XVI Jornadas de Ingeniería Telemática

JITEL 2023

La Salle – Universitat Ramon Llull

*Actas de las XVI Jornadas
de Ingeniería Telemática
(JITEL 2023),
Barcelona (España),
8-10 de noviembre de 2023.*

ISBN

Cryptographic approaches for confidential computations in blockchain

Daniel Morales, Isaac Agudo
Network, Information and Computer Security Lab (NICS)
Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga
damesca@uma.es, isaac@uma.es

Blockchain technologies have been widely researched in the last decade, mainly because of the revolution they propose for different use cases. Moving away from centralized solutions that abuse their capabilities, blockchain looks like a great solution for integrity, transparency, and decentralization. However, there are still some problems to be solved, lack of privacy being one of the main ones. In this paper, we focus on a subset of the privacy area, which is confidentiality. Although users are increasingly aware of the importance of confidentiality, blockchain poses a barrier to the confidential treatment of data. We initiate the study of cryptographic confidential computing tools and focus on how these technologies can endow the blockchain with better capabilities, i.e., enable rich and versatile applications while protecting users' data. We identify Zero Knowledge Proofs, Fully Homomorphic Encryption, and Secure Multiparty Computation as good candidates to achieve this.

***Palabras Clave*—blockchain, privacy, confidentiality, secure multi-party computation, zero knowledge proofs, fully homomorphic encryption**

I. Introduction

Blockchain technologies have emerged as a great solution for integrity, transparency, and decentralization. Broadly speaking, a blockchain network is a set of nodes with a P2P topology, which collaboratively maintain a unified ledger. Despite being conceived to manage cryptocurrency transactions (Bitcoin), other solutions have built a secure and distributed computing platform on top of the network, e.g., Ethereum. The key technology that has made such a secure ledger possible is Byzantine Fault Tolerant Consensus, in which a set of distributed and distrusted nodes can agree on what data is recorded in the ledger each time, resulting in a unified view of the ledger.

Since its conception, many use cases have been proposed [1], [2], e.g., financial, health, supply chain, or government.

Despite the benefits of blockchain, the lack of privacy hinders its adoption. Although it provides pseudonymity, it has been shown that users can be deanonymized [3]. Private connections, e.g., TOR [4], are recommended to mitigate this, at the expense of losing usability. Accessing blockchain data can also be a problem, because the most of the end-users do not own a blockchain node but delegate the access to a node provider[1], making them to become trusted third parties that can cheat on data provided, because end-users do not store all the blockchain data and cannot verify correctness. Also, the provider can perform a profiling attack, tracking all the activity by the user. Such issues directly ballast a real decentralization, which is the main contribution of blockchain.

Another issue is the lack of confidentiality. The evolution of blockchain has drifted towards programmable platforms, e.g., the Ethereum's Virtual Machine, which allows secure general-purpose computations. However, this approach loses its meaning when dealing with confidential data, as data must be decrypted to contribute to an on-chain computation. Different use cases, e.g., financial, or biometric data computation do not fit well with this public model. Finally, regulations such as GDPR can also contribute to restricting use cases.

Although blockchain's lack of confidentiality has been partially addressed, there are some misconceptions. One of the most trendy confidential computing technologies are Non-Interactive Zero Knowledge Proofs (NI-ZKP), which allow verifying that a computation has been performed correctly using specific data, without exposing them. However, NI-ZKP are mainly used in the blockchain ecosystem to achieve succinctness, e.g., in Layer 2 solutions [5]. While they can really help to acquire more capabilities while retaining more confidentiality, it is important to note that NI-ZKP must be computed directly on the plaintext data somewhere. This implies an overhead on

[1] https://www.infura.io/

the data owner's side, or a delegation to a trusted party to compute the proof if there are many data providers involved. There are other cryptographic solutions, e.g., Secure Multi-Party Computation (MPC) and Fully Homomorphic Encryption (FHE), that allow distrusted parties to compute on confidential data without exposing it. Unlike NI-ZKP, these technologies enable delegated computations on confidential data, as will be discussed in next sections.

