BlindSort: le tri en aveugle

Présentation de projet INF8750

Félix Larose-Gervais

1 Proposition de sujet

Implémentation de Blindsort, un algorithme de tri en aveugle. C'est-à-dire un programme qui, étant donné une liste d'éléments chiffrée, retourne le chiffré de cette liste triée. Pour ce faire, on utilisera le chiffrement homomorphe, en particulier le schéma TFHE [Chillotti *et al.*, 2020], ainsi que de nouvelles primitives d'accès matriciel et de permutation aveugles.

2 Chiffrement homomorphe

Un système de chiffrement complètement homomorphe [Marcolla *et al.*, 2022] est un crypto-système qui permet d'effectuer des calculs sur des données chiffrées. Un tel système permet donc notamment de mettre en place des services respectant la vie privée, puisqu'il ne nécéssite pas de connaitre les données des utilisateurs en clair pour opérer.

3 Implémentation

Nous projetons d'implémenter BlindSort en Rust, sur la base de plusieurs technologies existantes. Principalement, la librairie open source TFHE-rs [Zama, 2022], qui implémente le schéma TFHE et RevoLUT [Azogagh, 2024], une librairie construite sur TFHE-rs offrant de nouvelles primitives: l'accès matriciel aveugle [Azogagh *et al.*, 2023] et la permutation aveugle.

4 Algorithme

L'algorithme de tri est une adaptation TFHE d'une application similaire aux schémas BGV/BFV de [Iliashenko et Zucca, 2021], lui-même variante du Direct Sort dû à [Çetin *et al.*, 2015]. Nous adapterons l'algorithme pour exploiter les nouvelles primitives offertes par RevoLUT. L'idée initiale est de construire une matrice de comparaisons entre les paires d'éléments de la liste d'entrée, calculer les poids de Hamming de ses colonnes, et interpréter ses poids comme une permutation à appliquer à la liste d'entrée.

References

[Azogagh, 2024] Azogagh, S. (2024). Revolut. https://github.com/sofianeazogagh/revoLUT.

[Azogagh *et al.*, 2023] Azogagh, S., Deflour, V. et Killijian, M.-O. (2023). Oblivious Turing Machine. Cryptology ePrint Archive, Paper 2023/1643. https://eprint.iacr.org/2023/1643. Récupéré le 2024-02-05 de https://eprint.iacr.org/2023/1643

[Chillotti *et al.*, 2020] Chillotti, I., Gama, N., Georgieva, M. et Izabachène, M. (2020). TFHE: Fast Fully Homomorphic Encryption Over the Torus. *Journal of Cryptology*, *33*(1), 34–91. http://dx.doi.org/10.1007/s00145-019-09319-x. Récupéré le 2024-02-05 de http://link.springer.com/10.1007/s00145-019-09319-x

[Iliashenko et Zucca, 2021] Iliashenko, I. et Zucca, V. (2021). Faster homomorphic comparison operations for BGV and BFV. *Proceedings on Privacy Enhancing Technologies*. Récupéré le 2024-01-25 de https://petsymposium.org/popets/2021/popets-2021-0046.php

[Marcolla *et al.*, 2022] Marcolla, C., Sucasas, V., Manzano, M., Bassoli, R., Fitzek, F. H. P. et Aaraj, N. (2022). Survey on Fully Homomorphic Encryption, Theory, and Applications. *Proceedings of the IEEE*, *110*(10). http://dx.doi.org/10.1109/JPROC.2022.3205665

[Zama, 2022] Zama (2022). TFHE-rs: A Pure Rust Implementation of the TFHE Scheme for Boolean and Integer Arithmetics Over Encrypted Data. https://github.com/zama-ai/tfhe-rs.

[Çetin *et al.*, 2015] Çetin, G. S., Doröz, Y., Sunar, B. et Savaş, E. (2015). Depth Optimized Efficient Homomorphic Sorting. In K. Lauter et F. Rodríguez-Henríquez (dir.), *Progress in Cryptology – LATINCRYPT 2015*, volume 9230 61–80. Cham: Springer International Publishing. Series Title: Lecture Notes in Computer Science