Universidad Carlos III de Madrid
www.uc3m.es

Máster Universitario Ciberseguridad

2015-2016

Trabajo Fin de Máster

Shareinchain

Jaime Morales Rodríguez de Lope

Tutor/es

Sergio Pastrana Portillo

Madrid, 01 de julio de 2016

Shareinchain: An Anonymous Information Sharing Protocol using Blockchain

Jaime Morales Rodriguez de Lope[a], Sergio Pastrana[b]

[a]*Student of Master in Cybersecurity, University Carlos III de Madrid, Leganes, Spain*
[b]*Computer Security Lab, University Carlos III de Madrid, Leganes, Spain*

Abstract

Privacy and reputation loss are two of the main reasons stated by actual companies to refuse sharing information about cybersecurity incidents. However, both, the academy and industry consider information sharing and cooperativeness between parties a main strategic concept to fight current cyberthreats. Indeed, cyberattacks may harm all companies, but if they unify efforts to fight these attacks, the impact could be severely limited, thus increasing their safety. Accordingly, it is needed a communication protocol allowing information sharing by preserving privacy and anonymity of the sharing entities.

In this Master Thesis, we propose a protocol, dubbed Shareinchain, that uses the Blockchain to store and manage the information shared by different entities. Blockchain is an emerging technology that consists of a chain of blocks where transactions are stored. These transactions allow for metadata used in Shareinchain to store the shared information. In order to deal with selfish entities (i.e., those entities that are not willing to cooperate but want to access the information), the proposed protocol contains two chains, one of them has the encrypted information and is publicly available to all the partners, whereas the other chain is of private access and contains the decrypted information. On this way, companies are forced to share information in the public chain if they want to be authorized to access the information stored in the private chain. One single administrator, acting as a server, is in charge of the control and management of the system. This administrator can access all the information, but it cannot reveal who is sharing it, thus preserving the anonymity and privacy of the different partners.

Keywords: Threat management, Information Sharing, Blockchain, Anonymity, Distributed Ledger

Email addresses: 100277258@alumnos.uc3m.es (Jaime Morales Rodriguez de Lope), spastran@inf.uc3m.es (Sergio Pastrana)

1. Introduction

Nowadays companies receive a big amount of cyberattacks on a daily basis, which causes serious harms at different levels. Normally, most of the attacks can be prevented using cyberdefense systems such as firewalls, Intrusion Detection Systems, antivirus, etc. These cyberdefences are managed by companies and should be configured and patched properly to hinder new attacks (e.g. zero day exploits). Indeed, despite the efforts made by governments and organizations, many of the attacks successfully bypass the security barriers, thus causing serious impact on the companies' assets. Normally, after an attack has happened, an incident response team investigates the causes and consequences, usually leading to updates in the current cyberdefences (e.g. addition of entries in IP blacklisting, IDS rules, malware signatures databases, etc.). Unfortunately, in order to avoid losing reputation and to prevent losing confidence by its consumers, companies usually refuse to share information and lessons learned about attacks. This poses a disadvantage to other companies that cannot use information about the attack to update its defences and protect their assets. What many companies does not take into account is that, in many scenarios, sharing information can help themselves from future attacks. Indeed, a recent study showed that sharing information can be very helpful, mostly in scenarios where entities have functional dependencies from each others [1].

One way to incentive companies to share information is to provide mechanisms to allow for anonymous information sharing that preserves their privacy, where every company can share information about received attacks without damaging its reputation [2], [3]. On this way, participating entities could help each other, creating a symbiosis. Despite being anonymous, such mechanisms must fulfill other three properties. First, it must keep the integrity of the data shared, i.e. information must be real and should not be modified. Second, it must be scalable by allowing new entities to publicly access the information. Third, it has to be just in the sense that all companies must benefit from each others. The last feature is particularly important in order to prevent free-riders who benefit from the information shared by others without cooperating.

In this work we propose an anonymous communication protocol that allows different entities or users to share information. We have named this protocol Shareinchain. While we motivate our protocol in the field of cybersecurity information sharing, Shareinchain could be used to share any kind of information provided that it can be encoded properly, as we discuss later in this document. The key aspect of the protocol is to share information using Blockchain to resolve this problem.

The use of the Blockchain protocol has several benefits for the problem at stake. The Blockchain protocol is a chain of blocks which keeps a registry of all the transactions that have been realized in BitCoin. This protocol offers the possibility of adding metadata in the transactions and this is precisely the feature used by our protocol to share information. In summary, Blockchain inherently offers the following characteristics:

1. Anonymous: Each user of the chain has an address which is generated through a hash and ensures the anonymity in transactions.
2. Integrity: Blockchain is a chain of blocks where transaction integrity depends on previous ones. As such, the information cannot be forged unless the whole chain is corrupted.
3. Metadata: Each transaction allows the addition of metadata (encoded in hexadecimal), which allows for the addition of information.
4. Public: Every participant has access to the chain and can check the details of each transaction, such as the quantity of coins transacted or the metadata included within every transaction.

Using Blockchain, different entities (e.g. companies, organizations, etc.) are able to share information without revealing their identity, thus preserving their anonymity and reputation. Our protocol relies on a administrator user which acts as manager of the sharing community. This administrator has access to all the information gathered, but still he cannot identify the source entities sharing the information. For example, this administrator could be a government entity (e.g. from the health sector) and the different users or entities companies related with this administrator (e.g. hospitals, private clinics, etc.)

Given that the intrinsic characteristics of Blockchain solve the problems related to anonymity, integrity and scalability, the main purpose of Shareinchain proposed in this work is to encourage companies to share information and to avoid free-riders. Accordingly, participating entities will not just benefit from the information shared by others, but also they must share information in order to access the information. This way, Shareinchain is designed to ensure that only participating entities that actively share information are able to access information shared by others.

For such a purpose, the protocol relies on two chains. A public chain, where each entity uploads information encrypted using a combination of symmetric and public key encryption schemes, a semi-public chain, where the information is replicated unencrypted and whose access is managed by the administrator, which only grants access to those entities that actively participate and cooperate by sharing information. Section 3.4 provides further details on how authorizations to access this chain is granted and how trust is managed by the administrator.

The remainder of this document is structured as follows. In Section 2, Background and related work, we talk about the related work as well as applications that have been developed.

In section 3, Shareinchain is described and in a generic way we speak about the system and the algorithm that the protocol uses.

Section 4, Implementation, we talk about the developed client for users and administrator. In addition, we explain the necessary characteristics to use the library Multichain.

In section 5, Results, we make an analysis of the time which the server takes on different themes, in addition we make a performance testing and generation of results.

In section 6, Conclusion, we make a personal opinion explaining the reasons for the work.

