RIFT 2026 HACKATHON

Graph Theory / Financial Crime Detection Track

MONEY MULING DETECTION CHALLENGE

Graph-Based Financial Crime Detection Engine

Multi-city Hackathon • Graph Theory Track

PROBLEM OVERVIEW

Money muling is a critical component of financial crime where criminals use networks of individuals ("mules") to transfer and layer illicit funds through multiple accounts. Traditional database queries fail to detect these sophisticated multi-hop networks.

Build a web-based Financial Forensics Engine that processes transaction data and exposes money muling networks through graph analysis and visualization.

⚠ Your solution MUST be deployed as a live web application with CSV file upload functionality.

INPUT SPECIFICATION

Your web application MUST accept CSV file upload with the following exact structure:

Column Name	Data Type	Description
transaction_id	String	Unique transaction identifier
sender_id	String	Account ID of sender (becomes a node)
receiver_id	String	Account ID of receiver (becomes a node)
amount	Float	Transaction amount in currency units
timestamp	DateTime	Format: YYYY-MM-DD HH:MM:SS

REQUIRED OUTPUTS

⚠ CRITICAL: Outputs will be tested line-by-line against expected test case results. Exact format matching is required.

1. Interactive Graph Visualization

- All account nodes (sender_id and receiver_id from CSV)
- Directed edges representing money flow (sender → receiver)
- ALL identified money muling rings clearly highlighted
- Suspicious nodes MUST be visually distinct (different color/size/border)
- Interactive: hovering/clicking nodes shows account details

2. Downloadable JSON Output File

Provide a download button for a JSON file with the following EXACT format:

```
{ "suspicious_accounts": [
    { "account_id": "ACC_00123", "suspicion_score": 87.5,
      "detected_patterns": ["cycle_length_3", "high_velocity"],
      "ring_id": "RING_001" } ],
  "fraud_rings": [
    { "ring_id": "RING_001", "member_accounts": ["ACC_00123", ...],
      "pattern_type": "cycle", "risk_score": 95.3 } ],
  "summary": { "total_accounts_analyzed": 500,
    "suspicious_accounts_flagged": 15, "fraud_rings_detected": 4,
    "processing_time_seconds": 2.3 }
}
```

Mandatory fields in suspicious_accounts array:

- account_id (String)
- suspicion_score (Float, 0–100, sorted descending)
- detected_patterns (Array of strings)
- ring_id (String)

3. Fraud Ring Summary Table

Display a table in the web UI showing each detected ring with:

- Ring ID
- Pattern Type
- Member Count
- Risk Score
- Member Account IDs (comma-separated)

DETECTION PATTERNS — WHAT IS A MONEY MULING RING?

1. Circular Fund Routing (Cycles)

Money flows in a loop through multiple accounts to obscure the origin. Example: A → B → C → A

- Detect cycles of length 3 to 5
- All accounts in a detected cycle should be flagged as part of the same ring

2. Smurfing Patterns (Fan-in / Fan-out)

Many small deposits aggregated into one account, then quickly dispersed to avoid transaction reporting thresholds.

- Fan-in: Multiple accounts send to one aggregator (10+ senders → 1 receiver)
- Fan-out: One account disperses to many receivers (1 sender → 10+ receivers)
- Use temporal analysis: Transactions within a 72-hour window are more suspicious

3. Layered Shell Networks

Money passes through intermediate "shell" accounts with low transaction counts before reaching the final destination.

- Look for chains of 3+ hops where intermediate accounts have only 2–3 total transactions

PERFORMANCE REQUIREMENTS

Metric	Requirement
Processing Time	Upload to results display ≤ 30 seconds (datasets up to 10K transactions)
Precision Target	≥ 70% — minimize false positives
Recall Target	≥ 60% — catch most fraud rings
False Positive Control	MUST NOT flag legitimate high-volume merchants or payroll accounts

⚠ Your solution will be tested against hidden datasets containing both fraud patterns AND legitimate account traps designed to catch naive algorithms.

MANDATORY SUBMISSION REQUIREMENTS

⚠ ALL of the following are MANDATORY for evaluation. Incomplete submissions will be DISQUALIFIED.

#	Requirement	Details
1	Live Deployed Web Application URL	Must be publicly accessible (no authentication). CSV upload on homepage. Must stay live during evaluation. Platforms: Vercel, Netlify, Railway, Render, Heroku, AWS, Azure, GCP.

2	LinkedIn Video Post	2–3 min max. Must tag official RIFT LinkedIn page. Hashtags: #RIFTHackathon #MoneyMulingDetection #FinancialCrime. Post must be public.
3	GitHub Repository	Public repo with complete source code, well-organized folder structure, .gitignore (no node_modules or env files).
4	Comprehensive README.md	Must include: Project title, Live Demo URL, Tech Stack, System Architecture, Algorithm Approach (with complexity analysis), Suspicion Score Methodology, Installation & Setup, Usage Instructions, Known Limitations, Team Members.

SUBMISSION FIELDS

- Problem Statement selected (on RIFT website — 19th Feb, 6–8 PM window)
- GitHub Repository URL
- Hosted / Live Application URL
- Demo video link posted on LinkedIn tagging RIFT's official page

EVALUATION CRITERIA

Criterion	Description
Problem Clarity	Clear understanding of money muling and graph-based detection approach
Solution Accuracy	Correct detection of rings, valid JSON output, line-by-line test case matching
Technical Depth	Graph algorithm quality, cycle/smurfing/shell detection, complexity analysis
Innovation & Thinking	Novel suspicion scoring, temporal analysis, false positive handling
Presentation (Demo Video)	Architecture explanation, algorithm walkthrough, live demo quality
Test Cases	Exact match with expected account IDs, ring identification, JSON format
Documentation	Complete README including suspicion score methodology and known limitations

Good luck! Follow the money.

— RIFT 2026 Organizing Team