






Universitat Autònoma de Barcelona

**ADVERTIMENT.** L'accés als continguts d'aquesta tesi queda condicionat a l'acceptació de les condicions d'ús establertes per la següent llicència Creative Commons:  [http://cat.creativecommons.org/?page\\_id=184](http://cat.creativecommons.org/?page_id=184)

**ADVERTENCIA.** El acceso a los contenidos de esta tesis queda condicionado a la aceptación de las condiciones de uso establecidas por la siguiente licencia Creative Commons:  <http://es.creativecommons.org/blog/licencias/>

**WARNING.** The access to the contents of this doctoral thesis it is limited to the acceptance of the use conditions set by the following Creative Commons license:  <https://creativecommons.org/licenses/?lang=en>



**How to Win with Everyone Fighting its own Battles.  
Extending Opportunistic Networking to **Heterogeneous**  
Environments.**

Adrián Sánchez Carmona  
Bellaterra, February 2017

Directed by  
Dr. Sergi Robles Martínez

**UAB**

L<sup>A</sup>T<sub>E</sub>Xstyle:

*deic.uab.cat/~cborrego/thesis.tar.gz*

Printed in:

*UAB*

SUBMITTED TO UNIVERSITAT AUTÒNOMA DE BARCELONA  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE  
DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE



Creative Commons 2017 by Adrián Sánchez Carmona  
Cover design and art by Alicia García Carmona

This work is licensed under a Creative Commons  
Attribution-NonCommercial-ShareAlike 3.0 Unported License.  
<http://www.creativecommons.org/licenses/by-nc-sa/3.0/>

I certify that I have read this thesis entitled “How to win with everyone fighting its own battles. Extending Opportunistic Networking to heterogeneous environments.” and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Bellaterra, February 2016

---

Dr. Sergi Robles Martínez  
(Advisor)

*Committee:*

Dr. Jordi Herrera Joancomartí  
Dr. Juan Carlos Cano Escribà  
Dr. Yves Mahéo  
Dr. Ramón Martí Escalé (substitute)  
Dr. Pietro Manzoni (substitute)

**Program:** *Doctor en Informàtica.*

**Department:** Departament d’Enginyeria de la Informació i de les  
Comunicacions.



*A la meva família*



# Abstract

**O**PPORTUNISTIC Networking usually focuses on homogeneous environments. Therefore, most proposals process all messages the same way and take for granted that all participants will behave for the sake of the network without any consideration. This thesis claims that this approach constrains the applicability of Opportunistic Networking. This thesis aims to provide tools to extend Opportunistic Networking by allowing its operation in heterogeneous environments, where a wide set of different devices, applications and users may coexist.

This thesis is presented as a compendium of publications where every publication presents a system designed to face a different heterogeneous environment. At the first one, we consider application heterogeneity, a network that could be shared by various applications that may require a different routing for their messages. In order to deal with this, we present a network architecture based on messages that carry their own application-defined routing code and an access control system that makes feasible the usage of application-defined contextual information. In the second one, we consider users heterogeneity, a network where every node is ruled by a selfish participant, so they could not be interested in using their resources to store and forward others' messages. To face this, we present a mechanism to keep track of the nodes' actions and an incentive scheme that punishes or rewards nodes based on their behaviour. This incentive scheme is designed to make sure that the best strategy for every participant is to cooperate with the network. In the third and fourth publications, we consider node heterogeneity, a network of volunteer nodes that do not want to collapse their devices nor share any information that hurt their privacy. We deal with this using a privacy-preserving georouting protocol that learns about nodes usual whereabouts and preserves the privacy of this information.





## Resum

**L**A recerca en Xarxes Oportunistes habitualment està enfocada a escenaris de tipus homogeni. En conseqüència, la majoria de les propostes assumeix que tots els missatges han de ser tractats de la mateixa manera i que tots els participants actuaran pel bé de la xarxa. Aquesta tesi afirma que aquest enfoc constreny l'aplicabilitat de les Xarxes Oportunistes, i té com a objectiu desenvolupar eines que permetin estendre la seva aplicabilitat en ambients heterogenis, on una gran varietat de dispositius, aplicacions i/o usuaris poden coexistir.

Aquesta tesi es presenta com a compendi de publicacions. A cada publicació es presenta un sistema dissenyat per fer front a un tipus d'ambient heterogeni. A la primera, es planteja l'heterogeneïtat de les aplicacions, considerant aplicacions que requereixen que els seus missatges siguin tractats de formes diferents però conviuen en una mateixa xarxa. Per fer front a aquesta situació, es presenta una arquitectura de xarxa basada en missatges que porten el seu propi codi d'encaminament, proporcionat per l'aplicació, i un sistema de control d'accés que fa viable l'ús d'informació contextual per millorar l'encaminament. A la segona, s'estudia l'heterogeneïtat dels usuaris, a través d'una xarxa formada per usuaris egoistes que no tenen cap interès en utilitzar els seus recursos per ajudar els altres. A fi de reconduir aquest comportament, es presenta un mecanisme per controlar les accions dels nodes i un sistema de càstigs i recompenses que castiga o premia els nodes en funció de les seves accions. Aquest sistema d'incentius garanteix que la millor estratègia possible consisteix en cooperar amb la xarxa. A les dues últimes publicacions, es considera l'heterogeneïtat dels nodes, mitjançant una xarxa de nodes que volen ajudar a la xarxa, però no estan disposats a acceptar un perjudici per la seva privacitat ni un consum excessiu dels seus recursos. En aquest cas, es presenta un protocol d'encaminament geogràfic respectuós amb la privacitat dels nodes, que aprèn quines són les zones visitades amb major freqüència per cada node i protegeix la privacitat d'aquesta informació.



## Resumen

**L**A investigación en el campo de las Redes Oportunistas se suele centrar en el estudio de escenarios homogéneos. Por lo tanto, la mayoría de las propuestas da por hecho que todos los mensajes deben ser tratados de la misma forma y que todos los participantes actuarán por el bien de la red. Esta tesis afirma que este enfoque restringe la aplicabilidad de las Redes Oportunistas, y se marca como objetivo mejorar el desarrollo de herramientas que permitan su funcionamiento en ambientes heterogéneos, donde conviven una gran variedad de dispositivos, aplicaciones y/o usuarios.

Esta tesis se presenta en el formato de compendio de publicaciones. En cada publicación se presenta un sistema diseñado para hacer frente a un tipo de ambiente heterogéneo. En la primera, se plantea la heterogeneidad de aplicaciones, en una red compartida por aplicaciones cuyos mensajes deben ser tratados de formas diferentes. Para afrontar esta situación, se presenta una arquitectura de red basada en mensajes que contienen su propio código de encaminamiento, proporcionado por la aplicación, y un sistema de control de acceso que hace viable a la propia arquitectura y permite el uso de la información contextual de encaminamiento definida por la aplicación. En la segunda, se estudia la heterogeneidad de los usuarios, cuyos intereses egoístas no priorizan el uso de sus recursos para encaminar los mensajes de otros. Para tratar de reconducir este comportamiento, se presenta un mecanismo para controlar las acciones de los nodos y un sistema de castigos y recompensas que premia a los nodos en función de sus acciones. Este sistema de incentivos garantiza que la mejor estrategia posible consiste en cooperar con la red. En las dos últimas publicaciones, se tiene en cuenta la heterogeneidad de los nodos, en una red de nodos que quieren ayudar pero no por ello están dispuestos a que su privacidad se vea afectada, ni a permitir un consumo de recursos excesivo. En este caso, se presenta un protocolo de encaminamiento geográfico respetuoso con la privacidad de los nodos, que aprende cuales son las zonas más visitadas por cada nodo mientras protege su privacidad.



# Acknowledgements

**T**HE next text is intended, mostly, to people whose first language is catalan. Therefore, I feel that these acknowledgements must be written in catalan.

*Plovia molt. Tot i això, el tren estava gairebé buit, però aquella senyora va decidir que havia de seure just davant meu, i vaig haver d'arronsar les cames. Hauria quedat molt lleig aixecar-me i canviar de seient. Vaig mirar per la finestra, només era qüestió d'un parell d'estacions. Vaig tornar a mirar al meu voltant. Què collons hi feia, jo, allà? Com dimonis m'ho havia fet per arribat a aquella situació? Dues preguntes per a les que encara no tinc massa clar que tingui la resposta. . .*

*Bàsicament, un bon dia em vaig trobar fent un doctorat. No a punt de començar-lo, o plantejant-me'l, no, ja l'estava fent. Ja portava un temps, cada dia havia d'agafar un tren per venir al despatx, i em quedaven alguns anys així per endavant. I allà estava jo, que no sabia com o per què m'havia ficat en aquesta situació. No obstant, la sortida era molt evident. Havia de tirar cap endavant, tampoc no podia ser tan complicat, no?*

*En realitat no ho ha sigut gaire, de complicat. En bona part, gràcies a la gent del departament d'Enginyeria de la Informació i les Comunicacions (dEIC ), una gent que, sense adonar-se'n ha creat un ambient de treball agradable, on em vaig sentir acceptat amb tota naturalitat i a on, després de dinar, sempre podem seure una estona al voltant d'una taula per parlar del que sigui que s'ens acudeixi en aquell moment. I es que res és tan acollidor com la normalitat.*

*No convertiré això en un llistat de noms, ja que són molts, i probablement em deixaré algú i quedaré malament. A qui sí que esmentaré és als companys de despatx. Ells són els que, per motius evidents, més hores m'han aguantat, i encara*

*que només sigui per això, l'Abraham, el Lao, el Gerard i el Marc es mereixen el meu agraïment. Tot i que, també val a dir-ho, no he estat tan malament quan he estat sol. . .*

*També he d'agrair moltes coses al Carlos, que ha sigut un recolzament molt important a l'hora d'entrar a l'obscur món de la recerca. A ell mai li agrairé prou que hagi llegit i corregit una quantitat ingent d'esborranys, molts d'ells probablement molt dolents, i que tot i així sempre hagi sigut capaç de trobar un comentari constructiu al respecte.*

*Al Sergi, el meu director de tesi, li agraeixo tot en general. Tot i que tinc alguna sospita respecte que potser ell va tenir alguna cosa a veure amb que jo em trobés fent un doctorat sense adonar-me, és evident que també ha tingut molt a veure amb que valgués la pena continuar i acabar-ho.*

*Al grup CASA en general, però especialment al Fred, al Maël i al Mathieu, els he d'agrair que la meva aventura a la Bretanya fos un èxit inesperat. Com que el francès que vaig aprendre allà és clarament insuficient, a ells s'ho diré en anglès.*

I thank the whole CASA team, but I specially thank Fred, Maël and Mathieu, because they really contributed to the unexpected success of my adventure in Brittany.

*Finalment, i per sobre de tot, és evident que a la meva família mai els agrairé prou tot el que han fet, fan i faran per mi. Aquesta tesi simbolitza el tancament d'una etapa que en cap cas hauria sigut possible de no ser per ells.*

*PD: Gràcies a totes aquelles persones que, quan hi ha places lliures al tren, seuen a qualsevol lloc que no sigui el que hi ha just al meu davant!*

*Terrassa, 15 de desembre de 2016*

# Contents

<b>Abstract</b>	<b>vii</b>
<b>Resum</b>	<b>ix</b>
<b>Resumen</b>	<b>xi</b>
<b>Acknowledgements</b>	<b>xiii</b>
<b>List of Figures</b>	<b>xviii</b>
<b>Part I Context</b>	<b>1</b>
<b>Chapter 1 Introduction</b>	<b>3</b>
<b>Part II Compendium</b>	<b>9</b>
<b>Chapter 2 Contributions</b>	<b>11</b>



<b>Chapter 3</b>	<b>Identity-based access control</b>	<b>17</b>
<b>Chapter 4</b>	<b>An asynchronous incentive scheme</b>	<b>33</b>
<b>Chapter 5</b>	<b>PrivHab: A privacy preserving georouting protocol</b>	<b>53</b>
<b>Chapter 6</b>	<b>PrivHab+: A secure geographic routing protocol</b>	<b>67</b>
<b>Part III</b>	<b>Discussion</b>	<b>87</b>
<b>Chapter 7</b>	<b>Results</b>	<b>89</b>
7.1	Identity-based access control for pro-active messages DTN . . . . .	89
7.1.1	A network of pro-active messages . . . . .	90
7.1.2	Pro-active messages' identity . . . . .	91
7.1.3	Identity-based access control . . . . .	91
7.1.4	Main contributions . . . . .	93
7.2	An asynchronous incentive scheme for DTN . . . . .	94
7.2.1	Receipt exchange protocol . . . . .	95
7.2.2	Incentive scheme . . . . .	97
7.2.3	Main contributions . . . . .	101
7.3	PrivHab: A privacy preserving georouting protocol for DTN . . . . .	103
7.3.1	The Habitat . . . . .	103
7.3.2	A Habitat-based routing algorithm . . . . .	105
7.3.3	Mapping negatives to perform subtractions . . . . .	105

7.3.4	The PrivHab's exchange of messages . . . . .	108
7.3.5	A multiagent-based system . . . . .	108
7.3.6	Main contributions . . . . .	110
7.4	PrivHab+: A secure geographic routing protocol for DTN . . . . .	110
7.4.1	Usage of Taxicab Geometry . . . . .	111
7.4.2	The elliptic habitat . . . . .	111
7.4.3	PrivHab+ . . . . .	113
7.4.4	Main contributions . . . . .	115
<b>Chapter 8 Conclusions</b>		<b>119</b>
<b>Bibliography</b>		<b>127</b>



## List of Figures

7.1	Fields of a pro-active message. . . . .	90
7.2	Operation of the Identity-based access control. . . . .	93
7.3	The reward and punishment scheme. . . . .	99
7.4	The enforcing mechanism. . . . .	99
7.5	Flow chart of a node's strategy. . . . .	101
7.6	The circular model of habitat. . . . .	104
7.7	PrivHab's routing algorithm. . . . .	105
7.8	Negative's mapping illustration. . . . .	106
7.9	Schema of the multiagent system. . . . .	109
7.10	Taxicab geometry distances. . . . .	111
7.11	The elliptic model of habitat. . . . .	112



## Part I

### Context

*“That’s why he will be fired [...]”*



*“To rely on rustics and not prepare is the greatest of crimes; to be prepared beforehand for any contingency is the greatest of virtues.”*

*The Art of War, SUN TZU*

# 1

## Introduction

**T**HE match has just finished, the coach, César, enters the press conference, he's heart broken. It's been a hard defeat. The team doesn't seem able to react. The season started off on the wrong foot, with the departure of their best player to their most hated rival, but no one suspected relegation was going to be so close. It's a terrifying situation. He knows that he will be fired by the end of the week, if not tonight. He also knows that there is no need to keep up appearances, not anymore.

-We seem Pancho Villa's army -he confesses, -there is no way to win if everyone fights its own battles.

The press agrees, the players are not that bad, and they have tried really hard, most of them were exhausted at the end of the match. But they are disorganised. They don't seem a team, but a bunch of people wearing the same shirt. That's why he will be fired, because that was his job, to cohesion them, to design a common action plan. But he failed.



However, the *Divison of the North*, the army commanded by José Doroteo Arango Arámbula, best known as Pancho Villa, did obtain significant victories, and even deposed a president, during the Mexican Revolution. This revolutionary army was composed by peasants, ranchers, cowboys, foremen and other elements of the Mexico's rural people. They did not wear uniforms, which probably contributed to project an image of a heterogeneous and disorganised bunch of people instead of a regular army ready to pass in review; an image that their enemies used to make fun of them; an image that has survived until today.

