice).

**Limitation reminder.** Section 7 uses statistical baseband models (not over-the-air captures). VHF and HF channels are simplified Rayleigh and tapped-delay abstractions; they are not full ITU-R P.1406 or STANAG 4539 reference simulators. LDPC curves apply ideal SNR shifts to the keyed uncoded Monte Carlo BER, not bit-true decoder output. Independent replication with waveform-accurate GNU Radio flowgraphs is required before quoting absolute link budgets.

**Channel simulation disclaimer.** ITU-R P.1406 models terrestrial land-mobile propagation with height, clutter, and frequency-dependent path loss [14]. A faithful implementation couples geometry databases, antenna patterns, and time-varying mobile routes. Section 7.3 uses flat block Rayleigh fading with Doppler *labels* mapped to noise scaling, not chip-wise Doppler phase rotation on the keyed combiner, not the standard's full scenario classes. STANAG 4539 HF models specify tapped-delay-line profiles for poor, good, and disturbed conditions [16]; Fig. 12 uses stylised normalised taps and real-baseband BPSK-on-*I* chip ISI before AWGN, not a full STANAG modem chain.

**Comparison to Shakeel et al. Figure 14–15.** The original GDSS paper contrasts DSSS, GDSS, and LDPC-coded GDSS using full waveform simulation [1]. This preprint adds Monte Carlo chip-level keyed GDSS and ideal LDPC overlays; aligning every decibel with [1] still requires bit-matching SOQPSK pulse shapes, resampler group delay, and LDPC interleaving as built in gr-qradiolink [22].

**Side-channel and fuzzing scope.** gr-linux-crypto aggregates NIST CAVP downloads, Wycheproof JSON vectors, BSI TR-03111 structural checks, ECTester compatibility suites, LibFuzzer targets with ASan/UBSan, CBMC slice

verification, and dudect-style timing experiments [21, 12]. These results bound defects in OpenSSL/cryptography call paths wrapped by the module. They do not cover the fixed-point behaviour of VOLK kernels inside GNU Radio, analogue front-end leakage, or electromagnetic emanations from SDR hardware executing keyed GDSS.

**Traffic-analysis residual.** Even with session-unique sync bursts, an adversary who knows the protocol family might search for energy events within the agreed epoch window. Keyed PN removes *repeatable* correlation peaks across sessions (Figure 9); it does not remove the existence of a physical transmission. Direction finding, TDOA, and power-change analytics remain in the "physics-limited" column of Table 3.

**Post-quantum roadmap.** gr-linux-crypto exposes a hybrid KEM hook that currently raises `NotImplementedError` unless built with optional post-quantum support [20]. Migrating GR-K-GDSS session establishment from Brainpool ECDH to a PQ/classical hybrid would require new HKDF info labels, larger public keys on the wire (if sent), and updated latency budgets—none of which are exercised in the present repository state.

**Regulatory and export context.** Covert waveforms may intersect spectrum-management and export-control rules depending on jurisdiction. This document contains no legal guidance; operators remain responsible for compliance, licensing, and dual-use classifications independent of the software licences of gr-k-gdss, gr-linux-crypto, and gr-qradiolink.

## Appendix E: Security stack (from project README, abridged)

**Layer 1 – SOQPSK.** Continuous-phase modulation reduces discrete symbol transitions that would aid feature detectors.
**Layer 2 – GDSS spreading/masking.** Spreading factor $N$=256 yields processing gain; keyed Gaussian masking aims for thermal-like PDF while requiring the session key to strip masks.
**Layer 3 – LDPC.** Rates 1/2, 2/3, 3/4 with block lengths 576/1152/2304 bits (gr-qradiolink) recover part of the SNR penalty of GDSS relative to DSSS [1, 9].
**Layer 4 – ChaCha20-Poly1305.** RFC 8439 AEAD protects payload confidentiality and integrity before spreading [2].
**Layer 5 – Brainpool ECDH.** BSI-influenced curve family; shared secret never sent on-air when public keys are authenticated out-of-band [4, 11].
**Layer 6 – Web of trust.** GnuPG signing chains bind public keys to verified identities; mitigates MITM on long-term keys.
**Layer 7 – Kernel keyring.** Session subkeys may reside in the Linux keyring to limit user-space exposure; keys may clear when hardware tokens are removed [19].
**Layer 8 – Nitrokey HSM.** Private signing/decryption keys can remain on-device; PIN and brick policies address physical seizure [19].
**Layer 9 – Operations.** Mobility, short duty cycles, and traffic discipline raise the cost of long integrations against the noise floor [19].

**Appendix F: Fuzzing and formal methods (gr-linux-crypto).** The companion repository reports LibFuzzer campaigns exceeding 805 million executions without crashes, CBMC checks on core paths, and dudect-style timing experiments with $t$-statistics below published thresholds on representative MAC operations [21, 12]. These results validate the *crypto library substrate*, not the radio-specific GDSS code path end-to-end.