In section 7, References, links and documents that have obtained the information shown.

In section 8, Anex: Planning and Budget, the planning and costs inherent are shown through a Diagram of Gant and several graphics.

2. Background and related work

In this work, we have adapted the Blockchain protocol to the problem of cybersecurity information sharing, where different entities want to share information anonymously. Blockchain has become popular because it is the core of the Bitcoin payment system. As such, in this section, first we provide a description of bitcoin (Section 2.1) and Blockchain (Section 2.2), followed by a description of works where Blockchain has been used besides bitcoin (Section 2.3). Finally, in Section 2.5 we provide a brief background on information sharing from the cybersecurity community, which is an increasing topic of research.

2.1. Bitcoin

Bitcoin is a decentralized payment system invented by Satoshi Nakamoto in 2008 [4]. The system is peer to peer and transactions take place between users directly, i.e. there is not intermediaries. Bitcoin is a Proof-of-Work (PoW) based on crypto-currency that allows users to "mine" for digital coins by performing computations. Users digitally sign their transactions and are prevented from double spending their coins through a distributed time-stamping service. This service operates on top of the Peer to Peer (P2) network that ensures that all transactions and their order of execution are available to all Bitcoin users [5].

Bitcoin has different characteristics which make that it was a innovative idea. The main characteristics come from Blockchain protocol 2.2 for that reason we talk about it in the next section. Bitcoins are the coins that Blockchain uses for transactions, peculiarity, only there are 21 million of coins so anybody can have more coins.

Notably, each transaction makes more difficult the mining 1 for that reason is necessary a computational performance of mining. Basically, miners keep the Blockchain consistent and verify the transactions for each block. Each block contains a cryptographic hash of the previous block, using the SHA-256 hashing algorithm, which "chains" it to the previous block thus giving the block chain its name.

2.2. Blockchain

Blockchain [6] is the protocol where Bitcoin is based on. Each transaction completes a block which is mined, after a new block is added to Blockchain in a lineal and chronological order. On this way, every block forms the chain. The Blockchain has complete information about the addresses and their balances right from the genesis block to the most recently completed block.

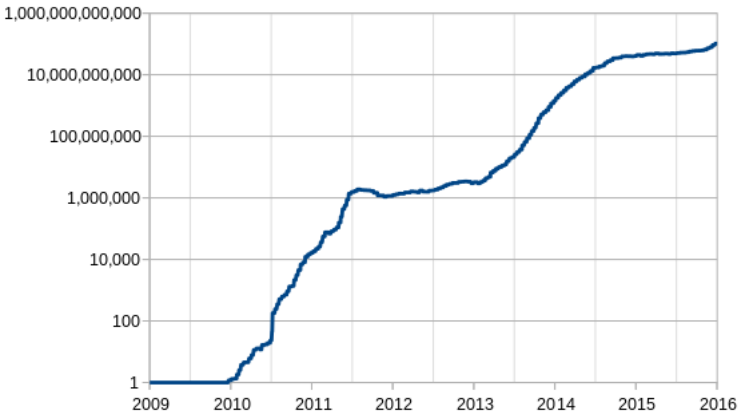

Figure 1: Difficulty Mine

Blockchain technology [7] has many utilities, it lets decentralized currencies, smart contracts and intelligent assets for example smart property. Blockchain could improve the governance systems with more democratic or participatory decision-making. These applications could complete different fields such as communications, business or even policies or laws.

Blockchain technology represents the next step in the peer-to-peer economy. By combining peer-to-peer networks, cryptographic algorithms, distributed data storage and a decentralized consensus mechanism, it provides a way for people to agree on a particular state of affairs and record that agreement in a secure and verifiable manner.

A Blockchain is simply a chronological database of transactions recorded by a network of computers. Each Blockchain is encrypted and organized into smaller datasets referred to as "blocks". Every block contains information about a certain number of transactions, a reference to the preceding block in the Blockchain, as well as an answer to a complex mathematical puzzle, which is used to validate the data associated with that block. A copy of the Blockchain is stored on every computer in the network and these computers periodically synchronize to make sure that all of them have the same shared database.

To ensure that only legitimate transactions are recorded into a Blockchain, the network confirms that new transactions are valid and do not invalidate former transactions. A new block of data will be appended to the end of the Blockchain only after the computers on the network reach consensus as to the validity of the transaction. The consensus within the network is achieved through different voting mechanisms, the most common of which is Proof of Work, which depends on the amount of processing power donated to the network. Figure 2 represents a graphical image of the Blockchain.

After a block has been added to the Blockchain, it can no longer be deleted and the transactions it contains can be accessed and verified by everyone on the network. It becomes a permanent record that all of the computers on the

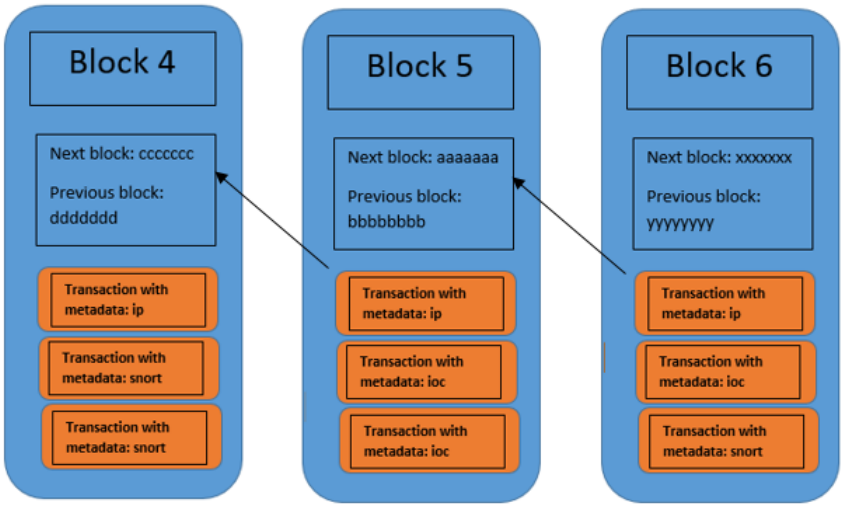

Figure 2: Structure of Chain

network can use to coordinate an action or verify an event.

2.3. Uses of Blockchain for other purposes

Blockchain can have other purposes, some of the most known are:

- DLT. Distributed Ledger Technology [8] is based on the transactions and ownership changes are made and verified by cryptography. DLT let make transactions between two parts without external agents. In addition, DLT can facilitate the registration of ownerships and custody of assets and it would facilitate the supervision by the competent authorities. This faster clearing and settlement results in a reduction of counterparty risk and lower need for collateral for a sales transaction. These possible benefits show that it is a more efficient technology in terms of costs.

