Blockchain to Improve Security and Knowledge in Inter-Agent Communication and Collaboration over a Restrict Domains of the Internet Infrastructure

Juliao Braga[1,2], Joao Nuno Silva[2], Patricia Takako Endo[3,4], Jessica Ribas[1], Nizam Omar[1]

[1]Universidade Presbiteriana Mackenzie (UPM)

[2]IST - INESC ID, University of Lisboa, Portugal

[3]Universidade de Pernambuco (UPE), Brazil

[4]Dublin City University (DCU), Ireland

{juliao.braga,joao.n.silva}@tecnico.ulisboa.pt, patricia.endo@upe.br

jessica.ribas@mackenzista.com.br, nizam.omar@mackenzie.br

Abstract. *This paper describes the deployment and implementation of a blockchain to improve the security, knowledge and intelligence during the inter-agent communication and collaboration processes in restrict domains of the Internet Infrastructure. It is a work that proposes the application of a blockchain, platform independent, on a particular model of agents, but that can be used in similar proposals, once the results on the specific model were satisfactory.*

Resumo. *Este documento descreve o desenvolvimento e implementação de uma blockchain para melhorar a segurança, o conhecimento e a inteligência durante os processos de comunicação e colaboração entre agentes em domínios restritos da Infraestrutura da Internet. É um trabalho que propõe a aplicação de uma "blockchain", independente de plataforma, em um modelo particular de agentes, mas que pode ser utilizado em propostas similares, uma vez que os resultados no modelo específico foram satisfatórios.*

1. Introduction

Autonomous System (AS) is the name given to the networks making up the Internet [Hawkinson and Bates 1996]. ASes establish interconnections through a protocol called *Border Gateway Protocol* (BGP) [Rekhter et al. 2006]. BGP is a complex protocol that requires a lot of knowledge from the administrators of an AS. Sometimes the human being also forgets to update information, especially those related to routing policy and that reside on important servers such as *Internet Routing Registry*[1] (IRR), for example. IRR is a distributed database of route and route-related information [Braga 2010]. The sometimes neglected participation of the human being during the creation and update IRR objects processes, was the motivation for creating a model of agents which could replace the human interventions (made by

[1]http://www.irr.net/

email). For this reason, was propose the *Autonomous Architecture Over Restricted Domains* (A2RD) into the restricted domain of an AS, applying as use case over the IRR [Braga et al. 2015]. A2RD replaces the human with your agents, *Intelligent Elements* (IEs), establishing a new IRR model, named *innovation IRR* (iIRR), shown in Figure 1.

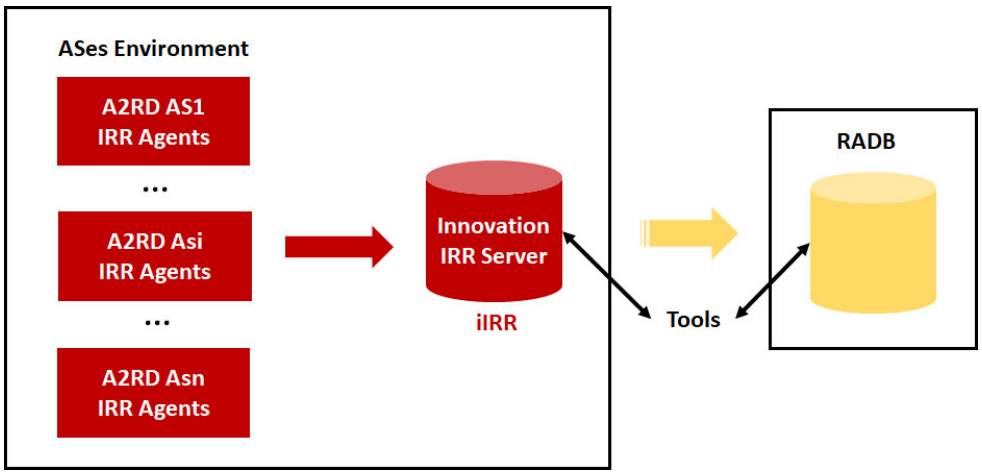

Figure 1. The innovation IRR model established by A2RD

A2RD specialized IEs, automatically create objects as defined by the *Route Policy Specification Language* [Alaettinoglu et al. 1999, Blunk et al. 2005]. Those objects that can not be created automatically will receive support from AS administrators through a human-computer cooperation mechanism. Nothing is changed in relation to the present and future IRR structure, characterized by the expectations recommended by the stakeholders to the *Internet Engineering Task Force*[2] (IETF) and *Internet Research Task Force*[3] (IRTF) disseminated through of their formal documents [Meyer et al. 1999, Villamizar et al. 1999, Newton 2004, McPherson et al. 2015, Kisteleki and Haberman 2016]. Neither does it affect the security concerns surrounding the IRR and Internet governance [Kuerbis and Mueller 2017]. Similarly, tools that use IRR databases can be used without any modification. A very useful, among others, is the IRR Powertools[4].

