Blockchain-Based Access Control Supporting Anonymity and Accountability

Gianluca Lax and Antonia Russo
DIIES Dept., University of Reggio Calabria, Reggio Calabria, Italy
Email: {lax, antonia.russo}@unirc.it

***Abstract*—In information security, access control is the selective restriction of access to an online resource or service. One of the most used access control models is Attribute-based Access Control, in which access rights are granted to users by evaluating suitable attributes (user attributes, resource attributes, and environment conditions). An important aspect of access control is to guarantee that the identity of the user accessing a service is preserved. In this paper, we deal with this problem and propose a new solution based on a blockchain to ensure that only authorized users can access a service, yet preserving anonymity and unlinkability of their accesses. Moreover, the cooperation among several trusted parties allows the identification of the user accessing a service in case of need.**

***Index Terms*—identity provider, service provider, un-linkability, authentication, Bitcoin, Ethereum**

I. Introduction

Blockchain has been recently proposed as a solution to several application problems [1], [2]. This emerging technology is a secure storage relying on a distributed consensus protocol able to validate the data added to it [3]. Indeed, blockchain is a distributed and transparent public repository of transactions executed by users and shared among a large number of nodes [4]. Transactions are stored inside a chain only if they are validated by blockchain nodes. Validation is done by a distributed consensus algorithm [5], on which the performance of the blockchain network depends.

Blockchain users create a wallet and are provided with a couple of private and public keys. The private one is used to sign the transactions and aims to guarantee security and authenticity. The public address of a wallet is generated starting from the public key. The users can perform blockchain transactions once they create their wallets [6]. Moreover, users can generate countless blockchain addresses in order to preserve their pseudo-anonymity, which is another important added value of blockchain technology. Generally, a transaction is a transfer of value among blockchain users. Inside a transaction, there is the reference to the recipient's public address and other suitable data, named transaction *payload*. The blockchain technology presents many advantages [7], such as the transparency and immutability of records and the pseudo-anonymity of transactions. These properties could be exploited in access control systems.

Access control systems regulate the accesses to protected resources or operations inside a computer system. The process of access control involves the authentication and the authorization of subjects through a series of security policies in such a way that only the legitimate accesses can take place. The security policies can rely on several security models proposed in the literature [8] and are shared among the different entities of the access control mechanism.

In the context of access control, some significant security research challenges are related to:

- How to guarantee the anonymity of a user accessing an online service supplied by a service provider;
- How to ensure that two different requests of a user to access an online service are not linkable;
- How to disclose the identity of a user who accessed an online service in case of need.

In this paper, we provide an access control scheme that provides the three features above. Our scheme is based on a public blockchain and relies on identity and access control providers. The users who request access to a protected resource supplied by an online service provider are provided with blockchain accounts. Also identity providers, access control providers, and service providers have blockchain accounts, in such a way that all information needed to implement the access control is on the blockchain and publicly available.

Any user exploits different blockchain addresses to interact with identity, access control, or service providers. The used blockchain addresses are linked to each other, and every entity can verify the transactions generated by the users by using the blockchain. Moreover, our solution stores the list of the blockchain addresses used by a user, and each provider involved in the access stores a part of this list (i.e., the link between two addresses adjacent in the list).

Concerning the anonymity, the user reveals his/her identity only to the identity provider, which maintains a mapping between the identity and the blockchain address generated by the user to be identified.

The unlinkability of different requests of the same user is reached by exploiting different blockchain addresses at each interaction with the other entities.

At the same time, we need the guarantee the accountability of the requests. We reached it by linking

Manuscript received May 11, 2020; revised October 9, 2020.

doi: 10.12720/jait.11.4.186-191

the users' blockchain addresses in a verifiable chain locally and partially stored by several parties. In case of need, a party in cooperation with other trusted parties can restore the chain and guess the identity of a user.

The structure of this paper is as follows. In the next section, we survey the access control concept, which is a key topic in our manuscript. The core of the proposal is presented in Section III. In Section IV, we provide a validation of our solution. Related work is discussed in Section V. Finally, we draw our conclusions in Section VI.

