Efficient comparison and sorting under encryption for anti money laundering protocols

Pranav Verma* and Roland Hostettler*
* Signals and Systems group, Department of Electrical Engineering
Uppsala University, Sweden

***Abstract*—The rise of financial crimes such as money laundering has led to the widespread adoption of anti-money laundering (AML) protocols. These protocols require handling sensitive personal and financial data, while ensuring data privacy and providing analytical capabilities. In this context, homomorphic encryption (HE) is a powerful cryptographic tool, enabling computations on encrypted data without the need for decryption. However, core operations necessary to AML, such as value comparisons, threshold checks, and sorting pose a significant computational challenge in the encrypted domain. Traditional comparison and sorting algorithms are incompatible with most HE schemes due to their deterministic and arithmetic-only nature. To overcome these challenges, recent research has focused on developing branchless, low-depth, and multiplicatively efficient circuits for encrypted comparisons and sorting. Our work explores the design and implementation of efficient comparison and sorting techniques over encrypted data, tailored for real-time AML applications. Furthermore, we explore the consensus mechanism in the architecture. Here, we propose that certain number of members must agree to make a decision about an account being suspicious of money laundering. This extended abstract is a work in progress and we are currently working on various architecture designs to find the one which simulates the real world scenario best.**

I. Introduction

As financial networks grow in complexity, so too do the methods employed to exploit them. Money laundering, in particular, presents a profound challenge—not merely because of the volume of transactions involved, but due to the layered, transnational strategies used to obscure illicit activity. Detecting these patterns requires access to granular, often sensitive financial data, and the ability to compute over it intelligently and collaboratively. Yet herein lies a paradox: the data most critical to identifying financial crimes is also that which is most legally and ethically constrained. Financial institutions cannot—and in many cases must not—share customer data openly. Thus, the core challenge becomes one of computing without revealing. This has positioned homomorphic encryption (HE) as a central tool in the development of privacy-preserving AML protocols, enabling computations directly on encrypted inputs while preserving the confidentiality of the underlying data.

Still, the adoption of HE in real-world AML systems is hindered not by theoretical limitations, but by practical bottlenecks. In particular, operations such as comparison (e.g., threshold checks, maximum detection) and sorting (e.g., prioritizing suspicious transactions) are non-trivial under encryption. These operations, which are trivial in plaintext, become complex due to the absence of native support for control flow, data-dependent memory access, and conditional branching in most HE schemes. This work is motivated by the need to rethink basic data operations—such as ordering and comparison—through the lens of encrypted computation. We aim to bridge a gap between cryptographic design and regulatory necessity.

II. Related work

In this section, we give a brief overview of each aspect of the the HE based AML protocols. We can divide the protocols in following four modules: privacy-preserving techniques in financial intelligence, homomorphic sorting techniques, private comparison, and HE based AML applications.

In [1], authors show a fairly detailed survey of how HE, among many other techniques, contributed to the the financial intelligence operations. It has allowed financial institutions to share their data with their peers (many of who are also their competitors in certain cases) with privacy assurances. Future of Financial Intelligence Sharing (FFIS) published a discussion paper [2] that shows the real world case studies where privacy-preserving techniques such as homomorphic encryption have helped tackle the financial crimes.

Homomorphic sorting techniques are a set of protocols that focuses on efficient privacy preserving sorting protocols. The important parameters in such protocol designs are the computation cost overhead on the client side devices. But, in case of AML, it can be different, since the clients here are generally banks and other financial institutions, they do not fall into a typical *client* definition that has limited computing power. A recent protocol [3], presents a k-way private sorting network that extends the 2-way sorting scheme. It shows that using a 5-way sorting techniques gives about 23% better result than 2-way technique while sorting a little over 16000 data. It is also possible to perform sorting using the Paillier cryptosystem which is a partial homomorphic encryption scheme [4].

Private comparison is another building block in the cryptography literature that has attracted researchers for a long time and there have been a consistent flow of new proposals and ideas in the domain. Comparison is the basic building block in sorting as well. DGK protocol is a well-known private comparison protocol [5] that uses HE. More recently, [6] proposed a New comparison methods based on composite polynomial approximation, achieving optimal asymptotic complexity. These methods significantly reduce computational overhead, making them suitable for applications requiring efficient encrypted comparisons.

The applications of HE in AML has also seen a lot traction in recent literature. In [7], authors discuss a scalable solution utilizing additively homomorphic encryption has been proposed to enable collaborative AML across financial institutions. This approach allows for the analysis of transaction graphs while preserving data privacy, addressing the limitations posed by the isolated data views of individual banks. Another recent work [8] proposes to use graph-based machine learning together with FHE to detect money laundering. It employs FHE over the Torus (TFHE), where computations are performed directly on encrypted data, facilitating secure data sharing across institutions and enhancing the detection of complex money laundering networks.

