Blockchain Applications with Privacy using Efficient Multiparty Computation Protocols

A. Anasuya Threse Innocent
Department of Computer Science and Engineering
Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham
India

G. Prakash
Department of Computer Science and Engineering
Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham
India
g_prakash@blr.amrita.edu

Abstract—**Blockchain technology provides a distributed solution, but not privacy of data used. Data privacy is included with the help of secure multiparty computation protocols and which in turn increases the complexity of application. This paper provides an efficient solution for blockchain technology with privacy by including a novel optimization for secure computation protocols.**

Keywords—blockchain, privacy, multiparty computation, efficiency

I. Introduction

Blockchains provide decentralized solutions for a variety of applications, ensuring honest parties to be part of the system. Blockchain allow the mutually distrusting parties to transact crypto-currency without the involvement of trusted third party. Transparency is maintained in every level and on each transaction leaving the data exposed to all the parties involved. In short we can say that, blockchains provides an excellent decentralized solution without privacy. On the other hand the emerging field of Cryptography, secure computation provides privacy of data on a distributed environment, but the parties involved in computation have to be carefully monitored. The above two concepts of blockchain and secure computation shall be combined which can lead to a number of practical solutions for decentralized distributed computing applications with privacy.

Research has been carried out to incorporate privacy to blockchain technology by the use of secure multiparty computation protocols. Platforms namely Enigma [1], Hawk [2], Ouroboros [3], AntNest [4], Raziel [5], PlatON [6], SoK [7], Wanchain [8, 9] and a number of works [10 - 15] have combined the power of multiparty computation with that of blockchain. In this paper we propose a novel framework for blockchain applications with efficient multiparty computation protocols.

II. Literature

Blockchain technology emerged with the invention of Bitcoin, the cryptocurency for Internet by Nakamoto [16] in 2008. The structure beneath consists of blockchain which is an open ledger of blocks linked by the hash values of subsequent blocks. Every action that happens on the blockchain is reflected on each node involved and the data and transactions are stored on every node. Before adding a new block, the actions performed should be published on the network, the transaction history has to be verified and confirmed by all the nodes and after that new block is added to the chain. This process eliminates the need for centralized trusted third party or a centralized authority to manage the transactions, and allows all the nodes to be part of a decentralized system only with honest parties. Blockchains can be public or private. The well-known public blockchains are Bitcoin [16], Ethereum [17], and the private ones are Eris [18], Hyperledger [19], and Ripple [20]. All these blockchains use cryptographic concepts to achieve security by means of access control, and identity authentication. But the problem here is privacy of private data, restricting the practical applications of blockchain.

Secure computation was developed by Yao [21] in 1982 for a two-party case to solve the Millionaires' problem without a trusted third party, and later developed for multiparty case [22, 23] and termed as multiparty computation (MPC). Even though secure computation does the private data computation without revealing it to the other parties involved, complexity of computation is very high. As well it is necessary to crosscheck if the parties involved are honest. The main components of secure computation protocols are garbled circuit construction (GC) followed by oblivious transfer (OT). Any application can be represented as a mathematical representation and which in turn can be implemented as a logical circuit. The circuit representation can be arithmetic or Boolean depending on the application. The size of underlying circuit is directly proportional to the complexity of the garbling protocol. Hence optimization of GC construction can lead to efficient secure computation protocol development. Extensive research has been carried out on optimization of GC construction and the comparative study based on Boolean circuits is given by Innocent et al. [24 - 27], which gives a short description of existing optimizations. The FastGarble framework [26] uses batch-key cipher optimization along with the existing optimizations on GC construction, and the FastGarble with Universal Gates framework [27] makes use of single type of gates along with batch-key cipher optimization, and proved to be ≅ 66 % more efficient in terms of time complexity than the other existing optimizations.

