**6. Munge Engine**

**6.1 What Is Munge?**

**Munge** is Mango's automated discovery engine for high-performance transform sequences. It systematically explores and evaluates millions of combinations to identify cryptographically strong pipelines tailored to a given InputType.

Munge is responsible for discovering the **god-sequences** that power Mango’s built-in production profiles.

**6.2 Why Munge Still Matters**

While Mango ships with four well-optimized InputProfiles, many industries deal with unique, domain-specific data formats. Running Munge on real-world input allows you to discover **custom sequences** that outperform generic defaults.

Munge is essential for extending Mango’s adaptability and achieving maximum encryption quality on structured, niche, or sensitive data.

**6.3 How Munge Works**

The Munge engine operates in several key phases:

1. **Sequence Generation**  
   All valid transform sequences are constructed up to the configured max length (L1–L5, where L5 = 5 transforms).
2. **Evaluation**  
   Each sequence is scored across Mango's cryptographic metrics: entropy, avalanche, bit variance, key sensitivity, and more.
3. **Filtering**  
   Sequences that fail to meet the minimum pass count are discarded early to reduce evaluation cost.
4. **Contender Ranking**  
   High-performing sequences are ranked and saved to a **Contenders file**.
5. **Snapshotting**  
   Periodic save points allow Munge to resume after interruptions (e.g., power failure, system crash, manual stop).

**6.4 Cutlists & Acceleration**

A full L5 Munge can explore over **90 million sequences** and run for several days.

To accelerate future runs, a **cutlist** is derived from the top 10 contenders. This shrinks the transform pool by excluding underperformers, making follow-up runs significantly faster.

For example:

* Rerunning L5 with a new transform might complete in hours instead of days.
* Poorly performing transforms are auto-added to the cutlist for exclusion in future searches.

**6.5 Batch Processing Munge Jobs**

Mango includes a script (RunBatch.cmd) to automate hands-off execution of Munge discovery runs.

This script performs full L1–L5 exploration across all four InputTypes.

**Notable command-line arguments:**

| **Argument** | **Description** |
| --- | --- |
| -RunCommand run munge / -restore | Starts or resumes a Munge run |
| -ExitJobComplete | Exits automatically after job finishes |
| -maxSequenceLen | Sets Munge level (e.g., L5 = 5) |
| -inputType | One of Combined, Random, Sequence, or Natural |
| -passCount | Required pass threshold (e.g., 6) |
| -mode | Evaluation mode: Cryptographic or Exploratory |
| -quiet | Disables verbose output for performance |
| -createMungeFailDB | Enables failed sequence logging to database |

**6.6 From Munge to InputProfile**

After Munge completes, top-ranked sequences are exported to text files. Typically, the **#1 sequence** is selected and incorporated into Mango’s official InputProfiles.

Optionally, you can run optimize sequence (or btr) to fine-tune the **per-transform rounds (TR)** and **global rounds (GR)**. However, in most cases, Munge alone produces sequences that are already near-optimal.

The result is a **domain-tuned**, high-scoring, reversible InputProfile—ready for use in production or research workflows.

**6.7 Closing Remarks**

Munge is the **discovery engine** behind Mango’s cryptographic strength. While end-users benefit from pre-tuned profiles, researchers and engineers can leverage Munge to unlock new performance and security optimizations.

In essence, Munge enables Mango to evolve with your data—whether you're protecting a sensor stream, medical archive, or custom file format.