For this paper, *blockchain* is a data structure whose components are chained, with guarantee of immutability of its contents, and consequent integrity of the chain preserved by a cryptography process, with difficult computational reversibility. This definition is much simpler but more computationally oriented than those in which blockchain is associated with crypto-economics or crypto-currencies, and often have confusing definitions, but when it is clear, blockchain is defined as a database [Nakamoto 2008, Pilkington 2015].

On the other hand, by abstracting from property of immutability, the data structure like blockchain is a well-known concept used in computer research and originated in the academic literature of the 1980s and 1990s [Narayanan and Clark 2017]. As

[2]https://ietf.org/
[3]https://irtf.org/
[4]https://github.com/6connect/irrpt

a simple data structure, for example, in works involving *provenance*, which is used as complementary data documentation containing the description of 'how', 'when', 'where', 'why' the data were obtained and 'who' obtained it [Braga and Banon 2008]. The blockchain model proposed in 2008 to meet the Bitcoin virtual currency has effectively aroused the interest of the research community mainly by the immutability property that ensures data integrity [Prusty 2017, Bashir 2017]. Immutability and integrity are obtained by a hash encryption mechanism [Bakhtiari et al. 1995, Rogaway and Shrimpton 2004]. The combination of these two factors and characteristics associated to the blockchain recommended the application in the A2RD model, with the aim of enhancing communication and collaboration among the IEs [Braga et al. 2017b]. This proposal is more simpler than those application of blockchain in Internet Infrastructure with fundamentals in Bitcoin technology, based in the appropriate fact that to run, Internet use resources such as numbers and names [Hari and Lakshman 2016].

There is no study directly related to this work and there are few blockchain works related to the Internet Infrastructure. Blockchain still is not a matured technology, there are challenges that need to be considered when designing a platform, to ensure security, reliability and usability. So, there is not related works associated with Internet Infrastructure, because is fact that due to the emergent nature of the topic, the reviewed literature was not published in high-ranking journals with prolonged review cycles [Xu et al. 2016].

The main goal of this paper is to present the *Internet Infrastructure Blockchain* (IIBlockchain), a blockchain architecture to improve the security, knowledge and intelligence in inter-agent communication and collaboration over a restrict domains of the *Internet Infrastructure*, developed specifically and therefore independent of any available blockchain platform. The next sections of this paper will be organized as follows. In section 2 we discuss the A2RD model and the needs for inter-agents communication and cooperation. In section 3 we present the architecture of IIBlockchain and the properties inherent to the blocks, their types and the characteristics of the designed chain. In section 4 we discuss the implementation of IIBlockchain showing the main associated properties. In section 5 we present the conclusions and in section 6 we present the proposals for future works.

2. The A2RD Model

A2RD is a project that initially proposed the creation of agents with automatic activities replacing human tasks in the environment restricted to the domain of an AS. The use case was the addition and update of objects in IRR server. The application was considered useful mainly because the tasks of the AS administrator did not guarantee the accuracy in its completion nor the permanent need to update the objects making the IRR an unreliable system from the point of view of its contents. A2RD solved this problem.

A new proposal for the A2RD model emerged from this experience [Braga et al. 2017b]. The Figure 2 shows this new proposal, in which the A2RD, (1), is represented as an agglomeration of IEs in a four layers model.