Combining the decentralization concept of blockchain with the privacy of secure computation leads to a number of decentralized applications with privacy [1 – 15]. All the platforms try to achieve efficient multiparty computation with secure blockchains. In this paper we combine the power of FastGarble with Universal Gates framework to the blockchain to achieve more efficient framework for blockchain applications with privacy.

III. FGUGChain

The proposed framework FGUGChain (pronounced as, [ɛf][dʒiː][juː][dʒiː] chain), the FastGarble Universal Gates Chain consists of on-line and off-line phases. The blockchain operations are executed in on-line phase and the complex MPC protocol executions are carried out in off-line phase. The overall structure of FGUGChain framework is shown in Fig. 1.

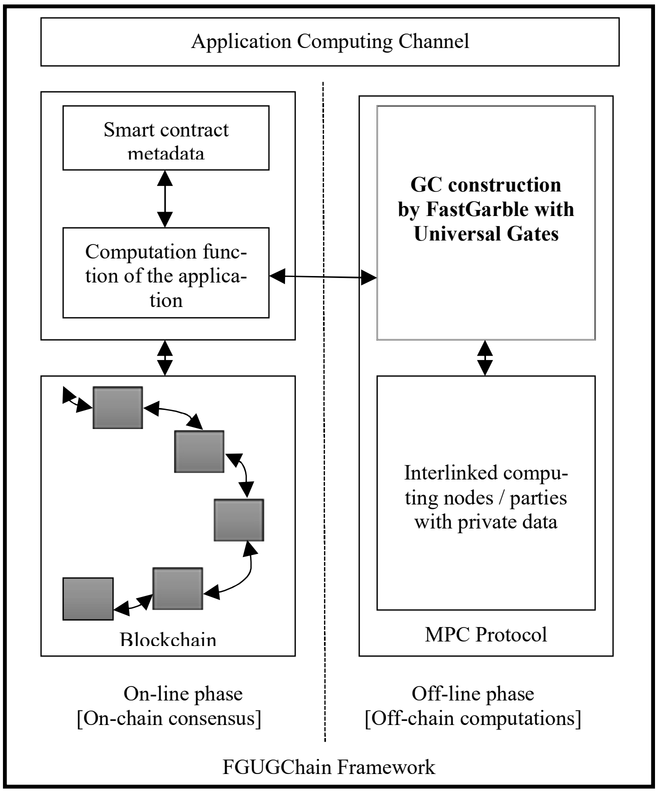
Application Computing Channel
Smart contract metadata
Computation function of the application
GC construction by FastGarble with Universal Gates
Interlinked computing nodes / parties with private data
Blockchain
MPC Protocol
On-line phase [On-chain consensus]
Off-line phase [Off-chain computations]
FGUGChain Framework

Fig. 1. Overall structure of FGUGChain framework

The FGUGChain framework consists of an application computing channel which combines the work on on-line and off-line phases of the framework to create a complete application.

A. On-line Phase

The on-line phase of the computation or the on-chain consensus does the blockchain processes without storing or exposing the data to the related parties.

a) Smart Contract Metadata: The smart contract used here is similar to that of stateless contract of PlatON [6]. It does not store any states of the chain, or reveal data to the computing nodes. When smart contract metadata is executed, the input data comes from the local databases of the off-chain nodes participating in MPC. The actual computation is broken down into multiple sub-tasks and are distributed to the computing nodes without revealing the data. MPC protocol used guarantee the ownership of the parties involved.

b) Computation Function of the Application: The computation function of the application is the underlying computation function, which is defined by the smart contract metadata along with the input and output parameters. This computation function is converted into a Boolean circuit, divided into multiple sub-circuits and is sent for GC construction in off-line phase. As well the GC generated are linked to it.

c) Blockchain: The computation function defined by the smart contract metadata is split and distributed to multiple computing nodes / parties by the block producers while generating the blocks. The computing nodes returns the result as well the proof of computation, and are also packed into the blocks by the producer. Validity of the block is determined by other nodes, just by verifying the proof, which reduces the complex, redundant computations on each node, and hence the block verification time. A new block is added to the chain once the verification process is completed.