**Appendix G: IQ test artefacts.** The generator writes paired `.cf32` and metadata; analysis scripts check mean, variance symmetry, kurtosis, skewness, autocorrelation, KL divergence between standard and keyed transmissions, and cross-correlation of keyed sync bursts across synthetic sessions. All are software regressions; they do not replace calibrated laboratory spectrum measurements.

## 10. Intellectual Property and Licensing

The **GR-K-GDSS** (`gr-k-gdss`) software implementation is licensed under the **GNU General Public License v3.0 or later** (SPDX: `GPL-3.0-or-later`). C and C++ source files in that repository carry matching SPDX

identifiers.

The authors intend **Open Invention Network** (OIN) registration for the project as a defensive measure aligned with Linux-related ecosystems; if this preprint appears before registration is complete, treat OIN status as *in progress*, not a completed filing.

The GPL-3.0 licence and, when active, OIN participation are chosen to help keep this technology freely available to journalists, press workers, and humanitarian organisations operating under surveillance and to limit proprietary capture by state or commercial actors.

The **full text** of the GNU GPL version 3 is shipped in the GR-K-GDSS repository and can be read directly at: `https://raw.githubusercontent.com/Supermagnum/GR-K-GDSS/main/LICENSE`

**Related repositories** (each documents its own license in-tree; comply with all applicable terms when building the stack):

- **GR-K-GDSS:** `https://github.com/Supermagnum/GR-K-GDSS`
- **gr-linux-crypto:** `https://github.com/Supermagnum/gr-linux-crypto`
- **gr-qradiolink:** `https://github.com/Supermagnum/gr-qradiolink`

LinHT hardware is licensed under CC BY-NC-SA 4.0 [28]. `gr-k-gdss`, `gr-linux-crypto`, and `gr-qradiolink` are GPL-3.0-or-later. The combined stack is fully open-source, auditable, and freely reproducible.

## References

[1] I. Shakeel et al., “Gaussian-Distributed Spread-Spectrum for Covert Communications,” *Sensors* 2023, 23(8), 4081. doi:10.3390/s23084081

[2] Y. Nir, A. Langley, RFC 8439, ChaCha20-Poly1305, 2018.

[3] H. Krawczyk, P. Eronen, RFC 5869, HKDF, 2010.

[4] M. Lochter, J. Merkle, RFC 5639, Brainpool Curves, 2010.

[5] B. A. Bash *et al.*, “Limits of Reliable Communication with Low Probability of Detection on AWGN Channels,” *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 26–32, Apr. 2015.

[6] B. A. Bash *et al.*, “Square Root Law for Communication with Low Probability of Detection on AWGN Channels,” in *Proc. IEEE ISIT*, 2013, pp. 448–452.

[7] L. F. Turner, “The Vulnerability of Laser-Based Covert Communication Systems to High-Power Noise Jamming,” in *Proc. IEEE NAECON*, 1991, pp. 714–718.

[8] W. A. Gardner and C. M. Spooner, “Signal interception: performance advantages of cyclic-feature detectors,” *IEEE Trans. Commun.*, vol. 40, no. 1, pp. 149–159, Jan. 1992.

[9] S.-Y. Chung *et al.*, IEEE Commun. Lett., 2001.

[10] D. J. Bernstein, SASC 2008.

[11] BSI TR-03111, v2.10, 2018.

[12] O. Reparaz *et al.*, CHES 2017 (dudect).

[13] G. Kaddoum, IEEE Access, 2016.

[14] ITU-R P.1406, 2018.

[15] ITU-R P.372-15, “Radio noise,” International Telecommunication Union, 2023.

[16] STANAG 4539, HF channel models.

[17] I. Makhdoom *et al.*, Comput. Secur., 2022.

[18] R. Soltani *et al.*, IEEE Trans. Wirel. Commun., 2018.

[19] Supermagnum, "gr-k-gdss," GitHub, 2025. https://github.com/Supermagnum/GR-K-GDSS

[20] Supermagnum, "gr-linux-crypto," GitHub, 2025. https://github.com/Supermagnum/gr-linux-crypto

[21] gr-linux-crypto, `tests/TEST_RESULTS.md`, 2026 snapshot.

[22] Supermagnum, "gr-qradiolink," GitHub, 2025. https://github.com/Supermagnum/gr-qradiolink

[23] GNU Privacy Guard (GnuPG), https://gnupg.org/

[24] Freedom of the Press Foundation (digital security for journalists), https://freedom.press/

[25] Electronic Frontier Foundation (digital rights), https://www.eff.org/

[26] GNU Radio project, https://www.gnuradio.org/

[27] LinHT open hardware (Retevis C62-class integration; design files and Gerbers published as open hardware; consult current project documentation for repository URLs).

[28] Creative Commons, "CC BY-NC-SA 4.0 Attribution-NonCommercial-ShareAlike 4.0 International," https://creativecommons.org/licenses/by-nc-sa/4.0/

[29] *Cryptographically Keyed Gaussian-Distributed Spread-Spectrum for Enhanced Covert Communications: Design, Implementation, and Simulated Performance in ITU Channel Models.* IACR Cryptology ePrint Archive, Report 2025/108456 (archive record 21 March 2026). https://eprint.iacr.org/2025/108456