III. Challenges

An AML solution is difficult to design as the problem has are many unique challenges. First, a huge digital network of transactions that is continuously growing. Second, the criminals are smart and layer the transactions to and from many accounts in small parts before converging to a few accounts. Third, multinational transactions have a separate set of challenges involving multiple countries and their regulations. Apart from these, from a research point of view, there are design challenges as well that we discuss here.

a) Using ML/AI techniques: These are the latest technologies and research show very strong and positive results. Such technological revolution will lead to better designs for AML protocols. But, they bring in their own challenges like computation cost and need for big servers. If we want to collaborate with multiple banks then operating over such a huge amount of encrypted data is a challenge in itself. A traditional anomaly detection based AML scheme has simpler design but may not be that effective in today's era. The future solutions have to strike a balance between this trade-off.

b) Architecture: In a centralized server architecture, the server is assumed to be trusted or semi-trusted, which is a difficult decision for financial institutions to make as they prefer to use in-house computations. They have fair reasons not to trust any *trusted* server as it comes with a risk of single point of failure architecture. On the other hand the distributed network requires all participating banks to be connected and communicating with each other. Here, they may have serious trust issues regarding data handling policies of the other banks.

c) Cryptosystem: The choice of cryptosystem to be used in the solution design depends on all the above mentioned parameters. Based on architecture, they *key management* will differ. Do we need a group key, is there a need for separate evaluation key, should there be a threshold based decryption policy, how to verify if a party has not violated the protocol. These are the question that one would ask before deciding on which cryptosystem to be used in the solution.

IV. Our work

We focus on optimizing a key component of any AML detection protocol, namely, the comparison and sorting under encryption. In such protocols, if the comparison operations are time consuming or computationally heavy, it may create a bottleneck for the whole system. Our work is motivated by this and we present our solution towards an optimal scheme to compare and sort values under encryption. In addition, we are working on a consensus mechanism in a serverless architecture, that requires more than a threshold number of participants (banks) to mark an account as suspicious and flag it for further inspections.

We are using CKKS cryptosystem which is an FHE scheme and allows computations over encrypted real numbers. We are using a threshold decryption variant proposed in [9]. Our solution considers a centralized server architecture where a semi-honest server is connected with the participating banks. The server uses an evaluation key to perform computations over the encrypted data and only the aggregate *risk score* of a suspicious account is shared with the banks. The protocol flow is as follows:

- There is a unique ID for each individual (like person number) that can be used to uniquely identify them across banks. This banks do not want to reveal to other banks.
- Banks individually compute a risk score for each client. The risk score computation is out of the scope of our work, we just take it as a real numbered value.
- Each bank sends a tuple (bank id, client id, person number, risk score) to the server. All values are encrypted except the "bank id".
- The server aggregates the risk score of each client (user) based on their person number.
  - Here the encrypted searching/comparison is used. The server has to find a user across banks with the same person number. Then using homomorphic operations it will compute the average risk score across all banks.
  - If the risk score is above the pre-defined threshold, it will inform all concerned banks.
  - At certain interval, the server will share the risk score of clients to the respective banks, and once a certain number of banks agree to that score, the server will update its long term database value. This is a periodic process and will not interfere with the main routine of AML detection.
- If multiple banks agree that a certain account (client) has crossed the threshold of the risk score, it is flagged as suspicious.
- All the participating banks, that has a transaction history with the suspicious account will be notified for further manual inspection.

V. Conclusion

We are working on the proposed solution and the initial results look promising. The idea is to explore the domain and come up with the architecture that is as close as possible to the real world applications.

References

[1] Y. Li, T. Ranbaduge, and K. S. Ng, "Privacy technologies for financial intelligence," 2024. [Online]. Available: https://arxiv.org/abs/2408.09935

[2] N. Maxwell, "Innovation and discussion paper: Case studies of the use of privacy preserving analysis to tackle financial crime," 2020. [Online]. Available: https://arxiv.org/abs/2408.09935

[3] S. Hong, S. Kim, J. Choi, Y. Lee, and J. H. Cheon, "Efficient sorting of homomorphic encrypted data with k-way sorting network," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4389–4404, 2021.

[4] P. Verma, A. Mathuria, and S. Dasgupta, "Efficient privacy preserving top-k recommendation using homomorphic sorting," *Cryptology ePrint Archive*, 2022.

[5] I. Damgård, M. Geisler, and M. Krøigaard, "Efficient and secure comparison for on-line auctions," in *Information Security and Privacy: 12th Australasian Conference, ACISP 2007, Townsville, Australia, July 2-4, 2007. Proceedings 12*. Springer, 2007, pp. 416–430.

[6] J. H. Cheon, D. Kim, and D. Kim, "Efficient homomorphic comparison methods with optimal complexity," Cryptology ePrint Archive, Paper 2019/1234, 2019. [Online]. Available: https://eprint.iacr.org/2019/1234

[7] M. B. van Egmond, V. Dunning, S. van den Berg, T. Rooijakkers, A. Sangers, T. Poppe, and J. Veldsink, "Privacy-preserving anti-money laundering using secure multi-party computation," in *International Conference on Financial Cryptography and Data Security*. Springer, 2024, pp. 331–349.

[8] F. Effendi and A. Chattopadhyay, "Privacy-preserving graph-based machine learning with fully homomorphic encryption for collaborative anti-money laundering," in *International Conference on Security, Privacy, and Applied Cryptography Engineering*. Springer, 2024, pp. 80–105.

[9] C. Mouchet, J. Troncoso-Pastoriza, J.-P. Bossuat, and J.-P. Hubaux, "Multiparty homomorphic encryption from ring-learning-with-errors," *Proceedings on Privacy Enhancing Technologies*, vol. 2021, no. 4, pp. 291–311, 2021.