SHA-3 Parameters with Machine Learning

Double Master Maths & Info MAFI 2023

1 Introduction, Motivation and Related Works

Links between machine learning and cryptography represent an emerging and exciting area of research, with the potential to unlock new applications and capabilities in both fields.

One example of the link between machine learning and cryptography is in the field of secure multi-party computation (MPC). MPC (cf. the seminal paper of Yao [13]) is a cryptographic technique that allows multiple parties to perform computations on their private data without revealing their inputs to each other. Machine learning can be applied in MPC scenarios to enable collaborative training of machine learning models on private data.

Machine learning (ML) can be used to optimize the parameters of cryptographic hash functions such as SHA-3. Specifically, ML algorithms can be trained on large datasets of input/output pairs for a given hash function, and then used to automatically search for the optimal parameters that maximize the performance of the hash function on this dataset. In [2], Abbas *et al* used evolutionary algorithms [8][1] to optimize the parameters of the SHA-3 hash function, including the number of rounds and the capacity and bitrate parameters of the sponge construction. The authors show that their approach is able to find parameter values that outperform the standard SHA-3 configuration in terms of speed and collision resistance.

For example, one could use ML to optimize the SHA-3 sponge construction's capacity and bitrate parameters [3], which determine the security level and efficiency of the hash function, respectively. By training an ML model to predict the hash output for a given input message and varying the capacity and bitrate parameters during training, one could identify the optimal parameter values that result in the highest prediction accuracy on a held-out validation set. Note that the authors of [5] used ML approach to optimize cryptographic parameters for elliptic curves.

Overall, ML can be a powerful tool for optimizing the performance and security of cryptographic hash functions such SHA-3, by enabling efficient exploration of the high-dimensional parameter space that governs their behavior.

Another area where machine learning and cryptography can be linked is in the development of privacy-preserving machine learning techniques [12]. Privacy-preserving machine learning aims to protect sensitive data while still enabling machine learning algorithms to extract insights and patterns from that data. Cryptography can be used in privacy-preserving machine learning techniques to encrypt and decrypt data, and to secure communication channels between different components of the machine learning system.

Observe that the SHA-3 Keccak-256 algorithm has been chosen for Ethereum Classic, Ethereum 2.0 (Proof of Stake), Maxcoin and Quark [4] as the algorithm resists to attacks from even quantum computers.

To summarize our introduction, proficiency in mastering SHA-3 and its parameters is a valuable asset for anyone involved in cryptography and blockchain. Furthermore, utilizing machine learning techniques to explore the interplay between SHA-3 parameters and performance can provide key takeaways and pertinent knowledges in our high level interdisciplinary works.

[1] In few words, evolutionary algorithms apply basic principles of natural evolution, such as selection, mutation, and recombination to optimize algorithms in the fields of engineering and computer science.

2 Proposed approach

To tune the parameters of a cryptographic algorithm using machine learning, the group can start with the following general steps:

1. Identify the performance or security metrics that need to be optimized for the cryptographic algorithm. For example, the metric might be the speed of encryption or decryption, the size of the encryption key (see [11, Chapter 5, Section 5.1.1] for a Theorem about key size and security), or the resistance to attacks (see [9, Chapter 3, Section 3.1.1] for a Theorem about resistance to attacks).

2. Collect a dataset of inputs and outputs for the cryptographic algorithm. The inputs could be plaintext or ciphertext, and the outputs could be the corresponding ciphertext or plaintext, or some other metric related to the performance or security of the algorithm.

3. Choose an appropriate machine learning algorithm that can learn from the dataset and optimize the parameters of the cryptographic algorithm. There are many machine learning algorithms to choose from, such as neural networks, decision trees, and genetic algorithms. The choice of algorithm will depend on the specific requirements and constraints of the problem.

4. Train the machine learning algorithm on the dataset of inputs and outputs, using the performance or security metrics as the target variable. This will involve running the cryptographic algorithm with different parameter settings and recording the corresponding performance or security metrics.

5. Evaluate the performance of the machine learning algorithm on a validation dataset, to ensure that it is generalizing well and not overfitting to the training data.

6. Use the optimized parameters to run the cryptographic algorithm on new data, and measure its performance or security using the same metrics as before. If possible, turn them into mathematical Theorem(s) as in [11, 9] for the choosen metrics.

3 Conclusion

In addition to conducting novel and innovative research, it is also important to present your results in a clear and concise manner, and to provide rigorous experimental evaluation and validation of your methods. This will increase the likelihood that your research will be accepted by the scientific communities. It is important to note that there are existing recent works relating ML and cryptographic protocols published in highly competitive and recognised conferences and journals [7, 6, 10]. For example, in [10] Khan and Zulkernine propose a machine learning framework to optimize the parameters of symmetric cryptography algorithms, using an objective function based on both security and performance metrics. They demonstrate the effectiveness of their approach on the Advanced Encryption Standard (AES) algorithm [1].

Note that this project is very similar to a twin project but on Diffie-Hellman protocols.

References

[1] Advanced encryption standard (AES), 2001. `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf`.

[2] Tariq Abbas, Farkhund Iqbal, Kashif Nisar, and Sajjad Haider. Optimizing SHA-3 hash function using evolutionary algorithms. In *2018 International Conference on Frontiers of Information Technology (FIT)*, pages 32–37. IEEE, 2018.

[3] Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. Keccak sponge function family main document. In *3rd International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pages 1–23. Springer, 2011.

[4] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Hashing with SHA-3. In *Proceedings of the 11th international conference on Cryptology and network security*, pages 1–21. Springer, 2013.

[5] Saptarshi Bhattacharya, Supratim Chakraborty, and Sandip Das. Using machine learning to optimize cryptographic parameters: A case study with elliptic curves. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6. IEEE, 2018.

[6] Rupam Chatterjee, Sudip Samanta, Indranil Ray, and Anupam Chattopadhyay. Automated parameter selection for cryptographic implementations using machine learning. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pages 557–569. ACM, 2019.

[7] Jia Chen, Wen Sun, Qing Li, Guojun Wang, and Jianwei Li. A machine learning framework for cryptographic parameter selection. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 61–72. ACM, 2017.

[8] John H Holland. Adaptation in natural and artificial systems. *MIT press*, 1(2):1–19, 1992.

[9] J. Katz and Y. Lindell. *Introduction to Modern Cryptography, Second Edition*. Chapman and Hall/CRC, 2014.

[10] Muhammad Ehtisham Khan and Mohammad Zulkernine. A machine learning approach for parameter tuning in symmetric cryptography. *IEEE Transactions on Information Forensics and Security*, 14(6):1624–1639, 2019.

[11] Alfred Menezes, Paul C van Oorschot, and Scott A Vanstone. *Handbook of Applied Cryptography*. CRC press, 1996.

[12] Reza Shokri and Vitaly Shmatikov. Privacy-preserving machine learning: threats and solutions. *IEEE Security & Privacy*, 13(1):68–75, 2015.

[13] Andrew Chi-Chih Yao. Protocols for secure computations. *Proceedings of the IEEE Symposium on Foundations of Computer Science*, 24:160–164, 1982.