II. Access Control

In this section, we introduce some important concepts related to access control as well as the access control models proposed in the literature, which are used in the rest of the paper.

Access control regards the processes carried out to protect users and resources from unauthorized accesses inside any information management systems. A subject is an entity able to access a protected object containing information. An authorized subject is provided with privilege, that is an authorization to carry out some actions on the objects.

In order to develop an access control system, three important abstraction layers have to be taken into account: the security policy, the security model, and the security mechanism [9]. The security policy defines high-level requirements related to the authorization rules that are formally stated in the security model. The security mechanism is the lower layer defining the functions that implement the control policies described in the security model. Many security models have been proposed in the literature to describe security properties in an access control system [8].

The Discretionary Access Control (DAC) is a flexible policy based on the identity of resources' owners. That is, a resource's owner can define the access rules and authorized operations of that resource and modify them anytime. The Access Control List (ACL) is an example of DAC. An access control list defines the authorized operations and the authorized users for every resource. This type of access control is not suitable in our case because we want to guarantee the user's anonymity.

Contrarily to the DAC, the Non-discretionary Access Control techniques (NDAC) rely on established and non-modifiable rules. An example of NDAC is the Mandatory Access Control (MAC). In the mandatory access control, the control policies are released by a central authority, such as the system's administrator, not by the single user able to access a resource.

In the Role-Based Access Control (RBAC), privileges are associated with the roles carried out by subjects. In an organization, a role is made of permissions or responsibilities referred to a subject or group of subjects. Therefore, the definition of roles is the central point of this model.

The Attribute-Based Access Control (ABAC) is defined as an access control methodology where authorization is determined by the possession of attributes associated with the subject, object, and policy or rules. The attributes generally describe these entities and are easily modifiable and verifiable by the authority in charge of releasing them.

It is evident that in some access control models, the identity of the subject is not necessary to gain the authorization for a resource. In the role-based access control, subjects have to demonstrate to perform a specific role, whereas the owned attributes are enough in the attribute-based access control. In our paper, we exploit an attribute-based access control scheme.

In the ABAC model, the entire process of access control can be summarized as follow. When a subject requests access to a protected object, the access control mechanism has to verify that the subject is authorized. That is, the subject possesses the attribute necessary to access the resource. Furthermore, also the object attribute and the environmental conditions (i.e., not related to the subject or object but linked to the environment, such as time and zone) have to be validated. If the conditions are fulfilled, the subjects gain access. Otherwise, the subjects are not authorized for that resource.

III. Our Proposal

In this section, we present the proposed solution: we start by describing the scenario considered in this paper, which is composed of the following actors:

- *Users (U)*, who are physical people whose anonymity in accessing a service should be guaranteed.
- *Identity Providers (IP)*, which create and manage digital identities.
- *Access Control Providers (ACP)*, which are in charge of verifying an access control policy.
- *Service Providers (SP)*, which offer online services only to authorized users.

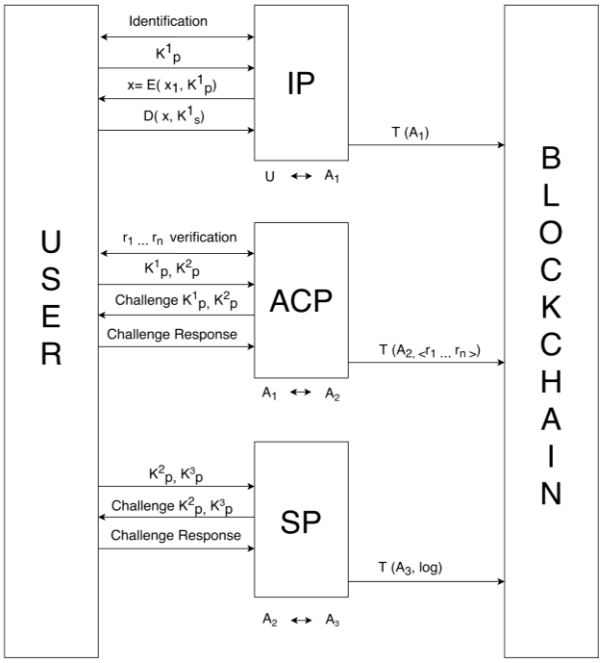

Figure 1. The scheme of our solution.

