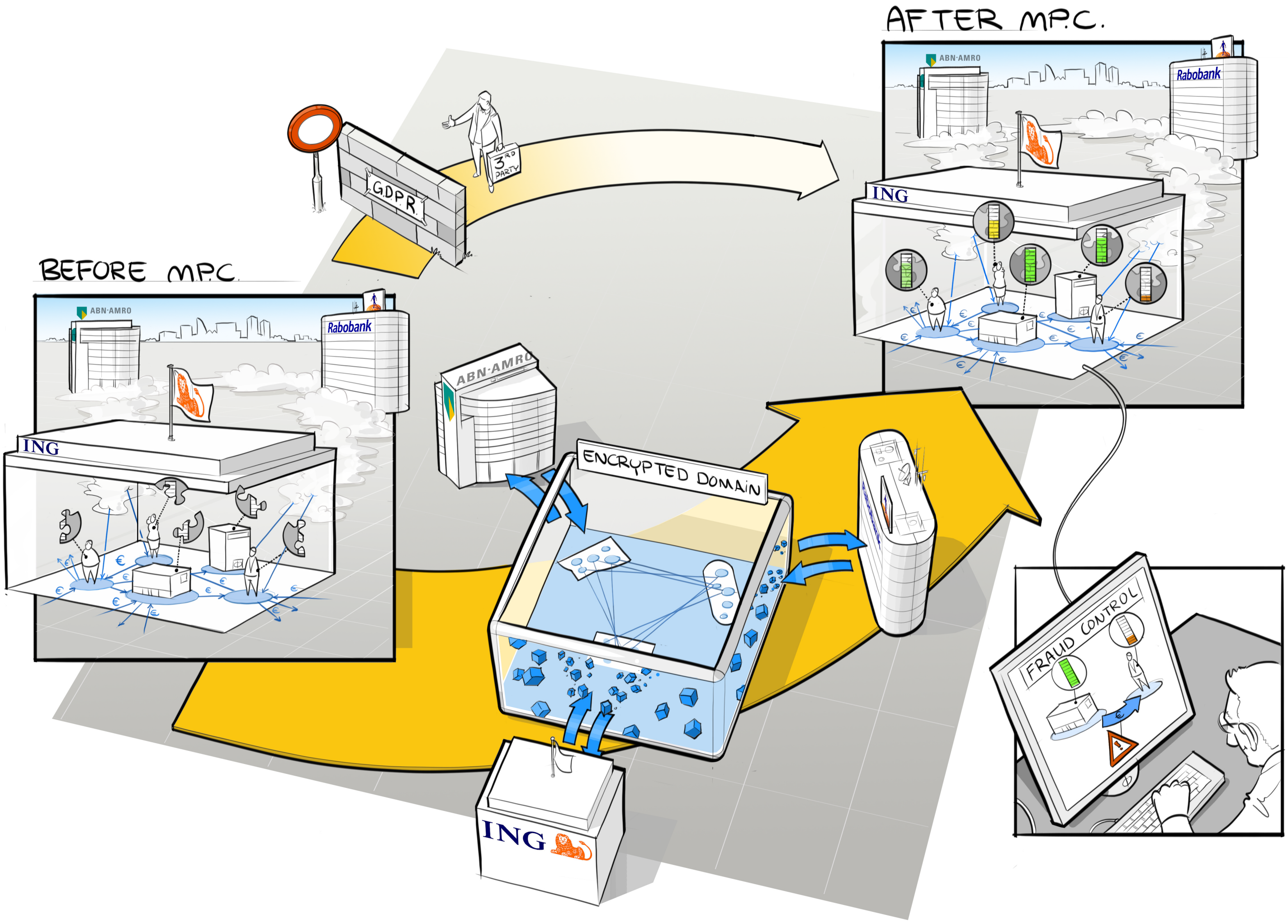
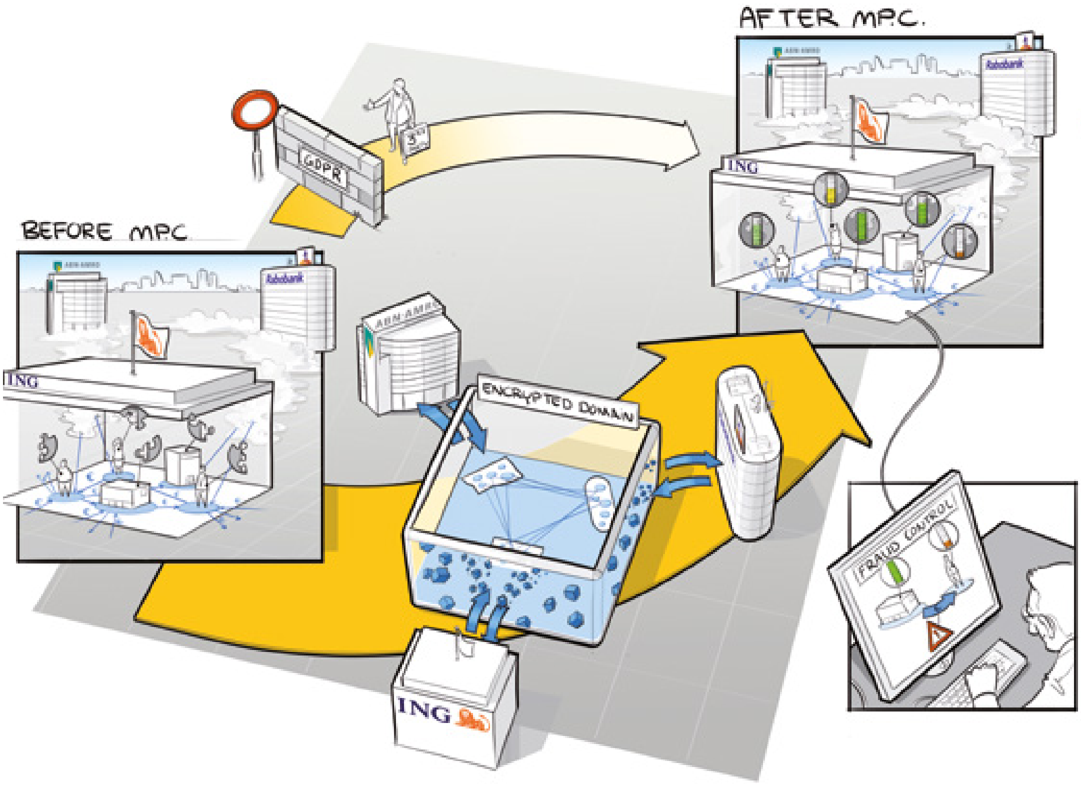
COLLABORATIVE FRAUD DETECTION USING SECURE MULTI-PARTY COMPUTATION