In Spanish, we use the expression “to seem Pancho Villa’s army” to describe “a disorganised group, a group that lacks coordination and where every element does as he or she wants”<sup>1</sup>. This expression has a crystal-clear negative connotation, because it assumes, as César meant, that a group of heterogeneous people, where every individual has his own goals and his own action plan, cannot be as efficient as a homogeneous group of people following a common course of action. That may be partially true, commanding a well-trained, homogeneous army with a strong chain of command is probably easier, and more effective, than commanding a division of revolutionary people. But the success of the latter demonstrates that being a heterogeneous group is not an obstacle to doing something. It forces to do things differently, and sometimes this could mean harder or slower, but it does not make things impossible.

At this point, the reader may think that this thesis is about sports coaching, the Mexican Revolution, Spanish idioms or military organisation. However, this thesis is about Opportunistic Networking<sup>2</sup>, and all the previous concepts and examples will be useful to introduce OppNet’s main problem. By definition, an OppNet is a network that lacks continuous connectivity, often due to the physical limitations of wireless communications, the sparsity and mobility of nodes, and energy or other environment-related constraints. OppNet uses a store-carry-and-forward routing approach to deal with connectivity disruptions, and needs to be flexible enough to operate with little or no infrastructure.

The OppNet research field usually focuses on homogeneous OppNets. Due to

---

<sup>1</sup>In French, there is a similar expression: “*ressemble à une armée mexicaine*”, that has a slightly different meaning.

<sup>2</sup>At the early days of this thesis, we considered Delay Tolerant Network (DTN) and Opportunistic Network (OppNet) to be synonyms. After the work done during the development of this thesis, we are convinced that it is more accurate to consider DTN as a subset of OppNet. However, the published works that form the compendium use the terms DTN and OppNet indistinctly.

this approach, OppNet deployment proposals usually assume that all nodes should process all messages the same way; and take for granted that all nodes will behave for the sake of the network. It is also assumed that many different applications can coexist without bothering each other. We build this thesis over the firm belief that this approach constrains the applicability of OppNet. The same way that, during a revolution, a leader cannot pick the army he wants and has to fight using his folk. Sometimes, a network has to deal with the scenario and the participants it has at hand. An available application-devoted devices' network cannot always be assumed, sometimes it is easier and cheaper, and therefore, more likely and feasible, to consider a heterogeneous network.

Besides, the increased miniaturisation of computers and mobile phones, and their last years' spread all over the society, has reinforced our claim. Nowadays, there is a vast amount of devices that do not need to be bought or deployed. However, to use them as a network, it is needed to respect and consider their own characteristics. Heterogeneity has to be considered even if it forces to do things differently. However, as we illustrated using the *Divison of the North's* example, different does not mean impossible.

The main objective of this thesis is to provide Opportunistic Networking with new tools to extend OppNet applicability by allowing its operation in heterogeneous environments. Concretely, the heterogeneity of the three main network's participants, nodes, applications and users, is considered. 1) Heterogeneity of applications: a network shared by applications that require a different routing in order to accomplish their mission. 2) Heterogeneity of users: a network where there are malicious users whose only interest is to mess with the network, to take advantage of its operation or to disrupt it. 3) Heterogeneity of nodes: a network where not all nodes are ruled by a central authority. So, they could not be willing to exchange private routing information or to use their resources to store and forward others' messages.

In summary, this thesis aims to provide tools to allow different applications to use the same network in different ways, to deal with the users' selfishness in networks where every user rules its own node, and to provide privacy to users that make their devices available to the network. The kind of tools that Pancho Villa used to command his army. The kind of tools that could have saved César's job.

## Objectives

In the following lines, we list the specific objectives of this thesis:

- Propose a security mechanism to allow a flexible OppNet architecture where the applications decide how their messages should be treated and routed.
- Propose a mechanism to incentivize nodes to use their resources to store, carry and forward others' messages, even if they are initially only interested in their own messages.
- Propose an efficient routing solution to a real application, using a heterogeneous OppNet where the privacy of the devices' owners is considered and protected.

## Structure

This thesis is presented as a compendium of publications. The rest of the thesis is organized as follows. Part II contains the compendium. Chapter 2 introduce every work and presents the argumentative thread that weaves the thesis as a whole. Chapters 3, 4, 5 and 6 reproduce the four published peer-reviewed international journal articles that form the core of this document. Then, Part III contains the discussion. Chapter 7 summarizes and discusses these publications' main systems, proposals and contributions. Finally, Chapter 8 concludes this thesis and provides some future lines of research.





## Part II

### Compendium

*“There is no way to win if everybody fights its own battles.”*



“Do not engage an enemy more powerful than you. And if it is unavoidable and you do have to engage, then make sure you engage it on your terms, not on your enemy’s terms.”

*The Art of War*, SUN TZU

# 2

## Contributions

**I**N this chapter, we introduce the contributions of the thesis, showing the argumentative thread that weaves it as a whole. After a little introduction, we refer to the articles of the compendium. The concrete background and the related work of every contribution are placed on the corresponding articles. The contributions are not ordered based on their date of publication, but following the order that best shows the progression of the overall work.

The first contribution of this thesis is a very specific access control system for an Opportunistic Network (OppNet) of pro-active messages. Based on the work done by *Borrego et. al.* in [BR13, BCR14], we define pro-active messages, a mechanism to adapt to the applications’ heterogeneity by using mobile code to let the applications customise their routing decisions. Obviously, to let the applications define their own routing code, but forcing them to make blind and generic decisions, does not make sense. For this reason, the keystone of this approach is a collection of contextual and application-related information that is first collected, maintained and stored by every node. Then, the routing code



accesses it. Finally, it is used by the application to make a routing decision.

However, this whole approach would lose all its purpose if we could not rely on the security of the information stored at the nodes. Identity-based access control provides feasibility to the pro-active message's OppNet by granting confidentiality and integrity of the information. This way, we avoid that any malicious message could access, delete or modify all the information stored in the nodes, disturbing this way the proper operation of the whole network. Identity-based access control is designed to operate in a concrete OppNet scenario, so it only makes use of cryptographic tools that are available in disconnected environments: two different hash functions and a symmetric key encryption algorithm.

*At this point, we encourage the reader to go to Chapter 3 for a better understanding of this thesis.*

The second contribution of this thesis is an asynchronous incentive scheme for OppNet. To spend some energy, occupy some buffer space, and use a fraction of an unknown-length connectivity window to store, carry and forward others' messages may lead to an early battery depletion, a filled buffer or the loss of a transmission's opportunity. That's not a problem when all the nodes are owned by a central authority that has deployed all the nodes and rules the whole network, but, what happens when every node is owned and operated by a different user? Individual users have their own needs and priorities. They may not be interested in the operation of the whole network, but only in using the network to send their own messages. These selfish users could perceive forwarding others' messages as a waste of resources.

However, if a big amount of users behave in a selfish way, the network will collapse and no one will accomplish their objectives because the messages will not be forwarded. Therefore, it is needed a mechanism that encourages the users to behave for the sake of the network, instead of trying to take advantage without collaborating. Our proposal revolves around an enforcing mechanism that rewards or punishes the users by giving them the Identity Based Cryptography (IBC) [Sha85] keys they need to send their messages. These keys are also used by a receipt exchange protocol to generate the receipts that proof that a user has been involved in the forwarding or delivery of a message. Those receipts are then asynchronously given to an element that uses them to re-build every message's chain of custody. This element follows the policy *guilty until proven otherwise* to decide the users that should be rewarded or punished. The rewards, punishments, and application of this policy, have been designed to ensure that a

Nash equilibrium [Nas50] is achieved when all users of the network behave in a fully cooperative way.

*At this point, we encourage the reader to go to Chapter 4 for a better understanding of this thesis.*

The next contribution of this thesis is a privacy preserving geographical routing protocol based on a multiagent system. It is straightforward that rural regions where the communication networks are unavailable or spotty are usually the ones where the access to knowledge and information would be more valuable to improve people's life conditions. Thus, they become excellent targets to apply the store, carry and forward techniques that characterise OppNet. Consequently, we identified an e-agriculture application of podcast distribution based on the work of the Non-governmental Organization (NGO) *Practical Action*<sup>1</sup> in Gwanda (Zimbabwe), and we proposed a way to improve it, by creating a cheap network of handheld devices carried by persons, and providing the required tools to achieve such a task.

The proposed protocol, named PrivHab, applies and develops some of the ideas presented by Martínez *et. al.* in [MCR<sup>+</sup>13] to this rural OppNet scenario. The Mobile Agent based DTN (MADTN) technology allows us to cope with the delays and disruptions of the network. At the same time, it provides enough flexibility and adaptability to deal with such an extreme environment thanks to the usage of mobile agents. PrivHab exploits the life-cycles of the persons carrying the nodes to make long-term predictions that could improve the decision making of the mobile agents. At this point, PrivHab has to face a common OppNet problem: the more useful it is a piece of information to a routing protocol, the more sensible it is to its owner's privacy. In order to solve this, PrivHab uses homomorphic cryptography [Gen09] to protect nodes' privacy by allowing the nodes to compare their usual whereabouts without sharing them with the rest of the network.

*At this point, we encourage the reader to go to Chapter 5 for a better understanding of this thesis.*

The last contribution of this thesis follows and expands the previous one. PrivHab+ is the result of the work done to improve PrivHab by overcoming some of its limitations and broadening its horizons. Mainly, PrivHab+ moves from Euclidean to Taxicab geometry in order to allow the usage of a higher variety

---

<sup>1</sup> The reader can find more details about this NGO at: <http://practicalaction.org/>

of shapes, *e.g.* the ellipse, to model the usual nodes' whereabouts. PrivHab+ is also focused on the usability and the applicability of the proposal, and provides a set of tools to improve it: *a)* a strategy to decrease the execution time by simultaneously processing multiple messages; *b)* an extensive security analysis proving that PrivHab protects the nodes' privacy against passive attacks; *c)* some simple countermeasures useful to greatly reduce the effectiveness of active attacks; and *d)* a re-design as a standalone geographical routing protocol that does not require of MADTN to operate.

*At this point, we encourage the reader to go to Chapter 6 for a better understanding of this thesis.*





“In all fighting, the direct method may be used for joining battle, but indirect methods will be needed in order to secure victory.”

*The Art of War*, SUN TZU

# 3

## Identity-based access control for pro-active messages DTN

Next, we reproduce the following article, which has been published on the international peer-reviewed journal *Security and Communication Networks*, a third quartile Journal Citation Reports (JCR) journal with an impact factor of 0.806.

*A. Sanchez-Carmona, S. Robles, C. Borrego. Identity-based access control for pro-active messages DTN. Security and Communication Networks (April 2016) vol 9, pp: 2323–2337. ISSN: 1939-0122. DOI: 10.1002/sec.1494*

This work was submitted for the first time on 01-Dec-2012, it was revised on 01-Sep-2014, and it was finally accepted on 16-Mar-2016.

## RESEARCH ARTICLE

# Identity-based access control for pro-active message's DTN

Adrián Sánchez-Carmona\*, Sergi Robles and Carlos Borrego

Department of Information and Communications Engineering (dEIC), Autonomous University of Barcelona (UAB), Bellaterra 08193, Spain

## ABSTRACT

Pro-active message's delay tolerant networks (DTNs) are based on the usage of mobile code to obtain messages that contain their own routing code. This architecture allows applications to use the same network in different ways. The keystone of this type of heterogeneous network is a collection of contextual and application-related information that is stored in every node and accessed by the messages' routing code. Access to that information must be protected in order to make the whole architecture feasible; the operation of the network has to be secure, and attacks of information poisoning have to be avoided. We propose an identity-based access control system for pro-active message's DTN based on tools that are available in DTN networks, like symmetric key encryption and hashes. Our system grants confidentiality and integrity to the contextual information and solves the question of messages needing to use distributed information stored in nodes to route properly. The proof of concept of identity-based access control in a certain kind of application demonstrates the feasibility of the proposal. The comparison between our proposal and other access control systems shows that identity-based access control is the only system that fits well with the special characteristics of pro-active message's DTN. Copyright © 2016 John Wiley & Sons, Ltd.

## KEYWORDS

cryptographic applications; heterogeneous communications network security; security for distributed networks; cryptographic mechanisms; DTN access control; security in DTN

## \*Correspondence

Adrián Sánchez-Carmona, Department of Information and Communications Engineering (dEIC), Autonomous University of Barcelona (UAB), Bellaterra 08193, Spain.

E-mail: adria.sanchez@deic.uab.cat

## 1. INTRODUCTION

Delay tolerant networks (DTNs) have a set of unique characteristics that make it very difficult to find a routing protocol that can be applied successfully in any situation. Some of them are the non-contemporary of communications, the existence of significant delays and mobility patterns that allow nodes to be isolated from their neighbours during variable lapses of time. Because of that, there is not a *de facto* routing standard broadly extended and used. This makes DTN routing a challenge.

In an environment where the topology of the network changes very quickly and the nature and characteristics of the applications that use it are very different, the one-fit-all solution could be a chimera. Flexibility is a key aspect, and the only proposal that tries to obtain it, Haggie [1], uses a dynamic approach with many pros and one important contra: the utilization of an array of routing algorithms

makes the deployment very hard and costly in any scenario where the set of applications grows, or simply changes, over time.

In response to that, we propose a model in which mobile code is used to obtain messages that contain its own routing code. This way we can offer applications the opportunity to differentiate themselves from one another and use the same network of different specific forms. From now on, there will be no need to treat different messages the same way.

Using a routing code carried by the messages to provide a blind routing where all applications dispose of the same, or no, information to make decisions would not be useful. It is necessary to let the applications use their own information in order to achieve the desired flexibility. This information should be application-oriented and designed to allow messages to decide the best route based on the specific criteria of each application.

Without the appropriate security model, this whole model is unusable. We cannot rely on a network if any malicious message could access, delete or modify all the information stored in nodes, thereby disturbing the proper operation of one or more applications. If the information used during routing becomes poisoned, then the whole architecture will lose all its purpose. Therefore, it is crucial to dispose of a system that secures the information and prevents possible damaging attacks occurring, a system that makes feasible this new approach.

The obvious solution for this problem is called access control. However, almost all access control proposals are thought to protect resources as an abstract term. This approach is general and can be used in many scenarios, with many different objectives, but the drawback of that generalization is that it restricts the tools that can be used to protect resources. For example, access control can be used to protect a file, the usage of a printer or the usage of the network, but we cannot cypher a printer or sign the usage of the network. Besides, access control systems usually rely on public key infrastructure (PKI) [2], a requirement that cannot be met in a DTN.

Therefore, an access control system that would grant only messages authorized by the application access to the information owned by that application is required. This system has to take into account:

- The system has to be capable of operating in a DTN environment, considering all of its unique characteristics.
- Every application must decide exactly the messages that are authorized to access to the information it manages. All other messages' access to this information must be prohibited.
- The system has to grant access to authorized messages fast enough to not diminish their routing opportunities.

We propose identity-based access control for pro-active message's system, a secure access control system for messages with routing code focused on protecting a particular resource: structured and organized information used by messages during routing. In our model, each application is an abstract entity that can use a set of different mobile codes to route messages to its destination. These codes usually work together with a common objective. The network provides distributed application-related information used by the mobile codes in order to communicate between them, gather context-aware data, use it during routing, store intermediate results or receive instructions or parameters provided by the application.