Now, we describe the protocol allowing us to solve the faced problem. It is schematized in Fig. 1 and is composed of the following phases:

Setup. This phase is used to initialize the environment and to perform some preliminary operations. First, the blockchain to be used is chosen. As we will see, we exploit the basic features of a blockchain (i.e., the distributed repository) so that any blockchain could be used. Although our solution is orthogonal to the used blockchain, for the sake of presentation, we will refer to the Bitcoin Blockchain when need. According to the chosen blockchain, the following two functions are defined: sign(M,k) and verify(S,M,k), where M is a message, k is a cryptographic key, and S is a signature. Specifically, the former returns the signature of the message M by the key k, the latter verifies the validity of the signature S of the message M by the key K and returns true if and only if S is a valid signature. For example, in Bitcoin, these two functions are based on elliptic curves.

All the actors know the chosen blockchain and the defined functions.

Blockchain Account Generation (BAG). Our solution is based on blockchain so that any actor needs an *account* to use the blockchain. This phase, which is carried out at least one time[1] by any actor of the scenario, aims to generate a pair of cryptographic (public and private) keys and a blockchain address. In most of the blockchains, the private key is a randomly generated string with a suitable large number of bits (typically, 256 bits), whereas the public key is obtained by applying a cryptographic function to the private key. Then, the associated blockchain address is obtained by applying a suitable function to the public key as $A = f(K_p)$. Typically, this function is implemented by applying a cryptographic hash function and by keeping the last n bits (e.g., $n = 160$ in Bitcoin).

Each actor joining the system generates a pair of cryptographic keys and obtains a blockchain address. Given an account K, K_s denoted the associated secret key, K_p the public key, and $A = f(K_p)$ the associated address.

User Identification (UI). This phase is carried out by each user to register his/her digital identity from an identity provider *ID*. First, the identity provider verifies the user identity by a recognition in person, via webcam, or online. Then, it collects the surname, name(s), date of birth of the user, and all the other personal data useful for identification.

Now, two (private and public) keys K_s^1 and K_p^1 are randomly generated by the user U, who needs to prove to be the owner of the blockchain address A^1. For this purpose, U sends K_p^1 to the identity provider. The identity provider generates a random x^1 and calculates $x=Encrypt(x^1,K_p^1)$, which is sent to the user. The user calculates $x'=Decrypt(x,K_s^1)$ and sends back this value to the identity provider. In turn, the identity provider verifies that $x' = x_1$, which means that U is the owner of both the public key K_p^1 and the address $A^1 = f(K_p^1)$. In this case, the identity provider generates the *registration* transaction, which is a blockchain transaction from the address of the identity provider to the address A^1, with no further data.

The purpose of this transaction is to store on the blockchain that A^1 is associated with an identified user (it will be clear later that this phase is run each time a user needs a new account). Moreover, the identity provider stores the mapping between A^1 and the personal data of U, collected in the identification step. This mapping is useful for accountability.

Access Control (AC). In this step, the user contacts a suitable access control provider *ACP* to receive a *proof* that he/she satisfies some requirements, say $r_1,...,r_n$. First, *ACP* verifies that the user satisfies the requirements $r_1,...,r_n$, and, as remarked in Section II, we use an attribute-based access control scheme. Then, the user U creates a new account K^2 by using the procedure described in the BAG step. U sends K_p^1 and K_p^2 to *ACP*.

The access control provider checks on the blockchain if there exists a registration transaction for the account K^1 (i.e., a transaction from any identity provider to the address A^1). If this transaction is not found, the procedure alts. Otherwise, *ACP* picks a random x^2 and calculates $a=Encrypt(x^2,K_p^2)$ and $x=Encrypt(a,K_p^{12})$.

