# An Architecture for Sharing Cyber-Intelligence Based on Blockchain

Rui Gonçalo[1(✉)], Tiago Pedrosa[2], and Rui Pedro Lopes[2]

[1] Polytechnic Institute of Bragança, Bragança, Portugal
rgoncalo@ipb.pt
[2] Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Bragança, Portugal
{pedrosa,rlopes}@ipb.pt

**Abstract.** Cyber-intelligence sharing can leverage the development and deployment of security plans and teams within organizations, making infrastructures resilient and resistant to cyberattacks.

To be efficient, information sharing should be performed in a trusted environment, ensuring both the integrity, privacy and confidentiality and the truthfulness and usefulness of the information. This paper addresses this issue with the development and deployment of an architecture based on blockchain technology. Each participant is granted a reputation level, that is used to assess and verify the information other actors produce. Each actor, then, is given an amount of credit, corresponding to the number and accuracy of the validation. Information is also organized in topics, instantiated in independent ledgers. The architecture was validated with a three organization scenario, for proof-of-concept.

**Keywords:** Blockchain · Hyperledger Fabric · Cybersecurity · Intelligence sharing

## 1 Introduction

The volume of cyberattacks to organizations grows exponentially everyday, caused not only by the increase of the hackers' creativity but also because some organizations lack the concern or the resources needed to raise proper defences. When it comes to cybersecurity, while the cybersecurity engineer has to plug every leak or vulnerability, the hacker needs only one successful exploit to steal data or to disable a system [1]. One of the key points in cybersecurity is the complete knowledge of the full organization's Information Technology (IT) infrastructure, including the topology, equipment, operating systems and applications, as well as the collection and analysis results of artifacts from network incidents, or from suspicious activity records [2]. These should be used as the starting point for a complete understanding and development of a threat model, used to prioritize the interventions and configuration of the infrastructure [3].

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2020
J. Prieto et al. (Eds.): BLOCKCHAIN 2020, AISC 1238, pp. 71–80, 2020.
https://doi.org/10.1007/978-3-030-52535-4_8

The gathered intelligence can also be useful for future reference and it can be leveraged by others, if shared properly and trustfully.

This paper proposes an environment where communicating entities enter and exchange information with an initial reputation level right from the beginning, with this level increasing with time. The use case for the architecture described herein is the collection and share of cyber-intelligence, used to leverage the organization's security programs, thus encouraging each participant to contribute.

Trust levels are associated with tiers in the infrastructure and each participant is placed in a tier according to the credit it gathers. Each participant is thus classified according to its reputation: the more relevant the intelligence it shares, the more it increases in reputation and, consequently, move up in the tier system. The infrastructure is built on a permissioned blockchain platform, to securely record transactions, and exchange information between one another [4]. The blockchain platform is based on Hyperledger Fabric, a modular and extensible open-source system for deploying and operating permissioned blockchains [5].

The rest of the paper is organized as follows. Section 2 discusses blockchain technology and the consensus model. Section 3 describes the architecture and implementation details. Section 4 presents the test scenario and Sect. 5 the conclusions and discusses future work.

## 2 Background

Blockchain technology has been, since it was announced by Satoshi Nakamoto, in 2008 [6], one of the mainstream scientific and technological research topic in many areas, such as financial, healthcare, insurance, internet of things and many others [7].

In its essence, blockchain is a distributed database that records transactions between parties or digital events that have been executed and shared in the network [8]. It is usually represented as a chain of blocks intrinsically connected to each other (Fig. 1).

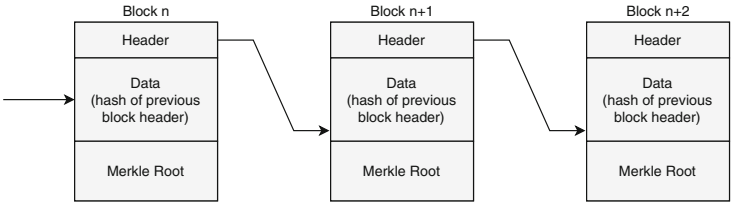


**Fig. 1.** Representation of a blockchain

Each block is cryptographically linked to the previous one, after validated by a consensus decision. As the chain grow, by the addition of new blocks, older blocks become more difficult to modify. Each block also stores a set of records, used to store transactions in a shared ledger. After a successful transaction, the block is added and replicated through the network.

### 2.1 Consensus Models

As mentioned before, the blockchain grows after consensus, which requires a broad agreement between several processes, even in the presence of faults or malicious nodes. This, known as the Byzantine generals dilemma, is characterized by the lack of trust between the participants. This dilemma describes the scenario of the lack of agreement in a group of generals, as to whether to attack or retreat from a siege to a city, and if there is no consensus it all falls apart [9].

