A Blockchain-based Architecture for Data Certification and Notarization

Giulia Rafaiani, Giacomo Zonneveld, Paolo Santini, Massimo Battaglioni, Franco Chiaraluce, Marco Baldi
Department of Information Engineering
Università Politecnica delle Marche, Ancona (60131), Italy
{g.rafaiani, g.zonneveld, p.santini, m.battaglioni, f.chiaraluce, m.baldi}@univpm.it

The increasing digitization of production processes in multiple sectors has generated an enormous amount of data that represent an essential value for different organizations and institutions. However, ensuring the integrity and certification of such data has become a crucial challenge in the digital age. Unauthorized manipulation of data could compromise the reliability of the information and undermine user trust. This is a crucial issue when we consider specific data that must be authentic and verifiable, such as traceability data or academic certificates. Nowadays, supply chains are extremely complex and this complexity is expected to grow more and more over the years. Therefore, it becomes essential to have traceability systems that follow the entire life cycle of a product, from its origin to the end users. The traceability process consists of several steps, including data identification, acquisition, recording, management and processing, as well as data transmission and communication. Blockchain emerges as an innovative technology for information sharing, ensuring a reliable environment that can be used for certify data integrity. The authors in [1] explored how to integrate blockchain with a distributed file system, aiming to provide a system with the security of a blockchain and the efficiency of a distributed files system. Several works are focused on the use of blockchain technology for food traceability [2]–[4]. Moreover, different approaches for academic certificates management have been proposed [5]–[7].

This work aims to study in a generalized way the problem of certification and data traceability by identifying an architecture that is able to exploit the inherent advantages of distributed ledger technologies. In the architecture we propose, data to be certified can come from different sources, such as embedded devices, mobile apps or web interfaces, as shown in Figure 1. These data of different types (strings, files, pictures, etc.) can be uploaded by users on a dedicated database through some ad-hoc application. The contribution of this work manly lies in the link between the database, containing the information to be certified, and the public blockchain (i.e., Ethereum), in order to immutably notarize data.

We propose two solutions for the considered problem. The first solution is the most straightforward: we include the hash of the data to be certified into an Ethereum transaction. In the second solution, instead, we organize data into one or more Merkle trees and one transaction containing the Merkle root is created for each tree. We then perform a comparative assessment of the two approaches in order to identify an optimal way to use blockchain technology for data certification.

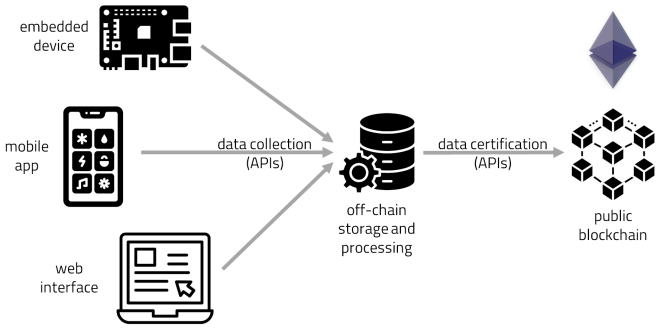

Fig. 1. Block diagram of the designed and implemented blockchain-based certification and notarization system.

I. First Approach: Single Data Certification

The first of the two schemes we propose is the simplest and most intuitive one. Basically, we generate a blockchain transaction that contains the hash of each data to be certified, allowing the information to be immutably notarized on the blockchain. These data (or hashes) are also saved off-chain, together with the corresponding parameters, such as the transaction ID, that are needed to verify data integrity.

The verification phase constitutes a crucial aspect of a certification protocol. In this case, when a user wishes to verify some data, the system takes the data as input, calculates the hash, locates it in the off-chain database, and returns the transaction identifiers, namely the certification timestamp, the transaction hash, the block number, and the hash of the block in the blockchain. By implementing this process, it is possible to invoke a verification function that identifies the appropriate blockchain transaction through these parameters, extracts the associated data field, and compares it with the hash that was locally calculated starting from the data uploaded by the user. If the two values match, the certification is confirmed as valid, attesting the integrity of the data. Conversely, if they do not match, the integrity verification fails, indicating a possible alteration of the information.

II. Second Approach: Multiple Data Certification

In the second approach, we propose the use of Merkle trees as an innovative solution for organizing documents hashes.

The primary goal of this approach is to limit the number of blockchain transactions to be generated, making the certification process more efficient and cost-effective. Compared to the approach analyzed in Section I, which requires a separate transaction for each data to be certified, the Merkle tree allows multiple data to be certified in a single transaction that contains the root of the Merkle tree, thus saving considerable cost and processing time.