How Google's PageRank inspired us to improve collaboration in fraud detection.[1]

Collaborative fraud detection using secure multiparty computation

Alex Sangers (TNO), Mark Wiggerman (ABN AMRO), Daniël Worm (TNO)



Introduction

Cyber security, anti-fraud and other anti-crime activities highly benefit from collaboration between involved parties like financial institutions, governments and law enforcement agencies. Public and private sectors are stimulated by regulators to perform joint activities and share data, e.g. threat intelligence, as there is a common goal to combat this type of crime. Relevant data to share includes lists of known criminals, confirmed money mules and known malicious IP addresses. Sharing operational data on customers, transactions and events between different organizations would be advantageous as well, but this has always been strictly restricted due to competition and privacy regulations, especially if it concerns personal data of customers and employees. The risks of sharing data for companies as well as public services are loss of trust in services, integrity, financial losses, societal damage, and damaged reputation.

> The risks of sharing data for companies as well as public services are loss of trust in services, integrity, financial losses, societal damage, and damaged reputation.

The financial sector is continuously fighting the misuse of the financial infrastructure for criminal activities like fraud and money laundering. Financial crime detection is a typical example of a

[1] This article is partly based on the more detailed scientific paper [2].

setting where multiple parties share a common interest, but confidentiality and privacy regulations prevent collaboration [1]. In a payment transaction a financial institution typically only knows details if it was involved in the payment. Financial institutions could be much more effective at combatting financial crime if they would be able to access results from analytics based on each other's data, as well as data from other related organizations. However, since such data is often too sensitive to share, there is no straightforward way of accomplishing this.

A possible solution would be to have a single trusted third party that all financial parties are willing to confide their financial secrets to. However, it may be difficult or impossible to find such a party. In addition, it may be very expensive. An important observation is that financial institutions do not need the data itself but only the result of the analytics performed on that data. Therefore we developed an alternative solution, without needing a trusted third party, but still achieving the same security goals. The cryptographic technology that overcomes the described dilemma is Secure Multi-Party Computation (MPC).

Financial institutions do not need the data itself but only the result of the analytics performed on that data.

Secure Multi-Party Computation (MPC)

MPC protocols are cryptographic techniques that allow multiple parties to collaboratively evaluate a function on private input data in such a way that only the output of the function is revealed, i.e. private input remains private. MPC could be explained as the implementation of a trusted third party that collects all relevant input data, evaluates the desired function and only reveals its output.

Already in the 1980s it was shown that any computable function can be evaluated securely, i.e. in an MPC fashion. However, early MPC protocols came at a cost as they introduced significant computation and/or communication overhead, deeming this protocols impractical in many situations.

Over the years progress has been made and research interests have shifted towards practical applicability making MPC ready for deployment.

PageRank for fraud detection

In the SRP, we selected a use case in the area of fraud detection in order to evaluate the possible application of an MPC approach. We focused on financial transaction networks. In mathematical terms a network is called a graph.

A financial transaction consists of a source bank account, a destination bank account, an amount and a timestamp. A set of transactions can be modelled as a graph where nodes (circles) represent bank accounts and links (arrows) represent the transactions between accounts.

It was shown that graph-based features can be used to reduce the false-positives of existing fraud detection techniques [3]. One of these graph-based features is PageRank, originally developed by Google to rank websites in their search engine results. PageRank estimates the popularity of a website, by considering how central a website is in the graph of all websites: the more central, the higher the PageRank value. PageRank can also be computed for transaction networks: For every account number (node) in the network (graph) one can calculate a value which is the PageRank of that account number. Although it is no silver bullet, it has been shown that transactions to accounts with high PageRank are less likely to receive fraudulent transactions. Because of this property, we have chosen to develop an MPC solution for PageRank.

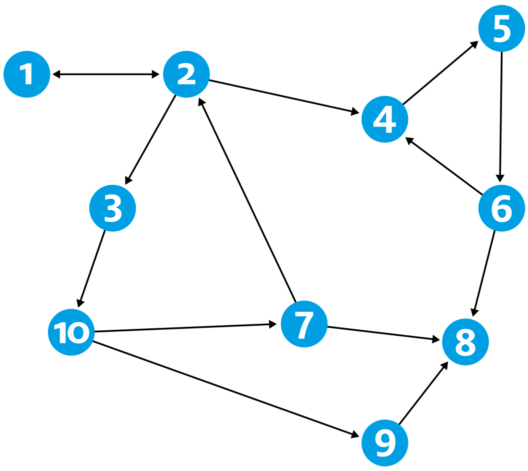


Figure 1: Example of a small transaction graph. Nodes represent bank accounts and links represent transactions.

The idea for detecting fraudulent transactions is to build a transaction graph based on historical transaction data. Based on this graph, the

PageRank value for each bank account is computed. As soon as a new transaction request comes in, this transaction request has to be assessed within milliseconds on whether it is fraudulent. This assessment can be based on various existing detectors, for example using geolocation. The graph-based features such as PageRank from the historical transaction graph can be used to improve this assessment. If the transaction request is assessed to be (likely) fraudulent, it can be declined or delayed for further investigation.