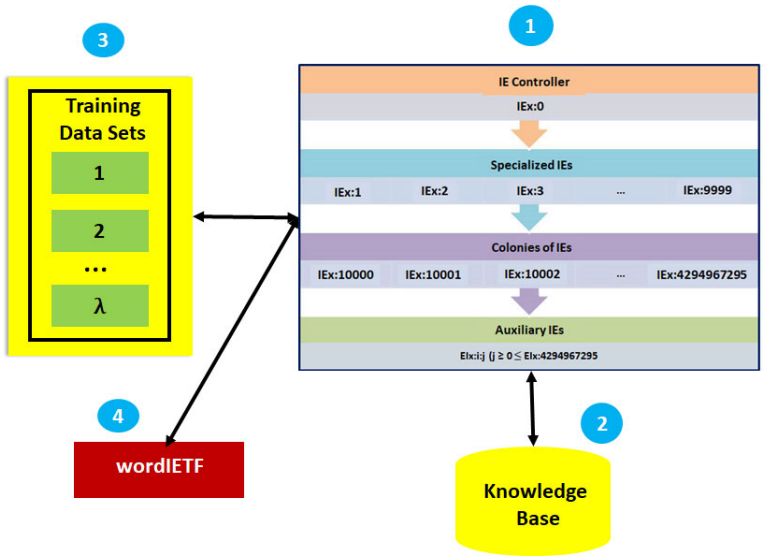

Figure 2. A2RD environment

A2RD IEs, reach their autonomy and intelligence aided by three components, the *Knowledge Base*, (2), the *Training Data Sets*, (3), and wordIETF, (4). These three components are obtained from non-structured databases, in particular, from the *Request for Comments* database, containing documents authored by network operators, engineers and computer scientists, documentary methods, behaviors, research, or innovations applicable to the Internet, all of them, working in groups of the IETF and IRTF, and maintained by RFC-Editor[5].

Each AS, of its own free will, may implement its respective A2RD, which is controlled by the IE named *IE Controller*, which receives the identification *x:0*, where x is the *AS Number* (ASN).

IEs need to communicate in order to collaborate, learn and cooperate with each other. This communication needs to be secure, that is, the respective *IE controller* must recognize the origin of each pair in their information exchanges. A mechanism called *Dark Think Security* (DTS) has been proposed to ensure the desired security [Braga et al. 2017a]. Although preliminary implementations have revealed that DTS is indeed secure, it has proved to be complex in implementation. In the search for a simpler alternative included *Pretty Good Privacy* (PGP) [Garfinkel 1995]. Using PGP, an ASx IE controller that wants to communicate with an ASy IE controller, will use the ASy *public key* to encrypt the message, for $\forall x$ *and* $\forall y$ *such that* $x \neq y$ *and* $x, \quad y = 1, ..., n, \quad n \leq$ total ASes present in the *Internet Routing Table*[6]. The ASy controller uses your *secret key* to decrypt the message. Thus, for this and for other reasons that we will see in the following section, the recommended solution was a variation of blockchain implementations proposed in the literature, that we named in this paper as *IIBlockchain*.

[5]https://www.rfc-editor.org/
[6]http://thyme.rand.apnic.net/current/data-summary

3. IIBlockchain Model and Implementation

The IIBlockchain model can be seen in Figure 3, which shows the implementation of A2RD in any two ASes (ASx and ASy).

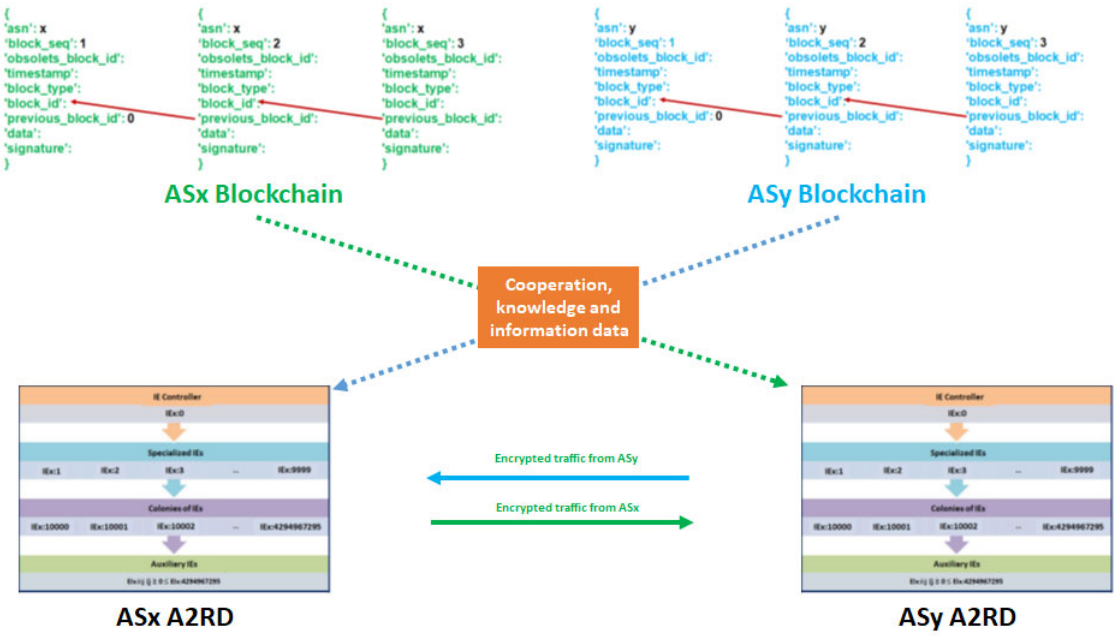

Figure 3. IIBlockchain Architecture implemented over ASx and ASy domains

This figure shows that the respective A2RD communicate through encrypted messages. Also, the A2RDs independently maintain a blockchain with properties characteristic of IIBlockchain. These chains contain, in their blocks, data inherent to each A2RD and about the environment of the AS in which they are implemented allowing the cooperation through the exchange of knowledge and information that can help in learning and maintaining the autonomy of their respective IEs. Each A2RD locally maintains a copy of IIBlockchain from each of the other ASes. There is no need to implement an A2RD for a chain to be constructed for an ASN. Specialized IEs of an ASx any guarantee that minimal information is included in chains of other ASes.