Our proposal brings feasibility to the whole network architecture by granting confidentiality and integrity of information. In order to achieve this, we implement access control using two different hash functions and symmetric key encryption when a mobile code requests access information. The hash functions are applied to the code itself

and are used to recover the cryptographic key needed to decrypt the information.

### 1.1. Structure of this paper

The following sections of this paper are structured this way: Section 2 explains the environment in which our research is focused, the new routing model of pro-active message's based DTN. Section 3 provides an overview of the state of the art of access control for mobile code and related work available in the literature. Section 4 studies the best way to identify and authenticate, which messages are authorized to access the information. Section 5 formally describes our proposal of identity-based access control, while Section 6 addresses the main security aspects and issues related to our proposal. Section 7 shows an application based on our proposal over a DTN routing problem and evaluates its performance and feasibility. Finally, Section 8 concludes this paper and provides some future lines of research.

## 2. PRO-ACTIVE MESSAGES, A PARADIGM'S SHIFT

In this section, we will explain the environment in which our research has been undertaken. We will explain first delay tolerant networks, and how the traditional approach deals with the challenges they present. Then we will focus on pro-active messages as our central concept of a new delay tolerant networks paradigm and we will study the utility of providing application-related information to be used during routing. Finally, we will explain the three phases of operation of pro-active message's based DTN.

### 2.1. Delay tolerant networks

Delay tolerant networks [3] are networks where the low connectivity rates, the high and variable delays and the impossibility to establish simultaneous end-to-end paths make communications very challenging. Delay tolerant networking is usually used in regions where the communication networks are unavailable or spotty, where the lack of a fixed infrastructure and the mobility of the nodes of the network allows them to be isolated from their neighbours during variable lapses of time.

The cornerstone of DTN is the store-carry-and-forward strategy [4], which is a way for turning a weakness as the mobility of the nodes into a strength. Using this strategy, nodes store and carry their messages until they opportunistically establish contacts with other nodes of the network, and they make use of these unpredictable contacts to forward the messages. This process is repeated until the messages eventually reach their destination. The simultaneous end-to-end paths that cannot be established are somehow substituted by a mix between physical distances travelled by nodes carrying messages and node-to-node transmissions when other nodes are contacted.



Delay tolerant network routing protocols focus on the decision-making when two or more nodes establish contact. What messages should be forwarded? How many copies of every message should be created? Is there a message not worth sending? What messages should be dropped last? etc... These questions are addressed by routing protocols, who try to ask these questions to maximize the performance of the network.

The traditional DTN approach consists in selecting a routing protocol and deploying it in every node of the network. This way all nodes behave the same way in terms of routing, and most important, all messages are treated equally all over the network.

## 2.2. Pro-active message-based delay tolerant network

Our proposal revolves around the concept of changing the traditional approach to routing. Instead of using routing algorithms deployed in every node of the network, we move the routing code from the nodes to the messages, using mobile code as is suggested in [5,6]. In this paradigm, nodes provide the necessary infrastructure to let the messages decide the way towards its destination, but the own message carries the routing algorithm that decides how to route it. The type of DTN where this research is focused is called pro-active message's DTN. The key concept of this networks is the **pro-active message**. As depicted in Figure 1, a pro-active message contains four fields:

- (1) **Source address:** address of the node that sent the message.
- (2) **Destination address:** address of the node where the message has to be delivered.
- (3) **Content:** the data from the application.
- (4) **Routing Code:** mobile code that has to be executed in every node the message arrives to in order to choose where the message should be forwarded to. Routing code executes a function  $f$  which, as defined below, operates above the list of the current neighbours and a set of contextual information and returns the subset of neighbours where the message has to be forwarded to.

$$forwardTo = f(neighbourList, information) \text{ forwardTo} \subseteq neighbourList$$

Source address	Destination address
Content	
Routing Code	

**Figure 1.** Schema of fields of a pro-active message, a message that carries its own routing code.

### 2.2.1. Application driven routing.

The most important advantage of pro-active message's DTN approach is that applications themselves can use the network any way they want. It is important to note that every application knows its own needs better than any other and, similarly, the best routing decision for an application can be different to the best routing decision for another application. And nobody but the applications themselves can know that. Consequently, by using pro-active messages, applications can take their own optimal routing decisions.

Very different applications can coexist inside the same network. However, pro-active messages cannot take optimal decisions without having enough information. In this model, the usage of an application-related information is crucial. Routing cannot be driven by applications unless they can decide everything about the information they want to use. The way to achieve this is to let the applications use ontologies to structure their information. Then, the same information is seen simultaneously in two very different ways: while the network sees it as a stream of bytes, the applications see it as a structured knowledge. In our proposal of pro-active message's based DTN, this routing information is stored in nodes in the Routing Information Database (RIDB). Eventually, nodes can exchange entries of the RIDB between them in order to allow applications to spread their updates of the entries.

Therefore, pro-active messages must access their application-related information in every node of their route. Pro-active messages must also be able to modify the information, updating its value in a way that can be known or understood by the application itself.

### 2.2.2. Operation of the network.

During the operation of the pro-active message's based DTN, we can differentiate between three phases, depending on the type of the messages sent and the actions performed by the users, the applications and the administrators of the network.

- (1) **Deployment phase:** nodes become initialized, and no pro-active messages are sent. Network administrators must be able to access all the nodes. Ideally, deployment should be applied only once before users and applications start using the network.
- (2) **Standard phase:** pro-active messages sent by users are the only kind of message that travels through the network. Nodes can become isolated during lapses of time, network topography varies quickly and there are not simultaneous end-to-end paths between nodes. This is the most common phase.
- (3) **Update phase:** applications update their information entries, and nodes exchange these entries between them in order to spread the updates of the routing information. Applications share the network with users during this phase. Therefore, pro-active messages continue travelling through the network.

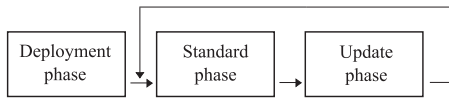


Figure 2. Flow diagram of the three phases.

This phase starts every time an application decides to share its update and ends when all nodes have received it or after a determined amount of time.

Figure 2 shows the flow chart of the three phases mentioned earlier; the deployment phase is the initial phase of the network followed by the standard and update phases which form an endless cycle.

### 3. STATE OF THE ART

In this section, we introduce the concept of mobile code related with access control. Following, we present some of the most representative research community's solutions for access control for mobile code and analyse their characteristics. We focus especially on those characteristics that can become problematic in a DTN.

#### 3.1. Access control for mobile code

Mobile code [7] is code sourced from remote, possibly unknown or untrusted systems or networks, but executed locally on the system. Some examples of mobile code are mobile agents [8], downloadable code, executable content, active capsules, remote code, etc. Because of its own nature, mobile code does not fit well with the traditional access control approach that runs any process with the same user privileges or capabilities that executed it.

In the following paragraphs, we provide an overview of some of the ideas and proposals from the research community currently used to solve the access control for mobile code problem.

#### 3.2. Role-based access control

Role-based policies [9], also referred to as role-based security, are the most extended and used approach in medium and large organizations [10]. Role-based policies regulate users' access to the information depending on the activities they carry out in the system. Roles are defined on the basis of the actions associated with a concrete working activity. Then, instead of giving authorizations to each user every time they want to access a resource, users are given authorization to adopt roles. Finally, each requested access is allowed or denied depending on the roles adopted by the user, and Figure 3 illustrates this process.

It is important to note that role-based access control is designed to be used inside a closed system. Using it in a DTN is problematic because the access control module

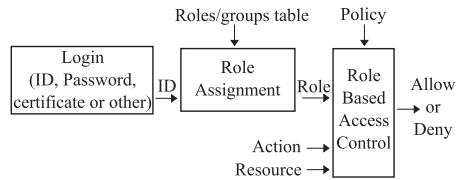


Figure 3. Schema of role-based access control.

needs a fair role assignment. This is hard to achieve if we have to separate the access control module and the role assignment module into different nodes that could become unlinked them during the access control process. As a result, although role-based access control does not use any cryptographic tool, it cannot be used in a DTN schema without using cryptography to secure transmissions.

#### 3.3. Access control based on trust

Cryptographic tools such as digital signature [11] are used by software authors to sign the code, which is distributed together with the author's digital certificate and signature. When the code arrives a host and is about to be executed, the signature is validated, and then the system grants the code access to all requested resources exclusively in the case that the author of the code is trustworthy.

There are lots of proposals designed to enhance this trust model. These proposals are based on the assumption that it is impossible for a user to know all the world's trustworthy software developers and trust their programmes. Trust management [12] tries to establish trust relationships between users and developers that do not know each other. The key point is not to try to establish the authorship of the code but establish the credentials of the code instead [13]. An example of access control based on trust is KeyNote Trust Management System [14], where users delegate into trusted credential issuers that are expected to have direct or indirect relationships with potential requesters. Other proposals [15] have dynamically updated trust relationships as more information is collected from code execution or uses recommendations from peers to calculate the scale of the trust on the unknown requester [16].

#### 3.4. Security-by-contract

Security-by-contract [17] goes a step further and uses digital signature to link together the code, not only with the author but also with a contract [18] that specifies the actions that the code will perform during its execution. This model also relies on trusting the code author, but the idea is that only permissions that are needed are granted, or at least, permissions that the author considers necessary for the code (open and write a file, create a socket, access a database, send packets to a specific domain, etc.). Thanks to security-by-contract, users are able to know exactly not

only who wrote a programme but also what actions will be executed if the contract is accepted.

Of course, security-by-contract needs to use cryptography to bind the code to a contract in order to guarantee the authenticity and the origin of the contract. Otherwise, a malicious entity could write its own contract and use it to gain access to a resource from unauthorized code.

Additionally, if an application decides to delete or revoke the permissions given to a code to access some information, it cannot be carried out with this system. When a contract is attached to a signed code, there is no way to revoke or change it unless the certificate used to sign the contract is revoked.

### 3.5. History-based access control

History-based access control for mobile code, such as Deeds [19], gathers and stores information about the actions executed and the resources accessed by the mobile code. Subsequently, this historical information is used to link the process to several pre-defined profiles (editor, browser, terminal, etc.). Finally, those profiles are used to decide about the requests made processes (e.g. a programme opens a local file is labelled as 'editor', but when the same programme tries to open a socket, the creation of the socket is not allowed because it is an action allowed for 'browsers' but not for 'editors').

This approach is extremely specific to control the execution of code downloaded from an unknown source towards Internet, or another similar environment, and it is hardly applicable to other situations. History-based access control does not match with the requirements of the environment. This proposal is designed to work in scenarios where the resources are very different and the only danger occurs when two or more are used together, or one after another. In pro-active message's DTN, the only action performed by codes usually is *access information*, and a 'safe' or 'dangerous' behaviour does not exist (based on the information accessed by the code).

### 3.6. Common characteristics

It is important to note that most of the access control proposals for mobile code are thought and focused on solving two specific cases: foreign code executed inside the browser and execution of code downloaded from an unknown source from the Internet. As a result, they usually assume some weaknesses derived from the nature of the open environment in which they are designed to operate (e.g. users cannot know all programmes they will execute and the amount of different programmes and software developers is enormous and keeps growing).

Another important aspect to consider is that security is, in almost most cases, cryptographically based on PKI, a very extended model that fits well with Internet's characteristics but not with DTN characteristics. The access to a trusted third party, to the certificate's repository or to

revocation lists cannot be available in a DTN, additionally, the distribution of the certificates among nodes remains unsolved [3,20–23]. The reason for this is that PKI is based on assumptions such as permanent point-to-point connectivity or the small delays at the link layer that cannot be applied in DTN.

## 4. IDENTITY OF PRO-ACTIVE MESSAGES

In this section, we will see the specific necessities of the pro-active messages identification's process and the need of that identification in order to provide access control to a given information. Then, we will discuss different ways of message authentication, and we will study the pros and cons of our solution.

### 4.1. Identification of pro-active messages

Supposing that a message arrives a node and requests access to the information of the application *A*, it is necessary to identify and authenticate that message in order to decide if it is authorized to access the information about this application.

Because the data field can contain any value, the quantity of different pro-active messages that an application can create is infinite. However, during routing, the data field is meaningless, and it does not play any role until the message is delivered to its destination.

We have to consider a hypothetical attacker that alters a pro-active message modifying its routing code that tries to, using a forged routing code, access or modify *A*'s information. This is one of the points we want to fight in our research, to develop an access control system that fits with this situation.

### 4.2. Authentication of pro-active messages

In general, the authentication process can be seen from four different perspectives:

- (1) to find out something anything that nobody else can know, for example, a password;
- (2) to find out something that nobody else has, for example, a key;
- (3) to find out how to do something in a particular way, for example, a signature; and
- (4) to have a unique characteristic, for example, a fingerprint or a DNA chain.

Here, we will examine why the first three approaches are not valid when considering the authentication problem of the pro-active messages in a DTN environment.

The first two points can be analysed together. A software entity cannot differentiate between *having something* and *knowing something*. A message cannot travel with

physical elements as a key or a card, but it can travel with any data, as a password or a digital key. The usage of portable passwords has an important drawback: the theft of a password compromises immediately the whole system, because from that moment, the attacker could use send a malicious message with the stolen password to access a protected information from its routing code.

The third approach makes us think directly about the usage of a well-known cryptographic technique: digital signature. However, digital signature leans in the PKI, a schema that cannot be applied in DTN, as we have seen in Section 3.

We find the solution to our problem in the fourth point, earlier. When a routing code tries to access an entry, we can determine if it is authorized by analysing it. In that case, we are using something that it is inherent to the code, something that cannot be copied or stolen because it forms part of what that routing code is. This is an idea previously pointed out in [24] to manage package distributions of software.

Using a simile with conventional identity-based access control system, we can say that we analyse the DNA of the message in order to identify and authenticate it, the routing code of the message, like the DNA of a living being. This way, the message does not need to *know* or *have* or *do* anything; it will be authenticated for what it *is*.

#### 4.3. Identify code using hash

Messages can use routing codes of different lengths. In order to use routing code to decide about access control, it is preferable to manage a fixed-size element. With just a hash function applied on the routing code, it is possible to obtain a binary sequence that identifies it and, at the same time, differentiates it from any other.

Our system uses the hash of the routing code to identify messages. This way, if a message is intercepted, the only way a hypothetical attacker could use the obtained data to access information is to create a message with the obtained routing code. Regardless, a behaviour like this would not compromise the security of the system. Note that if the routing code of the new message is exactly the same of the original message, then it would not cause any malicious action over the stored information.

## 5. IDENTITY-BASED ACCESS CONTROL

In this section, we will analyse the requisites that our access control system has to satisfy. Secondly, we will present the Authorized Hashes Set (*AHS*), the Entries Set (*ES*) and the algorithms used to add content to these sets. Thirdly, we will explain the way of using these sets and the algorithms to achieve an effective access control system. Finally, we will explain the characteristics of identity-based access control.

### 5.1. Notation

For the sake of clarity, we provide Table I, which contains the notation used to refer to each one of the different elements that will appear in this and the next Section, and a brief description of its meaning. From now on, we will use this notation.

