Secure computation of fan-in and fan-out degree of nodes using additive homomorphic encryption

Author: Darius-Eduard Floroiu (d.e.floroiu-1@student.tudelft.nl) Responsible Professor & Supervisor: Dr. Zeki Erkin, Dr. Kubilay Atasu

1. Introduction

- Financial crimes are a growing global concern
- Most systems use graph-based models, where accounts are nodes and transactions are edges
- Analyzing patterns in these graphs (such as unusually high income or outgoing connections) can help identify suspicious activity
- Privacy regulations limit the sharing of sensitive financial data between institutions
- Need for techniques that allow collaborative analysis without exposing raw data

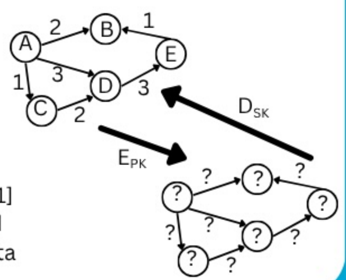
2. Background

- Financial institutions model transactions as graphs to detect patterns of fraud and money laundering [1]
- Important local graph features include fan-in and fan-out: indicators of unusual incoming/outgoing activity [1].
- Due to strict privacy regulations, institutions cannot share raw transaction data
- Homomorphic Encryption (HE) enables computations directly on encrypted data without revealing the content [3]
- There are multiple types of HE schemas

3. Existing solutions

Algorithms on encrypted graphs
- Use FHE - computational intensive!
- No support for basic local features
- Limited to static graphs

Graph-Based Machine Learning [1]
- High computational overhead
- Training requires plaintext data



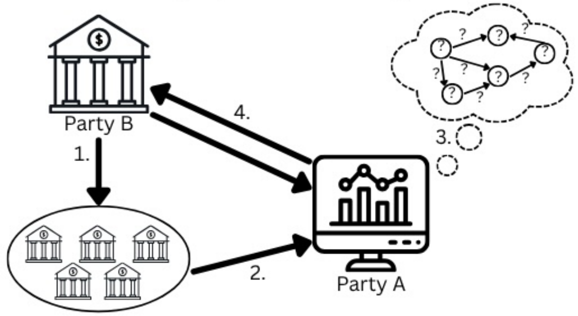
4. Research Question

How can fan-in and fan-out degrees of nodes in financial transaction graphs be computed using additive homomorphic encryption?

5. Our Protocol

- our proposed protocol has 4 phases:

1. Setup: Party B generates it's public-secret key pair and shares the public key to the other banks
2. Data Submission: banks encrypt their transactions (as seen in the table below) and send them to party A
3. Graph Construction: Party A builds the in-memory graph
4. Query Phase: Party B requests data about certain suspicious accounts and Party A provides the encrypted analytics



Parameter	Encryption
Timestamp, Receiving Currency, Payment Currency, Payment Format	n/a
From Bank, From Account, To Bank, To Account	**Deterministic** version of Paillier Cryptosystem
Amount Received, Amount Paid	**Non-Deterministic** version of Paillier Cryptosystem [2]

6. Analyses

- Let *n* - the number of unique accounts, *m* - the number of transactions, *k* - the average number of transactions per account, *l* - the bit length of the encryption key, and *e* - no. of unique edges
- Space complexity: $O(e * l)$ (on average), $O(m * l)$ (worst case)
- Time complexity (depending on the algorithm and phase as seen in the table below, where gc denotes graph construction, ma - multiplication algorithm and as - adjacency structure):

Phase	ma/as	Complexity
gc	Default multiplication (ma)	$O(m * l^2)$
gc	Karatsuba (ma)	$O(m * l^{\log_2 3})$
gc	Toom-Cook (ma) with *y* parts	$O(m * l^{\log_y (2*y - 1)})$
gc	Schönhage-Strassen (ma)	$O(m * l * \log l * \log(\log l))$
gc	Harvey-Hoeven (ma)	$O(m * l * \log l)$
query out	n/a	$O(k * l^2)$
query in	with reverse map (as)	$O(k * l^2)$
query in	no reverse map (as)	$O(n * k + k * l^2)$

7. Discussion & Future Work

- No timestamps - the current version of the protocol ignores the timestamps of the transactions. Thus, old and active accounts might be incorrectly flagged, while quick bursts of suspicious activities might be ignored
- Common currency - party A assumes all transactions have the same currency. Banks can use different ratios to achieve this, which can, although improbable, influence the final results
- Only basic patterns - the algorithm currently only supports basic patterns, which fail to detect more complex forms of fraud
- There is no comparison between our proposed protocol and other existent approaches, such as Centralized Data Collection, SWHE/FHE, Multi-Party Computing or Differential Privacy

8. References

[1] Fabrianne Effendi and Anupam Chattopadhyay. Privacy-preserving graph-based machine learning with fully homomorphic encryption for collaborative anti-money laundering. In Johann Knechtel, Urbi Chatterjee, and Domenic Forte, editors, Security, Privacy, and Applied Cryptography Engineering, pages 80–105, Cham, 2025. Springer Nature Switzerland.

[2] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In International conference on the theory and applications of cryptographic techniques, pages 223–238. Springer, 1999.

[3] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. On data banks and privacy homomorphisms. Foundations of secure computation, 4(11):169–180, 1978.