B. Off-line Phase

The off-line phase does the off-chain MPC protocol execution. Yao's GC construction [24, 28] plays a vital role in MPC protocols. The underlying computation function is implemented into a Boolean circuit, divided into multiple sub-circuits and then sent to computing nodes and are executed in parallel. Each sub-circuit is sent to multiple computing nodes to maintain certain level of redundancy and availability guarantee. The GC constructor selects random tokens for each wire in the Boolean circuit, encrypts the output token with that of input tokens for each gate, the encrypted tokens are garbled per gate and thus the garbled circuit is generated [24, 28, 29]. The GC evaluator uses OT protocol and deciphers the GC to obtain the required result [28].

a) GC Construction by FastGarble with Universal Gates: The GC construction here uses the FastGarble with Universal Gates framework [27]. Here, instead of different types of gates, any one of the universal gates (NAND / NOR) are used for Boolean circuit generation, and hence for GC construction. Also the batch-key cipher optimization [26] for GC construction is used along with point-and-permute [30], garbled-row reduction [31], dual-key cipher with fixed-key block cipher with Advanced Encryption Standards- New Instructions (AES-NI) and Simple Circuit Description (SCD) [32] optimizations. The FastGarble with Universal Gates framework is proved to be ≅ 66 % more efficient in terms of time complexity and also requires only 2 ciphertexts per gate for communications [27]. Overall, the use of FastGarble with Universal Gates framework improves the efficiency of MPC application.

b) Interlinked Computing Nodes / Parties with Private Data: These are the parties involved in MPC with their private data. With this FGUGChain framework, they are able to compute on their private data without revealing it to even a trusted third party on a decentralized environment.

IV. Conclusion and Future work

The FGUGChain framework proposed in this paper gives a theoretical exposure to the framework and can be used to develop various distributed computing applications. The FastGarble with Universal Gates Framework used for incorporating secure computation is 66 % more efficient than the existing practical implementations on secure computation. As well, it uses only 2 ciphertexts per gate for communication, and is proven to be secure under simulation-based and distinguishability-based security analysis.

The blockchain concept provides a better decentralization and allows only the honest parties to be part of computation. Thus privacy is maintained in the decentralized environment. The proposed FGUGChain framework will be developed into a fully functional system in near future.

References

[1] G. Zyskind, O. Nathan, and A. Pentland, “Enigma: Decentralized computation platform with guaranteed privacy”, arXiv preprint arXiv:1506.03471. 2015 Jun 10.

[2] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, “Hawk: The blockchain model of cryptography and privacy-preserving smart contracts”, In 2016 IEEE symposium on security and privacy (SP) pp. 839-858.

[3] A. Kiayias, A. Russell, B. David, and R. Oliynykov, “Ouroboros: A provably secure proof-of-stake blockchain protocol”, In 2017 Annual International Cryptology Conference, Springer, Cham, pp. 357-388.

[4] L. Zhou, L. Wang, Y. Sun, and T. Ai, “AntNest: Fully non-interactive secure multi-party computation”, IEEE Access, 6, pp.75639-75649.

[5] D. C. Sánchez, “Raziel: Private and verifiable smart contracts on blockchains”, arXiv preprint arXiv:1807.09484, 2018.

[6] PlatON: A high-Efficiency trustless computing network, whitepaper from www.platon.network

[7] J. Garay, and A. Kiayias, “Sok: A consensus taxonomy in the blockchain era”, Cryptology ePrint Archive, Report 2018/754, 2018.

[8] Wanchain – Yellow paper, https://wanchain.org/files/Wanchain-Yellowpaper-EN-version.pdf , accessed on 18-04-2019.