Achieving consensus in a distributed system is challenging. Algorithms have to be resilient to failures of nodes, partitioning of the network, message delays, out-of-order and corrupted messages [10].

The research on blockchain technology introduced several consensus models, an essential part of the operation as they ensure consistency and availability of the ledger. One of the most well-known consensus model is called Proof-of-Work (PoW), that consists in using computing power to calculate a hash value that is less than a certain number, referred to as the difficulty level, and thus execute transactions over the network, rewarding a specific individual with the privilege of inserting the next block of transactions in the chain.

This model requires each node to use a considerable computation power, and a few alternatives appeared. One example is consensus by Proof-of-Stake (PoS). This model dictates that the network participants who have a portion or stake of the network value, such as digital coins, receive a proportional allowed mining power, implying that participants with more stake would less likely harm or dominate the network because the cost needed would not be profitable over normal mining activities [9].

Another alternative is the Raft consensus algorithm. Raft implements consensus by first electing a distinguished leader, then giving it complete responsibility for managing the replicated log [11,12]. The existence of a leader simplifies the algorithm. If it fails or become disconnected from the other servers, a new leader has to be elected. Raft, thus, assumes three steps for consensus: leader election (a new leader must be chosen when an existing leader fails), log replication (the leader accepts log entries from clients and replicate them), and safety (if any server has a particular log entry in its state machine, then no other server may apply a different command for the same log index).

### 2.2 Smart Contracts

Smart contracts, or decentralized applications, are programs that are executed on the blockchain, in which their correct execution is enforced by the consensus model. This includes any set of rules that can be represented in the programming language. The executed code, although generic and able to be used in different areas and with different purposes, can assume the role of the middle man, verifying certain criteria before executing a transaction, therefore eliminating negotiation hindrances, such as having a third party to verify if a transaction is valid. This smart contracts are triggered upon an event in the network, which means that they are self-aware upon events and automatically execute an action,

written with rules agreed before hand by all participants, resulting in an practically unbreakable contract [7].

The implementation of permissioned blockchains usually requires to assemble a set of technologies in a common platform. For that, existing platforms and implementations may be used, saving the burden of implementing all the details of a blockchain operation.

### 2.3 Platforms and Implementations

There are several factors to consider in the selection of the best blockchain platform, such as development maturity, programming languages, consensus model, popularity and support, scalability, and smart contracts support.

One of the most popular platforms is the Ethereum platform and consists in a peer-to-peer network of virtual machines, that can be used to deploy and develop applications, although, it is more comfortable to support application programs based on rules and criteria [13]. Ethereum currently uses the PoW consensus model to reward the mining activity to the network participants. Furthermore, Ethereum also allows anyone to develop applications within the platform, as a permissionless system.

Along with permissionless blockchain platforms there are also permissioned blockchain implementations. One of this implementations is Ripple, which is a payment dedicated network that makes currency transactions rapidly. Ripple operates under the Ripple Protocol Consensus Algorithm (RPCA). This means that nodes have a list on which other nodes they trust rather than to accept any global assumption.

Another permissioned blockchain platform is the Hyperledger Fabric. This platform is modular since it allows for pluggable consensus models, although, raft is currently the default option. Usually, in the distributed ledger technology, consensus is considered to be a single algorithm applied to one part of the transaction flow. However, in Fabric, the consensus concept comprehends not only the algorithm itself but also most of the transaction process, from the transaction proposal until the transaction is verified and submitted into the ledger.

## 3 An Architecture for Sharing Information

The objective of this paper is to present an architecture for the exchange of reliable and meaningful intelligence in a partially trusted environment. A permissioned blockchain will be used as a communication mechanism to secure content and reward the participants that share information by increasing its reputation level.

The exchange of information relies on the concept of a channel, where each channel is specific for a context or topic. This is similar to message queues, although the content is not removed when read. Moreover, in addition to support the communication of content, the system also supports the validation of the content, performed by trustful peers (Fig. 2).