PageRank computation

Inspired by the original idea of Google, the PageRank model for transactions can be seen as money following transactions with some probability p, and jumping randomly to any bank account with probability 1-p. Each time the money ends up in a dead-end bank account, it will randomly jump to any bank account. If the money follows this behaviour infinitely long, then the PageRank value of the bank accounts is the proportion of time spent by the money in the bank accounts. The lower the PageRank value of a bank account, the more likely that such a bank account is fraudulent if it receives large sums of money.

PageRank

Mathematically, the PageRank is the stationary distribution of a Markov chain. An efficient algorithm to compute the PageRank is given by the power method. The PageRank value of node j at the k^{th} iteration of the power method is denoted as x_k^j and the power method is given by the following iterative scheme.

$$X^{j}_{k+1} = \frac{1-p}{n} + p \sum_{i \in s\,(j)} \frac{x^{i}_{k}}{c_{i}}$$

where *p* is a fixed probability, *n* the total number of nodes, c_i the outdegree (number of outgoing links) of node *i*, and *S(j)* is the set of nodes linking to *j*. This formula is linear and consists mostly of additions, which is a nice property for some MPC solutions. Under mild conditions, it can be shown that the power method has a convergence rate of *p*. For *p*=0.85 the power method converges within 50 till 100 iterations independent of the graph size.

Cooperation is required to analyse the combined transaction graph

The PageRank algorithm can be deployed at each individual financial institution to detect fraudulent transactions. However, each financial institution oversees only a part of the global transaction graph. To be precise, a bank only sees the transactions if the source and/or destination bank account is managed by that bank. Figure 2 shows an example of how the transaction graph consists of parts that are visible to each financial institution. The transaction graph can be constructed by combining the transaction data that is available to each individual financial institution.

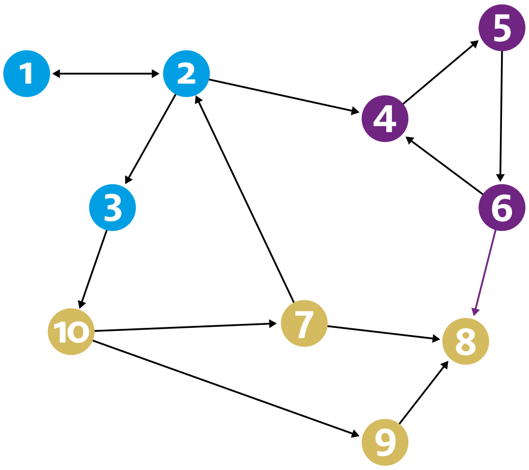


Figure 2: Small example of how three subgraphs (indicated by three colors) can be coupled to the combined transaction graph.

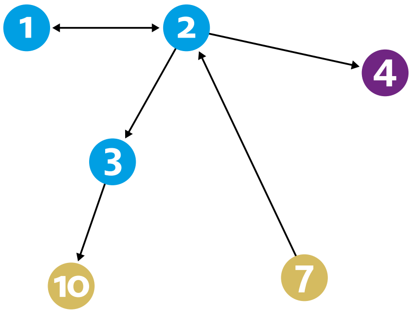


Figure 3: The part of the transaction graph that is visible to the blue financial institution.

The idea for detecting fraudulent transactions is to build a transaction graph based on historical transaction data.

We designed a secure PageRank solution that is able to collaboratively compute PageRank of coupled transaction graphs without leaking information about individual transactions.

During the project we have shown that the PageRank values more accurately represent the true PageRank values (of the whole financial transaction network) if financial institutions would collaboratively compute the PageRank values compared to them doing this separately. This effect is even stronger for financial institutions with a relative small number of bank accounts. Furthermore, the PageRank values of bank accounts with many interbank transactions are significantly more accurate if the PageRank is computed collaboratively.