3.1. Block Properties

A block of any chain type is equivalent to a dictionary structure of the Python language, whose configuration and summary description of the respective keys are shown in Figure 4.

The detail description of block keys are in Table 1.

3.2. Chain Properties

Any chain only exists if it has a 'Genesis' block type as its first block ('block_seq' = 1). Suppose that ASx wants to add in its chain, a block that will contain its *PGP public key* with which any ASN can encrypt messages that only ASx will understand. At this point, the ASx chain is empty. Suppose x = 18782. So, using the IIBlockchain

```
{
'asn': Autonomous System Number,
'block_seq': Position into the chain,
'obsoletes_block_id': 'Block ID of the block that may have been replaced by this',
'timestamp': 'Current time the block was add to the chain',
'block_type': 'Type of the block',
'block_id': 'Hash of the entire block that will identify it',
'previous_block_id': 'Block ID of the previous block of this block',
'data': 'Data of the related with the block type',
'signature': 'Signature that ensures the owner of the data'
}
```

Figure 4. Block Structure

Python class available at GitHub[7] if we add the block of type *PublicKey* we will have a two block chain as can be seen in Figure 5. It is important to note that block numbers (*block_seq*) are sequential (1 and 2, respectively).


```
{
'asn': 18782,
'block_seq': 1,
'obsoletes_block_id': '',
'timestamp': 'Wed, 31 Jan 2018 10:57:19 +0000',
'block_type': 'Genesis',
'block_id': 'b663e678b4a96432e66ad45018680a93f3345b87490b6c49a0b791df61b3932a',
'previous_block_id': '0',
'data': '',
'signature': ''
}

{
'asn': 18782,
'block_seq': 2,
'obsoletes_block_id': '',
'timestamp': 'Wed, 31 Jan 2018 10:57:19 +0000',
'block_type': 'PublicKey',
'block_id': 'c18c6625f918b75b6292c62ee2949f83d53d5d8207174673bd08b8d4c7635657',
'previous_block_id':
'b663e678b4a96432e66ad45018680a93f3345b87490b6c49a0b791df61b3932a',
'data':
['-----BEGIN PGP PUBLIC KEY BLOCK-----',
'Version: GnuPG v1',
'',
'mI0EWnGhDwEEAKyP9NkCL3Fjs5x2FOtgrpJOC35U12JQI6MsudUYGA5vrMsvkq1L',
'W8oATQMUDzcPr2r6N+Y/PJKCDs9D81UlbtZJrUDgCBWnMpgxnngfSmEzgCZVrCev',
'QLQusEQBSwAaHxkDYnvK7vYznLkxd0v59qGwream5Byq3hvu5uGio7A9ABEBAAG0',
'LkFTMTg3ODlgKFVzZWQgaW4gQTJSRCBtb2RlbCkgPEFTMTg3ODJAYTJyZC5wdD6l',
'uAQTAQIAIgUCWnGhDwIbLwYLCQgHAwIGFQgCCQoLBBYCAwECHgECF4AACgkQfpNr',
'HVGA5oRR9gQAn94XRaAzAUZqiVVK48FVDzRP1Zf+73BABGNGXx9SfMLM5xQ7w8Pq',
'YREtJJGu0Ax8nBBK1crRC1RBSIXGWyWx6xqZCM2NsNCViDFVOVyxp3sR5kVrcLwW',
'0PW2V0kepdzN+jPjNwJTlrzqQu4STl9iALcv4bURiB9Kd2Zzuv/Hv/8=',
'=MLP4',
'-----END PGP PUBLIC KEY BLOCK-----',
''],
'signature': ''
}
```


Figure 5. Initial chain that, necessarily, has the block type 'Genesis'

Continuing and add a third block, now an *mntner* IRR object type (*irr_mntner*). The data is transformed into a string to be signed by the PGP, ensuring data properties to AS18782. Once this is done, the block is added to the chain as the third block. The block added can be seen in Figure 6.

