EAI Endorsed Transactions
on Future Internet
Research Article EAI.EU

Approaching decentralized non-repudiation

H. Kamdem Fezeu[1,*], T. Djotio[2] and R. Oulad Haj Thami[3]

[1]University of Yaounde I, harry.kamdem@polytechnique.cm
[2]University of Yaounde I, tdjotio@gmail.com
[3]Mohammed V University, rachid.ouladhajthami@gmail.com

Abstract

Securing data exchanges is a major preoccupation, and several techniques have been developed to reach that aim. The predominant model for such exchange is that which relies on trusted third-parties. Meanwhile, emerging technologies such as IoT are set to broadcast growing amounts of sensitive data, thereby making centralized architectures problematic for privacy and performance reasons and making decentralized networks ever more relevant. However, these third-parties play an important role in securing brokered communications and are essential in providing Authentication and Non-Repudiation according to current models, and cannot be used in peer-to-peer networks. Hence there is need for a simple model applicable in fully decentralized networks to provide Non-Repudiation. This document proposes such a model, presents an implementation and discusses its application, particularly in implementing irrefutable trustless transaction mechanisms – similar to blockchain – with limited resources.

Keywords: Information Security, Internet of Things, Mesh Networks, Blockchain

Received on 15 December 2017, accepted on 5 January, 2018, published on 12 January 2018

doi: 10.4108/_____

1. Introduction

The basic requirements of Information Security, are generally referred to as the "CIA triad" [1], comprising Confidentiality (ensuring that information is not disclosed to non-authorized third-parties), Integrity (ensuring that information is not altered during transit) and Availability (ensuring that information reaches the correct recipients). Several cryptographic techniques have been adapted to autonomous distributed networks and as such, the basic tenets of the "CIA triad" can successfully be fulfilled in a Mesh Network. Nonetheless, there exists a number of issues which re-quire different forms of security such as Authentication and Non-repudiation which are not generally addressed.

On the internet, the implementation of such characteristics generally involves the introduction of a Trusted Third-Party allowing for brokered communication, typically in a client-server model. Such an entity serves as middle-man or notary, and may leverage other technologies to further secure communications (SSL, PGP, etc.). However, in a peer-to-peer net-work, such a solution would be impractical as any node may leave the network at any moment and no node is to be inherently trusted.

This paper presents a cryptographic approach to securing communications, which allows for Non-repudiation and Authentication in addition to Confidentiality, Integrity and Availability, and which can readily be implemented in decentralized networks, as well as overlaid on existing infrastructure. It starts out with a brief overview of the existing literature, and then presents the proposed communication model, before discussing the principle and results of an experimental implementation, and goes on to discuss the pertinence of the said model as an alternative to blockchain in scenarios with little computing resources before ending with concluding remarks.

*Corresponding author. Email:kamdemharry@gmail.com

2. Literature review

With the growing number of connected devices, centralized cloud services require increasing resources and are becoming performance bottle-necks. Also, according to industry leaders, one of the main factors hindering success of the Internet of Things is lack of adequate information security associated with Machine-to-Machine communication [2].

These assertions tend to suggests that there is a real need for security models which are specifically designed to be implemented upon decentralized peer-to-peer networks. However, even vendor applications which promise consumers offline networking capabilities (such as home automation and smart connected devices) still greatly rely on internet servers as trusted third-parties for coordination and centralized security. [3]

Moreover, although numerous security mechanism have been designed and implemented to protect the various aspects of mesh networks, none has been specifically designed to provide general-purpose non-repudiation of messages transmitted in a completely decentralized network. [4] [5]

Furthermore, although Blockchain technology is a promising solution to securing Decentralized Communication (a subset of our domain of interest) using a cryptographically secured shared ledger, it is worth noting that it has high resource costs and scalability issues [6], which make it inapplicable or at least inefficient in low-resource scenarios, such as wireless sensors or the internet-of-things [7].

3. System Model

3.1. Components

The proposed system consists of four main components,

- An identification component (optional).
- A message encryption component (optional).
- A broadcast system for control messages.
- A message transfer component.

The components are defined as follows:

Identification

Participants in this communication system are each required to generate a key-pair for asymmetric encryption. The public key from this key-pair is used to uniquely identify its holder on the system and serves as his/her address.