Then, *ACP* sends a to the user as a challenge. In response to this challenge, the user calculates $a'=Decrypt(x,K_s^1)$ and $x'=Decrypt(a',K_s^2)$. The latter value is returned to the access control provider, which checks that $x'=x^2$. In the positive case, *ACP* has the proof that U is the owner of both the address A^1 and A^2.

In this case, the *ACP* generates the *authorization* transaction, which is a blockchain transaction from the address of the access control provider to the address A^2, having $r_1,...,r_n$ as payload (which is the data field in a transaction).

This transaction saves on the blockchain the information that A^2 satisfies the requirements is $r_1,...,r_n$. Moreover, the access control provider stores the mapping between A^1 and A^2, which is used for accountability.

SA. Service Access. In this step, the user contacts the service provider to require a service. First, the user creates a new account K^3 as described in the BAG step. To prove to be authorized to this service, the user sends the public keys K_p^2 and K_p^3 to the service provider *SP*. *SP* verifies that the account K_2 satisfies the requirements $r_1,...,r_n$ by searching on the blockchain for an authorization transaction sent to the address $f(K_p^2)$, having in the payload at least $r_1,...,r_n$. If this transaction is not found, the procedure alts. Otherwise, *SP* generates a random x^3 and calculates $a=Encrypt(x^3,K_p^3)$ and $x=Encrypt(a,K_p^2)$. Then, *SP* sent a to the user as a challenge.

In response to the challenge, the user calculates $a'=Decrypt(x,K_s^2)$ and $x'=Decrypt(a',K_s^3)$. The value x' is sent back to the service provider, which can verify that $x' = x^3$. In this case, the service is granted to the user, and a log transaction is generated. This transaction is sent from the address of the service provider to the address $f(K_p^3)$ and has in the payload the log information (typically, it contains the id of the service and the timestamp).

In the next section, we discuss how our solution reaches the expected goals.

IV. VALIDATION

Our solution aims at providing an access control mechanism that guarantees anonymity, unlinkability, and accountability in accessing online services. In our security analysis, we assume the following properties hold:

1) The random generated numbers x_1, x_2, and x_3 are never re-generated and used by any actor. This can be guaranteed provided that the number domain is suitably large (for example, 128-bit random numbers currently satisfy this requirement);
2) The cryptographic primitives Encrypt and Decrypt are robust and cannot be broken. The elliptic curves used in several blockchains (e.g., Bitcoin adopts secp256k1 [10]) currently satisfy this requirement;
3) Private keys are kept secret and cannot be guessed;
4) The information about the mapping of the addresses stored by each entity is not shared or made publicly available;
5) The user discloses personal information only during the Identification Step;
6) Identity Provider, Access Control Provider, and Service Provider do not collude.

Under these assumptions, we show how the expected security properties are guaranteed. We start from anonymity, which requires that the name of the user accessing the service is not given or known from the service provider, the access control provider, or any third party (except the identity provider, which is the entity that knows the user identity by the scheme). Observe that the identity provider publishes the blockchain address of the user so that the user is identified by a public key only. In blockchain this is called pseudo-anonymity and differs from anonymity because an attacker wishing to de-anonymize a user tries to construct the one-to-many mapping between users and public keys [11]. We prevent this attack by storing the association between user and address on the identity provider only. As a consequence, there is no possibility to break pseudo-anonymity and anonymity.

The second property is unlinkability, which means that a user may make multiple uses of services without other parties being able to link these uses together. This is achieved by forcing the user to generate a new account after each iteration. This way, at each iteration, the providers see a new blockchain address that appears randomly generated.

The last requirement is to guarantee accountability, that allows a party in cooperation with other trusted parties to guess the identity of a user in cases of need. The identity of a user who accessed a given service can be guessed as follows. First, from the log transaction of this service, the address A^3 is extracted. Then, the service provider returns the address (say A^2) associated with A^3, by using the locally stored mapping. Now, an authorization transaction to A^2 is searched on the blockchain and let *ACP* the access control provider that generated this transaction. Again, by using the *ACP*'s local mapping, the associated address A^1 is found. A new search for a registration transaction sent to the address A^1 returns the identity provider that identified the user and that can provide the requested information. Observe that all these transactions are found if the protocol has been correctly run by the identity provider, access control provider, and service provider.