To complete these example that illustrate some properties of the chain, let's assume

[7] https://github.com/juliaobraga/a2rd/IIBlockchainCode

Table 1. Description of block dictionary keys

Key	Description
asn	ASnumber: Identifies the owner number of the string. For same string, the value of this key is always the same
block_seq	Identifies the position of the block within the chain. If block i preceding or immediately preceding block j than $i < j$ and not necessarily $j = i + 1$. This is due to the fact that a block can be removed, from an *ASN* chain, if it becomes obsolete. Upon removal, the block is added to the *obsolete* chain. The immutability and integrity of this *ASN* chain must be restored.
obsoletes_block_id	If the value of this key is not empty, so this references the **block_id** that will be obsoletes
timestamp	Time moment the block was add in the chain
block_type	Type of the block: block types are not necessarily predefined. IEs can create different types of blocks through agreements between them during their normal activities. Important blocks are, however, predefined. For example, the *Genesis* block, which is necessarily the first block of any chain. Blocks that represent IRR objects always prefix the usual object name with *irr_*
block_id	Hash that will identify the block, obtained on the whole block, after it is completely filled
previous_block_id	**block_id** of the previous block of this block
data	Data of the related with the block type
signature	Signature that ensures the owner of the data

a change in the object *irr_mntner*. A new data is signed via PGP, and included in the chain, not without first identifying in the *obsoletes _block_id*, the block that it is rendering obsolete. The new block is added as 4th block in the AS18782 IIBlockchain and your configuration is shown in Figure 7.

3.3. Chain Transfer

The chains are compressed and named as *ASxVaaaammddhhmmss.zip*. A specialized IE will take care of this activity and follow up by compacting the chain, sending it to GitHub[8] and update the respective version in *wordIETF*. All chains are public, but the *secret keys* are not.

4. IIBlockchain Implementation

In this section we make considerations on important topics that deserved special attention during implementation.

[8]https://github.com/juliaobraga/a2rd

```
{
'asn': 18782,
'block_seq': 3,
'obsoletes_block_id': '',
'timestamp': 'Wed, 31 Jan 2018 10:57:24 +0000',
'block_type': 'irr_mntner',
'block_id': '5cc58ab3650c8ea443efddd1a4f748381e80e2c7ed325421750ec3f36f9b20ea',
'previous_block_id': 'c18c6625f918b75b6292c62ee2949f83d53d5d8207174673bd08b8d4c7635657',
'data':
{
'mntner': 'MAINT-AS18782',
'descr': 'Pegasus',
'admin-c': 'Juliao Braga',
'tech-c': 'Juliao Braga',
'upd-to': 'jb@pegasus.com.br',
'mnt-nfy': 'jb@pegasus.com.br',
'auth': 'CRYPT-PW ZocCkOH/zCkQw',
'mnt-by': 'MAINT-AS18782',
'changed': 'jb@pegasus.com.br 20090302'
},
'signature':
'-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1\n

{
"mntner": "MAINT-AS18782", "descr": "Pegasus", "admin-c": "Juliao Braga",
"tech-c": "Juliao Braga", "upd-to": "jb@pegasus.com.br",
"mnt-nfy": "jb@pegasus.com.br", "auth": "CRYPT-PW ZocCkOH/zCkQw",
"mnt-by": "MAINT-AS18782", "changed": jb@pegasus.com.br 20090302
}
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1

iQEcBAEBAgAGBQJacaEUAAoJEIH4EAKGjl9ZzUUIAJ7q5iIUob8w0cvTKhNF8vq/
wzEh+lw3SF85nvV5GvplXql2dmgtddZD+aQVCXeRxENlO0ClbTPR0pAIlcyAG4O0
7IYxjF3nVYK2p44aCIJLYu5siUVRYexeyUVRVLVQj4qLV6p4S3VxsoTROa9avXoF
Cqa64wmDRJOepVSZMeclhjKeQRprSemOGf9SfwjZZDs8TZHFKVvAUQlgkOfuAjLn
vf6ehTfD+WM+zm+nNpXAbWx+ql+uYua/wacA5G4p0R81GVEQpmuF19Z9/BFi5gKP
qLd+modeEx0UIWr1DZle6kDAMqJFC5Q4telmTsnWKqNFvFEf6pwcR+Xzy6RD9N8=
=FLpY
-----END PGP SIGNATURE-----'

}
```


Figure 6. Block 3: Adding an IRR object

4.1. Space Analysis

Table 2 displays some data about storage values, considering the chain created for the example in this paper.