  This new technology, like everything, has the risk of cyberattack or the risk of fraud or money laundering if no controls are established suitable. They can also increase the operational risks arising from the full automation. In addition, market volatility may be enhanced by interconnection through the network. Finally, a noteworthy aspect of the DLT is that it could lead to the accumulation of risks in the segments less regulated markets, creating new areas pockets of risks) in the financial markets.

- Digital notary. One of the possibilities is the digital notary. This concept opens a new world, people would not need pay for a notary person and people could avoid the corruption saving the document integrity. In addition, Blockchain could be used in:

  - Mortgages.
  - Inheritances.
  - Certification of funding
  - Digital democracy.
  - Crowdfunding.

- Financial sector. Banks are thinking about the importance that BitCoin can have in the future, the problem is the identity and the most of the transactions are related to illegal themes like drugs or weapons.

  Although the motives of Bitcoin are negative, it is true that the Blockchain concept is really interesting for banks. In fact, there are banks like Santander or BBVA that are investing in it. Banks are cognoscente of the potential which Blockchain could have and the saving management cost.

  Santander bank says that they could save 20 billion in cost or if people used BitCoin would not have a crisis because nobody could have stolen.

  Banks could substitute SWIFT network for this new protocol, in fact, Nasdaq is investing in this technology and for this reason we can assert that it has an important value.

- Connection IoT: There are several Blockchain applications for IoT and smart systems [9]. When you apply to Internet of Things, the concept of Blockchain can open up the possibilities of new innovations. Blockchain technology can be used in tracking the history of individual devices. It can enable the processing of transactions and coordination between devices that are involved. The technology will make the IoT devices independent by keeping a record of the ledger of data exchanges between the device and other devices, services and human users and by enabling them to conduct transactions.

  Various established technology firms and startups have been exploring multiple ways of using these use cases. They are investing and extensively researching into the various possible solutions that the technology could be leveraged into. The main aim of these use-cases is to link the home network to the cloud and electrical devices nearby (home automation).

- Voting, Authorization, Authentication. An increasing number of organizations [10] and political parties have proposed the creation of a Blockchain-based system to build a fairer and more transparent voting environment. In 2014 the Danish political party, Liberal Alliance, proposed using the technology for e-voting.

- Ethereum: Is a decentralized platform [11] that runs smart contracts: applications that run exactly as programmed without any possibility of downtime, censorship, fraud or third party interference.

  These apps run on a custom built Blockchain, an enormously powerful shared global infrastructure that can move value around and represent

the ownership of property. This enables developers to create markets, store registries of debts or promises, move funds in accordance with instructions given long in the past (like a will or a futures contract) and many other things that have not been invented yet, all without a middle man or counterparty risk.

The project was crowdfunded during August 2014 by fans all around the world. It is developed by the Ethereum Foundation, a Swiss nonprofit, with contributions from great minds across the globe.

2.4. Security and problems of the Blockchain

It is necessary include a part in this work talking about the Blockchain security and the problems that it can have.

- Scalability: The bandwidth that allows Blockchain are only 7 transfers per second while others like VISA currently performed some 56.000 transfers per second. Still, it is considering expanding it.
- Permissions: Anyone can read and write, this combined with no identity makes banks may not like the idea.
- 51% Attack: The fact that someone would control more than half of the nodes that mine transfers cause it had control of the chain may be able to fake transactions, but it is really complicated as it shows the Figure 3 Different groups of miners.
- Privacy information: There is no possibility of privacy, really it is not a fail because Blockchain doesn't offer this capability, but this is the reason why banks don't want to invest more than they are doing now. However, there are works which are studying in this to relation the identity with the Blockchain address.
- Reputational question: It is not willingly that while legal transfers are made, someone is buying illegal issues.
- Software bugs: Blockchain software has had several bugs, they have been resolved but if banks change their technology and a bug appears it would be a big problem.
- 25% Attack: This attack what it says is that 25% of miners are controlled by the same person and when a block comes this group mines it before the rest of the honest miners which would always be one step in front.
- Hawk: A framework [6] for building privacy preserving smart contracts. "A non-specialist programmer can easily write a Hawk program without having to implement any cryptography. Hawk compiler is in charge of compiling the program to a cryptographic protocol between the Blockchain and the users". As show in Figure 4.

  The program contains two parts [6]:

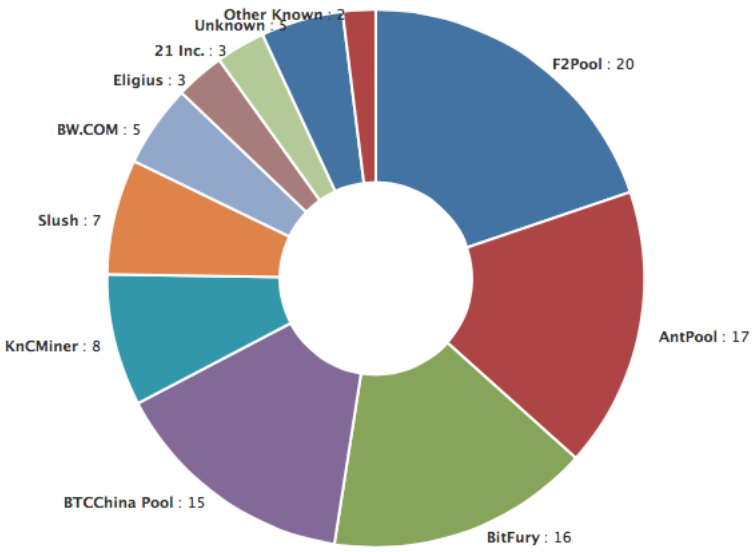

Figure 3: Groups Mining

1. A private party which takes input data as well as currency units. "This part performs computation to determine the payout distribution amongst the parties."
2. A public part does not touch private data or money.

2.5. Info sharing

Regarding the sharing of information and cooperation in the field of cybersecurity, there are studies which analyse and defence cooperation mechanisms for collaborative security with the goal of making systems more resilient and efficient security and identify challenges in designing such systems[12]. On the other hand, in line with the complexity of the environments in which you try to put into practice, there is a high number of concepts to consider when information exchange is cybersecurity. In found a good summary of the related issues [13] and where we can draw some key concepts for effective ecosystem of information exchange, to note:

- The importance of thinking in an environment where everyone wins (win-win environment).
- The reputation and image as an asset to be protected.
- Understand the relevant information as an asset of value.