### 5.2. Requirements

The developed system uses the hash value of  $c_i$  from each pro-active message in order to identify it and provide control access to the protected information  $I_j$ , which is stored at the custodian. The following requisites need to be granted:

- The system should grant access to all authorized  $c_i$  to information  $I_j$ .
- The system should grant secrecy and integrity to all protected information  $I_j$ . No unauthorized  $c_i$  (or other processes) should be able to access or modify it.
- The system should allow nodes to send entries between them to spread among the nodes of the network the updates made by the applications.
- The system should add a minimum impact in terms of resource's consumption and execution time. This is necessary to avoid conflicts with small connectivity windows, typical in DTN scenarios.

### 5.3. A system based on two sets

To decide if  $c_i$  is authorized to access  $I_j$ , our proposal is based in the usage of two sets, the set of entries and the set of stored information.

#### 5.3.1. Authorized Hashes Set.

The *AHS* is the key element of our proposal, and it is the only set that has to be deployed into nodes during the

**Table I.** Notation of all elements used from now on.

Notation	Meaning
$i$	Identifies a message.
$c_i$	Routing code of message $i$ .
$c'_i$	Routing code forged to replace $c_i$ .
$j$	Identifies an application.
$I_j$	Information of application $j$ stored in the RIDB.
$I'_j$	An updated version of $I_j$ .
$h()$ and $h'()$	Two different hash functions.
$E_k()$	Symmetric key encryption function of algorithm $E$ using key $k$ .
$D_k()$	Symmetric key decryption function of algorithm $E$ using key $k$ .

deployment phase. *AHS* contains a collection of triplets as explained next:

$$(j, h'(c_i), E_{h(c_i)}(k_j)) \quad (1)$$

where

- $j$  identifies the application to which the Information  $I_j$  belongs and is used to identify the triplet together with the result of applying a hash algorithm to  $c_i$ .
- $k_j$  is the symmetric key needed to cipher and decrypt the protected information  $I_j$ , and it is cyphered with  $E$  using the result of another hash algorithm over  $c_i$  as key.

### 5.3.2. Entries Set.

The *ES* is a collection of pairs as follows:

$$(j, E_{k_j}(I_j)) \quad (2)$$

where

- $j$  identifies the application to which the Information  $I_j$  and allows us to make a quick search of a stored entry identified by  $j$  without trying to decrypt every entry in the set.
- $I_j$  is the protected information, ciphered using a symmetric encryption algorithm using a key  $k_j$  that is stored in the *AHS* (see following Sections 5.4 and 5.5 in order to know how the key  $k_j$  is obtained and stored).

### 5.4. Creation of the Authorize Hashes Set

When new information  $I_j$  has to be stored and protected in one or more nodes, Algorithm 1 is used to add to the *AHS*, and *ES* sets the needed triplets and pairs.

---

#### Algorithm 1 Storage of protected information

---

**Input:**  $I_j$ : Information to be protected.

$j$ : Identifier of the information.

$M$ : Set of messages  $i$  with access to  $I_j$ .

**Output:**  $\emptyset$

- 1: Generate a random key  $k_j$ .
  - 2: Cypher  $E_{k_j}(I_j)$ .
  - 3: Ad to *ES* the pair  $(j, E_{k_j}(I_j))$ .
  - 4: **for**  $i \in M$  **do**
  - 5:   Obtain routing code  $c_i$ .
  - 6:   Calculate  $h(c_i)$  and  $h'(c_i)$ .
  - 7:   Cypher  $E_{h(c_i)}(k_j)$ .
  - 8:   Add  $(j, h'(c_i), E_{h(c_i)}(k_j))$  to *AHS*.
  - 9: **end for**
  - 10: **return**  $\emptyset$
- 

### 5.5. Using the set of entries to control access to an entry

When a message  $i$  arrives to a node and its routing code  $c_i$  requests access the information of application  $j$ , Algorithm 2 has to be applied. This algorithm recovers the keys needed to decrypt that information only if  $c_i$  is authorized to access  $I_j$ .

---

#### Algorithm 2 Access Control

---

**Input:**  $i$ : Message requesting access to the information.

$j$ : Identifier of the information.

**Output:**  $I_j$ : Information requested, identified by  $j$ .

- 1: Obtain  $c_i$  from message  $i$ .
  - 2: Calculate  $h'(c_i)$ .
  - 3: Search in *AHS* a triplet that matches with  $(j, h'(c_i), \textit{Sciphered\_key})$ .
  - 4: Store in *Sciphered\\_key* the corresponding value.
  - 5: Calculate  $h(c_i)$ .
  - 6: Decrypt  $D_{h(c_i)}(E_{h(c_i)}(\textit{Sciphered\_key})) = k_j$ .
  - 7: Search in *ES* a pair that matches with  $(j, \textit{Sciphered\_info})$ .
  - 8: Store in *Sciphered\\_info* the corresponding value.
  - 9: Decrypt  $D_{k_j}(E_{k_j}(\textit{Sciphered\_info})) = I_j$ .
  - 10: **if**  $I_j$  has been successfully decrypted **then**
  - 11:   **return**  $I_j$
  - 12: **else**
  - 13:   **return**  $\emptyset$
  - 14: **end if**
- 

### 5.6. Modification of a protected entry

Algorithm 3 is used when a message  $i$ , previously authorized to access  $I_j$ , modifies it and decides to overwrite  $I_j$  with the newer version  $I'_j$ .

---

#### Algorithm 3 Modification of an entry

---

**Input:**  $i$ : Message trying to modify the entry.

$j$ : Identifier of the entry.

$I'_j$ : New version of  $I_j$ .

**Output:** **true** or **false**.

- 1: Execute first 9 lines of algorithm 2.
  - 2: **if**  $I_j$  has been successfully decrypted **then**
  - 3:   Cypher  $E_{k_j}(I'_j)$ .
  - 4:   Remove the pair  $(j, E_{k_j}(I_j))$  from *ES*.
  - 5:   Add the pair  $(j, E_{k_j}(I'_j))$  to *ES*.
  - 6:   **return true**
  - 7: **else**
  - 8:   **return false**
  - 9: **end if**
- 

### 5.7. Optimization

Algorithms 2 and 3 are very similar (the first nine lines are the same). Furthermore, in any situation, Algorithm 2 (used to access to an entry) has always

been executed before Algorithm 3, which is used to update the entry. Therefore, Algorithm 3, the most time-consuming of the three, can be optimized and executed quicker if certain values (obtained from intermediate calculations by Algorithm 2) are kept in the memory for a while. This way, we can avoid executing repeatedly the same operations.

**5.8. Characteristics of identity-based access control system**

Identity-based access control system uses two different hash functions; the output of one of them is used to identify the subject, and the output of the other is used to cypher and decrypt the key needed to access the information.

Our proposal uses a discretionary policy, and it is designed loosely following the guidelines of an access control list (Figure 4). This approach consists on associating each system's object with all the authorized subjects that can access it and the actions they can perform over the object. Subsequently, the list is consulted when a subject requests access to an object in order to know if it is authorized or not.

Instead of relating the name of every subject with its permissions in order to allow or deny access to information, our system relates the identifier of each subject (the result of a hash function) to the key needed to decrypt the information (Figure 5).

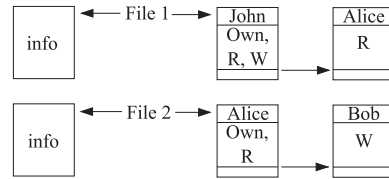


Figure 4. Schema of an access control lists implementation.

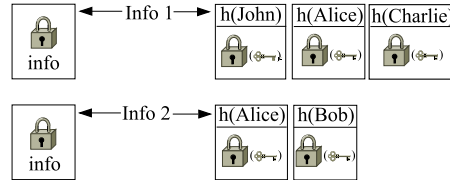


Figure 5. Schema of identity-based access control.

This key can be retrieved only by the proper subject (because it is encrypted with the result of another hash function is key). Moreover, the identifier of each piece of information is related with the encrypted information itself; therefore, the information cannot be decrypted and accessed without getting the key.

Table II. Comparison between identity-based access control and other systems.

Access Control System	Identity	Contract	RBAC	History-based
<b>Cryptographic tools used</b>	Hash and symmetric key encryption	Hash and digital signature X	Nothing	Nothing
<b>Distribution of keys</b>	Deployment	Continuous X	N/A	N/A
<b>Need secure transmissions</b>	No	No	Yes X	No
<b>Application requirements</b>	Yes	Yes	Yes	No X
<b>Type of resource</b>	Structured information	Any resource	Any resource	Any resource
<b>State of resource</b>	Cyphered	Original state	Original state	Original state
<b>Initial Deployment</b>	Platform and sets	Platform	Platform	Platform
<b>Add new permissions</b>	Add to set	Create new contract	Change policies	Change profiles
<b>Delete old permissions</b>	Remove from set	Impossible X	Change policies	Change profiles
<b>Add a new resource to the system</b>	Add to set	Add file	Change policies	Modify platform X
<b>Applicability</b>	✓	X	X	X

Using the identity-based access control instead of simply an access control list system has two advantages: it provides security against a certain type of situations that we need to avoid, such as a remote attack using routing code forgery, and it allows nodes to spread updates of routing information securely (see Section 6 for more details).

**5.9. Access control in pro-active message's delay tolerant network**

We encourage the reader to look at Table II, which provides a qualitative comparison between our proposal and other access control systems. Table II is structured according to the following format: each column references one of the four compared systems (identity-based access control, security-by-contract, role-based access control and history-based access control). Each row of the table refers to a specific characteristic of access control systems, chosen based on the needs of the environment in which we are working.

Because of this comparison, we can conclude that there are no other systems that fit well with the characteristics of DTNs because they are designed to solve problems in other types of scenarios and present issues when they have to operate in a pro-active message's DTN. We cannot find any system that improves identity-based access control system in that environment.

**6. SECURITY OF THE ACCESS CONTROL**

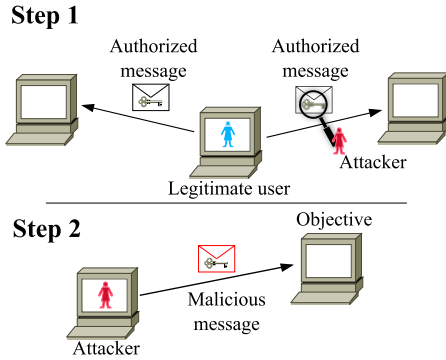
In this section, we will discuss security offered by our identity-based access control system. With this in mind, we will analyse three different scenarios. In the first scenario, an attacker tries to create a malicious pro-active message that could access a protected information. In the second scenario, an attacker intercepts an exchange of routing information between nodes and tries to access these entries. In the last scenario, an attacker that has compromised a node tries to access all entries, using the sets *AHS* and *ES*. Finally, we provide some conclusions about the three scenarios.

**6.1. Security against routing code forgery**

A remote attacker that wants to compromise the entry *I<sub>j</sub>* needs to make a pro-active message *i'* with a routing code *c'<sub>i</sub>* that is passed as an authorized code *c<sub>i</sub>* in order to access an entry (Figure 6).

The first step of this attack consists of intercepting a message *i* with an authorized *c<sub>i</sub>* that can access *I<sub>j</sub>*. Then, the attacker can use the non-secret functions *h* and *h'* to calculate *h(c<sub>i</sub>)* and *h'(c<sub>i</sub>)*. At this point, the attack will be successful if the attacker finds a *c'<sub>i</sub>* that accomplishes both *h(c'<sub>i</sub>) = h(c<sub>i</sub>)* and *h'(c'<sub>i</sub>) = h'(c<sub>i</sub>)* with *c'<sub>i</sub> ≠ c<sub>i</sub>*.

Therefore, the attacker has to realize a double pre-image attack [25,26] against two different hash algorithms.



**Figure 6.** Schema of the attack. (1) Attacker intercepts an authorized message. (2) Attacker generates a malicious message using the information obtained in the previous step.

Assuming that *h* and *h'* are in line with the following statements:

- They are safe against pre-image attacks. A pre-image attack consists in finding a value *a* such as *h(a) = b* when *b* is known beforehand and a hash function is considered safe against it if the probability of finding a value *a* that fits this condition is  $\frac{1}{2^n}$ , where *n* is the length in bits of the output of *h* (so  $2^n$  is the total amount of possible *h* outputs).
- The length of the output of *h* and *h'* is *n* and *m*, respectively.
- The two algorithms are totally independent, so an attacker cannot obtain any information from either of them using the cryptanalysis of the other.

Thereby, the double pre-image attack consists in finding a *c'<sub>i</sub>* such as *h(c'<sub>i</sub>) = h(c<sub>i</sub>)* and *h'(c'<sub>i</sub>) = h'(c<sub>i</sub>)*. The probability of finding this value is  $\frac{1}{2^n} \cdot \frac{1}{2^m} = \frac{1}{2^{n+m}}$ , because both condition need to be met simultaneously.

If the attacker intercepts *r* different messages authorized to access the same entry, then the probability of a successful attack goes up to  $r \cdot \frac{1}{2^n} \cdot \frac{1}{2^m} = \frac{r}{2^{n+m}}$  because every message authorized and intercepted provides a new target for a double pre-image attack, and all the attacks can be made in parallel to maximize the probability of success. This growth does not compromise security in any sense because any *r* possible is some orders of magnitude smaller than  $2^{n+m}$ .

Choosing two hash functions *h* y *h'* whose output's size *n* and *m* are considered safe and for which there are not any known algorithm that can reduce the complexity of a pre-image attack, then our system is safe against that kind of attack.

**6.2. Security against update interception**

In order to improve the operation of a pro-active message's DTN, nodes spread routing information updates among

the network during the update phase by sending pairs  $(j, E_{k_j}(I_j))$  from the *ES* to other nodes. This is a delicate situation, because an attacker that intercepts this transmission can obtain lots of entries of the *ES*.

The first thing the attacker has to do is to find the pair that contains the identifier  $j$  of the entry he wants to compromise. Next, the attack consists of breaking the ciphering provided by the symmetric key ciphering algorithm  $E$ , because there is no way to obtain the key  $k_j$ , which is stored in the *AHS* and has not been transferred anywhere.

If the chosen  $E$  algorithm is safe, the probability of a successful attack is  $\frac{1}{2^n}$  where  $n$  is the size of the key  $k_j$  used to cipher the information (so the attack consists in trying with all possible combinations of  $n$  bits to find the key). Thereby, we can conclude that the system is safe against these kind of attacks and that nodes can send data from the *ES* without compromising the security of the network.

### 6.3. Compromising the node

This is the worst possible case. In this scenario, an attacker compromises the infrastructure of a node and wants to access all routing information of the RIDB. In this situation, the success of the attack is dependent upon acquiring pro-active messages with authorization to access all the information. A situation like this is improbable and can compromise the security of any access control system.

The attacker tries to compromise all entries of *ES* using the data that can find in sets *AHS* and *ES*. The data from the *AHS* are not useful to the attacker unless he has intercepted some messages. When the attacker intercepts a message  $i$  with a routing code  $c_i$  that is authorized to access  $I_j$ , then this entry can be compromised. In that case, the attacker can use  $c_i$  and the non-secret hash algorithms  $h$  and  $h'$  to calculate  $h(c_i)$  and  $h'(c_i)$  and access  $I_j$  using the algorithm explained in Section 5.