The principal function of this component is to establish and certify the link between user-recognizable identifiers (such as e-mail addresses) and the public keys of their respective holders. This can readily be achieved outside the system and as such, the component is optional.

This component could function follow declarative approach (where the sys-admin or user directly registers a list authorized contacts, as in PGP). Or an iterative approach (where new addresses are automatically recognized, thanks to some form of Central Authority, as in S/MIME).

Additionally, this subsystem could be used to restrict access to the system to authorized users only, by requiring some proof-of-identity on arrival. However, such a setting could help message recipients to determine the identity of the sender before accepting a transmission, thereby allowing them to filter incoming messages and allowing them to undermine overall information Availability on the network.

Message Encryption

Decentralized communication depends on messages being transferred successively between the nodes in the pathway between the sender and the receiver. As such for each message sent, one is to assume that any client in the system could access the payload.

This component serves to encrypt messages before transmission. The receiver's public key is used to convert the initial message into a self-contained cipher-text, which can only be decrypted by the intended receiver. This cipher-text is transmitted in-lieu of the original message.

In certain cases, one may wish to allow a message to be publicly readable (like an open letter), hence this component is optional and even when it is present its action should be optional.

Broadcast

The system is coordinated via a number of control messages. These messages indicate the users involved and the hash fingerprint of the payload, and are designed to allow any client to verify, for each sent message, the identity of the sender and receiver, the sending and receiving time and the validity of the sender's description of the payload, all without necessarily allowing a receiver to know the identity of the sender prior to receiving a message.

Ideally, each participant possesses and processes a full copy of this store of control messages, with metadata from the very first message to the last, so as to detect and report fraudulent data and duplicates. Nonetheless, this is impractical due to storage and computing limitations, hence it may be necessary to implement an obsolescence policy.

Message Transfer

This component is responsible for the transmission of the payload from the emitting node to the receiving node, in a way which allows all other nodes to inspect any given packet in order to verify the correctness of metadata contained in control messages.

3.2. Operation

Assuming a system comprises the functional elements discussed, the transfer of messages using this model involves five steps,

- Encryption (optional).
- Cyphering.
- Emission.
- Reception.
- Deciphering.

What follows is an account of the successive steps in the transfer of a message from one hypothetical user, Alice, to another hypothetical user, Bob:

Encryption (optional)

Alice may desire to keep her message confidential. To achieve this, she would generate a symmetric key with which she would encrypt the initial message (M'), and which she would later encrypt (the key), together with a hash fingerprint of M', using Bob's public key. The encrypted message and the encrypted key-and-hash would be combined into a single file M, similarly to the encryption process in PGP [8].

Cyphering

Alice creates a file containing the message (M) and her identification (her public key and optionally her user-recognizable address such as email). She then generates a one-time symmetric encryption key (transmission key) with which she encrypts the said file, to obtain the "payload".

Emission

Alice makes the payload available to the transmission system, and broadcasts a control message containing:

- Bob's identification (his public key and possibly his user-recognizable address such as email).
- The identifier of the payload (which could allow any node to retrieve it).
- A hash fingerprint of the payload.
- A timestamp.

This "emission message" serves to announce the payload publicly, and allows all intermediary nodes to verify the accuracy of the timestamp and unicity of payload identifier. Also, given that this emission message bears no mention of Alice's identity and is distributed from peer to peer, it impossible for Bob to identify the sender at this point.

Reception

Bob, after receiving the emission message and later retrieving the payload, submits a control message (reception message) which contains the pay-load identifier and a timestamp, and is signed using his private key.

Deciphering

Alice, after receiving Bob's control message and verifying its authenticity (using the signature), broadcasts a control message (deciphering message) containing:

- The transmission key.
- A hash of the message (M).
- A description of the contents of M.

This control message is signed with Alice's private key. It allows all nodes to unlock the payload and verify that Alice was indeed the initial sender, simply by comparing her public key to the key written inside the payload.

Moreover, Bob cannot deny having received this message, due to his reception message. Also, the message hash and description can later be used by Alice to reasonably prove that she sent a given message to Bob, even if encryption was applied.

4. Theoretical implications

4.1. From non-repudiation to atomicity

The proposed protocol, if correctly applied, allows us to ensure the non-repudiation of messages in a fully decentralized setting. However, there is a point where it is unfair: It gives an unfair advantage to the receiver, who may attempt – alone or with accessories along the chain of transmission – to retrieve and use the deciphering message whilst pretending he did not get it.