In this paper we initiate research on blockchain's confidentiality problem, where our main contribution is a gathering of different technologies that can contribute to solve it. We briefly discuss their main features and argue to what extent they can actually achieve a confidentiality-preserving blockchain.

The rest of the paper is organized as follows: Section II gathers some surveys on blockchain privacy. Next, Section III analyzes three key characteristics of blockchain and their relation to confidentiality. The main technologies available for confidential blockchain solutions are briefly described in Section IV, and later discussed in Section V, emphasizing their relations and caveats w.r.t. blockchain. Finally, some conclusions and future work are presented in Section VI.

II. RELATED WORK

Blockchain privacy has been addressed in different works [6], [7], [8], mainly distinguishing between private payments and confidential computations ([8] also covers function privacy). However, private payments have been much more covered than confidential computations, mainly due to the maturity of the solutions. In addition, [8] states that confidential computations are much more difficult to achieve than private payments.

To achieve confidential computations in blockchain, the three works above claim NI-ZKP, FHE and Trusted Execution Environments (TEE) as the most extended building blocks, but [7], [8] also consider (briefly) MPC. In fact, [8] is the only work that deeply covers usability and interoperability of these techniques, identifying as open problems the handling of multi-user inputs (partially solved by MPC or multi-key FHE) and the development of case-specific cryptographic primitives to achieve more efficient solutions.

III. BLOCKCHAIN AND PRIVACY

This Section introduces some concepts that provide an understanding of how a standard blockchain (with public data) works and how confidential data can be related to it.

A. Blockchain state model

Roughly speaking, each node (or most of them) in a blockchain maintains a state S, which is computed from all recorded data. Each time new data x arrives on the blockchain, the state is re-computed using a state transition function $S' \leftarrow Transition(S, x)$. This is typically implemented in batches (a set of transactions forms a block) and the "checkpoints" of the state are computed using hash functions, which also link the blocks together. The specific details vary from blockchain to blockchain (in Bitcoin hashing transactions is enough, while Ethereum also maintains accounts and smart contracts). Although chaining blocks by hashing is the classic and most widespread option, there are new solutions that compact the whole state to constant size thanks to recursive ZKP[2].

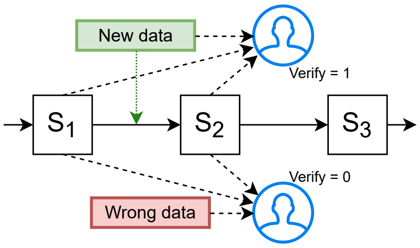

Fig. 1. The blockchain state model

Figure 1 depicts how the blockchain state evolves when new data is stored. More precisely, the new data (a block) triggers the transit from S_1 state to S_2 state. Any node in possession of S_1, S_2, and the block data can verify the correctness of the transition. In contrast, an incorrect block (due to an error or a modification attack) does not pass verification.

As for confidential data, the way the blockchain state is computed presents a first barrier, since every piece of data included in a state transition phase must be available in the verification process. Given a confidential value, including it locally in the owner's state leads to a different state from the rest of the network, losing the sense of consensus, while making it available to everyone means losing confidentiality. Confidential data can be added to the state using ciphertexts, however it is interesting to consider what value that actually adds versus storing data off-chain.

B. Blockchain storage model

Blockchain storage is problematic by nature, due to its high cost, as the ledger view must be the same for each node (data replication enables availability and eliminates deletion). Figure 2 compares a centralized storage system with a decentralized one. The centralized system allows deploying a central computer with a large amount of memory (in contrast to constrained clients) more cheaply than the decentralized one, where each node must store the same amount of information.

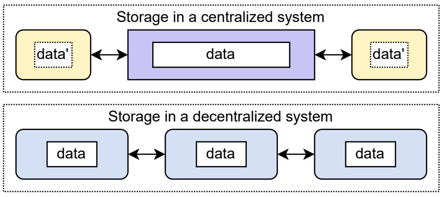
Storage in a centralized system
data'
data
data'
Storage in a decentralized system
data
data
data

Fig. 2. Storage model in a centralized and decentralized system

In practice, different types of nodes can be deployed depending on the amount of data they store, e.g., Ethereum distinguishes between full nodes (which store all data and

[2] https://minaprotocol.com/

can verify states) and light nodes (which only store block headers and have to request data from full nodes).