Although the system is not safe against an attack of this type, identity-based access control makes the success of the attack harder. Storing the information cyphered with a key that it is not present in the node forces the attacker to obtain the key using a different method, sniffing the network traffic or waiting until an authorized message arrives the node.

### 6.4. Security results

We can conclude that identity-based access control for pro-active message's DTN makes the network safe against routing code forgery and update interception and message interception attacks in the active adversary mode. Thanks to identity-based access control, an attacker cannot create a malicious message with a routing code that compromises the security of information stored in a node by accessing it without permission.

Furthermore, nodes can easily spread updates of the routing information among the network safely, because an attacker that intercepts one or more of these updates cannot be able to decrypt it nor to access the needed keys. This

contribution improves both the security of the network and the performance of its operation.

Even in the worst-case scenario attack, a compromised node where the attacker gains full control of the system, our proposal would make the success of the attack harder, and it would force the attacker to wait until the interception of some authorized messages before being able to access any cyphered information.

## 7. PERFORMANCE EVALUATION

To test the feasibility of our proposal and to evaluate its operation and its performance, we have tested identity-based access control on a specific application. In this section, we will present the chosen application and examine the most important implementation decisions taken during the development of a proof-of-concept software of our system. Finally, we will provide some conclusions obtained from the proof of concept.

### 7.1. Scenario of application: PROtocols for the Single European Space

The year 2020 will mark a turning point in the field of European air traffic management (ATM) and control, as the next evolution in ATM is expected to become fully operational and deployed. The Single European Sky initiative will unify the heterogeneous air traffic control models used by each country, transforming the European airspace into a single integrated air management scenario. In order to achieve this, the system will require an unprecedented level of connectivity between all the participants to support the massive increase in data exchanges taking place between the terrestrial, aerial and satellite platforms. In conclusion, a sort of 'aerial ATM Internet' is being created, composed of mobile and collaborative nodes that will integrate distributed and/or geographically sparse services.

The scenario where we have tested and monitored our system is based on PROSES (PROtocols for the Single European Space) [27]. In PROSES, a network of heterogeneous nodes, which in this case are aircraft and ground control centers, along with unmanned vehicles, for non-critical data exchange is created. This network is based on pro-active message's DTN, and a dozen of different applications with varied routing needs coexist.

The authors deployed a small-scale version of the proposed delay tolerant network scenario in a small aerodrome near Seville at the end of 2011 [28], where 2 days of flying tests were performed using two mobile nodes, located in an Radio Controlled (RC) fixed-wing aircraft and an RC helicopter, and a stationary ground station. Statistics about the scenario and the characteristics of the network utilization were collected and used here to study the feasibility of the presented identity-based access control system inside PROSES environment. The average number of messages carried by every node is 10,



with an average size of 10 KB. There are 10 different types of information with sizes between 50 KB and 4 MB. Field tests showed that smaller pieces of information are more commonly used than bigger pieces of information, in 82.5% of the cases access takes place with information between 100 and 750 KB. Connectivity windows in PROSES are typically located between 20 and 30 s.

**7.2. Implementation decisions**

In order to implement a pro-active message's DTN, we have modified MobileC [29], a standard IEEE FIPA compliant mobile agent's platform to allow its use as the central element of a pro-active message-based DTN. MobileC was chosen because it is specially designed for real-time and resource constrained applications and because it provides support to the execution of mobile code. The modifications are based in the usage of the libraries OpenSSL [30], Libxml2 [31] and Raptor [32]. The most important modifications are as follows:

- Change of approach: MobileC is no longer a mobile agent's platform and now is an element that can send, receive and route pro-active messages with its own routing code.
- Implementation of a module that performs the neighbour discovery. This module is very necessary to use the platform in DTN environments where the neighbour's list of any node could change very quickly.
- Inclusion of the identity-based access control system presented in this paper, implemented using the algorithms described in Section 5.

The chosen hash and symmetric key cypher algorithm are  $h = \text{SHA2-256}$ ,  $h' = \text{SHA1}$  and  $E = \text{AES-256 CBC}$ . This choice was made taking into account that  $h$  and  $h'$  must be independent algorithms. Besides, the size of the key of  $E$  and the output of  $h$  was considered in order to avoid having to truncate or pad these values. Notice that they are related because  $h(c_i)$  is used as a key of algorithm  $E$  when cyphering and decrypting  $E_{h(c_i)}(k_j)$ .

**7.3. Implementation of the routing information**

We have defined a collection of routing information related to different applications, which is a set of files with Resource Description Framework (RDF) statements. Each statement can be stored in an unprotected form, which means it is a public entry formatted as plain text, or can be protected by our system, which means that it has to be decrypted before it is accessed. Both public and protected entries are stored in the corresponding file using Base64. This is a way to assure that both kinds of entries are accessed the same way, and the results and differences of execution's times obtained in experimentation are not affected by the parsing process.

Box 1 shows an example of a public entry and a cyphered one.

```
<rdf:Description rdf:about="http://
  test.com/publ">
  <field:data1>Value1</field:data1>
  <field:cyphered> 0 </field:cyphered>
</rdf:Description>
<rdf:Description rdf:about="http://
  test.com/priv">
  <field:data2>h+Iy3+RwDfn/qcPEF2Y5oA=
  </field:data2>
  <field:cyphered> 1 </field:cyphered>
</rdf:Description>
```

Box 1: Example of cyphered and unencrypted entries.

**7.4. Deployment**

Figure 7 shows the deployment diagram of the developed proof-of-concept software. As seen earlier, pro-active MobileC is made up of three main elements: the neighbour discovery module, the identity-based access control system and the original MobileC core. The two sets *AHS* and *ES* are stored in two different files, and there is a collection of RDF [33] files that contain the routing information. These files cannot be accessed directly by pro-active messages, and they need to send an 'information request' to the platform when they want to consult the information.

**7.5. Test environment**

The laboratory environment where we have tested our system is a machine with an Intel Pentium 4 3.3-GHz processor of 32 bits and 512 MB of RAM memory and a

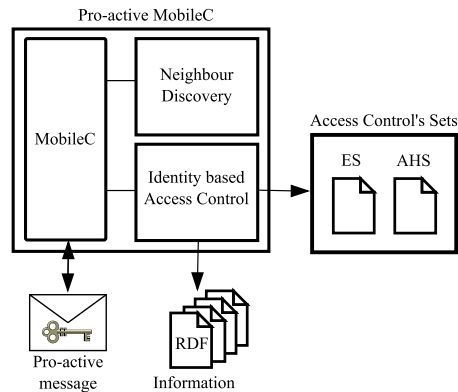
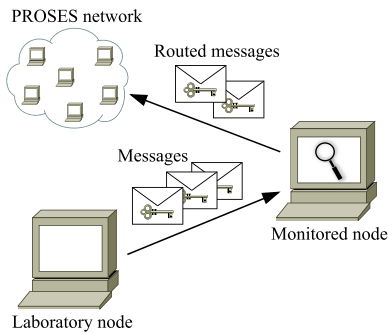


Figure 7. Deployment diagram of pro-active MobileC. AHS, Authorized Hashes Set; ES, Entries Set.



**Figure 8.** Schema of the proof-of-concept tests. PROSES, PRO-protocols for the Single European Space.

GNU/Linux O.S. with a 3.3.2 kernel. The node is equipped with the pro-active MobileC platform, and it is connected to a network interface through which it receives and sends pro-active messages.

We have modelled the incoming traffic of pro-active messages we injected in that node in the basis of the statistics exposed in the previous paragraph and information obtained in PROSES field tests. Figure 8 shows a schema of the operation of the test.

In order to obtain conclusive results, we have routed 2200 messages, and we have measured the routing time spent by each one. Half of these messages were routed using public entries, while the other half was routed using protected entries, routing codes of those messages, which have a structure like the one shown in Algorithm 4.

---

#### Algorithm 4 Structure of message's routing code

---

**Input:** destination: Message ultimate destination.

identifier: Identifier of the information.

**Output:** nextHop: Where to forward the message.

```

1: information = getInformation(identifier, this)
2: // Process the information to decide the next hop
3: // ...
4: return nextHop or ∅

```

---

The sizes of the entries range included 50, 100, 200, 300, 500 and 750 KB and 1, 2, 3, 4 and 5 MB. Measurements allowed us to obtain average routing times and their corresponding standard deviations. We can evaluate the impact of our system by analysing the routing time taken by each message according to the size and the type of information accessed during routing.

## 7.6. Results

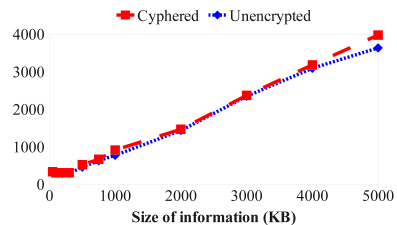
To obtain Figure 9, we have grouped the samples according to the size and the type (cyphered or unencrypted) of

the accessed information. We can observe how the overhead included by our access control system becomes linear as the size of the information increases. Also, we can see that the time needed for routing when the accessed entry is cyphered is slightly longer (6.67% in average) than when it is unencrypted.

The obtained average routing times and its standard deviations for access to unencrypted information are shown in Table III. Table IV shows the same information for access to cyphered entries. In both cases, we have calculated the average number of milliseconds per kilobyte taken during access by dividing the average routing time (ms) by the size of the entry (KB).

The amount of time spent per kilobyte is especially important to be able to see the performance's trend of the system for larger sizes of information. Almost all operations executed during the access of the information are independent of its size. Thereby, we can see that when the information accessed is small, each kilobyte is very costly because of the overhead of these operations. However, from 750 KB, the time per kilobyte is stabilized around the 70–80 ms/KB (a few milliseconds more for the cyphered entries).

Figure 10 shows the routing time according to the size of the cyphered entry of the most commonly used sizes



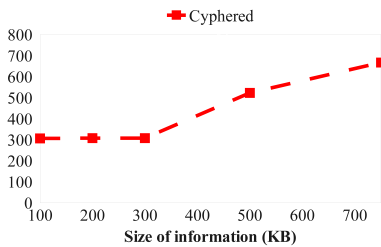
**Figure 9.** Routing time (ms) according to the size and the type of the accessed entry.

**Table III.** Routing times when the accessed entry is unencrypted.

Size (B)	Average routing time (ms)	Standard deviation (%)	Time per KB (ms/KB)
50 K	326.25	6.20	6.53
100 K	299.99	6.00	3.00
200 K	285.08	5.79	1.43
300 K	290.33	5.77	0.97
500 K	457.31	4.56	0.91
750 K	629.03	6.34	0.84
1 M	771.05	4.63	0.75
2 M	1434.15	12.72	0.70
3 M	2346.62	13.60	0.76
4 M	3092.70	8.63	0.76
5 M	3636.26	8.16	0.71

**Table IV.** Routing times when the accessed entry is cyphered.

Size (B)	Average routing time (ms)	Standard deviation (%)	Time per KB (ms/KB)
50 K	335.61	5.88	6.71
100 K	306.35	5.78	3.06
200 K	308.06	6.02	1.54
300 K	307.91	5.64	1.03
500 K	523.24	6.45	1.05
750 K	667.38	5.68	0.89
1 M	912.76	5.12	0.89
2 M	1465.50	9.99	0.72
3 M	2369.94	11.64	0.77
4 M	3181.93	13.75	0.78
5 M	3978.00	9.69	0.78

**Figure 10.** Routing time (ms) according to the size of the entry.

in PROSES, from 100 to 750 KB. We can see that the needed routing time is stabilized during the first half and then grows linear during the second half. Considering the size of PROSES' connectivity windows (10–30 s), differences of 0.3 s cannot be considered as significant, even if an increment of 0.3 s represents a 120% growth in routing time (note that, in the same interval, the size of the entry has grown a 650%).

The average routing time without accessing any routing information is 4.8ms (with a huge standard deviation of 39%). That time grows to 0.3–0.6 s when messages access cyphered entries of 100–750 KB. Considering the PROSES characteristics previously explained, we conclude that the overhead introduced by our system is totally acceptable. Ten messages that access cyphered entries of less than 1 MB can be routed in less than 10 s, one half of the smallest connectivity window. Identity-based access control system could be used even if the size of the used information or the number of queued messages grows, for example, 10 messages that access entries of 4 MB can be routed in 31.8 s, and up to 22 messages that access entries of 500 KB–1 MB can be routed in less time than the minimum connectivity window.

Therefore, we can say that our proposal is feasible and its performance is good enough to be used in the studied DTN application; besides, as explained in Section 6, the

usage of identity-based access control for pro-active message's DTN would make the PROSES network safe against routing code forgery and update interception.

## 8. CONCLUSIONS AND FUTURE WORK

Most of DTN routing algorithms do not take into account the close relationship between applications and the way these use the network. DTN based on pro-active messages solve that problem by allowing applications to define their own routing code and the information they need to consult during routing. This way, every application can chose differently the best way to get to its destination and can even decide how to spread over the network creating copies of the messages. This is what we call 'application driven routing'.

In this paper, an access control system for routing information in a pro-active message's based DTN has been presented. The fundamental contribution of this proposal lies in the usage of the own identity of the routing code that tries to access the information in order to decide if it is authorized. This system uses two different hash functions, one to identify the routing code that requests access to information and the other to, together with a symmetric cyphering algorithm, protect the information and grant its confidentiality and integrity.

This proposal will not only improve security in pro-active message's based DTN but also make the whole paradigm become feasible. Thanks to identity-based access control, an attacker will not be able to create a malicious message with a routing code that compromises the security of information stored in a node by poisoning it. Furthermore, nodes can spread updates of the routing information among the nodes of the network safely. In the worst-case scenario attack, our proposal would not provide security if an attacker compromised the whole infrastructure of the attacked node. However, it would make the success of the attack harder, forcing the attacker to acquire (by intercepting authorized messages) the keys to access cyphered information. So it provides a last line of defence even in a situation where all other systems fall.

The application of our proposal in a specific context, PROSES, allowed us to evaluate the feasibility and the performance of the identity-based access control system. In conclusion, the performance of the system is good enough to be used in the pro-active message's DTN scenario.

We also studied the characteristics of our proposal and other access control systems in order to analyse if they can be applied to the environment of our research, pro-active message's DTN. We found that only identity-based access control system can be used properly in that environment because of its unique characteristics and design.

Future lines of research include but are not limited to widening the access control system to make it usable out of DTN scope, developing a mechanism to spread the information and enhance the operation of the network and improving the system to allow *a posteriori* modifications

of the set of rules used to decide which codes are authorized or not to access information. It is also possible to use the same principles used here to control mobile agent's access to local resources in DTN scenarios.

## ACKNOWLEDGEMENT

This work has been partially funded by the Science and Innovation Ministry of Spain towards the reference project TIN2010-15764.

