Securely Characterising Fraudulent Cycles

1. Introduction

Problem
- $800 billion to $2 trillion dollars laundered annually
- Origin of fund often tied to violence and organised crime

Challenge and Motivation
- Privacy issues: money laundering often involves multiple institutions, collaboration limited due to laws such as GDPR
- Privacy-preserving cycle detection tries to solve this by detecting cycles in financial graphs, but some cycles are benign
- Filtering out non-fraudulent cycles helps preserve privacy

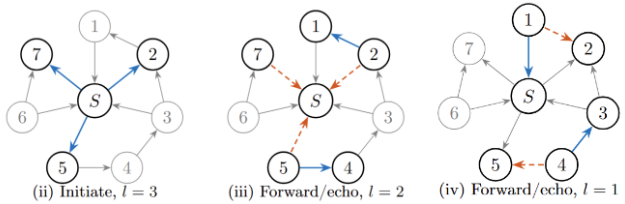
Research Question: *How can we characterise detected cycles in privacy-preserving financial crime detection?*

Contributions
- Protocol to enable evaluation of financial risk before exposing cycle
- Privacy-preserving implementation of protocol
- Theoretical and empirical performance analysis

2. Background

Homomorphic Encryption: Security scheme allowing operations to be performed directly on ciphertexts. We only need addition, so the Paillier scheme is used.

Juno Jense's Cycle Detection: Privacy-preserving algorithm which sends 'propagate' and 'echo' messages to determine if a cycle has been detected. Every step of the algorithm, the propagate routine explores neighbours, and the echo iteratively propagates back to the initiating node for it to determine if a cycle has been detected. The function 'initiate' starts the algorithm by sending a 'propagate' message to neighbouring nodes.



3. Characterising Cycles

Characterising Cycles with Jense's Algorithm
- Use echo routine to make each node in a cycle add risk values
- Risk value is determined by financial experts due to complexity of laundering

Propagate Modification
- Initiator sends 'propagate' message with depth ℓ
- Messages propagated along graph with (ℓ - 1) by each node until ℓ = 0
- Upon receiving 'propagate' message send 'echo' message back with an encrypted risk value of 0

Echo Modification
- When nodes receive an echo they add their locally calculated risk value using function 'risk' with the received one
- This is repeated until the echo reaches the nodes that initiated the protocol
- If the initiator detects a cycle it decrypts the value

End Results
- The final value received by the initiating node is the encrypted value of $\sum_{v_i \in V_c} \text{risk}(v_i)$, where V_c is the set of nodes in the cycle (except the initiating node)
- Value gets decrypted by trusted third party and can be compared against threshold

4. Analysis

Security Analysis
- Secure under honest-but-curious security model
- Paillier prevents adversaries from linking values it receives together (and cycle membership unless explicitly revealed)
- Only final result of aggregation can be read and only by initiating node

Worst Case Complexity Analysis

Function	Space	Time	Communication
initiate	$O(n(\kappa_q + \kappa_r))$	$O(n \log^3 \kappa_p)$	$O(n(\kappa_p + \kappa_r))$
propagate	$O(n^\ell(\kappa_q + \kappa_r))$	$O(n^\ell(n \log^3 \kappa_p + \kappa_h^3))$	$O(n^\ell(\kappa_p + \kappa_r))$
echo	$O(\ell n^\ell(\kappa_r + \kappa_q + \kappa_p))$	$O(\ell n^\ell(\kappa_h^3 + \log^3 \kappa_p))$	$O(\ell n^\ell(\kappa_h + \kappa_r + \kappa_p))$
Overall (simplified)	$O(\ell n^\ell(\kappa_r + \kappa_q + \kappa_p))$	$O(\ell n^\ell(\kappa_h^3 + \log^3 \kappa_p))$	$O(\ell n^\ell(\kappa_h + \kappa_r + \kappa_p))$

Table 1: Worst case complexity analysis of the protocol, where n represents the total number of nodes, ℓ the maximum cycle length, and $\kappa_r, \kappa_q, \kappa_p$, and κ_h are security parameters for the nonces, private key, public key, and Paillier cryptographic scheme respectively.

5. Experiments

- C++ implementation of the protocol run on an Intel Core i7-13700H (14 cores, 20 threads), and 16 GiB of RAM
- Experiment runs parallelised on scale-free graphs, 2048-bit Paillier is used, the other security parameters are toy values
- Results show exponential runtime with regards to average node degree, and significant increase in runtime with addition of Paillier

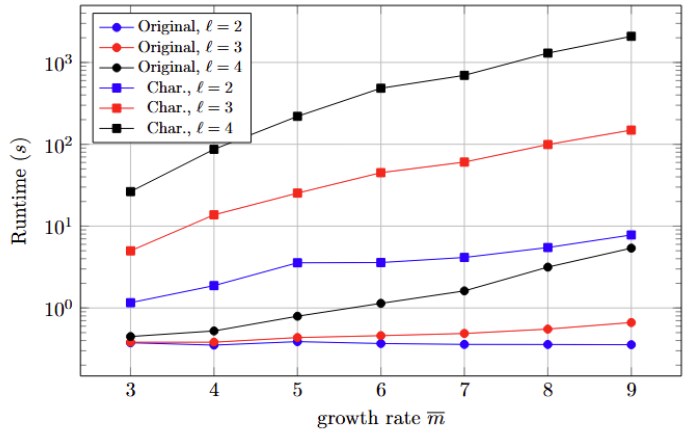


Figure 1: Relationship between the growth rate $\overline{m}$ (average node degree) and the runtime (in seconds), shown on a logarithmic scale. Results are reported for the original implementation (with improvements) and for the characterisation approach with Paillier encryption. The experiment uses $n = 50$ and varies $\ell \in \{2, 3, 4\}$.

6. Conclusions and Future Work

Protocol has the potential to be used for characterisation, and thereby could reduce false positives but comes with a performance trade-offs.

Risk Evaluation: Investigating the optimal risk algorithm for propagation needs to be researched.

Honest-but-curious security model: malicious nodes is a threat to protocol, side-channel attacks need to be taken into consideration.

Runtime Limitation: Exponential runtime with regards to maximum cycle length ℓ.

Rodrigue Vande Capelle
r.h.t.vandecapelle@student.tudelft.nl

Supervisor: Dr. Zekeriya Erkin