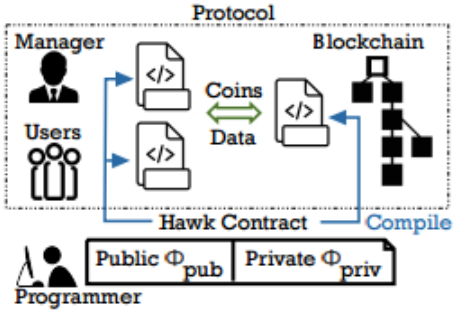

Figure 4: Hawk

- Build a system based on trust.
- Try to maintain the confidentiality of the information.

3. The protocol Shareinchain

This section presents the details of the proposed protocol. The protocol manages two separate Blockchains. The first one is public and, i.e. everybody can connect to share the information through metadata (as shown in Figure 5) in the chain (as shown in Figure 2 in Section 2.2). The second one is semi-private since only trusted users are allowed to connect. In our work, we consider a trust node as an entity that cooperates by sharing information regularly, without disrupting the regular operation of the protocol. Section 3.4 provides more details about how trust is managed.

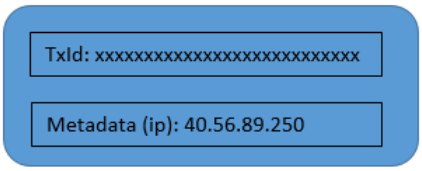

Figure 5: Structure of Transaction

A key difference between the two chains is in how metadata is stored. In the public chain, the metadata is encrypted with the public key of the administrator, so only the administrator can access to the information. Meanwhile, in the private chain the metadata is stored in cleartext to allow authorized (trusted) nodes to get the information that has been shared by the community. See Figure 6.

	Public Chain	Private Chain
Connect	✓	✓
Send	✓	✗
Receive	✓	✗
Issue	✓	✗
Mine	✗	✗
Admin	✗	✗
Metadata size	8MB	8MB

Figure 6: Compare Public and Private Chain

Before starting the protocol the administrator must configure the chain with some required settings. First, the set of allowed IPs that are going to have permissions to connect through Java RPC client as shown in Figures 7, 8. Second, the administrator must set the parameters of the public and private chains. The most important are:

- Connect: This parameter means if anyone can connect or not, so before starting the chain, the administrator can set this parameter (true or false).
- Send: This parameter means if anyone can send or not. If the administrator set this parameter to true, everybody can send transactions.
- Receive: This parameter means if anyone can receive transactions or not. On this way, the administrator can create a public or private chain.
- Issue: Anyone can issue new assets. This parameter is set to false if the administrator wants that only he can create new assets.
- Mine: Anyone can mine blocks. This parameter means if everybody can mine blocks, in our case only the administrator can mine the blocks.
- Admin: Anyone can perform administrator actions. In our chain, only the administrator is the unique administrator.
- Metadata size: Configure size the metadata. There is a parameter where you can configure the size of metadata, in this case, the administrator put the biggest one.

Once the chain is configured, the protocol is ready to start. As a public distributed ledger, the chain keeps every transaction done by the nodes. Both, the IP of the public chain and the address of the administrator must be public to allow connection to all interested users that want to join the sharing community.

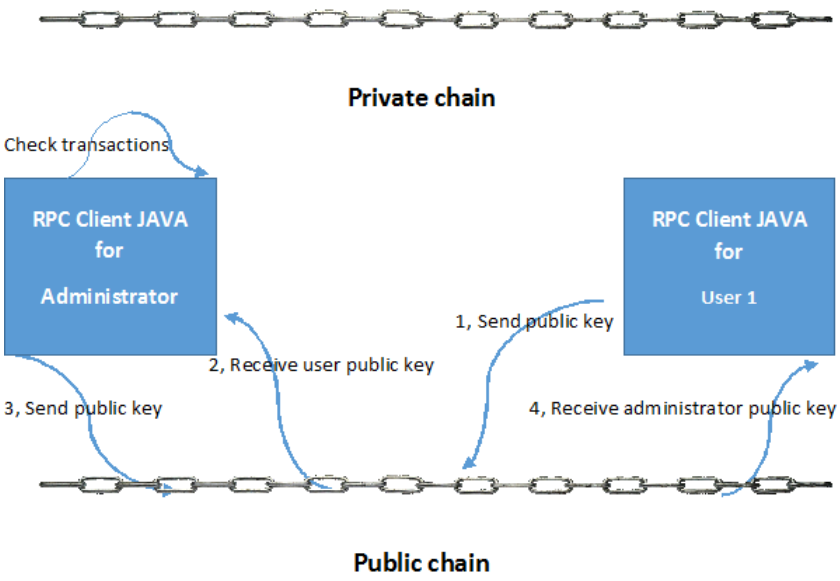

Figure 7: Start Protocol

3.1. Initialization

The initialization scheme is shown in Algorithm .

Initialization Entity U_i wants to join the sharing community

1: U_i: Creates an asset $joined_i$ with 2 coin.
2: U_i: Transacts in the public chain
$(joined_i, 1, [K_{pub}^{U_i}]) \rightarrow Ad$
3: Ad: Upon processing the transaction, prepares λ_{ij} coins for each asset A_k, with $k = 1...j$
4: Ad: Transacts in the public chain
$(A_k, \lambda_{ij}, [\tau]) \rightarrow U_i$
5: Ad: Transacts in the public chain its public key
$(joined_i, 1, [K_{pub}^{Ad}]) \rightarrow U_i$

In order to start the communication, a user or node (we use these terms indistinguishably) has to connect to the chain and, since the system is distributed, the database is replicated. However, this node will have restricted permissions on the chain as we have explained above.

Upon connection, the new node creates an asset whose name is $joined_j$, having two coins. Then, in order to send its public key to the administrator, it