Sharding [9] is a recent idea that aims to minimize the problem of replicated storage by dividing the network into logical subnets with independent data and validators, which are synchronized through a main network.

Despite sharding minimizes the exposure of data to network nodes, it does not really aim at confidentiality, but at performance, as specific data fragments can be requested if needed. In fact, the replicated storage does not pose a problem for confidentiality when using, e.g., ciphertexts, despite they will be publicly available as long as the blockchain lives, which increases the attack surface.

C. *Blockchain computation model*

Ethereum introduced a computational model that allows the use of data on the blockchain. Roughly speaking, its virtual machine accepts data and smart contract opcodes that enable general-purpose computations. The main difference with the centralized model is that data, contracts, and computation must be managed by each node, i.e., a node must re-compute a function to verify its correctness, leading to a secure, reliable, and expensive system.

In general, there exist two models (see Figure 3) regarding how a computation is executed in a blockchain:

On-chain. The computation is executed by the blockchain, i.e., any node executes two phases: (1) $result \leftarrow Compute(x)$, and (2) $\{0,1\} \leftarrow Verify(result, state)$.

Off-chain. The computation is not replicated, i.e., inputs and outputs can be stored on-chain, but the computation cannot be verified by the blockchain nodes.

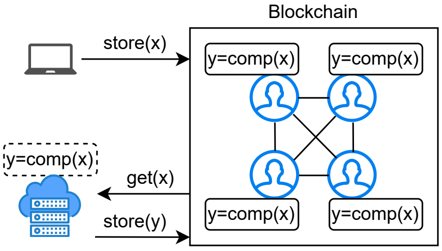

Fig. 3. On-chain (solid) vs off-chain (dashed) computation

It is easy to realize that on-chain computations are more expensive, but very secure, since to alter the result a malicious adversary must corrupt most of nodes. On the other hand, the off-chain computation model implicitly assumes a trust relation in the delegated computational party, but is cheaper. It is this area that NI-ZKP has contributed most, by storing publicly verifiable proofs of correctness on the chain.

Finally, as far as confidential computing is concerned, the on-chain model does not allow data to be protected by default, as it must be publicly available to allow verification of correctness. The only option available is to perform the computation on the user side, or to delegate it to a trusted third party, assuming they will not expose the data.

IV. TECHNOLOGIES FOR CONFIDENTIAL BLOCKCHAIN

In this Section, we gather a set of technologies that enable confidential computing and can be deployed in blockchain scenarios.

A. *NI-ZKP*

A NI-ZKP [10] allows a prover to convince a verifier (with one message) that a statement is true using some confidential data and without exposing it. These protocols can be formalized as follows:

$\pi \leftarrow Prove(setup, st, x, w)$: the prover generates a proof π using public data x and private data w that computes the statement st.

$\{0,1\} \leftarrow Verify(st, x, \pi)$: the verifier checks whether the statement is true when computed on x and w.

B. *MPC and Proactive-MPC*

MPC protocols allow a set of distrustful parties $\{P_1, ..., P_N\}$ to compute a function f on some private data $\{w_0, ..., w_N\}$ without exposing w_i to a party P_j with $j \neq i$. At the end, the computation outputs $y \leftarrow f(w_0, ..., w_N)$ as if it had been computed in clear. There exist different approaches for MPC, e.g., Garbled Circuits [11] for 2-party and Secret Sharing Schemes [12] for N-party, where security relies on the adversary inability to corrupt $t < N$ nodes. We remark that FHE (explained below) is typically understood as a specific form of MPC.

Proactive-MPC [13] is a variation in which every m operations of the computation the secret-shares are moved from a committee of holders C_1 to C_2 using a handover and re-sharing protocol. This approach limits the adversary time to corrupt parties, as secrets will not always reside in the same place.

MPC solutions for blockchain [14], [15], [16] tend to coordinate the computation on-chain and execute it off-chain, using a pool of designated nodes.