[9] O. Birch, "Secure multiparty computation and Shamir’s secret sharing on Wanchain," https://medium.com/wanchain-foundation/secure-multiparty-computationand-shamirs-secret-sharing-on-wanchain-e502012b80ef, accessed on 31-01-2019.

[10] G.W. Peters, and E. Panayi, “Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the internet of money”, In Banking beyond banks and money, Springer, Cham, 2016, pp. 239-278.

[11] G. Zyskind, Efficient secure computation enabled by blockchain technology, Doctoral dissertation, Massachusetts Institute of Technology, 2016.

[12] MPC on Ethereum, https://ethresear.ch/t/mpc-on-ethereum/311, accessed on 20-11-2018.

[13] D. Shrier, W. Wu, and A. Pentland, Blockchain & Infrastructure (Identity, Data Security), Massachusetts Institute of Technology, connection.mit.edu, May 2016, accessed on 28-04-2019.

[14] D. Cerezo Sánchez, “The Valuation of Secrecy and the Privacy Multiplier, Available at SSRN 3103343, 2018.

[15] D.W. Archer, D. Bogdanov, Y. Lindell, L. Kamm, K. Nielsen, J.I. Pagter, N.P. Smart, and R.N. Wright, “From Keys to Databases — Real-World Applications of Secure Multi-Party Computation”, 2018, The Computer Journal, 61(12), pp.1749-1771.

[16] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system”, 2008, https://scholar.google.com.

[17] V. Buterin, “A next-generation smart contract and decentralized application platform”, white paper, 2014.

[18] Eris: The smart contract application platform, https://erisindustries.com/, accessed on 28-04-2019.

[19] Hyperledger Blockchain, https://www.hyperledger.org/, accessed on 28-04-2019.

[20] Ripple, https://ripple.com/, accessed on 28-04-2019.

[21] A.C. Yao, “Protocols for secure computations”, In: FOCS, 1982, pp. 160–164.

[22] A. C. Yao, “How to generate and exchange secrets”, In: FOCS, 1986, pp. 162–167.

[23] O. Goldreich, M. Silvio, and A. Wigderson, “How to play any mental game”, In: STOC, 1987, pp. 218–229.

[24] A. A. T. Innocent, and K. Sangeeta, “Secure two-party computation with AES-128: Generic approach and exploiting specific properties of functions approach”, In: ICADIWT, 2014, pp. 87-91.

[25] A. A. T. Innocent, and K. Sangeeta, “Secure Two-Party Computation: Generic Approach and Exploiting Specific Properties of Functions Approach”, JISR, 2014, vol. 5(1), pp. 19-27. www.dline.info

[26] A. A. T. Innocent AAT, K. Sangeeta, and G. Prakash, “An Efficient Garbled Circuit Construction for Secure Computation Protocols”, manuscript under submission.

[27] A. A. T. Innocent AAT, K. Sangeeta, and G. Prakash, “Universal gates on garbled circuit construction”, Concurrency Computat Pract Exper. 2019, e5236, https://doi.org/10.1002/cpe.5236.

[28] Y. Lindell, B. Pinkas, “A proof of security of Yao’s protocol for two-party computation”, Journal of Cryptology, 2009, vol. 22(2), pp. 161–188.

[29] M. Bellare, V. T. Hoang, and P. Rogaway, “Foundations of garbled circuits”, In CCS, 2012, pp. 784–796.

[30] M. Naor, B. Pinkas, and R. Sumner, “Privacy preserving auctions and mechanism design”, In ACM EC, 1999, pp. 129-139.

[31] B. Pinkas, T. Schneider, N. P. Smart, and, S. C. Williams, “Secure two-party computation is practical”, In Asiacrypt, Berlin, Springer; 2009, pp. 250-267.

[32] M. Bellare, V. T. Hoang, S. Keelveedhi, and P. Rogaway, “Efficient garbling from a fixed-key blockcipher”, In IEEE SSP, 2013, pp. 478-492.