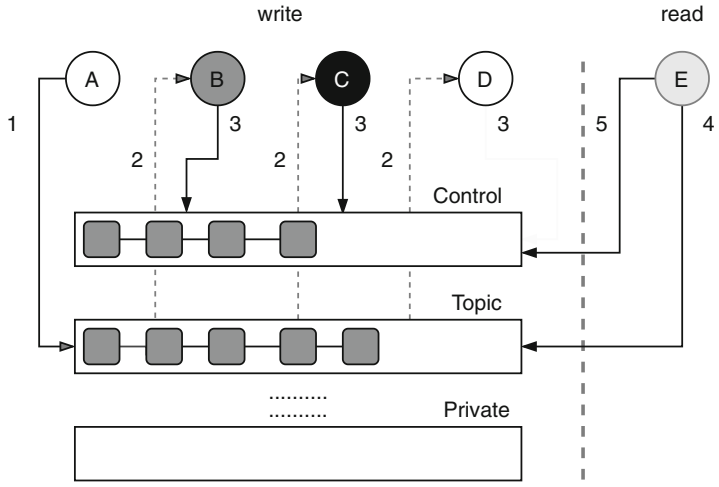


**Fig. 2.** Channel architecture flow

Although this approach relies on some tools and mechanisms used in some studies and articles [14], what makes it diverge from the rest is that its essence is based on a tier system. In the beginning of the network all organizations assume the same basic level, from which they are able to climb, by submitting reliable intelligence.

Each organization is assigned a reputation level, among five possibilities: *Others*, *Vendors*, *Sec. Researchers*, *CSIRT*, and *ICSIRT* (International CSIRT). The higher the reputation level, the more trustful is the information it asserts. The level depends on the credit and reputation that each peer has managed to gather. In Fig. 2, nodes are numbered from A to E, and the color represents the reputation level: from white (*Others*) to black (*ICSIRT*). Specifically, an entity placed in the *Others* level has lower reputation and lower credit value to sign or evaluate an asset committed to the channel by another entity than a entity placed one tier above (e.g., *Sec. Researchers*). Furthermore an entity which earns a certain level of reputation moves up in the tier system, notifying the rest of the network that it is an entity that shares and collects reliable intelligence.

When the system first starts, there is only a single channel, responsible for the coordination between peers. This channel, designated Control channel, is accessible to all the peers and used for membership management, asserting content and checking reputation. The Topic channel corresponds to a specific context, such as "Hardening" or "Vulnerabilities". A peer starts by placing information in a specific topic (1), which executes a transaction in the ledger. This starts the execution of smart contracts in all the remaining peers (2), according to the policies defined in the system. Each smart contract will then check the content and assert its validity (3). When a peer wants to read, it checks the topic (4) and asserts its validity by checking the control channel (5).

Within the network, it is possible to create more channels (Fig. 3). In a stationary situation, peers are constantly interacting with the control channel and, eventually, specific topic channels (1). When necessary, a participant can deploy a new channel by sending a request to the Platform (2). This will then handle the deployment of the genesis block for the channel intended (3) and communication can start in the new channel (4).

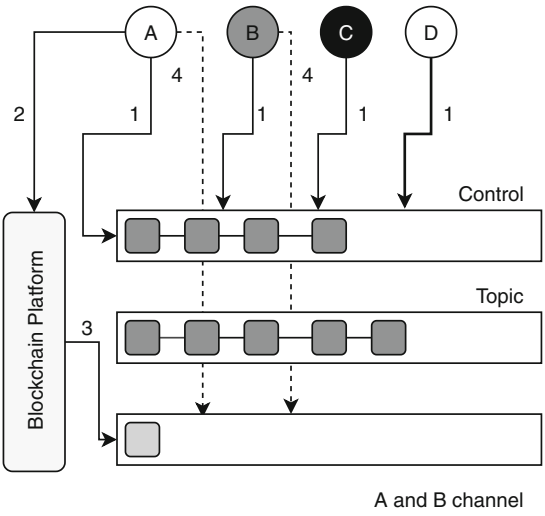


**Fig. 3.** Deployment of a private channel

These channels can be used in two perspectives. They can be a topic channel, which then originates an open channel for all the participants who want to share intelligence, i.e., a channel dedicated to the submission of vulnerabilities. On the other hand it can be used as a private channel, for confidentiality purposes, between two or more organizations. The creation and deployment process occurs in the same way in both cases, although the channel configurations like policies and chaincode may differ.

It is not imperative for the proper functioning of the system that entities join in private channels. These private channels are new instances of a ledger, that can hold new chaincode and policies but, in a more theoretical sense, they are channels of communication seen only by those who have access to it. They are a kind of secret rooms inside a open house where only few who want to do business in the "dark" know its whereabouts.

### 3.1 Implementation Details

The implementation and validation of the architecture relied on Hyperledger Fabric, where its unique features, fit perfectly into the main requirements of this

work. Furthermore, being a highly modular platform, allows developers to adapt the solution for their needs. Fabric does not impose the use of tokens or digital currency to work. Moreover is an open source platform, actively developed and with a strong and active community.