Table 2. Storage Costs Parameters

#	Discrimination	Value
1	Block 1	1,300
2	Block 2	1,365
3	Block 3	3,451
4	Block 4	3,571
5	Total	9,687
6	ASes in routing table (12 Fev 2018)	59,789
7	IRR objects number (ARIN)	10
8	Number of protocols in TCP/IP	51

We used the *sys.getsizeof* function to determine the amount of bytes occupied by the Python dictionary structure, chosen to represent IIBlockchain. The result is not very good and so we evaluated two alternatives versions. The preferred version was that

```
{
'asn': 18782,
'block_seq': 4,
'obsolets_block_id': '5cc58ab3650c8ea443efddd1a4f748381e80e2c7ed325421750ec3f36f9b20ea',
'timestamp': 'Wed, 31 Jan 2018 10:57:49 +0000',
'block_type': 'irr_mntner',
'block_id': 'b65ab41ee6e8675a5a75e5be5d3eb99a335b1c1073e0617a4d92cc4e6650350d',
'previous_block'_id: '5cc58ab3650c8ea443efddd1a4f748381e80e2c7ed325421750ec3f36f9b20ea',
'data':
{
'mntner': 'MAINT-AS18782',
'descr': 'Pegasus',
'admin-c': 'Juliao Braga',
'tech-c': 'Juliao Braga',
'upd-to': 'info@a2rd.pt',
'mnt-nfy': 'jb@pegasus.com.br',
'auth': 'CRYPT-PW ZocCkOH/zCkQw',
'mnt-by': 'MAINT-AS18782',
'changed': 'jb@pegasus.com.br 20180203'
},
'signature':
'-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

{
"mntner": "MAINT-AS18782", "descr": "Pegasus", "admin-c": "Juliao Braga", "tech-c": "Juliao Braga", "upd-to": "info@a2rd.pt",
"mnt-nfy": "jb@pegasus.com.br", "auth": "CRYPT-PW ZocCkOH/zCkQw", "mnt-by": "MAINT-AS18782",
"changed": jb@pegasus.com.br 20180203
}
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1

iQEcBAEBAgAGBQJacaEtAAoJEEtF6Eiepna407UH/jh4Y5TEr2A72ROOEihrCUZf
TtXe69kAxbM4rFEOu5YyNKuVn/nosiOgRC46XowymC1Oben0Wwy2VVnt7jPjuImg
08vQYVrIBSYEConCiRSIAi9O1k0qBt8JjFoX1kySNwrg2TwWHu/ggQ7fwu+Bt73o
17w9/ZUBWBZyfXiPlt6PokQW000AwfDAPQ5xUfRsDE6xHfZUz/7mrQUtDBQNAWmp
5YP84Gzyjfv6BLuFCpHetq2pzXDpDDOt13cGWGKMreR0TgAS7jpZlGXtdFjBcOUx
pisjnQc4EyQdNSUyNWMWOm7CBEGdwhvPDb1FYjPxI+SaecYrr3pwJIIlOr9H7uU=
=+/Yx
-----END PGP SIGNATURE-----'
}
```


Figure 7. Block 4: Makes block 3 obsolete

of larger result values[9] (lines 1-4 on the table). Suppose each block of the string to be constructed occupies twice as many bytes as the largest block in our example (line 4). So our block occupies **7,124 bytes**. *American Registry for Internet Numbers*[10] (ARIN) identifies ten objects to populate its IRR (line 7). Thus, only with IRR objects, the IIBlockchain of an AS spends $7,124 \times 10 =$ **71,124 bytes** ~ **70 Kbytes**. So, the total bytes to represent the IRR objects for all ASn are: $59,789 \times 70$ Kbytes $= 4,285,675,520 \sim$ **4 Gbytes**. Let us now assume that for each TCP/IP protocol[11] (line 8) we will need 20 blocks with the largest known double size (knowledge information, for example): $20 \times 7,124$ *bytes* = **139 Kbytes**, value that corresponds to **0.003%** of the space spent by IRR objects. Certainly there are other types of blocks that IEs will produce. But the largest number of them are obsolete blocks. Very difficult to measure the space to be occupied by obsolete blocks. Only an inaccurate estimate would be possible. One estimate is that 25% of the blocks will be obsolete. So the total estimated storage space for the IIBlockchain is **5 Gbytes**. Any operation on IIBlockchain do not require additional space. Therefore the space complexity is $\mathbf{O(1) \sim O(n)}$ [Costa 2015].

[9]https://goshippo.com/blog/measure-real-size-any-python-object/
[10]https://www.arin.net/resources/routing/templates.html
[11]http://www.comptechdoc.org/independent/networking/protocol/protnet.html

4.2. Time Complexity

The heaviest algorithm we have in operations on IIBlockchain is to search linearly over an array or eventually over a linked list. Then, in the worst case, the complexity of time is O(n) [Costa 2015].