Hence, we have effectively transformed our original problem (non-repudiation of communications) into one of ensuring atomicity of each exchange (either it works in a way that both can testify to, or it fails in way which does not allow the receiver to know the message) between successive nodes (directly in contact with each other), taking into account their limited resources.

This problem is actually an embodiment of the "Byzantine Generals' problem", for safely communicating instructions between multiple actors whilst taking into consideration the potential actions of adversarial third-parties, as well as those of "traitors" amongst the actors, and based on previous research [9][10], it possible to design the successive broadcast-spaces and model the network traffic so as to make effectively mitigate risk of "receiver fraud" and hence make this last step more "just".

It is worth noting that the approached solutions proposed to the Byzantine Generals' problem each require strict pre-conditions to be applicable, and although we design the routing algorithm to favour the occurrence of such conditions (for example, by making each step to be witnessed by multiple, varying third-party nodes), these tweaks rapidly add to the footprint of the protocol in terms of metadata and extra processing, hence ultimately requiring compromise.

4.2. Towards blockchain-lite

It is worth noting that, whilst offering non-repudiation of electronic communications, this proposed model is specifically designed for low-resource scenarios. As such, it can be construed as a light-weight version of blockchain technology, applicable in resource-constrained scenarios, allowing for better scalability and transaction speed, with less stringent security.

4.2. Proposing an implementable solution

Previous works of research [11], [12] provided us with formalized goals of full non-repudiation and associated verification methods. Furthermore, blockchain technology has been the focus of the bulk of the research concerning actual implementations of decentralized non-repudiation [13]. However, this approach (blockchain) carries a great computational and storage cost and is not suited for in a decentralized setting.

As such, our primary contribution was to propose an implementable solution which embodies an approached solution to the problem, and which can readily be applied to low-resource environments. Hence it follows that our experimental data could not readily be compared with pre-existing datasets.

5. Experimental implementations

5.1. Presentation

Working from the model specified above, a basic experimental implementation was devised and simulated. The focus was on simple nodes with little computing and memory resources, and direct node-to-node links, such as in WSANs (Wireless Sensor-Actor Networks).

Table 1. Set-up of the experimental system

Component	Implementation
Identification	None. Nodes are addressed directly using a hash of their public key
Encryption	Each node is capable of encrypting and decrypting messages
Broadcast	Each node keeps queue of control messages in memory. These messages are delivered via successive network broadcasts
Message transfer	To get the payload, one simply broadcasts a certain control message. This is echoed node-to-node and any node possessing the payload sends it back along the echo route

5.2. Results

The system satisfies the requirement of Irrefutability in a fully decentralized network. However, in order to evaluate the performance of the pro-posed model in this setting, the total time taken for the successful delivery of a message and the total amount of traffic generated on the network were calculated, as they both varied with respect to the number of intermediary nodes between sender and receiver, and with respect to the size of the payload. This was first done via a mathematical model, and then a computer simulation was built to include realistic delay, jitter and net-work failure phenomena [14].

The simulated mesh was set to operate using the protocol described above, and was designed with the following properties:

- 100 Mbps node-to-node link speed.
- 512 bytes packet size.
- Failure rate of 1.5%.
- Sideways propagation (transmission to non-path nodes) limited to a maximum of 2 steps.

The results of the experiment were as shown below (Numerical results in Appendices):

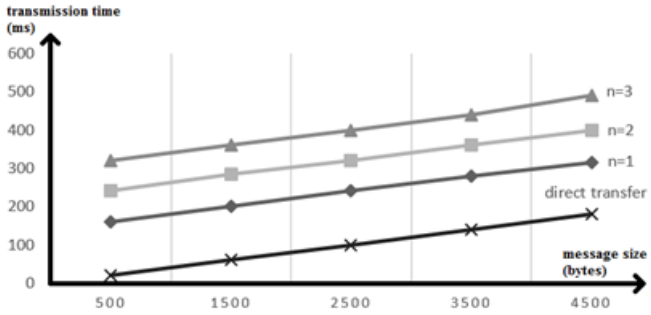

Figure 1. Message Size against Transmission Time, for varying number of Intermediary Nodes