However, the transaction data is sensitive data and cannot be shared between financial institutions for this purpose. Given Figure 3, assume that the blue bank wants to compute the PageRank values of its nodes. In order to compute the PageRank values of the blue nodes, the blue bank requires information of the nodes that have a link directed to the blue nodes. For example, node 7 contributes to the PageRank of node 2, so the blue party should know the number of outgoing links of node 7 and the intermediate PageRank value of node 7. However, these values are known by the yellow bank but it is a secret to the blue bank. Additionally, the intermediate PageRank values also leak information about transactions. We designed a secure PageRank solution that is able to collaboratively compute PageRank of coupled transaction graphs without leaking information about individual transactions.

Secure PageRank algorithm

Developing an MPC solution for PageRank is non-trivial for several reasons. MPC may introduce a significant overhead. Furthermore, most cryptographic protocols work over finite groups[3], rings or fields and not over real numbers. These challenges can be tackled by developing a specific and efficient MPC protocol. We developed an MPC protocol using additive homomorphic encryption.

In depth: the secure PageRank solution

Some encryption schemes have the property that computations can be performed with ciphertexts. Such schemes can form important building blocks for MPC solutions. This property is called fully homomorphic encryption (HE) if both additions and multiplications can be performed with ciphertexts. However, fully HE performs poorly in practical applications as a general solution. Additively HE is much faster but only allows for additions in the encrypted domain. Additively HE has the following properties:

- Two ciphertexts can be multiplied, with the same result as if adding the decrypted ciphertexts;
- A ciphertext can be exponentiated with a known/plaintext integer, with same result as if multiplying the decrypted ciphertext with the known integer.

Recall the PageRank formula:

$$x^{j}_{k+1} = \frac{1-p}{n} + p \sum_{i \epsilon s\,(j)} \frac{x^{i}_{k}}{c_i}$$

The PageRank algorithm has to be adjusted to efficiently use additive HE. Firstly, the formula should be adapted to work with integers instead of real numbers. This is solved by multiplying the formula with a large value f_x. All values are then rounded. Secondly, the division by c_i in every iteration is too expensive in practice, requiring an approximate integer division. This is solved by multiplying x_k^i each iteration with a large value f_c. This way, the division by c_i can be replaced by a multiplication with f_c/c_i. Thirdly and lastly, the outdegree c_i of node i is a privately known number and cannot be shared between parties. A crucial observation is that all nodes are managed by one of the parties participating in the protocol. Therefore, the number c_i is known to the party that manages node i and this party can execute the multiplication by f_c/c_i. For more details we refer to the scientific conference paper [2].

[3] Groups, rings and fields are mathematical objects often used in cryptography.

Our developed solution achieves computational security in the semi-honest model, i.e. under the assumption that all parties follow the prescribed protocol no party will be able to learn any more information other than the output of the algorithm and any information that follows from that. In our setting, each party will learn the final PageRank values of its own nodes (bank accounts). The solution is implemented using the Paillier Homomorphic Encryption library in Python [4]. The key generation involves a public key, and a partial private key for each party. The private keys ensure that ciphertexts can only be decrypted collaboratively. Generating the keys is a onetime effort and is implemented using a trusted third party. Currently, a distributed key generation algorithm is in development in the Shared Research Program, removing the need for a trusted third party all-together. The number of communication rounds, excluding the key generation, equals the number of PageRank iterations plus 1.

Results

For practical application, it is important to show the accuracy and scalability of the secure PageRank algorithm. The results are based on sampled, anonymized transaction data. Bank accounts, amounts and times are hashed or randomized but in such a way that bank accounts can be coupled in different transactions. Four different datasets are sampled from the transaction data with 100, 1.000, 10.000 and 100.000 bank accounts and the transactions between themselves. The sampled transaction data is divided among three artificial parties, each of whom only see a transaction if the source and/or destination bank account is managed by that artificial party.

Firstly, the accuracy of the secure PageRank algorithm is measured by comparing the results to the outcome of normal PageRank. A maximum relative error below 0.05 is acceptable. As can be seen in Figure 4, the effect of rounding errors in the secure PageRank algorithm is small. Secondly, the computation time increases linearly with the number of nodes in the transaction graph, as shown in Figure 5. This is consistent with the theoretical scalability.