4.3. Security

IIBlockchain is public. The security that matters to IIBlockchain will only be verified when a non obsolete block needs to be used. In two stages this is necessary: (a) the integrity of the block and (b) the reliability of the information contained in the block. Stage (a) consists of checking the validity of the hash that identifies the *block_id*. Stage (b) is the verification that the signature guarantees ownership of the information by the respective AS. If any of the above stages fails, an alert is sent to all implementations of A2RD. Immediately look for the block in the previous version and use it. The existence of the block in the previous version can be verified by the parameter *timestamp* and the name of the version. Meanwhile, specialized IEs will analyze the chain, in order to identify the cause of the breach of trust in the block.

5. Conclusions

The authors consider that the objective of allowing a mechanism of relationship between IEs of the various A2RD implementations was achieved. Also, Blockchain is effective in ensuring co-operation and distribution of knowledge that can be shared among IEs in the various domains of ASes. It is a simple, easy-to-understand, and implementation-oriented design with no additional effort required in any programming language. The IIBlockchain has both public and private characteristics and has no inherent concerns or additional difficulties, for this reason. Also, it is worth remembering that IIBlockchain is oriented to the application of Blockchain by agents and not by humans, which certainly decreases complexity.

6. Future Works

At this point, it is not possible to determine how the presence of obsolete blocks will influence the operations on an IIBlockchain of some ASN. Implementations in programming languages like Python and others one, does not seem to be a big problem, because dictionaries are indexed and obsolete blocks can be ignored. However, it is necessary to evaluate the possibility of creating a new type of chain: the chain of obsolete blocks, that is to say, the chain consisting of blocks that become obsolete in each ASN chain.

At some point, one A2RD IE may checking the state of the chain and remove obsolete blocks, passing it to the obsolete chain considering:

- The chain from which the block was removed will be reconstituted to maintain the immutability and integrity. This is achieved by having the next block point to the previous block removed, and a new hash is calculated to identify the next block and successively to the blocks thereof until the end of the chain.

- The block removed will be inserted in the *obsolete chain* pointed to the last block of this chain. The block's block number ('block_seq') should be concatenated by a hyphen and another sequence number to the number of the last block of the *obsolete chain*. After this a new hash will be determined to identify this block and the block can be inserted in the *obsolete chain*

Complementary, the IIBlockchain design is simple enough for applications in several other networking areas or not. New versions of the implementation will seek to establish independence from block structure and coding.

7. Thanks

From Juliao Braga and Jessica Ribas: Supported by CAPES – Brazilian Federal Agency for Support and Evaluation of Graduate Education within the Brazil's Ministry of Education.

References

Alaettinoglu, C., Villamizar, C., Gerich, E., Kessens, D., Meyer, D., Bates, T., Karrenberg, D., and Terpstra, M. (June 1999). Routing policy specification language (rpsl). Technical report, RFC Editor. RFC2622. `https://www.rfc-editor.org/info/rfc2622`. (Obsoletes RFC2280) (Updated-By RFC4012, RFC7909) (Status: PROPOSED STANDARD) (Stream: IETF, Area: ops, WG: rps) (DOI: 10.17487/RFC2622) . Acessado em 03/02/2018.

Bakhtiari, S., Safavi-Naini, R., Pieprzyk, J., et al. (1995). Cryptographic hash functions: A survey. Centre for Computer Security Research, Department of Computer Science, University of Wollongong, Australie.

Bashir, I. (2017). Mastering Blockchain. Packt Publishing Ltd.

Blunk, L., Damas, J., Parent, F., and Robachevsky, A. (March 2005). Routing Policy Specification Language next generation (RPSLng). Technical report, RFC Editor. `https://www.rfc-editor.org/info/rfc4012`. (Updates RFC2725, RFC2622) (Updated-By RFC7909) (Status: PROPOSED STANDARD) (Stream: IETF, WG: NON WORKING GROUP) (DOI: 10.17487/RFC4012) . Acessado em 03/02/2018.

Braga, J. (2010). Curso IRR – Parte I a Parte X. Internet Infrastructure Blog: accessed: 19.01.2018.

Braga, J., de Amorim Silva, R., Endo, P. T., and Omar, N. (2017a). Dark Think Security: Enhancing the Security for the Autonomous Architecture over a Restricted Domain. In Proceeding of CSBC 2017, page 8, Mackenzie Presbiterian University.

Braga, J., Omar, N., and Granville, L. Z. (2015). Uma proposta para o uso de elementos inteligentes em domínios restritos da infraestrutura da internet. In Anais CSBC 2015 - WPIETFIRTF, Recife, Pernambuco, Brasil.

Braga, J., Omar, N., and Thome, L. F. (2017b). Acquisition and use of knowledge over a restricted domain by intelligent agents. In Proceedings of the SouthEast Conference, ACM SE '17, pages 203–207, New York, NY, USA. ACM.

