**Chapter 12: Transform Reference**

**🧩 Substitution Layers**

**SubBytesFwdTx**

Applies a simple global S-box substitution to each byte. Fast, consistent diffusion layer.

**SubBytesInvTx**

Inverse of SubBytesFwdTx, restoring the original bytes using the inverse S-box.

**SubBytesXorMaskFwdTx**

Combines S-box substitution with random XOR masking. Adds variability while preserving reversibility.

**SubBytesXorMaskInvTx**

Reverses the combined S-box and XOR masking process.

**MaskedDoubleSubFwdTx**

Performs S-box substitution, then XORs with a random mask, then applies a second S-box. Strong nonlinear mixing.

**MaskedDoubleSubInvTx**

Reverses both substitution layers and removes the applied mask.

**CascadeSub3xFwdTx**

Applies a three-tier substitution: CoinTable → ActiveCBox → Global SBox. Highly layered diffusion.

**CascadeSub3xInvTx**

Reverses the three-layer substitution path in reverse order.

**MaskedCascadeSubFwdFbTx**

Applies mask-driven XOR followed by ActiveCBox and S-box lookup. Extremely nonlinear and resistant to prediction.

**MaskedCascadeSubInvFbTx**

Inverts the cascade by unwinding the masked and substituted layers.

**🔄 Bit, Byte, Nibble Shuffling**

**ShuffleBytesFwdTx**

Randomly rearranges all bytes in the input using a seeded shuffle. Destroys positional correlation.

**ShuffleBytesInvTx**

Restores the original byte order using the same seed-guided shuffle path in reverse.

**ShuffleNibblesFwdTx**

Splits all bytes into upper/lower nibbles and randomly shuffles the resulting nibble pool. Boosts fine-grained diffusion.

**ShuffleNibblesInvTx**

Reverses the nibble shuffle, restoring original byte composition.

**ShuffleBitsFwdTx**

Breaks down all bits and shuffles them across the entire bitspace. Extremely high entropy and diffusion.

**ShuffleBitsInvTx**

Reconstructs the original bit arrangement from the shuffled version.

**NibbleSwapShuffleFwdTx**

Randomly exchanges upper/lower nibbles between bytes. Enhances intra-byte disruption while preserving most magnitude.

**NibbleSwapShuffleInvTx**

Undoes the nibble swaps using a deterministic reverse path from the same PRNG seed.

**MicroBlockShufflerFwdTx**

Divides input into small fixed-size blocks (e.g. 4 bytes) and shuffles bytes within each block. Retains local data groupings but breaks order.

**MicroBlockShufflerInvTx**

Restores the intra-block byte order using the same PRNG sequence.

**MicroBlockSwapFwdTx**

Reorders bytes within each small block using a custom permutation. Good for structured scrambling.

**MicroBlockSwapInvTx**

Reverses the small-block swap pattern, restoring local byte sequences.

**⚡ Bit/Pattern Alteration Transforms**

**BitFlipCascadeTx**

Flips bits in a patterned cascade using both positional math and randomness. Creates dense, semi-predictable noise.

**BitRandFlipTx**

Randomly flips 1–4 bits per byte. Light but chaotic data distortion layer.

**XorTx**

XORs each byte with a PRNG-derived mask. Simple, fast, and fully reversible.

**AdditiveScatterFwdTx**

Adds a pseudo-random value to each byte (mod 256). Distorts values while preserving reversibility.

**AdditiveScatterInvTx**

Subtracts the original mask value (mod 256) to restore original data.

**SlidingMaskOverlayTx**

Applies a sliding XOR mask across the input with controlled drift. Introduces correlated chaos with temporal structure.

**🎭 Mask & S-Box Hybrids**

**ApplyMaskBasedMixingTx**

Applies a derived mask (from S-box values) to only the upper nibble of each byte. Subtle structure manipulator with masked substitution bias.

**MaskBasedSBoxFwdTx**

XORs each byte with an S-box value derived from a PRNG index. Efficient non-linear mixing.

**MaskBasedSBoxInvTx**

Uses the same masking logic; since XOR is self-inverse, this undoes the transform.

**🦋 Butterfly & Multi-Layer Mixing**

**ButterflyTx**

Swaps the upper and lower nibble of every byte. Simple, low-cost symmetry disruptor.

**ButterflyWithPairsFwdTx**

Performs a nibble swap, then XORs each byte with a portion of the previous byte. Builds pairwise dependency and cumulative diffusion.

**ButterflyWithPairsInvTx**

Reverses both the XOR influence and the nibble swap, carefully preserving the pair-linked structure.

**ButterflyWithRotationFwdTx**

Starts with a nibble swap, then applies a directional bit rotation depending on byte position. Delivers layered, high-entropy rearrangement.

**ButterflyWithRotationInvTx**

Undoes the directional rotation and then reverses the nibble swap.

**BitFlipButterflyFwdTx**

Combines nibble swapping with targeted bit flipping on odd-indexed bytes. Adds non-linear interference on top of symmetry disruption.

**BitFlipButterflyInvTx**

Reverses the bit flip and then re-symmetrizes each byte with an inverse nibble swap.

**🔐 AES-Inspired Transforms**

**AesSubBytesFwdTx**

Applies the classic AES S-box to each byte. Strong non-linear substitution and resistance to linear cryptanalysis.

**AesSubBytesInvTx**

Restores the original values using AES’s inverse S-box.

**AesShiftRowsFwdTx**

Treats the data as a 4x4 matrix per 16-byte block and cyclically shifts each row left by its index. Introduces byte misalignment and disrupts column consistency.

**AesShiftRowsInvTx**

Reverses the row shifts to realign the matrix to its original form.

**AesMixColumnsFwdTx**

Applies AES's MixColumns operation per 4-byte column in each block using Galois Field arithmetic. Diffuses data across columns.

**AesMixColumnsInvTx**

Inverts MixColumns using GF(2^8) inverse multipliers. Losslessly restores column structure.

**🧪 Specialized & Miscellaneous**

**PassthroughTx**

A no-op transform. Used for benchmarking, testing, or as a sequence placeholder.

**ChunkedFbTx**

Divides data into fixed-size chunks and applies feedback-style XOR masking with evolving PRNG seed. Strengthens resistance to repeating pattern attacks.

**PatternEqualizerTx**

Analyzes rolling bit patterns, then injects random bit flips to break dominant patterns. Designed to flatten frequency spikes and neutralize repeated byte structures.

**MaskedCascadeSubFwdFbTx**

Applies a masked XOR, followed by a two-tier S-box substitution (input-derived + global). Heavy entropy generator with trace resistance.

**MaskedCascadeSubInvFbTx**

Reverses the cascade by peeling off S-box layers and XORing with the original random mask.

**CascadeSub3xFwdTx**

Performs a 3-layer substitution: coin-derived, input-derived, then global S-box. Each stage increases non-linearity and context dependence.

**CascadeSub3xInvTx**

Sequentially undoes all three substitutions in reverse. Restores the exact original byte.

✅ **All transforms above are reversible**, either inherently (e.g. XOR, shuffle) or via defined inverse functions. The choice and combination of these transforms allow Mango to create cryptographic sequences that are adaptive, highly entropic, and resistant to known plaintext attacks.