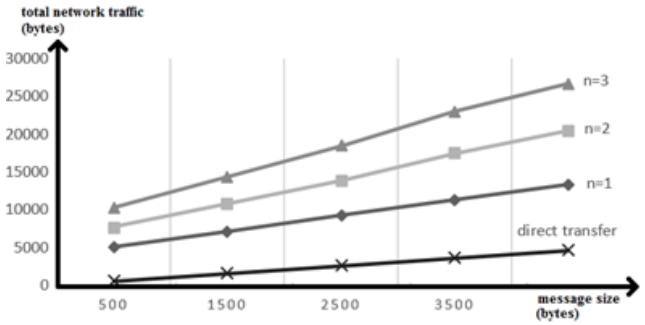

Figure 2. Message Size against Total Network Traffic, for varying number of Intermediary Nodes

5.3. Results Analysis

For a given number of intermediary nodes, transmission time and net-work traffic are both directly proportional to message size. Hence, for any given network setting, one could easily, starting from limited test data, extrapolate network performance under hypothetical conditions.

Moreover, there appears to be a sharp increase in transmission time due to the introduction of the first intermediary node, with all subsequent additions of intermediary nodes causing smaller and fairly constant increments. Transmission time does not vary substantively in response to the actions of nodes outside the transmission chain.

In contrast, network traffic increases at an increasing rate and could greatly be accentuated if multiple independent nodes decide to "pull" the payload so as to verify it cryptographically. Hence, network traffic is a potential limiting factor which is to be monitored.

5.4. Extensions to the model

In addition to the transmission of massages across a distributed network, our implementation was enhanced to be able to store-and-forward messages for offline nodes, and to handle multi-receiver messages. These features can readily be implemented as follows:

Offline messages

A simple solution to store-and-forward offline messages lies in selecting a "forwarder" amongst the existing nodes. Clients would then be configured to know the public key of the forwarder and when they send a message, they would automatically encrypt the "deciphering message" with the forwarder's key and send that to the shared store. The forwarder would automatically collect these messages and the associated payload. When a receiver issues a "reception message", it would then be the for-warder's responsibility to complete the protocol exchanges.

For greater resilience, message forwarding can be designed as follows:

- Multiple forwarders: The first forwarder to catch a "reception message" would issue the "deciphering message", and the others, seeing this re-action would know the message has been delivered. If however multiple forwarders respond simultaneously, the subsequent responses will be discarded as duplicates by node all who receive them.
- The forwarder role could be transmissible amongst nodes, possibly by implementing a line-of-succession on a first-come-first serve basis, or by organizing elections amongst the nodes

Nonetheless, forwarding raises a number of issues which need to be addressed:

- A node could spoof another identity, so as to pose as a forwarder to itself, thereby allowing it to receive a message without issuing a "reception message" and thus undermining non-repudiation. This can be addressed by stringent authentication policies.
- A node could fail to forward messages entrusted to it, thereby under-mining availability of data. This can be addressed by assigning the forwarder role collectively to a large number of nodes. Possibly, all nodes on the network could act as forwarders simultaneously.

Multi-receiver

Given that the payload is publicly accessible, it is possible to allow for multiple recipients by simply issuing multiple "emission messages". Nonetheless, this would undermine the Non-repudiation requirement for all except the very first receivers to acquire the message. This is because the "deciphering record" would have been made public on first-acquisition and subsequent receivers would not need to submit a "Reception Record".

Thus, this technique is only well-suited to sending a message to multiple devices belonging to the same user or group of users, or in cases where non-repudiation is not important beyond the first receiver.

6. Conclusion

The present work proposes a novel system for securing decentralized communications, in mesh networks or overlay networks implemented atop existing infrastructure, in a way which ensures Irrefutability, Confidentiality, Integrity and Availability. It is functional and simulations have been successfully conducted.

Theoretically, the proposed model could be built upon to devise a scalable alternative to Blockchain specifically for low-resource decentralized computing environments.

Moreover, this model holds several opportunities yet to be tapped. For example, with the rise of cheap wireless devices (accompanied by a strong BYOD trend in most organizations [3]), it is common for groups of individuals within organizations (such as businesses and governments) to user their consumer computing devices in local networks, without any reliable internet connection. Hence, this model could be implemented as an auto-configuring tool which could work transparently at the lower OSI layers (5 and below) [15], handling communications on the physical net-work. Thanks to POP's simplicity and its default download-and-delete behavior [16], such an implementation could readily provide a valid virtual POP interface on localhost, such that the user's existing e-mail soft-ware could function transparently.