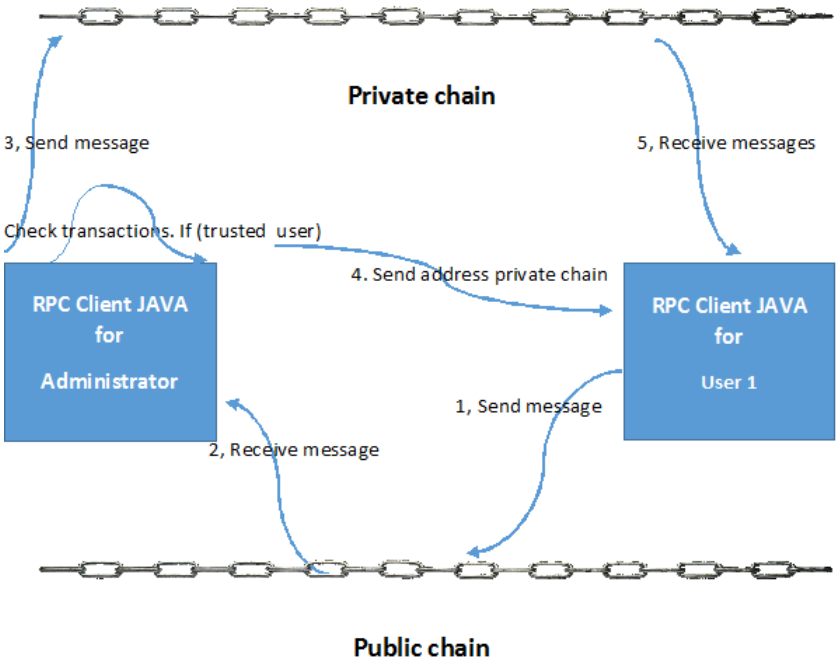

Figure 8: Share Info

performs a transaction to the administrator (whose address is public) into the public chain using the two coins and integrating the public key as metadata of the transaction. Additionally, it indicates the symmetric encryption algorithm to be used. Then, the administrator runs a process which periodically checks if there are new nodes connected to the public chain by looking at transactions done using new asset $joined_j$.

The administrator answers to the new joined node by transacting back one coin of the $joined_j$ asset and including the public key of the administrator as metadata. Additionally, the administrator creates the set of assets related to the topics defined for which information may be shared. For example, if the community were intended to share three topics of information (e.g., malicious IP addresses, IDS rules and leaked accounts), then the administrator will create three different assets (e.g. IP, IDS_rules and $passwd$). For each asset, the administrator transacts an initial, fixed quantity to the new user (λ_{ij}). The purpose of providing a fixed quantity is to control how many pieces of information are being shared by each user, which is important to decide which users are cooperating and which may be free-riders. Thus, whenever a user shares a piece of information, it must do so by transacting a single coin of the corresponding asset, as explained in next section.

3.2. Information sharing

The information sharing scheme is shown in Algorithm .

Data sharing Entity U_i wants to share information I_k about asset A_k

1: U_i: Creates a random session key K_s.
2: U_i: Prepares and encrypts the information I_k with K_s
$data = E_{K_s}(I_k)$
3: U_i: Transacts 1 coin of the asset A_k including as metadata the encrypted info and the encrypted session key
$(A_k, 1, [data, E_{K^{Ad}_{pub}}(K_s)]) \rightarrow Ad$

Once user are connected to the public chain and the initialization protocol has finished, the user is ready to shared information. For such a purpose, it must chose the asset corresponding to the topics of the information by looking at the assets provided by the administrator in the initialization phase. Then, it transacts one coin of such asset including the information as the metadata. This information is encrypted using a symmetric encryption algorithm with a unique session key (this algorithm was established during the initialization phase) and the session key is protected by an RSA encryption using the public key of the administrator. This encryption prevents all users but the administrator to access the information. In summary, the steps to share a message are:

1. Generate random key session.
2. Encrypt the message with the symmetric encryption algorithm selected during the initialization and the session key.
3. Encrypt the session key with public key administrator using RSA.
4. Send a transaction to the administrator over the asset corresponding to the topic, using just one coin and including the encrypted message as metadata.

Only transactions made using just one coin are considered valid. Thus, given that each piece of information increases the quantity of coins for this asset by one, by looking at its currency on this asset the administrator can know how much information is being shared by each user on each asset. This information is used during the trust management phase as explained later.

3.3. Info management

The scheme of the information management carried out by the Administrator is shown in Algorithm .

Info management The administrator periodically lists transactions of the public chain

1: **for all** T_i in *transactions* **do**
2: *Ad*: decrypts *data* and gets info I_k about asset A_k.
3: *Ad*: verifies I_k and the quantity q of the transactions.
4: **if** isFalse(I_k) OR $q > 1$ **then**
5: *Ad*: Obtains who the cheater is (U_c)
$U_c \leftarrow get\_issuer(T_i)$
6: *Ad*: penalizes U_c (transacts β coins for asset A_k)
$(A_k, \beta, [\emptyset]) \rightarrow U_c$
7: **if** exceeds τ AND U_c is in *trustedEntities* **then**
8: *Ad*: creates a new private chain and excludes U_c
9: *Ad*: notify all other from *trustedEntitites* with the new chain address
10: *Ad*: create and transact an *obsolete* into the old chain
11: *Ad*: removes U_c from *trustedEntitites*
12: U_c: must re-transact 1 coin of the asset $joined_c$
13: **end if**
14: **else**
15: *Ad*: Copies info I_k into private chain
16: **end if**
17: **end for**

The administrator periodically lists the transactions of the public chain that has been done on each of the assets regarding topics of information (e.g., transactions on assets IP, IDS_rules and *passwd*) For each transaction, the following steps are performed:

1. Obtain the metadata and decrypt the information.
2. Verify the quantity of the transaction. If the quantity is bigger than one coin, the administrator penalizes the user by transacting back the double amount of coins to the user and continues with the next transaction.
3. Verify that the information is true.
4. In case, that the previous steps are correct (i.e., the information is true and just one coin has been transacted), then the administrator sends the information to the private chain (by performing a transaction and including the information in clear text as metadata).

With this approach, all nodes that have access to the private chain can list the transactions and access the information. Thus, in order to avoid free-riders and motivate users to share information, the access to this chain must be restricted only to those nodes that have actively cooperated, i.e., that have share information to the community.

3.4. Trust management

The scheme of the trust management carried out by the Administrator is shown in Algorithm .

To determine which users are able to access the private chain, the proposed protocol uses a node confidence measured by the amount of information shared by this user. When such amount reaches a threshold for any user, then the administrator sends the private chain address to this user. This is a key part since the use of such threshold motivate users to share information and to behave properly, i.e. according to the protocol specifications. That means that if they do not meet the protocol requirements they will be penalized. The reasons by which users can be penalized are:

- Make transactions with more than one coin. The threshold is checked through the amount of coins on the different assets held by the administrator, so if the user share information with more than one coin it could reach the threshold sharing less information. As explained above, the administrator checks the amount of the transactions and if there is an invalid amount, the user will be penalized doubling the amount of the transaction.
- Once the user has access to the chain with the clear-text information, it must continue to share information, otherwise it will be punished. Thus, periodically the administrator checks if users share information and it penalizes them if during a long period they are inactive. This penalization is done by transacting back one coin of each asset. The node stops being trusted node if the administrator penalizes a lot (it is removed from the set of trusted nodes). In this case, the administrator will replicate the private chain into a new one and it will inform all the users but the one which has been penalized. This causes that the user will no longer have access to the updated information that may be shared.
- Share false information. Though checking that information is not fake is out of the scope of this project, we assume that some kind of information verification process can be done [14].