## REFERENCES

1. Diot C *et al.* *Haggle project*. Available from: <http://www.haggleproject.org/> [Accessed on 12 April 2016].
2. Herzberg A, Mass Y, Michaeli J, Ravid Y, Naor D. Access control meets public key infrastructure, or: Assigning roles to strangers. *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, IEEE Computer Society, Washington, DC, USA, 2000; 2–14.
3. Farrell S, Cahill V. *Delay- and Disruption-tolerant Networking*. Artech House, Inc.: Norwood, MA, USA, 2006.
4. Scott K, Burleigh S. *Bundle Protocol Specification*, November 2007. RFC 5050 (Experimental).
5. Castillo S, Robles S, de Toro M, Borrell J. Seguridad en protocolos de encaminamiento para redes dtn. *Actas de la XI Reunión Española de Criptología y Seguridad de la Información*, Tarragona, September 2010; 383–388.
6. Borrego C, Robles S. Seguridad en la planificación de agentes móviles en redes dtn. *Actas de la XI Reunión española de criptología y seguridad de la información*, Tarragona, September 2010; 389–394.
7. Cugola G, Ghezzi C, Picco G, Vigna G. Analyzing mobile code languages. In *Mobile Object Systems Towards the Programmable Internet*, vol. 1222, Vitek J, Tschudin C (eds), Lecture Notes in Computer Science. Springer: Berlin / Heidelberg, 1997; 91–109.
8. Lange DB, Oshima M. Seven good reasons for mobile agents. *Communications of the ACM* 1999; **42**(3): 88–89.
9. Sandhu RS. Role-based access control. In *Advances in Computers*, Vol. 46. Elsevier, 1998; 237–286.
10. O'Connor AC, Loomis RJ. Economic analysis of role-based access control. *Technical Report*, National Institute of Standards and Technology, 2010.
11. Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM* 1978; **21**(2): 120–126.
12. Ruohomaa S, Kutvonen L. Trust management survey. *Proceedings of ITRUST 2005, Number 3477 in LNCS*, Springer-Verlag, Paris, France, 2005; 77–92.
13. Chakraborty S, Ray I. Trustbac: integrating trust relationships into the rbac model for access control in open systems. *Proceedings of the Eleventh ACM Symposium on Access Control Models and Technologies*, ACM, New York, NY, USA, 2006; 49–58.
14. Blaze M, Ioannidis J, Keromytis AD. Experience with the keynote trust management system: Applications and future directions. In *Proceedings of the 1st International Conference on Trust Management*, Springer-Verlag, Hamburg, Germany, 2003; 284–300.
15. Lin C, Varadharajan V, Wang Y, Pruthi V. Trust enhanced security for mobile agents. *Seventh IEEE International Conference on E-Commerce Technology, 2005. CEC 2005*, Washington, DC, USA, 2005; 231–238.
16. Giang PD, Hung LX, Lee S, Lee YK, Lee H. A flexible trust-based access control mechanism for security and privacy enhancement in ubiquitous systems. *International Conference on Multimedia and Ubiquitous Engineering*, Berlin, Germany, 2007; 698–703.
17. Bielova N, Dragoni N, Massacci F, Naliuka K, Siahaan I. Matching in security-by-contract for mobile code. *Journal of Logic and Algebraic Programming* 2009; **78**(5): 340–358, The 1st Workshop on Formal Languages and Analysis of Contract-Oriented Software (FLACOS'07).
18. Helm R, Holland IM, Gangopadhyay D. Contracts: specifying behavioral compositions in object-oriented systems. *SIGPLAN Not.* 1990; **25**(10): 169–180.
19. Edjlali G, Acharya A, Chaudhary V. History-based access control for mobile code. *Secure Internet Programming*, 1999; 413–431.
20. Asokan N, Kostianinen K, Ginzboorg P, Ott J, Luo C. Towards Securing Disruption-Tolerant Networking. *Nokia Research Center, Tech. Rep. NRC-TR-2007-007*, 2007.
21. Aniket K, Gregory M. Z, Urs H. Anonymity and security in delay tolerant networks. *Third International Conference on Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007*, Nice, France, 2007; 504–513.
22. Seth A, Keshav S. Practical Security for Disconnected Nodes. *1st IEEE ICNP Workshop on Secure Network Protocols, 2005.(NPSec)*, Boston, Massachusetts, USA, 2005; 31–36.
23. Asokan N, Kostianinen K, Ginzboorg P, Ott J, Luo C. Applicability of identity-based cryptography for disruption-tolerant networking. *MobiOpp '07: Proceedings of the 1st International Mobisys Workshop on Mobile Opportunistic Networking*, ACM, New York, NY, USA, 2007; 52–56.

24. Hollingsworth JK, Miller EL. Using Content-Derived Names for Caching and Software Distribution. *Tenichal Report*, Technical Reports of the Computer Science Department, 1998.
25. Hoffman P, Schneier B. *Attacks on Cryptographic Hashes in Internet Protocols*, 2005. RFC 4270.
26. Kelsey J, Schneier B. *Second preimages on  $n$ -bit hash functions for much less than  $2^n$  work*, Cryptology ePrint Archive, Report 2004/304, 2004. <http://eprint.iacr.org/>.
27. Giuditta N, Robles S, Viguria A, Castillo S, Cordero M, Fernández L. Proses - network communications for the future european atm system. *Proceedings of the International Conference on Application and Theory of Automation in Command and Control Systems*, Barcelona, Spain, May 2011; 94–98.
28. Martínez-Vidal R, Castillo-Pérez S, Robles S, Sánchez-Carmona A, Borrel J, Cordero M, Viguria A, Giuditta N. Mobile-agent based delay-tolerant network architecture for non-critical aeronautical data communications. In *Distributed computing and artificial intelligence*, vol. 217, Advances in Intelligent Systems and Computing. Springer International Publishing: Salamanca, Spain, 2013; 513–520.
29. *MobileC*. Available from: <http://www.mobilec.org/> [Accessed on 12 April 2016].
30. *OpenSSL Project*. Available from: <http://www.openssl.org/> [Accessed on 12 April 2016].
31. *The XML C parser and toolkit of Gnome*. Available from: <http://xmlsoft.org/> [Accessed on 12 April 2016].
32. *Raptor RDF Syntax Library*. Available from: <http://librdf.org/raptor/> [Accessed on 12 April 2016].
33. Klyne G, Carroll JJ. *Resource Description Framework (RDF): Concepts and Abstract Syntax*, 2004. W3C Recommendation.

“All warfare is based on deception.”

*The Art of War*, SUN TZU

# 4

Endeavouring to be in the good books.

Awarding DTN network use for  
acknowledging the reception of bundles

Next, we reproduce the following article, which has been published on the international peer-reviewed journal *Computer Networks*, a second quartile JCR journal with an impact factor of 1.446.

*A. Sanchez-Carmona, S. Robles, C. Borrego. Endeavouring to be in the good books. Awarding DTN network use for acknowledging the reception of bundles. International Journal on Computer Networks (June 2015) vol. 83, pp: 149-166. ISSN: 13891286. DOI: 10.1016/j.comnet.2015.03.007*

This work was submitted for the first time on 21-May-2014, it was revised on 03-Feb-2015, and it was finally accepted on 05-Mar-2015.



## Endavouring to be in the good books. Awarding DTN network use for acknowledging the reception of bundles



Adrián Sánchez-Carmona\*, Sergi Robles, Carlos Borrego

Department of Information and Communications Engineering (dEIC), Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain

### ARTICLE INFO

#### Article history:

Received 21 May 2014

Received in revised form 3 February 2015

Accepted 5 March 2015

Available online 18 March 2015

#### Keywords:

Incentive schemes

Delay Tolerant Networks

Nash equilibrium

Non-repudiation

Receipt exchange

Cooperation

### ABSTRACT

This paper describes an incentive scheme for promoting the cooperation, and, therefore, avoiding selfish behaviours, in Delay Tolerant Networks (DTN) by rewarding participant nodes with cryptographic keys that will be required for sending bundles. DTN are normally sparse, and there are few opportunistic contacts, so forwarding of other's bundles can be left out. Moreover, it is difficult to determine the responsible nodes in case of bundle loss. The mechanism proposed in this paper contributes to both problems at the same time. On one hand, cryptographic receipts are generated using time-limited Identity Based Cryptography (IBC) keys to keep track of bundle transmissions. On the other hand, these receipts are used to reward altruistic behaviour by providing newer IBC keys. Finally, these nodes need these IBC keys to send their own bundles. When all nodes behave in a cooperative way, this incentive scheme works as a virtuous circle and achieves a Nash equilibrium, improving very much the network performance in terms of latency. The scheme is not difficult to implement, and it can use an already existing IBC infrastructure used for other purposes in a DTN.

© 2015 Elsevier B.V. All rights reserved.

## 1. Introduction

Delay Tolerant Networks (DTN) [1] are networks with low connectivity rates and high and variable delays. They support two main networking operations: (1) *to route* own traffic, to transmit a message from its origin to any intermediate node and (2) *to forward* other's traffic, to receive a message, store and carry [2] it for some time to transmit it when it is possible to its destination or to another intermediate node.

In these networks, all nodes are usually interested in routing and use their resources for their own benefit. On the other hand, all nodes demand that others forward their messages, but no one has a special interest in forwarding because it consumes energy and fills buffer space without

any direct benefit. Therefore, it is necessary a mechanism to keep track of their behaviour: to know if they are forwarding, if they are refusing to forward or if they are losing or dropping messages. This knowledge about the performed actions of nodes must be used to encourage them to be cooperative and behave for the benefit of the network.

To solve this situation, we created an incentive scheme where nodes are required to forward if they want to route. The incentive scheme is based on a receipt exchange protocol. The receipt exchange protocol makes use of the principles of non-repudiation protocols to provides a way to discover which nodes are suspect of non-cooperative behaviour. The exchanged receipts are used by an incentive scheme that requires nodes to forward if they want to route, and punishes non-cooperative behaviours.

In the presented scheme, nodes need cryptographic keys, not only to forward messages and perform the receipt exchange protocol but also to route their own messages, because running out of keys means becoming isolated.

\* Corresponding author.

E-mail addresses: [adria.sanchez@deic.uab.cat](mailto:adria.sanchez@deic.uab.cat) (A. Sánchez-Carmona), [Sergi.Robles@uab.cat](mailto:Sergi.Robles@uab.cat) (S. Robles), [Carlos.Borrego@uab.cat](mailto:Carlos.Borrego@uab.cat) (C. Borrego).

When the incentive scheme detects suspicious nodes, it punishes them by delivering them lesser amounts of keys or even forcing these nodes to wait a while without keys. Therefore, Identity Based Cryptography (IBC) [3] keys act as an enforcing mechanism, because nodes are forced to forward messages to obtain keys, and they want the keys to route their messages.

Our main contributions can be summarised as follows.

- A receipt exchange protocol designed to overcome the limitations that the non-repudiations protocols present when applied in DTNs. The cryptographic receipts are generated by the incentive scheme using IBC keys that are used to track the actions of the nodes.
- An asynchronous incentive scheme for DTN that uses the policy “guilty until proven innocent” to punish and reward the cooperative nodes. This scheme uses the receipts generated by the receipt exchange protocol and rewards nodes by delivering IBC keys to the nodes.

In this article, we proof that, on the presented incentive scheme, node behaviours form a Nash equilibrium when all participants behave in a fully cooperative way. Besides, the simulations show that, even if nodes have low demand of keys and try to be as uncooperative as they can afford, our system improves the performance of the network in terms of latency.

The remainder of this paper is organised as follows: Section 2 presents the related work, in the field of incentive schemas and in the field of non-repudiation protocols and signature exchanges. Section 3 presents a receipt exchange protocol designed to overcome the limitations of non-repudiation protocols when applied to DTNs. Section 4 explains the incentive schema, its asynchronous operation and how we relate the amount of keys given to the nodes with their balances. Section 5 analyses the choices to be made by the network’s participants and demonstrates that all nodes cooperating and being honest form a Nash equilibrium. Section 6 details the performance evaluation. Section 7 details the simulations and presents the obtained results. Finally, Section 8 concludes the article and provides some future lines of research.

## 2. Related work

In this section, we will present the state-of-the-art of incentive schemes. As our proposal relies not just on the incentive scheme but also on the receipt exchange protocol to build the chain of custody of every message, we will summarise how other incentive schemes keep track of the actions performed by the nodes to reward them. Finally, we will briefly summarise some non-repudiation protocols, a field that we used to develop the receipt exchange protocol presented in Section 3.

### 2.1. Incentive schemes

Incentive schemes have been an active research field; Mobile Ad Hoc Networks (MANET) [4] and DTN are usually the kinds of networks where this research is focused.

There are proposals that are heavily related to the concrete application they were designed to solve: dissemination of advertisements, special offers, discount coupons, and so on over a MANET. In [5], a central authority approves and marks each advertisement to track it, nodes that obtain the advertisements deliver receipts to the relaying node, and relaying nodes use these receipts to claim a reward for their work, but the central authority only rewards relaying nodes when the advertisement is used by an end user. *Coupons* [6] is based on the simple idea of adding the name of each relaying node to the transferred coupon, when the coupon is finally used a central authority rewards all nodes that had relayed it. *SMART* [7], is based on the same principles, but it is adapted for general purpose messages in DTN.

The incentive schema called *Pi* [8] includes the policy of payment-rewarding inside each message, giving to the relaying nodes the opportunity to choose, at every message, if the reward will be enough to compensate the usage of resources. As in almost all schemes, a central authority does the credit clearance after the message arrives at its final destination.

Other proposals, such as *Nuglets* [9], are based on the idea of a counter of virtual currency that every node maintains and updates when they send messages, subtracting the cost of sending a message or relaying others messages, adding a payment for relaying. Obviously, nodes are motivated to cheat and alter the content of the virtual currency counter, therefore these proposals are supported by a trusted and tamper resistant hardware module that provides security to the incentive schema.

In [10,11] the performance of the network is improved by forcing nodes to exchange messages one by one in a *Barter* manner, this way nodes are incentivised to accept and carry messages they are not interested in but they could exchange later by more interesting ones. In this proposal, nodes are restricted to exchange sets of messages of the same size, and no measures are taken against cheating, so in each transaction one party can deliver one message less than the other without being punished. Selfish nodes could benefit from this weakness to obtain all messages they are interested in without forwarding any other one, performing transactions where they receive one message and do not deliver one.

Several works present incentive schemes that, from a game theory perspective [12–14], grant that nodes should behave honestly and provide services to others because it is in its own interest. These kind of schemes, like *Sprite* [15], a scheme designed for Ad Hoc Networks, base their operation on the rationality of nodes. In *Sprite*, relaying nodes obtain a receipt of a message together with the message, and deliver the receipt to a central authority. The central authority re-builds the chain of custody of a message to charge the sender and reward the relay nodes when the message arrives at its final destination.

*RAPID* [16,17] is a DTNs incentive schema strongly related to a routing algorithm. This proposal, and many others, such as [18–21] are based on the Tit-for-tat principles: nodes reciprocate good or bad behaviour on part of the peer, they low service to a neighbour when they detect that a neighbour is misbehaving.



This research topic has been studied even from an economic point of view. In [22], the fear of an audit that proves that a node has been misbehaving becomes the only incentive for nodes to behave honestly. A similar approach is used in *iTrust* [23], where the audit is substituted by a probabilistic inspection that reduces a 90% the effort that the *Trusted Authority* has to do. Other works [24,25] are focused on the global aspects of the network's economy like taxes, inflation, deflation, "feast and famine" cycles, and effects of isolation and usually do not care about how to track the actions performed by each node.

There are proposals that do not try to incentivise selfish nodes to act in an unselfish way, but try to mitigate the impact of such behaviours in the network. For example, in [26], authors try to mitigate routing misbehaviours in DTN using random nodes of the network as witnesses of each transaction to detect nodes that do not relay messages. Then, the results of these observations are used to re-send messages across another path, or to decrease the reputation of selfish nodes.

## 2.2. Tracking the actions of the nodes

All incentive schemas need to track the actions done by the nodes of the network. It is needed to distribute rewards to nodes with fairness. The most used mechanism is called layered coin.

