

# Milcodec Project: Technical Methodology

## Abstract

This paper details the architecture and implementation of the Milcodec System, a covert signaling platform designed for field operations. The system utilizes Direct Sequence Spread Spectrum (DSSS) steganography to hide encrypted command data within a -20dB noise floor, effectively bypassing standard intercept methods. The platform consists of a receiver ("Night Watch") and a transmission commander ("Glass Cockpit"), both secured by a simulated Post-Quantum Cryptography (PQC) layer.

## 1. Introduction

Modern battlefield communications face the constant threat of Electronic Warfare (EW) jamming and interception. Traditional encryption is necessary but insufficient if the transmission itself is detected. The Milcodec project aims to solve this by achieving "Low Probability of Intercept" (LPI) properties via steganography.

The system hides low-bandwidth command data (approx. 32 bps) underneath a carrier signal that mimics Gaussian noise, rendering it indistinguishable from background static to casual observers.

## 2. Signal Processing Architecture

The core of the system is the implementation of DSSS (Direct Sequence Spread Spectrum).

### 2.1 Spreading Algorithm

We utilize a 31-bit pseudo-noise code (PNC), specifically a Barker-like M-sequence, to spread the spectrum of the input data. The spreading factor (SF) is 31.

```
PNC_KEY = [1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, ...]
Input Bit (0/1) -> Mapped to +/- 1
Spread Chip Sequence = Input_Scalar * PNC_KEY
```

# Milcodec Project: Technical Methodology

## 2.2 Masking and Modulation

The spread chips are modulated onto a 12kHz carrier wave using simple amplitude modulation (Simulated BPSK). To achieve covertness, the signal is mixed with high-amplitude Gaussian noise.

The mathematical model for the transmitted signal  $S(t)$  is:

$$S_{tx}(t) = 0.1 * S_{mod}(t) + 0.9 * N(t)$$

Where:

- $S_{mod}(t)$  is the modulated data signal
- $N(t)$  is Gaussian White Noise

This results in an effective Signal-to-Noise Ratio (SNR) of approximately -20dB, burying the data below the noise floor.

## 3. Cryptographic Layer

Security is handled by a hybrid "Store-and-Burn" architecture.

- **Key Exchange**: Simulated CRYSTALS-Kyber KEM is used for ephemeral session key establishment.
- **Payload Encryption**: ChaCha20-Poly1305 provides high-speed, authenticated symmetric encryption.
- **Burn Protocol**: A dedicated "Panic" subroutine (Protocol Zero) allows immediate wiping of private keys from RAM.

## 4. Implementation Details

The system is implemented in Python, leveraging `numpy` for vector math and `PyQt6` for the tactical interface. The architecture is split into three standalone modules:

1. **milcodec\_receiver.py**: The "Night Watch" field unit.

## Milcodec Project: Technical Methodology

2. `milcodec_commander.py`: The C2 uplink terminal.
3. `milcodec_masker.py`: The standalone signal masking engine.

### 5. Conclusion

The Milcodec system successfully demonstrates that high-security, low-probability-of-intercept communications can be achieved using standard consumer hardware. The -20dB noise floor masking provides a robust layer of obfuscation, ensuring that field commands can be received securely even in contested electromagnetic environments.