The idea is to independently consider the data from different sources and organize them into Merkle trees, whose numerosity (i.e., the number of leaves) is a design choice that takes into account various factors aimed at minimizing costs. To avoid the unnecessary occupation of database space through the redundancy storage associated with storing the Merkle proofs for every data leaf, we opted to store the entire Merkle tree structure. Saving the tree allows the nodes that are part of the proof to be quickly identified, speeding up the local root calculation during the verification phase. The verification phase, indeed, is more complex than in the approach presented in Section I. It requires the local computation of the Merkle root, starting with the document whose certification is to be verified and the corresponding proof. The proof is extracted after locating the correct tree and the position of the considered data, and obtaining the nodes with which the tree is retraced back to the root. When the user has the proof and the data needed to locate the transaction that contains the in-chain root, the actual verification function begins, which includes the following operations:

1) calculate the hash of the file,
2) extract the root from the blockchain,
3) locally calculate the root through the proof and the hash found in step 1),
4) compare the values obtained in steps 2) and 3); in case of equality, the verification is successful.

III. Comparison of the Proposed Approaches

In the above sections we have introduced two approaches to data certification: one based on individual transactions for each data entry and the other based on their organization in Merkle trees. Although both schemes perform the same function, there are significant differences in terms of cost and performance. The parameters considered for the performance evaluation are:

- *CPU usage during the certification process*. The results show that, fixed the amount of data to be certified, there is less CPU usage when Merkle trees are not used. Logically, as the data increases CPU usage increases.
- *Storage to save Merkle trees*. Defining by N the number of data to be certified and by M the number of Merkle trees, it follows that the number of hashes to be saved X is given by $X = (2 \cdot N - M)$. Therefore, for the same N, the total storage is inversely proportional to the number of trees.
- *Execution time required for the certification process*. The biggest factor affecting this parameter is the number of transactions: once N is fixed, the solution with single transactions has a greater execution time than the case with one Merkle tree.
- *Transactions cost*. Using Ethereum, the cost is calculated by multiplying the cost in ether by the value of the ether/€ exchange rate.

Finally, we evaluated the performance of the two approaches during the certification integrity verification phase. In the first approach, the hash of the file is directly compared with the content of the Ethereum transaction. Instead, using Merkle trees, one needs to locally recompute the root from the proof before verifying its integrity; this operation involves the computation of $\log_2 N'$ hashes, where N' denotes the number of leaf nodes per generated tree and is defined as $\frac{N}{M}$.

The results highlight the simplicity of the process regardless of the design choice; the execution time increases as the size of the trees increases, due to the increase in the number of hashes that need to be calculated to obtain the proof. In contrast, in the first method, the number of entries does not affect the process since each data item is certified independently of the others. The approach that makes use of Merkle trees is also efficient during verification, while maintaining an excellent balance between resource consumption and performance considering the entire process.

Conclusion

This work aims to propose methods for data certification and traceability and evaluates their performance to find the most advantageous solution to accomplish such a purpose. The key advantage of organizing data in one or more Merkle trees is to limit the number of transactions to be sent to the blockchain, reducing the costs of transactions, at the expense of an increased storage in the off-chain database and an integrity verification phase that requires more operations. A parametric cost function can be used to identify the optimal solution for the different applications, since there is not an a priori ideal configuration.

More details concerning each one of the above stages will be provided in the presentation.

References

[1] E. Nyaletey, R. M. Parizi, Q. Zhang, and K.-K. R. Choo, "BlockIPFS-blockchain-enabled interplanetary file system for forensic and trusted data traceability," in *2019 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2019, pp. 18–25.

[2] A. Mendi, "Blockchain for Food Tracking," *Electronics*, vol. 11, p. 2491, 2022.

[3] IBM (International Business Machines Corporation), "Food Trust," https://www.ibm.com/blockchain/solutions/food-trust.

[4] M. P. Caro, M. S. Ali, M. Vecchio, and R. Giaffreda, "Blockchain-based traceability in Agri-Food supply chain management: A practical implementation," in *2018 IoT Vertical and Topical Summit on Agriculture-Tuscany (IOT Tuscany)*. IEEE, 2018, pp. 1–4.

[5] "Blockcerts," https://www.blockcerts.org/.

[6] E. F. G. Dias, "Ethereum smart contracts for educational certificates," 2018.

[7] A. Rustemi, F. Dalipi, V. Atanasovski, and A. Risteski, "A systematic literature review on blockchain-based systems for academic certificate verification," *IEEE Access*, 2023.