V. RELATED WORK

In this section, we survey the most important proposals of the state of the art related to our approach.

In [9], the authors discuss various access control policies already proposed in the literature. Access control is considered as a relevant requirement of any information management system to protect users and resources from unauthorized accesses. Various access control models have been studied to preserve the information protection. The authors of [8] propose a model named T–RBAC and based on the role-based access control.

The name of this model denotes the importance of the task in an enterprise environment where T–RBAC is supposed to be used. The Usage Control is a promising approach to handle the access control process in an information system [12]. This model puts together access control, trust management, and digital rights management for controlling the usage of digital information objects. The proposed solution enables finer-grained control with privacy issues in enterprise and non-enterprise environments. The components involved in the systems are the subjects, objects, and the policies.

The paper [13] provides a definition of Attribute-Based Access Control (ABAC) to understand the real applications of this mechanism. The model is analyzed in real use cases to improve scalability, feasibility, and performances of applications in which the information sharing within and between organizations is expected. In [14], the authors provide a literature review and a taxonomy of the current ABAC models. They highlight the open or unexplored problems, such as the scalability, the delegation, and the suitability of proposed solutions.

The paper [15] deals with the implementation of an anonymous authentication in a decentralized access control scheme in the cloud for secure data storage. In the proposed system, the cloud is in charge of verifying the users' authorization without knowing their identity. Moreover, the access policy for each stored record is managed by the cloud. With respect to our proposal, the use of the cloud reduces the pervasiveness of the solution. In our proposal, the adopted blockchain technology allows us to include in the solution any party provided with a blockchain account.

The authors of [3] propose a new approach to access control based on the blockchain technology. The authorization to access a resource is publicly stored and visible among users that are able to check if policies and resources match. This way, every party can verify the right to access a resource, and this right can be transferred through a blockchain transaction. The approach is validated inside the Bitcoin blockchain and is integrated with the XACML reference architecture. Differently from

our solution, the authors do not solve the problem of anonymity.

The authors of [16] highlight the importance of an anonymity-based authentication and implement a blockchain-based RBAC model that provides role-based access control. The model is simulated on an Ethereum-based through the use of smart contracts, and the authors claim that their technique is more efficient in gas use than the existing RBAC model. Our solution is even more efficient because we do not use smart contracts, so that we do not have gas cost.

Nowadays, the security and privacy issues in the Internet of Things are enormous. The paper [17] presents a distributed access control framework, named FairAccess, which is based on blockchain. The authors exploit blockchain to enforce access policies in distributed environments using smart contracts. In this case, smart contracts need gas to be executed, and this is the price to pay to make authorization decisions. In contrast, our solution does not support a fine-grained access control policy but is more affordable because there is not gas to pay.

It is evident that the security aspects of access control are relevant topics in the literature. Our solution combines the accountability and anonymity requirements with the blockchain technology for the access control of a service delivery. We have shown the advantages of our proposal with respect to the state of the art.

VI. Conclusion

In this paper, we faced some relevant security challenges in the context of access control, by proposing an access control scheme relying on the blockchain technology. For this reason, users who request access to a protected resource supplied by an online service provider are provided with blockchain accounts. Also the other entities involved in the scenario, which are identity providers, access control providers, and service providers, have their blockchain accounts. The aims of our solution is to guarantee the anonymity of a user and the unlinkability of different requests of the same user. This is reached by exploiting different blockchain addresses at each interaction with the other entities. At the same time, we need the guarantee the accountability of the requests. We reached it by linking the users' blockchain addresses in a verifiable chain locally stored in a distributed way by several parties.