Transactions are key in Fabric and are of the responsibility of the ordering service. A transaction starts with a proposal, sent from the client and received by the peer component in the network. The peer then simulates the execution of that transaction against the current version of the values in the state or world state of the ledger (Fig. 4). The result of this simulation is a *read/write* set. Since all peers are synchronized, they should execute the proposal with the same values and obtain the same result. The peer now signs the transaction and sends it back to the client in a endorsement response format along with cryptographic materials.

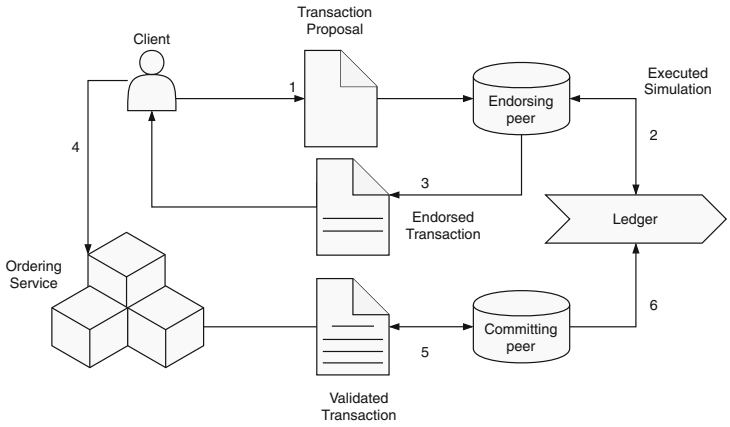


**Fig. 4.** Fabric consensus in transaction lifecycle

The client sends a invocation request to the ordering service onward with the endorsed response that will be analyzed against the policies defined in the network. If the transaction does not meet the established criteria, it will not be classified as valid and it will not update the state of the ledger, rather, it will be stored in the blockchain part of the ledger. However, if it meets the policies and if all endorsement responses from the peers have the same result, it is assigned as a valid transaction and submitted to the ledger to update the world state.

The endorsement policies are also a key component, by determining if a transaction is valid or not. In the most common endorsement policies it is usually specified that for a transaction to be valid it must be endorsed or signed by a specific set of peers or by the majority of the peers [5]. Only designated administrators from the network have the permission to update or modify this policies that are agreed by all network participants.

## 4 Test Scenario

The Fabric platform uses docker containers as the baseline for their components deployment. Therefore, it was also necessary to implement docker orchestration tools. To this end, docker-compose was used, mainly for simplicity.

The test scenario included the development of three organizations ("orgA" - CSIRT, "orgB" - Sec. Researcher, and "orgC" - ICSIRT), were each peer was deployed in a different host, simulating three real world entities and an additional node for the ordering service. In addition, three channels where also created: the control channel, where entities reflect their intention to join the network and create their reputation level; one private channel between the organizations "orgA" and "orgB"; one private channel between organizations "orgA" and "orgC".

The chaincode used to create the tier profile for each organization is deployed only in the control channel. Therefore, entities can only execute queries to obtain other member reputation level in that specific channel.

The first test was used to implement a specific channel for the exchange of the PGP public key fingerprint of each organization. The logic behind this choice is that because PGP public keys are widely used in email encryption and signing, and so trust in this information is paramount. Furthermore, in future services additions to the system, the use of the PGP public key fingerprint will be imperative to execute and interact in those topics, as a way for identity verification.

Peers on both private channels could not execute nor see the chaincode for the reputation profile creation, proofing that chaincode can only be called in the specific channel that is deployed. For testing purposes simple chaincode samples were deployed in the private channels. These created two variables to hold integer values and make simple value transactions between the two variables.

The private channels were not visible for the specific organization that did not had permission, e.g., "orgB" could not see nor make any kind of connection to the private channel between "orgA" and "orgC", and vice versa.

Participants for each organization were able to create their tier profile, give credit to one another for the submission of their PGP public key fingerprint, make simple values transactions executing the chaincode in each private channel and finally, they were also able to request to move up in the tier system once the reputation level was enough to do so.

## 5 Conclusion and Future Work

The work described in this paper describes an architecture for the storage and exchange of cyber-intelligence using a blockchain. For this, Hyperledger Fabric was used, with the Raft consensus model. Each actor in the communication is granted a reputation level, that increases with the amount of credit received from the other actors. Also, each actor is required to validate the information it finds in the channels and, the higher the reputation level, the higher the trust on the information. This gives an indication to the other actors of the overall quality

and correctness of the information present in the channels. The system rewards actors that contribute for the network, with access to privileged intelligence.