The layered coin consists of two or more layers, the first, which is also named the base layer, is generated by the source of the message and is sometimes used to indicate payment policies, the class-of-service requirements, or other remuneration conditions. During the subsequent message relaying process, each intermediate node will generate a new layer based on the previous layers by appending a non-forgable digital signature. This new layer is also called the endorsed layer, which implies that the forwarding node agrees to provide forwarding service.

Using endorsed layers, it is easy to track the propagation path and determine each intermediate node by checking the signature of each endorsed layer, but the layered coin is only complete when a message arrives at its final destination, and intermediate nodes do not have any proof of their cooperation by forwarding a message. The usage of the layered coin always leads to a synchronous schema where relaying nodes are all rewarded at the same time, after the message has arrived at its final destination.

Another mechanism to track the actions of the nodes is the *watchdog*. In [27], each node monitors the next node in the path of a message to check if the message is relayed or not. This solution is related to the characteristics of Ad Hoc Networks and is not applicable in Delay Tolerant Networks.

## 2.3. Non-repudiation protocols

Our approach is to provide the nodes with a mechanism to obtain a receipt in exchange for its cooperation. The receipt must be changed with the message in a fair way to avoid cheating. A situation where a node obtains a receipt but does not relay the message is as undesirable as a situation where a node relays the message but does

not obtain the receipt. This leads us to consider non-repudiation protocols [28].

Non-repudiation protocols provide ways to exchange messages with receipts in a fair way. The majority of the proposals are based on a Trusted Third Party (TTP) that acts as a moderator or intermediary of each transaction, to ensure the protocol is performed correctly by all participants (online TTP) or to repair damages when one participant cheats to obtain an advantage over the other (offline TTP). Proposals that use a TTP, either online or offline, are not viable due to lack of end-to-end connectivity in DTNs.

Non-repudiation protocols without TTP are based on the idea of splitting the message into  $n$  parts and send the parts one by one, receiving an acknowledgement for each one [29]. A variation of this idea can be found in [30], where the message is cyphered and sent at the beginning of the transmission and the key needed to decrypt it is sent by parts. These kinds of protocols are called probabilistic because the receiver can, with a probability of  $1/n$ , guess what part is the last one and there is no need to send the last acknowledgement in order to obtain the whole message. These probabilistic protocols are not viable due to the extremely variable (and usually unpredictable) size of connectivity windows in DTNs.

Unfortunately, to our knowledge, there are not non-repudiation protocols that could be used in Delay Tolerant Networks.

## 3. Receipt exchange

The core of this proposal is divided into two different and complementary parts, a receipt exchange protocol presented in this section, and an incentive scheme presented and discussed in Sections 4 and 5.

In this section, we explain the two fundamental inputs we have considered during the design of the receipt exchange protocol. Then, we present the notation used during this section, and we provide an extensive description of all the algorithms and steps involved. After this, we explain the evidences created during the execution of the protocol, and we discuss some security aspects of the usage of IBC.

### 3.1. Receipt exchange protocol's design

The receipt exchange system we propose is based on combining the Fair Exchange Signature Scheme (FESS) [31] with IBC, a cryptographic scheme where the identity of nodes is used to build their public keys. On one hand, we chose this signature scheme because it needs to exchange a low number of messages; it does not require the involvement of a third party during the transaction; and because when the algorithm finishes, the two signatures arouse and become effective simultaneously. On the other hand, we chose IBC because this cryptographic scheme avoids key management issues in DTN scenarios [32].

However, we have not only combined FESS with IBC. Firstly, and most important, we have transformed a protocol where two nodes sign a document they know

beforehand into a protocol where two nodes forward a message and generate evidences about the transaction done. We have achieved this by changing the goal of the protocol and using the last step of the protocol to send the message, instead of a random *keystone*. Besides, we have introduced the concept of a *voucher* as a description of a transaction; and we have modified the structure of the FESS receipts, adding the needed fields to make it store unequivocal information about the transaction they are related. We have made sure that nodes cannot reuse past receipts or parts of them on future transactions of the same message. Note that this is something not considered in FESS, where reusing parts of a past receipt to sign the same document again is not a problem. Finally, we have benefited from hash functions properties to optimise the protocol and reduce the amount of space needed by nodes to store the receipts.

### 3.2. Definitions

Firstly, we present the notation of the elements and the definitions of the concepts used in the receipt exchange protocol. Table 1 contains the notation used to refer to each element and a brief description of its meaning.

A voucher  $v = \langle \text{sender}, \text{receiver}, \text{whosigns}, \text{type} \rangle$  of a transaction contains four fields: *sender* is the identity of the sender; *receiver* is the identity of the receiver; *whosigns* indicates who is the issuer of the voucher; and *type* is a flag used to indicate the type of the transaction (*origin*, *relay* or *delivery*). From now on, we use *transaction* to refer indistinctly to the next three cases: a message  $m$  sent by its origin to any non-final destination node (*type*: *origin*); a message  $m$  sent from a node that is not its origin to a node that is not the destination (*type*: *relay*); and a message  $m$  delivered to its final destination from any node (*type*: *delivery*).

It is important to differentiate between a voucher and a receipt. A voucher is the description of a transaction between two nodes while a receipt contains a voucher and a signature that binds it to the issuer and to the message.

**Table 1**  
Elements used in our receipt exchange protocol.

Notation	Description
$sk_i$	Private key of user $i$
$pk_i$	Public key of user $i$
$SK_i$	Private IBC key of user $i$
$PK_i$	Public IBC key of user $i$
$m$	Message
$v$	Voucher of a transaction
$\sigma$	Receipt of a transaction
$H(m)$	Hash function applied on message $m$
$ID_m$	Unique identifier of message $m$
$E_k(m)$	Cypher of $m$ using key $k$
$D_k(m)$	Decrypt of $m$ using key $k$
$S_k(m)$	Signature of $m$ using key $k$
$V_k(m, s)$	Verification of signature $s$ associated to message $m$ with key $k$

### 3.3. Algorithms

The receipt exchange protocol that we present in this paper uses of the following algorithms: Algorithm 1, that generates the public key of each participant; Algorithm 2, that generates the exchanged receipts; Algorithm 3, that validates the exchanged receipts; and Algorithm 4, that validates a receipt when executed *a posteriori* by a third node.

#### Algorithm 1. SystemSetup

---

**Input:**  $\emptyset$   
**Output:**  $\emptyset$

- 1: Choose  $p$  and  $q$ , big prime numbers so that  $q|p-1$ .
- 2: Choose  $g$  with order  $q$  so that  $g \in \mathbb{Z}_p^*$ .
- 3: **for**  $i$  in  $\langle \text{All participants} \rangle$  **do**
- 4:   Generate the pair of keys  $(sk_i, pk_i)$  so that  $pk_i = g^{-sk_i} \bmod p$ , where  $sk_i$  is the private key and  $pk_i$  is the public key.
- 5: **end for**

---

#### Algorithm 2. FSign

---

**Input:**  $v$ : Voucher of the transaction.  
 $pk_A$ : Issuer's public key.  
 $sk_A$ : Issuer's private key.  
 $SK_A$ : Issuer's private IBC key.  
 $PK_B$ : Receiver's public IBC key.  
 $k$ :  $H(H(m||ID_m))$ .

**Output:**  $\sigma = \langle a, v, k, s \rangle$ : Receipt of the transaction.

- 1: Choose  $w$  so that  $w \in \mathbb{Z}_p^*$ .
- 2: Calculate  $a = \langle E_{PK_B}(pk_A), S_{SK_A}(H(pk_A)) \rangle$ .
- 3: Calculate  $r = g^w \bmod p$ .
- 4: Calculate  $e = H(a, v, k, r)$  where  $H$  is a one way hash function.
- 5: Calculate  $c = w + sk_A e \bmod q$ .
- 6: **return**  $\sigma = \langle a, v, k, s \rangle$  where  $s = \langle r, e, c \rangle$ .

---

#### Algorithm 3. SVerify

---

**Input:**  $\sigma = \langle a, v, k, s \rangle$ : Received receipt, where  $s = \langle r, e, c \rangle$  and  $a = \langle E_{PK_B}(pk_A), S_{SK_A}(H(pk_A)) \rangle$ .  
 $SK_B$ : Receiver's private IBC key.

**Output:** **true** or **false**

- 1: Decrypt  $pk_A = D_{SK_B}(E_{PK_B}(pk_A))$ .
- 2: Calculate  $r_s = g^c pk_A^e \bmod p$ .
- 3: **if**  $e == H(a, v, k, r_s)$  **AND**  $V_{PK_A}(H(pk_A), S_{SK_A}(H(pk_A)))$  **then**
- 4:   **return true**
- 5: **else**
- 6:   **return false**
- 7: **end if**

---

**Algorithm 4.** KVerify

---

**Input:**  $\sigma = \langle a, v, k, s \rangle$ : Received receipt.  
 $\langle m || ID_m \rangle$ : the message and its identifier.

**Output:** true or false

- 1: **if**  $SVerify(\sigma) == \text{true AND } k == H(H(m || ID_m))$  **then**
- 2:   **return true**
- 3: **else**
- 4:   **return false**
- 5: **end if**

---

**3.4. Steps of the exchange**

Let  $A$  be a node that wants to send a message  $m$  to node  $B$  and wants to generate and exchange the receipts related to this transaction. Fig. 1 shows the schema of the protocol, which we explain in detail in the next paragraphs:

**Step 1.** At the deployment phase, before the firsts messages are sent, values  $p, q$  and  $g$  must be generated by the Public Key Generator (PKG) of the network and delivered to all system nodes. Every node needs, also, its pair of keys  $\langle sk_i, pk_i \rangle$ , as can be seen in Algorithm 1.

**Step 2.** Every time node  $A$  starts a transmission of message  $m$  to node  $B$ , the sender begins creating a voucher  $v_A = \langle PK_A, PK_B, A, type \rangle$ . Note that IBC public keys  $PK_i$  are used to identify nodes. Then, the sender uses Algorithm 2 to generate the receipt  $\sigma_A = \langle a_A, v_A, k, s_A \rangle$  and sends it to  $B$ .

**Step 3.** Node  $B$  receives  $\sigma_A$  from  $A$ .  $B$  checks that the voucher  $v_A$  is correct and verifies  $\sigma_A$  using Algorithm 3. If the voucher is valid and the algorithm returns true,  $B$  can proceed to Step 4; otherwise, the transmission is aborted.

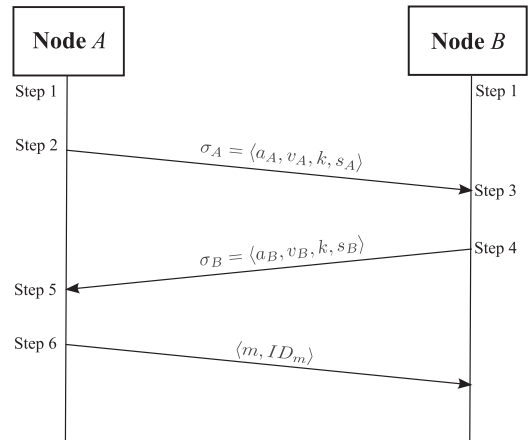
**Step 4.** Node  $B$  accepts the receipt  $\sigma_A$  issued by  $A$ . If  $B$  accepts the transaction, then it creates a voucher  $v_B = \langle PK_A, PK_B, B, type \rangle$ , uses Algorithm 2 to generate the receipt  $\sigma_B = \langle a_B, v_B, k, s_B \rangle$  and sends it to  $A$ .

**Step 5.** When  $A$  receives  $\sigma_B$ , it verifies it using Algorithm 3.  $A$  also checks the voucher  $v_B$ . If the algorithm returns true and the voucher is correct, it can proceed to Step 6; otherwise, the transmission is aborted.

**Step 6.** When both nodes have exchanged and verified the receipts,  $A$  sends  $\langle m, ID_m \rangle$  to  $B$ . This last transmission allows  $B$  to obtain the message and its identifier  $\langle m || ID_m \rangle$ . Node  $B$  verifies that  $H(H(m || ID_m)) = k$  to accept and end the transaction. The keystone links together each participant with the message itself and the receipt issued on the previous steps. The keystone also provides both nodes unequivocal evidence that they have been in contact because of the transfer of  $m$ .

**3.5. Created evidences**

From the moment the exchange has been completed,  $B$  has the receipt  $\sigma_A$  and  $H(m || ID_m)$ , that acts as the keystone.  $\sigma_A$  and  $H(m || ID_m)$  together form a piece of origin non-repudiation evidence that compromises  $A$  as the sender of the message  $m$ , so  $B$  can prove it has received  $m$  from  $A$  revealing  $\sigma_A$  and  $H(m || ID_m)$  to a third party that executes



**Fig. 1.** Schema of the receipts exchange protocol. Node  $A$ , that initiates the transaction sends the receipt  $\sigma_A$  to  $B$ .  $B$  receives it, checks if it is correct and sends the receipt  $\sigma_B$  to  $A$ . When  $A$  receives it and checks its correctness, sends the message, that provides validity to the two receipts.

Algorithm 4 and returns true. This way the protocol provides non-repudiation of origin.

Moreover,  $A$  has the receipt  $\sigma_B$  and the keystone  $H(m || ID_m)$ .  $\sigma_B$  and  $H(m || ID_m)$  together form a piece of evidence that  $A$  and  $B$  have been in contact due to the transfer of  $m$ . Node  $A$  can prove it revealing  $\sigma_B$  and  $H(m || ID_m)$  to a third party that executes Algorithm 4 and returns true, but cannot prove by itself that the transaction of  $m$  has ended correctly.

Notice that, when one of the nodes involved in a two-party transaction acts dishonestly by not forwarding the message, neither of the two evidences are sufficient to prove, unequivocally, which node is guilty and which node is innocent. However, when we reconstruct the chain of custody of this message using receipts from some other two-party transactions, we can identify the two nodes that are suspicious of having lost the message. Then, our incentive scheme punishes both nodes, even when one of them is probably innocent. Later, when we look at the big picture, by reconstructing the chains of custody of lots of messages, we can tell apart the innocent nodes from the guilty ones. The incentive scheme is designed to, in the long run, identify and punish guilty nodes that do not forward messages. We provide discussion about the asymmetric nature of this receipt exchange in Section 4.

**3.6. IBC keys obtention: security aspects**

The presented proposal has to face all IBC inherent issues related with the obtention of new keys, due to the usage of Identity Based Cryptography. They are out of the scope of this article, so, for the sake of simplicity, we will briefly describe the ones that are related with our incentive scheme and point the suggested method to fight them.

Nodes not always may ask the PKG for new keys before their keys have expired, if this has happened, they do not have a valid secret key that could be used to demonstrate that they are whom they claim to be. To avoid an

**Table 2**

Notation of deliverable proofs. Proofs are formed by the pieces of evidence created during the receipt exchange protocol when sending, forwarding, or delivering a message. We use  $type(v_A)$  to denote the value of the field  $type$  of the voucher  $v_A$  inside the receipt  $\sigma_A$ .