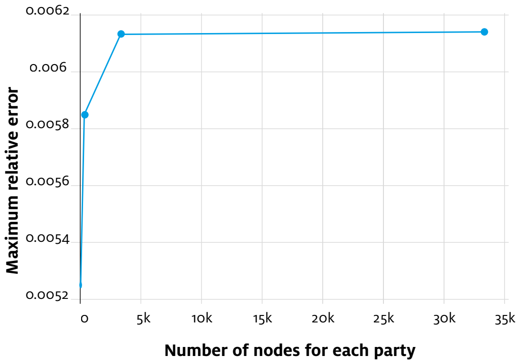


Figure 4: The maximum relative error for increasing sample size of the transaction graph.

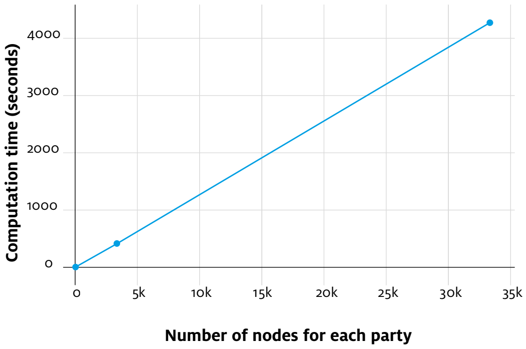


Figure 5: The computation time for increasing sample size of the transaction graph.

The secure PageRank algorithm is also highly parallellizable (for each party). Extrapolation of these results indicate that, for a secure 3-party PageRank computation with 30 million nodes and average outdegree of 80, the current Python implementation would require less than 11 days to compute the PageRank values. When implemented in C++, this can be further improved to within 1 day.

Conclusion

Financial institutions can highly benefit from collaborative fraud detection. Relevant data exchange, however, is limited due to privacy and legal restrictions. Collaborating organizations actually do not need the data itself, only the result of the analysis, the computation. Some techniques such as PageRank detect fraudulent transactions using historical transaction data in order to find anomalous patterns that deviate

The application of the MPC solution is not limited to fraud detection but can e.g. also be applied in Security monitoring & detection.

Each financial institute learns the PageRank values of its own bank accounts using a collaborative decryption scheme.

from normal transaction patterns. Individually, financial institutions only see a part of the global transactions and would benefit from a more complete view on all the transactions. In the Shared Research Program a secure PageRank algorithm has been developed to compute the PageRank of the combined transaction graph of collaborating financial institutions, without sharing any data on transactions. Each financial institute learns the PageRank values of its own bank accounts using a collaborative decryption scheme. The algorithm has been implemented in Python and experiments show that securely analyzing features of a large-scale network that is distributed over multiple parties is feasible. The application of the MPC solution is not limited to fraud detection. Current research focuses on generalizing and extending the solution for secure collaborative money laundering detection. And on how to enable secure following of cash flows and propagate risk metrics across transaction networks. The possibilities with MPC are countless. Think of opportunities such as securely sharing Indicators of Compromise between organizations or collaboratively detecting botnets. Do you have a suggestion on a possible MPC application? We are always interested in exploring new ideas!

Bibliography

[1] Poortwachter bank ziet veel, maar mag weinig – Financieel Dagblad 27 november 2018

[2] Sangers, A., Heesch, M. van, Attema, T., Veugen, T., Worm D., Wiggerman M., Veldsink J., Bloemen O.: Secure multiparty PageRank algorithm for collaborative fraud detection. In: Financial Cryptography and Data Security, February 2019. See https://fc19.ifca.ai/preproceedings/61-preproceedings.pdf

[3] Molloy, I., Chari, S., Finkler, U., Wiggerman, M., Jonker, C., Habeck, T., Park, Y., Jordens, F., van Schaik, R.: Graph analytics for real-time scoring of cross-channel transactional fraud. In: Financial Cryptography and Data Security, February 2016.

[4] A Python 3 library for Partially Homomorphic Encryption using the Paillier crypto system, https://python-paillier.readthedocs.io/en/develop/ - 2 June 2016.