The channels are implemented as a distributed ledger, based on blockchain. Each actor can be allowed to access a channel and can also be allowed to create new channels, that can either be public or private.

For proof-of-concept, a test scenario with three organizations was implemented. It successfully demonstrated the architecture for the exchange and validation of information. The initial architecture is still considerably complex to assemble, and this is an issue that should be addressed in the future, along with the choice for the format to share the intelligence used in the system. Since there is no world accepted standard this is still a object in study, although all directions point to STIX 2.0.

Another problem lies in encouraging entities to sign information submitted by the other participants. This can become a problem because if information is not being signed the actor's reputation cannot increase, resulting in a tier stagnation.

## References

1. Bavisi, S.: Penetration testing. In: Managing Information Security, pp. 177–200 (2013). Elsevier. https://doi.org/10.1016/B978-0-12-416688-2.00007-6. https://linkinghub.elsevier.com/retrieve/pii/B9780124166882000076. Accessed 27 Jan 2020. ISBN: 978-0-12-416688-2
2. Nurse, J.R.C., Creese, S., Goldsmith, M., Lamberts, K.: Guidelines for usable cybersecurity: past and present. In: 2011 Third International Workshop on Cyberspace Safety and Security (CSS), ISSN: null, 2011 September, pp. 21–26. https://doi.org/10.1109/CSS.2011.6058566
3. Marback, A., Do, H., He, K., Kondamarri, S., Xu, D.: A threat model-based approach to security testing. Softw.: Pract. Exp. **43**(2), 241–258 (2013). https://doi.org/10.1002/spe.2111. http://doi.wiley.com/10.1002/spe.2111. Accessed 27 Jan 2020. ISSN: 00380644
4. Vukolić, M.: Rethinking permissioned blockchains. In: Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts - BCC 2017, pp. 3–7. ACM Press, Abu Dhabi (2017). https://doi.org/10.1145/3055518.3055526. http://dl.acm.org/citation.cfm?doid=3055518.3055526. Accessed 28 Jan 2020. ISBN: 978-1-4503-4974-1
5. Androulaki, E., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolić, M., Barger, A., Cocco, S.W., Yellick, J., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference on - EuroSys 2018, pp. 1–15. ACM Press, Porto (2018). https://doi.org/10.1145/3190508.3190538. http://dl.acm.org/citation.cfm?doid=3190508.3190538. Accessed 28 Jan 2020. ISBN: 978-1-4503-5584-1
6. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008)

7. Mohanta, B.K., Panda, S.S., Jena, D.: An overview of smart contract and use cases in blockchain technology. In: 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–4, July 2018. IEEE, Bangalore. https://doi.org/10.1109/ICCCNT.2018.8494045. https://ieeexplore.ieee.org/document/8494045/. Accessed 28 Jan 2020. ISBN: 978-1-5386-4430-0
8. Yaga, D., Mell, P., Roby, N., Scarfone, K.: Blockchain technology overview. National Institute of Standards and Technology, Gaithersburg, MD, Technical report NIST IR 8202, October 2018, NIST IR 8202. https://doi.org/10.6028/NIST.IR.8202. https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf. Accessed 02 Jan 2020
9. Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H.: An overview of blockchain technology: architecture, consensus, and future trends, pp. 557–564 (2017). https://doi.org/10.1109/BigDataCongress.2017.85
10. Fischer, M.J.: The consensus problem in unreliable distributed systems (a brief survey). In: Karpinski, M. (ed.) Foundations of Computation Theory. Lecture Notes in Computer Science, pp. 127–140. Springer, Berlin (1983). https://doi.org/10.1007/3-540-12689-9_99. ISBN: 978-3-540-38682-7
11. Ongaro, D., Ousterhout, J.: In search of an understandable consensus algorithm, p. 18 (2014)
12. Howard, H., Schwarzkopf, M., Madhavapeddy, A., Crowcroft, J.: Raft reoated: do we have consensus? In: ACM SIGOPS Operating Systems Review, vol. 49, no. 1, pp. 12–21, January 2015. https://doi.org/10.1145/2723872.2723876. http://dl.acm.org/citation.cfm?doid=2723872.2723876. 02 May 2020. ISSN: 01635980
13. Buterin, V.: A next generation smart contract & decentralized application platform. Technical report, p. 36 (2013). https://www.weusecoins.com/assets/pdf/library/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf
14. Homan, D., Shiel, I., Thorpe, C.: A new network model for cyber threat intelligence sharing using blockchain technology. In: 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), June 2019, pp. 1–6. https://doi.org/10.1109/NTMS.2019.8763853. ISSN: 2157-4960