Braga, J. C. and Banon, G. J. F. (2008). Data provenance: Theory and application to image processing. IEEE Latin America Transactions, 6(2).

Costa, E. (2015). Programação em Python. FCA, Lisboa, PT, 1 edition.

Garfinkel, S. (1995). PGP: pretty good privacy. " O'Reilly Media, Inc.".

Hari, A. and Lakshman, T. (2016). The internet blockchain: A distributed, tamper-resistant transaction framework for the internet. In Proceedings of the 15th ACM Workshop on Hot Topics in Networks, pages 204–210. ACM.

Hawkinson, J. and Bates, T. (March 1996). Report on MD5 Performance . Technical report, RFC Editor. RFC1930. `https://tools.ietf.org/rfc/rfc1930.txt`. (Updated-By RFC6996, RFC7300) (Also BCP0006) (Status: BEST CURRENT PRACTICE) (Stream: IETF, Area: rtg, WG: idr). Acessado em 06/09/2014.

Kisteleki, R. and Haberman, B. (June 2016). Securing Routing Policy Specification Language (RPSL) Objects with Resource Public Key Infrastructure (RPKI) Signatures. Technical report, RFC Editor. RFC7909. `http://www.rfc-editor.org/rfc/rfc7909.txt`. (Updates RFC2622, RFC4012) (Status: PROPOSED STANDARD) (Stream: IETF, Area: rtg, WG: sidr) (DOI: 10.17487/RFC7909). Acessado em 29/07/2017.

Kuerbis, B. and Mueller, M. (2017). Internet routing registries, data governance, and security. Journal of Cyber Policy, 2(1):64–81.

McPherson, D., Amante, S., Osterweil, E., Blunk, L., and Mitchell, D. (December 2015). Considerations for Internet Routing Registries (IRRs) and Routing Policy Configuration . Technical report, RFC Editor. RFC7682. `http://www.rfc-editor.org/rfc/rfc7682.txt`. (TXT = 47996) (Status: INFORMATIONAL) (Stream: IETF, Area: ops, WG: grow) (DOI: 10.17487/RFC7682). Acessado em 29/07/2017.

Meyer, D., Schmitz, J., Orange, C., Prior, M., and Alaettinoglu, C. (August 1999). Using RPSL in Practice. Technical report, RFC Editor. RFC2650. `https://tools.ietf.org/rfc/rfc2650.txt`. (Status: INFORMATIONAL) (Stream: IETF, Area: ops, WG: rps) (DOI: 10.17487/RFC2650). Acessado em 29/07/2017.

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.

Narayanan, A. and Clark, J. (2017). Bitcoin's academic pedigree. Communications of the ACM, 60(12):36–45.

Newton, A. (February 2004). Cross Registry Internet Service Protocol (CRISP) Requirements. Technical report, RFC Editor. https://www.rfc-editor.org/info/rfc3707. (Status: INFORMATIONAL) (Stream: IETF, Area: app, WG: crisp) (DOI: 10.17487/RFC3707). Acessado em 03/02/2018.

Pilkington, M. (2015). Blockchain technology: Principles and applications. In esearch Handbook on Digital Transformations, pages 11–39. Edward Elgar. Available at https://ssrn.com/abstract=2662660.

Prusty, N. (2017). Building Blockchain Projects. Packt Publishing Ltd.

Rekhter, Y., Li, T., and Hares, S. (January 2006). A Border Gateway Protocol 4 (BGP-4). Technical report, RFC Editor. RFC4271. http://www.rfc-editor.org/rfc/rfc4271.txt. (Obsoletes RFC1771) (Updated-By RFC6286, RFC6608, RFC6793) (Status: DRAFT STANDARD) (Stream: IETF, Area: rtg, WG: idr) . Acessado em 07/09/2014.

Rogaway, P. and Shrimpton, T. (2004). Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In International Workshop on Fast Software Encryption, pages 371–388. Springer.

Villamizar, C., Alaettinoglu, C., Meyer, D., and Murphy, S. (December 1999). Routing Policy System Security. Technical report, RFC Editor. RFC2725. http://www.rfc-editor.org/info/rfc2725. (Status: PROPOSED STANDARD) (Stream: IETF, Area: ops, WG: rps) (DOI: 10.17487/RFC2725) . Acessado em 03/02/2018.

Xu, X., Pautasso, C., Zhu, L., Gramoli, V., Ponomarev, A., Tran, A. B., and Chen, S. (2016). The blockchain as a software connector. Proceedings - 2016 13th Working IEEE/IFIP Conference on Software Architecture, WICSA 2016, 11(2016):182–191.