Trust management Users that share information are allowed to read the private chain

---

1: **for all** A_k in *assets* **do**
2: *Ad*: checks remaining quantity λ_{ik} for all U_i
3: **if** any $\lambda_{ik} \leq \tau$ **then**
4: $trustedEntities \leftarrow U_i$
5: *Ad*: Creates a random session key K_s.
6: *Ad*: Prepares and encrypts *address* of the private chain with K_s
$data = E_{K_s}(I_k)$
7: *Ad*: Transacts 1 coin of the asset $joined_i$ including as metadata the encrypted address and the encrypted session key
$(joined_i, 1, [data, E_{K^{U_i}_{pub}}(K_s)]) \rightarrow U_i$
8: **end if**
9: **end for**

---

4. Implementation

During this project, we have implemented a prototype of this protocol using a library called MultiChain [15]. Multichain allows to create chains and configure the parameters to read, write, create assets, manage or undermine. The prototype is based on the following features:

- Users and administrators are implemented as virtual machines. The operative system used is Ubuntu Server 14.04. We chose this because two main reasons. First, because MultiChain is only available for UNIX based systems. Second, the use of a server environment, free of graphical interface, reduces the memory consumption and allows to run several instances simultaneously.
- All the participating entities (i.e. users and administrators) generate a RSA key pair with at least 1024 bits.
- The logic of the users is managed by a RPC client developed in Java using the MultiChain library. The methods that have been developed in Java are in this reference [16].
- The logic of the administrator is managed by a RPC administrator developed in Java MultiChain [16].

The public chain is configured considering that the new users can connect to the chain, send transactions, receive them and create an issue. There is another file that must be configured in the multichain configuration file (multichain.conf) in order to permit the connection of a RPC Client. Concretely, this file holds the following variables:

- rpcuser: the name of the user to connect to the server.
- rpcpassword: the password of the user.
- rpcallowip: the ips that allowed for the connection.
- rpcport: the port for the connection.

As explained above, the administrator and every user need a key pair. In our implementation we use the library java.security to generate these pairs. The use of this library allows the user to decide the algorithm type and key size. The administrator by default uses RSA and the key size is 1024. In addition, the protocol uses a symmetric encryption. The protocol generates a random key which is used to encrypt the message using a symmetric algorithm and this key is encrypted with the administrator public key. On this way, only the administrator can read the message and send it to the other chain.

4.1. RCP client for users

The RPC client for users is an application developed in Java which connect with the chain to participate like anonymous user. To start the communication it is necessary to change the password and the name user in the program, this is a configurable parameter as explained above. This application provides 4 options:

- Join community: This is the first option and allows the user to join the community. First, a new key pair is created if needed. Then, it implements the user behaviour during the initialization phase of the protocol. Once it has joined the community, it waits for the reception of the administrator's public key.
- Share info: Once it has joined, the user can share the information that he only knows. First, he has to select the asset regarding the topic about he wants to share and after that he writes the message. This message is encrypted and only the administrator will be able to read it.
- Query info: When the user is considered trustworthy, he can read the shared messages for all the users. If it is still not trusted, his access will be denied and he will obtain a message indicating that he does not have that available option.
- Exit: When he wants to go out of the application he has an available option.

4.2. RCP client for administrator

RPC client for the administrator is a different application. In this program, the administrator does not have any interaction and it runs continuously in background. Periodically, the program performs the following processes:

- Get all assets: This method gets all the assets of the chain.
- Get new transactions: This method get all new transactions which have been done.
- Create assets: If there are new assets from the initialization phase, it means that there are new users. These new assets will have the name joined and the number of the user. In this case, the administrator has to do:
  - Check new users.
  - Create assets to the new users.
  - Send his public key. The administrator sends the public key, so the user will be able to encrypt the messages.
- Check metadata: Check the new transactions regarding the assets with the topics of interest and process the metadata.

- Trust management: Each transaction with metadata is checked. If there is a transaction with a quantity bigger than 1 coin or the metadata is not true, the administrator will penalize to the user. In case of everything is correct, the administrator perform next step.
- Send information to the other chain.

5. Results

We have conducted a series of performance tests to get results about the protocol. These tests measure the time overhead incurred by different steps of the protocol. The characteristics of the machine where we have run the experiments are the follows one:

- Processor: Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz.
- RAM: 8GB.
- Operating System: Windows 8.1.
- Type of operating system: 64 bits.

The tests have been separated in two parts: Server time and User time.

5.1. Server time

One of the critical measures to allow the protocol to scale is the time taken by the server to do all steps. To measure this time we have tested three steps. One cycle is the time that the application takes to check everything.

1. Creation of a new user. This is the step that more time consume, because the server has to save the user's public key, create the different assets (e.g. ioc, snort, IP) and send back its public key. In addition, the server has to check that every new asset is mined. In our experiments, we have seen that the time consumed in average to process a new user is 14.706,16ms for 40 cycles.
2. Processing of a new message. When an anonymous user sends a message, the server has to decrypt the message and replicate it in the private chain. The time consumed in average for this case is: 2.695,5ms for 40 cycles.
3. Cycle without incidents. If the cycle is without incidents is because nobody has sent anything. In this case, the server only checks each of the previous steps to look for new users or new messages. If none of this applies, it takes on average 808.45ms to complete the processing.

Overall, we have seen that the server takes 3.345,5 ms on average to perform all the checks for each lap, as shown in Figure 9.

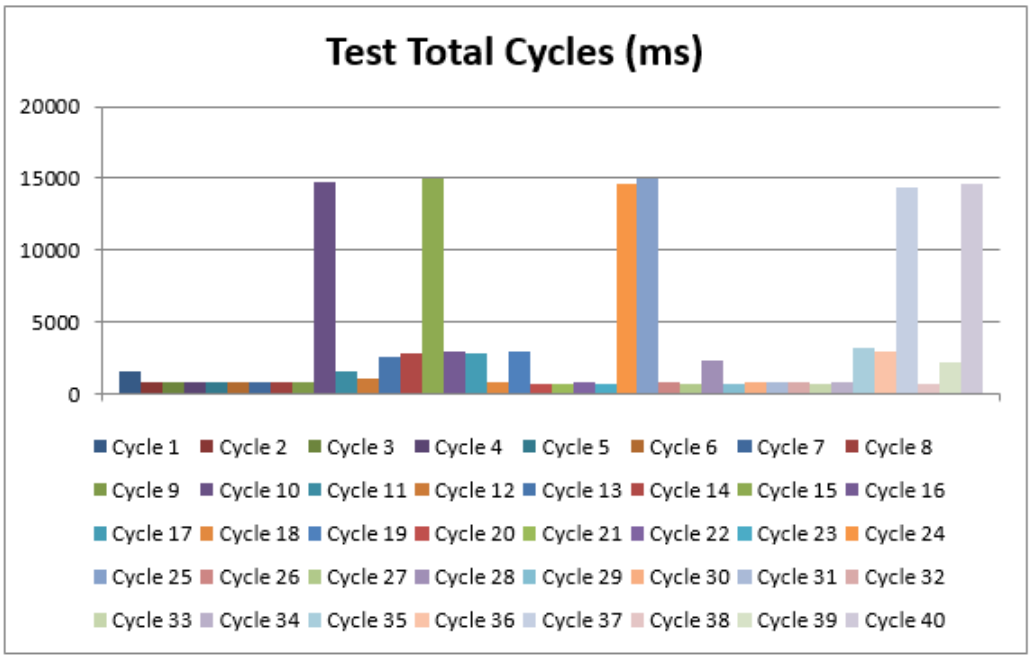

Figure 9: Test Total Cycles

5.2. User Time

For the user time, there are four different steps where we have measure the time.