C. *FHE*

Roughly speaking, an FHE scheme allows to compute on ciphertexts as if it was computed on plaintexts, i.e., given $c_1 = Enc_k(m_1)$ and $c_2 = Enc_k(m_2)$, it is possible to compute $c_{add} = Enc_k(m_1 + m_2)$ or $c_{mul} = Enc_k(m_1 \cdot m_2)$. FHE was inefficiently introduced in [17], but it has been largely improved [18]. The most extended FHE settings work on public key cryptography and allow multiple clients to delegate the computation to a single server.

FHE solutions for blockchain [19] enable on-chain computations on encrypted data, however they struggle with multi-user inputs.

D. *Multi-prover NI-ZKP*

A multi-prover NI-ZKP [20] allows a set of parties $\{P_1, ..., P_n\}$, each with a private witness w_i, to compute a NI-ZKP in a collaborative way. More specifically, they run an MPC to compute $\pi \leftarrow Prove(st, x, \{w_1, ..., w_n\})$, where no party other than P_i learns w_i.

E. Threshold-key FHE and Multi-key FHE

Threshold-key FHE (Th-FHE) [21] is similar to public key-based FHE, but the secret key sk is secret-shared to a set of holders. The decryption process is computed interactively through an MPC, where $t \leq n$ key shares are needed to recover the plaintext. On its part, in multi-key FHE (Mk-FHE) [22] each party holds a different key pair and decryption is also done interactively using MPC.

V. Discussion

The main difference between blockchain and cloud computing is that the former enables public verifiability, so achieving verifiable confidential computations should be considered. Verifiable computation is still novel, but NI-ZKP and its multi-prover version seem to be useful to add public verifiability to MPC and FHE. As for how confidential computations relate to state, we note that fully on-chain computations [19] are possible, however their difference from publicly available ciphertexts that are computed off-chain lies in additional issues, e.g., control and verification of computation steps (also input commitment and output disclosure). Th-FHE and Mk-FHE, e.g., rely on MPC for output disclosure, and key handling is not straightforward ([19] leads the blockchain to handle the decryption key, so security relies on majority honesty, i.e., FHE is reduced to MPC). On the other hand, in solutions like [15], [16], the on-chain overhead is avoided, but relating computation to state is more difficult and the benefit of replicated storage is lost.

As a summary of this discussion, we could offer an informal definition of what confidential data means in the context of blockchain: *confidential blockchain data is only accessible by designated parties, linked to the blockchain state, and verifiable by publicly available mechanisms in relation to the computation executed*. We note that this is a broad definition, difficult to achieve in its entirety and highly dependent on the specific technologies used.

VI. Conclusions

In this work, we have reviewed the blockchain model with respect to confidential data, and outlined the main lines of research and barriers to bring closer these two scenarios, which seem opposed by design. We have presented the main technologies for blockchain confidential computations (NI-ZKP, MPC, FHE, and some advanced variations), and briefly discussed their pros and cons.

As future work, we focus on providing a formal model for confidential computing in blockchain that gathers the main requirements and links them with the specific enabling technologies. We envision that it will be necessary to combine different cryptographic tools to provide sufficiently secure and usable solutions.

Referencias

[1] A. Alketbi, Q. Nasir, and M. A. Talib, "Blockchain for government services — use cases, security benefits and challenges," in *2018 15th Learning and Technology Conference (L&T)*, 2018, pp. 112–119.

[2] P. Gonczol, P. Katsikouli, L. Herskind, and N. Dragoni, "Blockchain implementations and use cases for supply chains-a survey," *IEEE Access*, vol. 8, pp. 11 856–11 871, 2020.

[3] A. Biryukov and S. Tikhomirov, "Deanonymization and linkability of cryptocurrency transactions based on network analysis," in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2019, pp. 172–184.

[4] M. Reed, P. Syverson, and D. Goldschlag, "Anonymous connections and onion routing," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 482–494, 1998.

[5] L. T. Thibault, T. Sarry, and A. S. Hafid, "Blockchain scaling using rollups: A comprehensive survey," *IEEE Access*, vol. 10, pp. 93 039–93 054, 2022.