This, and multiple other applications of this model are yet to be implemented, so as to be used to protect communications between people and machines.

Appendix A. Numerical Results

This section contains the numerical observations obtained from the experimental setup, from which the conclusions were drawn.

A.1. Variations in data speed

Table 2. Total Transit time (in milliseconds) with respect to number of intermediary nodes (n) and payload size (in bytes)

Payload size	n=0	n=1	n=2	n=3
500 b	12 ms	150 ms	221 ms	312 ms
1500 b	46 ms	186 ms	246 ms	347 ms
2500 b	83 ms	222 ms	265 ms	390 ms
3500 b	114 ms	256 ms	298 ms	421 ms
4500 b	149 ms	290 ms	344 ms	463 ms

A.2. Variations in data throughput

Table 3. Total Traffic (in bytes) with respect to number of intermediary nodes (n) and payload size (in bytes)

Payload size	n=0	n=1	n=2	n=3
500 b	504 b	5009 b	7704 b	10200 b
1500 b	1502 b	6744 b	10401 b	14274 b
2500 b	2532 b	8880 b	13609 b	19206 b
3500 b	3550 b	10196 b	16737 b	23750 b
4500 b	4790 b	12109 b	19504 b	27300 b

Acknowledgements.
This work was partially supported by the ERMIT Project and conducted at ENSIAS - Mohammed V University, Rabat, Morocco.

References

[1] Cherdantseva, Y. and Hilton J. (2013) A Reference Model of Information Assurance & Security (International Conference on Availability, Reliability and Security (ARES). Pages 546 - 555)

[2] IBM Institute for Business Value (2015) Device Democracy: Saving the fture of the internet of Things (http://www-935.ibm.com/services/us/gbs/thoughtleadership/internetofthings)

[3] Challenges stemming from the growth of the Internet of Things (https://bdtechtalks.com/2016/05/28/challenges-stemming-from-the-growth-of-iot/)

[4] Jaydip S. (2014) Security and Privacy Issues in Wireless Mesh Net-works: A Survey. Innovation Labs, Tata Consultancy Services Ltd. Kolkata

[5] Gerkis A. and Purcell J. (2006) A Survey of Wireless Mesh Networking Security Technology and Threats. SANS Institute

[6] R. Beck et al. (2016) Blockchain – The Gateway to Trust-Free Cryptographic Transactions. *Twenty-Fourth European Conference on Information Systems (ECIS), İstanbul,Turkey*

[7] M. Conoscenti et al., Blockchain for the Internet of Things: a Systematic Literature Review

[8] Zimmermann, Philip (1995). PGP Source Code and Internals. MIT Press. ISBN 0-22-2039-4.

[9] L. Lamport et al. (1982) The Byzantine Generals Problem, ACM Transactions on Programming Languages and Systems, Vol. 4, No. 3, July 1982, Pages 382-401.

[10] M. Vukolc, The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication, IBM Research – Zurich

[11] Zhou J and Gollmann D, Towards Verification of Non-repudiation Protocols, In Proceedings of 1998 International Refinement Workshop and Formal Methods Pacific

[12] Wei K., Heather J. (2006) Towards Verification of Timed Non-repudiation Protocols. In: Dimitrakos T., Martinelli F., Ryan P.Y.A., Schneider S. (eds) Formal Aspects in Security and Trust. FAST 2005. Lecture Notes in Computer Science, vol 3866. Springer, Berlin, Heidelberg

[13] M Crosby et al, BlockChain Technology, 2015, Sutardja Center for Entrepreneurship & Technology Technical Report
[14] Pingman Tools, Pingman Tools – Knowledgebase. Retrieved on August 12 2016, (http://www.pingman.con/kb/42)
[15] Application Layer ISO OSI Functionality and Protocols. Retrieved on August 2 2016. (http://www.highteck.net/EN/Application/ Application_Layer_Functionality_and_Protocols.html)
[16] Dean, Tamara (2010). Network+ Guide to Networks. Delmar. p. 519