TRANSFEM

1. Introduction

TRANSFEM extends the core principles of the MEME scheme—matrix perturbation and modular arithmetic—to construct an eight-stage encryption pipeline. Each stage contributes diffusion, confusion, or randomness, yielding strong resistance to linear and algebraic attacks.

2. Acronym Definition

Stage	Name	Description
T	Tensorization	Embed plaintext into vectors over Z_m.
R	Randomization	Apply random invertible matrix P and add noise.
A	Affine Transform	Compute A*x + b, where A is invertible.
N	Normalization	Reduce all entries modulo m.
S	Scrambling Field	Mix data with secret matrix S.
F	Extension	Operate in GF(p^k) for algebraic complexity.
E	Encoding (S-Box)	Apply nonlinear bijective substitution.
M	Masking	Combine with pseudorandom keystream; final mod m.

3. Key Generation

3.1 Select a large modulus m (prime or power of two). 3.2 Generate invertible matrices P, A, S in Z_m. 3.3 Choose offset vector b and a noise distribution for randomization. 3.4 Define a bijective S-Box function sigma: Z_m -> Z_m. 3.5 Initialize a cryptographically secure PRNG for masking.

4. Encryption Process

4.1 Tensorization: Flatten plaintext M into vector x.

4.2 Randomization: x1 = P * x + n (noise).

4.3 Affine: x2 = A * x1 + b.

4.4 Normalization: x3 = x2 mod m.

4.5 Scrambling: x4 = S * x3 mod m.

4.6 Field Extension: Interpret x4 in GF(p^k).

4.7 Encoding: x5 = sigma(x4).

4.8 Masking: C = (x5 + keystream) mod m.

5. Decryption Process

5.1 Unmask: x5 = (C - keystream) mod m.

5.2 Decode: x4 = sigma^{-1}(x5).

5.3 Unscramble: x3 = S^{-1} * x4 mod m.

5.4 Project back to Z_m if in GF(p^k).

5.5 Denormalize: x2 = x3 mod m.

5.6 Inverse Affine: x1 = A^{-1} * (x2 - b) mod m.

5.7 Inverse Randomization: x = P^{-1} * (x1 - n).

5.8 De-tensorize: Recover M from x.

6. Security Considerations

- Layered linear operations (P, A, S) ensure complete diffusion.
- Noise injection resists chosen-plaintext attacks.
- Nonlinear S-Box provides strong confusion.
- Field extension complicates algebraic cryptanalysis.
- Fresh masking per message achieves IND-CPA security.