In the future, we aim at implementing the validation of blockchain addresses chain through a smart contract running in a Blockchain 2.0. This way, we can maintain a trusted real-time mapping of transactions.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

GL and AR conceived the presented idea and developed the theory. AR wrote the manuscript with support from GL. All authors had approved the final version.

References

[1] F. Buccafurri, G. Lax, L. Musarella, and A. Russo, "Ethereum transactions and smart contracts among secure identities," in *Proc. Distributed Ledger Technology Workshop*, 2019, pp. 5-16.

[2] F. Buccafurri, V. De Angelis, G. Lax, L. Musarella, and A. Russo, "An attribute-based privacy-preserving ethereum solution for service delivery with accountability requirements," in *Proc. the 14th International Conference on Availability, Reliability and Security*, 2019, pp. 1-6.

[3] D. D. F. Maesa, P. Mori, and L. Ricci, "Blockchain based access control," in *Proc. IFIP International Conference on Distributed Applications and Interoperable Systems*, 2017, pp. 206-220.

[4] M. Swan, *Blockchain: Blueprint for a New Economy*, O'Reilly Media, Inc., 2015.

[5] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *CoRR*, vol. abs/1904.04098, pp. 1-34, 2019.

[6] M. Crosby, *et al.*, "Blockchain technology: Beyond bitcoin," *Applied Innovation*, vol. 2, no. 6-10, p. 71, 2016.

[7] Z. Zheng, S. Xie, H. N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352-375, 2018.

[8] S. Oh and S. Park, "Task-role-based access control model," *Information Systems*, vol. 28, no. 6, pp. 533-562, 2003.

[9] P. Samarati and S. C. D. Vimercati, "Access control: Policies, models, and mechanisms," in *International School on Foundations of Security Analysis and Design*, Springer, 2000, pp. 137-196.

[10] H. Mayer, "Ecdsa security in bitcoin and ethereum: A research survey," *CoinFaabrik*, vol. 28, 2016.

[11] F. Reid and M. Harrigan, "An analysis of anonymity in the bitcoin system," in *Security and Privacy in Social Networks*, Springer, 2013, pp. 197-223.

[12] J. Park and R. Sandhu, "Towards usage control models: Beyond traditional access control," in *Proc. the Seventh ACM Symposium on Access Control Models and Technologies*, 2002, pp. 57-64.

[13] V. C. Hu, *et al.*, "Guide to Attribute Based Access Control (ABAC) definition and considerations (draft)," *NIST Special Publication*, vol. 800, no. 162, 2013.

[14] D. Servos and S. L. Osborn, "Current research and open problems in attribute-based access control," *ACM Computing Surveys (CSUR)*, vol. 49, no. 4, pp. 1-45, 2017.

[15] S. Ruj, M. Stojmenovic, and A. Nayak, "Decentralized access control with anonymous authentication of data stored in clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 384-394, 2013.

[16] Y. Lee and K. M. Lee, "Blockchain-based RBAC for user authentication with anonymity," in *Proc. the Conference on Research in Adaptive and Convergent Systems*, 2019, pp. 289-294.

[17] A. Ouaddah, "A blockchain based access control framework for the security and privacy of IoT with strong anonymity unlinkability and intractability guarantees," in *Advances in Computers*, Elsevier, 2019, vol. 115, pp. 211-258.

Gianluca Lax is an Assistant Professor of Computer Science at the University Mediterranea of Reggio Calabria, Italy. In 2005, he received his PhD in computer science from the University of Calabria. In 2013, he got the habilitation as Associate Professor of Computer Science by the Italian National Scientific Qualification Procedure and, in 2018, he got the habilitation as full professor of computer science. His research interests include information security and social network analysis. He is an

author of more than 100 papers published in leading international journals and conference proceedings. He serves as a referee for many international journals and is in the program committee of many conferences. He is also included in the editorial board of several international journals and participates in many funded projects.

Antonia Russo is PhD Student in Computer Science at the University Mediterranea of Reggio Calabria. She received her MsC Degree in Telecommunication Engineering from the University Mediterranea of Reggio Calabria in July 2018. Her research interests include security, privacy, and social network analysis.