[6] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *Journal of Network and Computer Applications*, vol. 126, pp. 45–58, Jan. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804518303485

[7] R. Zhang, R. Xue, and L. Liu, "Security and Privacy on Blockchain," *ACM Computing Surveys*, vol. 52, no. 3, pp. 51:1–51:34, Jul. 2019. [Online]. Available: https://dl.acm.org/doi/10.1145/3316481

[8] G. Almashaqbeh and R. Solomon, "SoK: Privacy-Preserving Computing in the Blockchain Era," 2021, publication info: Published elsewhere. Minor revision. IEEE Euro S&P 2022. [Online]. Available: https://eprint.iacr.org/2021/727

[9] H. Dang, T. T. A. Dinh, D. Loghin, E.-C. Chang, Q. Lin, and B. C. Ooi, "Towards scaling blockchain systems via sharding," in *Proceedings of the 2019 International Conference on Management of Data*, ser. SIGMOD '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 123–140.

[10] E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza, "Succinct non-interactive zero knowledge for a von neumann architecture," Cryptology ePrint Archive, Paper 2013/879, 2013, https://eprint.iacr.org/2013/879.

[11] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, ser. STOC '87. New York, NY, USA: Association for Computing Machinery, 1987, p. 218–229.

[12] V. Goyal, Y. Song, and C. Zhu, "Guaranteed output delivery comes free in honest majority mpc," in *Advances in Cryptology – CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, Proceedings, Part II*. Berlin, Heidelberg: Springer-Verlag, 2020, p. 618–646.

[13] A. R. Choudhuri, A. Goel, M. Green, A. Jain, and G. Kaptchuk, "Fluid MPC: Secure Multiparty Computation with Dynamic Participants," 2020, report Number: 754. [Online]. Available: https://eprint.iacr.org/2020/754

[14] F. Benhamouda, C. Gentry, S. Gorbunov, S. Halevi, H. Krawczyk, C. Lin, T. Rabin, and L. Reyzin, "Can a Public Blockchain Keep a Secret?" in *Theory of Cryptography*, ser. Lecture Notes in Computer Science, R. Pass and K. Pietrzak, Eds. Cham: Springer International Publishing, 2020, pp. 260–290.

[15] M. de Vega, A. Masanto, R. Leslie, A. Yeoh, A. Page, and T. Litre, "Nillion network: Whitepaper," Tech. Rep., 2022. [Online]. Available: https://docsend.com/view/7bkgvzagr6ifhwrc

[16] P. Blockchain, "Partisia blockchain: Whitepaper," Tech. Rep., 2022. [Online]. Available: https://drive.google.com/file/d/1_doKDtMuY1YPDJ8LgKCiOqZvjoYkTmx4/view

[17] C. Gentry, "Computing arbitrary functions of encrypted data," *Commun. ACM*, vol. 53, no. 3, p. 97–105, mar 2010.

[18] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "Tfhe: Fast fully homomorphic encryption over the torus," Cryptology ePrint Archive, Paper 2018/421, 2018, https://eprint.iacr.org/2018/421.

[19] M. Dahl, L. Demir, and L. Tremblay Thibault, "Private smart contracts using homomorphic encryption," 2023.

[20] A. Ozdemir and D. Boneh, "Experimenting with Collaborative zk-SNARKs: Zero-Knowledge Proofs for Distributed Secrets," 2022.

[21] D. Boneh, R. Gennaro, S. Goldfeder, A. Jain, S. Kim, P. M. R. Rasmussen, and A. Sahai, "Threshold Cryptosystems from Threshold Fully Homomorphic Encryption," in *Advances in Cryptology – CRYPTO 2018*, 2018, pp. 565–596.

[22] P. Ananth, A. Jain, Z. Jin, and G. Malavolta, "Multi-key fully-homomorphic encryption in the plain model," Cryptology ePrint Archive, Paper 2020/180, 2020, https://eprint.iacr.org/2020/180. [Online]. Available: https://eprint.iacr.org/2020/180