- Joining to the community: The user has to create a new asset to start the protocol, then to waits for the mining of the asset and finally he generates the pair of keys and sends the public key. The user is finally joined to the community when the server sends back its public key, so the user has to wait for it. The total average is 28.641,83ms, as showed in Figure 10[1], but the time consumed by each step are:
  - Generate pairs of keys and send public(if they are no pairs of keys): 616ms for 6 new users.
  - Send public key (if pair of keys already existed): 277ms for 6 new users.
  - Issue an asset: 273ms for 6 new users.

[1]For the sake of illustration and to denote the difference between data, we only show in Figure 10 the time below 27.000.

– Waiting server public key: 15789ms for 6 new users.

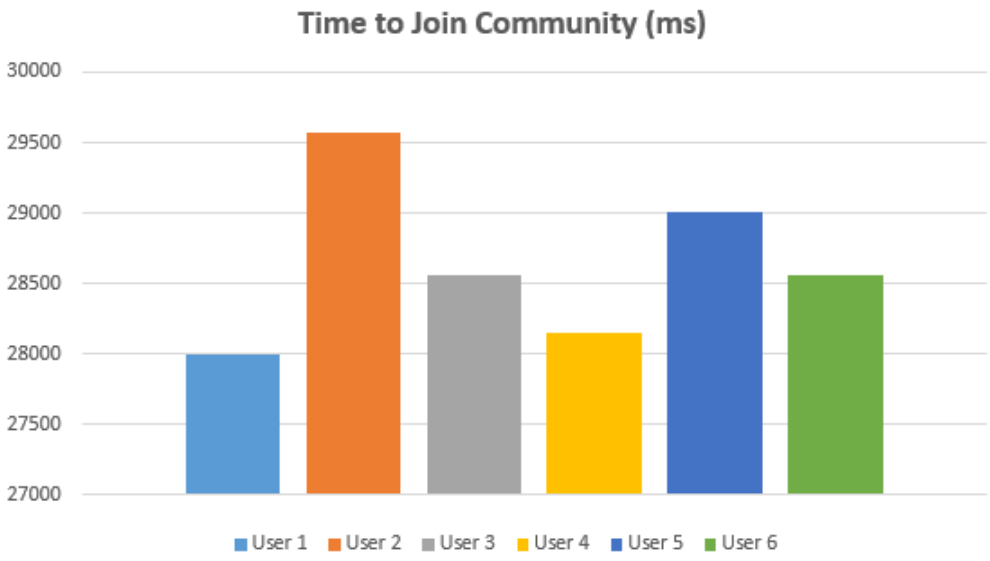

Figure 10: Time to Join Community

6. Conclusion

Nowadays cyberattacks are becoming more frequent and cause serious harm to public and private organizations. Information sharing is projected as a strategic concept to fight these attacks. However, there are still a lack of mechanisms to incentive information sharing. Companies refuse to share information from their attacks mainly due to privacy concerns and to avoid losing reputation. Reputation is the idea or concept that people have about a person or thing. In case of businesses, loss of reputation involves loss of money and if a company loses its reputation is very difficult to win it back. This problem suggest the need for a protocol to share details about attacks anonymously. In this work we have proposed Shareinchain. This protocol relies on Blockchain as a distributed ledger to store the information shared. However, due to the problem of free-riders, we propose the use of two chains with restricted permissions so as to avoid those entities that do not share information to access the information shared by others. This both incentives and motivates entities to share details about incidents and cyberattacks. The use of Blockchain allows Shareinchain to maintain anonymity and privacy of the participating entities, integrity and non-repudiation of the information shared.

7. References

[1] R. Garrido, Modelo de propagación de ciberataques e intercambio de información de ciberseguridad, University Carlos III de Madrid, 2016.

[2] E. Ajayi, The impact of cyber crimes on global trade and commerce, Available at SSRN.

[3] N. Wilding, Cyber resilience how important is your reputation? how effective are your people?, Business Information Review 33 (2) (2016) 94–99.

[4] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system (2008).

[5] G. Karame, E. Androulaki, S. Capkun, Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin., IACR Cryptology ePrint Archive 2012 (2012) 248.

[6] A. Kosba, A. Miller, E. Shi, Z. Wen, C. Papamanthou, Hawk: The blockchain model of cryptography and privacy-preserving smart contracts, Tech. rep., Cryptology ePrint Archive, Report 2015/675, 2015. http://eprint. iacr. org (2015).

[7] A. Wright, P. De Filippi, Decentralized blockchain technology and the rise of lex cryptographia, Available at SSRN 2580664.

[8] M. Walport, Distributed ledger technology: Beyond blockchain, UK Government Office for Science, Tech. Rep 19.

[9] K. Christidis, M. Devetsikiotis, Blockchains and smart contracts for the internet of things.

[10] J. Young, Voting Systems, `http://bravenewcoin.com/news/factom-dreams-of-a-new-generation-of-blockchain-based-voting-systems/`, [Online; accessed 07-June-2016] (2015).

[11] G. Wood, Ethereum: A secure decentralised generalised transaction ledger, Ethereum Project Yellow Paper.

[12] G. Meng, Y. Liu, J. Zhang, A. Pokluda, R. Boutaba, Collaborative security: A survey and taxonomy, ACM Computing Surveys (CSUR) 48 (1) (2015) 1.

[13] H. Borchert, It takes two to tango: Public-private information management to advance critical infrastructure protection, Eur. J. Risk Reg. 6 (2015) 208.

[14] M.-C. Boudreau, D. Gefen, D. W. Straub, Validation in information systems research: a state-of-the-art assessment, MIS quarterly (2001) 1–16.

[15] D. G. Greenspan, Multichain, `http://www.multichain.com/download/MultiChain-White-Paper.pdf`, [Online; accessed 07-June-2016] (2013).

[16] D. G. Greenspan, Multichain, `http://www.multichain.com/getting-started/`, [Online; accessed 07-June-2016] (2013).

8. ANEX: Planning and Budget

In order to comply with the regulations of MsC Thesis by University Carlos III, this Anex presents the Planning and Budget of the Thesis.

First, it is going to be presented a planning about this project defining the different tasks that have been carried out. To do this, it has been used a tool called GanttProject, which generates the typical Gantt chart presenting in a graphical form the duration of each of the activities. This project has been divided into tasks, which are showed in Figure 11 with its start and end date. Gantt chart with detailed information is shown in Figure 11 too.

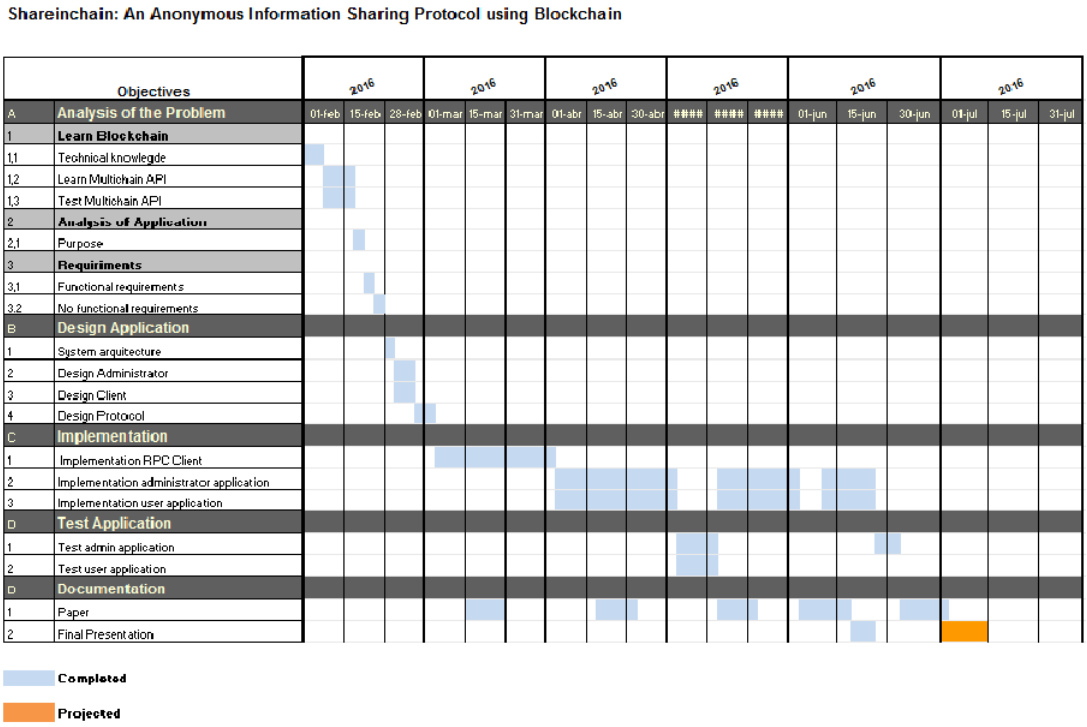

Figure 11: Diagram of Gantt

The budget is going to be broken down into various estimations, Personal cost (see Figure 12); Hardware cost (see Figure 13); Indirect cost (see Figure 14); Total cost (see Figure 15). All estimations are in last page.

Person	Price/hour	Dedication	Cost of phase (€)
Analyst	1.900,00	1.7	3.230,00
Designer	1.650,00	2.3	3.795,00
Engineer	2.150,00	3	6.450,00
Engineer	1.100,00	1	1.100,00
Documenter	1.350,00	1	1350,00
		Total	15.925,00

Figure 12: Personal Cost

Element	Quantity	Cost (€)	Subtotal (€)
Computer	1	750,00	750,00
Laptop	1	650,00	650,00
	Total		1.400,00
	Total amortized		116,67

Figure 13: Hardware Cost

Description	Company	Cost (€)
Internet	Telefonica	430,00
Gasoline	Lecrer	80,00
Print documents	Bookshop	45,00
	Total	555,00

Figure 14: Indirect Cost

CONCEPT	COST (€)
Personal cost	15.925,00
Amortized	117,00
Indirect cost	555,00
TOTAL	16.597,00

Figure 15: Total Cost

UNIVERSIDAD CARLOS III DE MADRID
Escuela Politécnica Superior

PRESUPUESTO DE PROYECTO

1.- Autor:
Jaime Morales Rodríguez de Lope

2.- Departamento:
IT Department

3.- Descripción del Proyecto:

- Titulo: An Anonymous Information Sharing Protocol using Blockchain
- Duración (meses): 5 meses
- Tasa de costes Indirectos: 20%

4.- Presupuesto total del Proyecto (valores en Euros):
Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar - solo a titulo informativo)	Categoría	Dedicación (hombres mes) [a]	Coste hombre mes	Coste (Euro)	Firma de conformidad
Jaime Morales Rodríguez de Lope		Analista	1,7	1.900,00	3.230,00	
Jaime Morales Rodríguez de Lope		Diseñador	2,3	1.650,00	3.795,00	
Jaime Morales Rodríguez de Lope		Ingeniero Seni	3	2.150,00	6.450,00	
Jaime Morales Rodríguez de Lope		Ingeniero	1	1.100,00	1.100,00	
Jaime Morales Rodríguez de Lope		Documentació	1	1.350,00	1.350,00	
		Hombres mes 9		**Total**	15.925,00	

[a] 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable [d]
Ordenador de sobremesa	750,00	100	5	60	62,50
Portátil	650,00	100	5	60	54,17
		100		60	0,00
		100		60	0,00
		100		60	0,00
					0,00
				Total	116,67

[d] Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \text{xCxD}$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado
B = periodo de depreciación (60 meses)
C = coste del equipo (sin IVA)
D = % del uso que se dedica al proyecto (habitualmente 100%)

SUBCONTRATACIÓN DE TAREAS

Descripción	Empresa	Coste imputable
Impresión documentos	Copistería	45,00
	Total	45,00

OTROS COSTES DIRECTOS DEL PROYECTO[e]

Descripción	Empresa	Costes imputable
Servicio de Internet	Telefónica	430,00
Gasolina	Lecrer	80,00
	Total	510,00

[e] Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y

6.- Resumen de costes

Presupuesto Costes Totales	**Presupuesto Costes Totales**
Personal	15.925
Amortización	117
Subcontratación de tareas	45
Costes de funcionamiento	510
Total	16.597