Notation	Description	Definition
$A \xrightarrow{B} C$	Proof of forwarding. Proves that $B$ has forwarded message $m$ between $A$ and $C$	$\sigma_A + \sigma_C + H(m  ID_m)$ ( $type(v_A) = relay$ , $type(v_C) = relay$ )
$\{A \xrightarrow{B} C$	Proof of forwarding from the origin. Proves that $B$ has forwarded message $m$ between its origin $A$ and $C$	$\sigma_A + \sigma_C + H(m  ID_m)$ ( $type(v_A) = origin$ , $type(v_C) = relay$ )
$A \xrightarrow{B} \{C$	Proof of forwarding to the destination. Proves that $B$ has forwarded message $m$ between $A$ and its final destination $C$	$\sigma_A + \sigma_C + H(m  ID_m)$ ( $type(v_A) = relay$ , $type(v_C) = delivery$ )
$A \rightarrow B\}$	Proof of delivery. Proves that $B$ , the destination of a message $m$ , has received it from $A$	$\sigma_A + H(m  ID_m)$ ( $type(v_A) = delivery$ )

impersonation attack where a node claims himself to be another, usually with a higher score, when asking for keys at the PKG, the PKG and each node may share a secret and update it, using the Diffie–Hellmann protocol, each time they obtain new keys.

When a new node arrives to the network and wants to become part of it, the node can ask the PKG for a new identity in order to become part of the network. To avoid that a node with a low score could benefit from this and reset it to a higher amount by changing their identity, new identities should be created setting their score equal to the lowest of the network.

If a node loses the secret that proves their identity and their keys expire, it will maintain their identity and do not lose their score only if the network can provide an alternative mechanism that demonstrate the identity of the node; otherwise, it will be treated as a new node.

#### 4. Incentive scheme

In this section, we explain how to convert the pieces of evidence generated during the receipt exchange phase into proofs of behaviour that could be used by the Incentive Manager. We also discuss the role played by the Incentive Manager, how it rewards and punishes nodes due to their behaviour, delivering a higher or lesser amount of keys, and the policies of the asynchronous incentive scheme.

##### 4.1. Definitions

In order to make the incentive scheme easier to understand, we define a proof of behaviour as a set of receipts and keystones used to prove the behaviour of a node towards one concrete message. We also define *deliverable* proof as any proof that can be delivered to the manager of the incentive scheme.

Table 2 contains the notation used to refer to each deliverable proof, a brief description of its meaning and the items that compose it, using the notation explained in the previous section.

##### 4.2. Guilty until proven innocent

The receipt exchange protocol we propose is specifically designed to operate in DTNs, it does not need any third

party during the transaction, and the number of messages exchanged is minimal. This protocol allows us to obtain non-repudiation proof of origin and reception when all steps are completed. However, when the last message of the protocol is not sent by the sender, discarded by the receiver, or lost during transmission, there is no way of knowing exactly what happened with that transaction. Here, we face the classic Two Generals' Problem<sup>1</sup>: to be sure that the last acknowledgement has been sent and received, another acknowledgement needs to be sent, but that new last acknowledgement has the same issue, and so on.

To deal with this, our incentive scheme uses the policy “guilty until proven innocent”, meaning that the two nodes involved with the loss of a message will be marked as suspicious nodes and punished until it is demonstrated that they behaved honestly. From the moment their innocence is proven, punishment is removed, and they are rewarded for their behaviour.

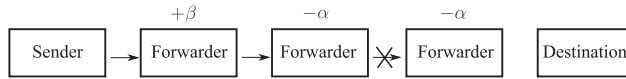
##### 4.3. System parameters

Conceptually, the incentive scheme is very simple. Nodes are rewarded or punished with Cooperation Points (CP) depending on their actions. We define Level of Cooperation (LC) as the amount of CP that a node has obtained with its behaviour and from now on we will use this terminus or its acronym indistinctly. The amount of CP that it is added to or subtracted from the LC of the nodes is defined by the next three parameters:  $\alpha$ ,  $\beta$  and  $\epsilon$ , which have the following meaning:

- $\alpha$ : The punishment applied to nodes that are suspicious of not forwarding a message,  $\alpha$  must satisfy<sup>2</sup>  $\alpha > 0$ .

<sup>1</sup> Suppose there is a valley surrounded by two hills. General  $A_1$  is on one. General  $A_2$  is on the second hill. The enemy,  $B$ , is in the valley. If either  $A_1$ 's or  $A_2$ 's army attacks  $B$  independently they would lose, but together they would win. The problem for  $A_1$  is to communicate a coordinated attack time to  $A_2$  and be sure that  $A_2$  received the message.  $A_2$  also needs to know the acknowledgement got through to avoid attacking alone and lose the battle.

<sup>2</sup> Restrictions for the values of  $\alpha$ ,  $\beta$  and  $\epsilon$  have been chosen in order to satisfy the Nash equilibrium. See Section 5 for more details.



**Fig. 2.** Illustration of the reward and punishment scheme. The arrows depict the relays of the message. The first node, the sender of the message, is not rewarded nor punished. The second node is rewarded with  $+\beta$  CP because it is a confirmed relay. The third and the fourth nodes are the two last nodes of the chain of custody, so they are suspected of having lost the message; therefore, they are punished with  $-\alpha$  CP.

- $\beta$ : The reward given to nodes when it is proven that they have forwarded a message. The value of  $\beta$  must satisfy  $\beta > 0$ .
- $\epsilon$ : The reward given to a node for each proof delivered to the manager. The value of  $\epsilon$  must satisfy  $\epsilon > \frac{13}{20}\alpha + \frac{4}{5}\beta$ .

It is important to note that, when it is proved that a node had forwarded a message, it is not considered suspicious anymore, so it is rewarded with  $\alpha + \beta$  CP (remove the punishment and add the reward).

Fig. 2 illustrates the basic idea of the incentive scheme, and how the suspicious nodes are punished while other nodes are rewarded if it is proven that they have forwarded a message.

#### 4.4. Reward and punishment

The chain of custody of every message is indexed by its message identifier  $ID_m$ , and it is updated automatically using the proofs of forwarding or the proofs of delivery by applying the following rules:

- Punish with  $-\alpha$  CP those nodes that have become suspected of not complying as a result of the last update. Suspicious nodes are the last nodes and the second-to-last nodes of each chain of custody.
- Reward with  $+\alpha$  CP those nodes that were suspicious before the update but not after it.
- Reward with  $+\beta$  CP those nodes that have become a confirmed relay with the last update. A confirmed relay is a node that is, at least, the third last node of a chain of custody, or any node in the chain of custody of a message that has been delivered. There are two exceptions that do not obtain this reward:
  - The sender, the node that has created the message.
  - Any node that has appeared at least once before in the chain of custody. This way we avoid that nodes can obtain high amounts of CP by colluding with other nodes to forward a message between them an arbitrary amount of times.
- Finally, reward the node that has delivered the proof with  $+\epsilon$  CP if the chain of custody has changed thanks to it.

Note that uncooperative nodes that do not accept messages to forward them, do not obtain CP, and that nodes that drop messages are punished with  $-\beta$  for every lost message. This punishment is proportional to the damage done to the overall performance of the network by each one of these two behaviours. Fig. 3 shows an example of updating a chain of custody and rewarding nodes when a new proof is used. We provide lots of examples in Section 5.

#### 4.5. Incentive manager and the enforcing mechanism

IBC-based DTNs are based on the assumption that nodes will, eventually, connect with the Private Key Generator to obtain a set of private IBC keys. Besides, as seen in [33], some DTN routing protocols base their operation on some infrastructure assistance, either via mobile data mules or through the deployment of stationary nodes.

Our proposal benefits from this assumption and uses an offline third-party called Incentive Manager (IM), which must be located on the same node that acts as the PKG.<sup>3</sup> The IM receives the proofs of all transactions, tracks the chain of custody of every message and rewards or punishes nodes due to their behaviour. This way, nodes use the trip to upload proofs and to obtain new IBC keys.

IBC systems usually use keys with a small duration. Depending on the needs of the nodes, the PKG may deliver them sets of a high number of their next keys instead of delivering them only the next one. This way nodes do not need to contact the PKG during a while and can continue routing their messages without running out of keys. Therefore, nodes prefer to obtain more keys when they contact the PKG because this way they obtain more independence and they can operate for more time without asking the PKG again for more keys.

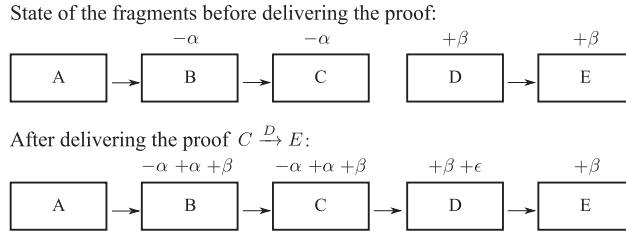
As an enforcing mechanism for our incentive system, we relate the amount of IBC keys given to a node to its Level of Cooperation, as briefly depicted in Fig. 4. We calculate  $K$ , defined as the number of keys of a fixed duration given by the IM to a node in the basis of the LC of the demanding node related to the LC of all other nodes.

We define  $max_{LC}$  as the higher LC of the network and  $min_{LC}$  as the lower LC of the network. Then, we normalise the Level of Cooperation of the demanding node inside the interval  $[min_{LC}, max_{LC}]$  and map that value to its corresponding value inside the interval  $[min_K, max_K]$ , defined by the minimum and maximum possible values of  $K$ . This way nodes will not be excluded of the network because they will obtain, at least,  $min_K$  and the possibility of increase their LC using these keys. This procedure is formalised in Eq. (1).

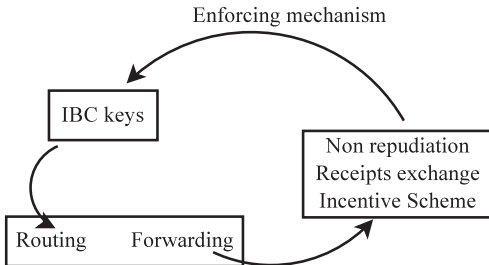
$$K = min_K + \frac{LC - min_{LC}}{max_{LC} - min_{LC}}(max_K - min_K) \quad (1)$$

Note that every node will approximately have the same opportunities to forward messages as their neighbours. A node that does not forward because it has no chance will not obtain any CP, but their neighbours neither, so it will not be punished when keys are delivered based on their

<sup>3</sup> In networks where an alternative communication channel exists, two or more Incentive Managers and/or PKGs can coexist, using this channel to connect between them in order to share the state of the chain of custody of all messages.



**Fig. 3.** Example of updating a chain of custody. On the upper chain of custody, nodes B and C are suspected of having lost the message, so they are punished with  $-\alpha$  CP. Then, node D delivers  $C \xrightarrow{D} E$ , D gains the reward  $+\epsilon$  CP for delivering a valuable proof, and B and C are now confirmed relays (they are previous steps of the confirmed relay D). Therefore, B and C are rewarded with  $+\alpha + \beta$  CP to remove the punishment and reward their behaviour. Note that a proof delivered by D that involves C and E has affected too the Level of Cooperation of node B.



**Fig. 4.** The enforcing mechanism. Nodes need IBC keys to route their messages, and they obtain the keys by forwarding other’s messages.

relative LC. But a node that decides not to forward messages will remain with the same LC, but will obtain a lesser amount of keys on the basis of their relative LC because their neighbours will probably be forwarding messages and obtaining CP.

Besides, we use an exponential decay function to gradually decrease Nodes’ Levels of Cooperation over time. This way we avoid *selfish bursts*, defined as the behaviour of a node accumulating CP and then using it to behave in a very uncooperative way without being punished. Decreasing the LC of all nodes periodically, we allow nodes with a very small (or negative) LC to quickly recover from their past uncooperative behaviour if they start being cooperative, because past actions will weigh less than present actions. With that purpose, we update the LC of every node every  $t$  seconds using Eq. (2), using the previous value  $LC_{-1}$  of their Level of cooperation to calculate the new value  $LC_t$ , where  $T$  is the time constant.

$$LC_t = e^{-t/T} LC_{-1} \tag{2}$$

Finally, our system includes a mechanism designed to ward off nodes that conform with receiving  $K \simeq min_k$ . When the IM calculates  $K$  and it is lesser than a threshold  $thr_k$ , then the IM gives no key to the demanding node and waits an amount of time defined by  $tout_k$ . When the node demands keys again after  $tout_k$  seconds, the IM gives him  $thr_k$  keys.

4.6. Asynchronous incentive scheme

To the best of our knowledge, this is the only proposal where the incentive manager works asynchronously and

does not need to build the whole chain of custody before rewarding nodes. Every time a node delivers a proof to the IM, it updates the state of the chain of custody, and it distributes rewards and punishments as if the new state will be the last.

This signifies that, despite heavy punishment is applied to a node when other nodes deliver proofs - because it has become suspicious for not forwarding lots of messages - it does not matter. This is so because own node’s balance is re-calculated when it contacts the IM and uploads its proofs, and all punishment will be removed if it proves that it has forwarded all the messages.

5. Nash equilibrium

In this section, we start by making some assumptions about the rationality of the network’s participants their knowledge about other’s behaviours. Afterwards, we discuss the different strategies than can be played by nodes. Finally, we demonstrate that all nodes behaving honestly, forwarding messages, and delivering proofs of forwarding and proofs of delivery, form a unique Nash Equilibrium.

5.1. Previous assumptions

We start assuming that nodes cannot make guesses about other’s Levels of Cooperation. Besides, we assume all participants of the network to be rational [34]. As rational nodes never behave in a way that could turn against their interests, they always want to maximise its utility function. During this section, we define the utility function of nodes as their Level of Cooperation.<sup>4</sup>

Moreover, when a node delivers a proof to the IM, it is rewarded or punished depending on the previous state of the chain of custody. The chain of custody tracked by the IM is updated every time a node delivers a proof. Therefore, it depends on the proofs previously delivered by other nodes. This means that a node cannot know *a priori* the state of the chain of custody. During all the current section, we will assume that nodes cannot make any guesses about the state of the chain of custody. As a result, we will consider all possible states as equally likely.

<sup>4</sup> In Section 7 we consider a slightly different utility function where nodes try not to maximise Level of Cooperation, but to obtain at least a set number of keys each time they contact the PKG.

Summarising, nodes are interested in obtaining a higher Level of Cooperation and act according to their interest, and no guesses can be made about the state of the chains of custody.

5.2. Simplified notation

In all tables of this section we used a simplified notation  $X..YZ$  to summarise a chain of custody in which  $X$  is the first known custodian node, and  $Y$  and  $Z$  are the second-to-last and the last nodes. We use ‘...’ to represent a chain of custody where no node matches the nodes involved in the received proof.

5.3. Game theory analysis

In order to analyse the game generated by the presented incentive scheme, we have split the whole game into a set of subgames. Every subgame takes into consideration one of the decisions that a participant of the network has to take: to participate on the network; to accept other’s messages to forward them; to cheat or be honest when exchanging receipts; to deliver proofs of forwarding to the IM; and to deliver proofs of delivery to the IM. Therefore, a strategy profile of one player for the whole game contains his strategy profile for every subgame.

**Definition 1.** The game generated by the presented incentive scheme is

$$G = \langle N, \{S_i\}, \{\pi_i\}, \{p_i\} \rangle.$$

- $N = \{N_0, N_1, \dots, N_n\}$  is the set of the nodes of the network.
- $S_i = \{S_{i0}, S_{i1}, S_{i2}, S_{i3}, S_{i4}\}$  is the strategy set of the player  $N_i$ , where every  $S_{ij}$  corresponds to the strategy player  $i$  has chosen for subgame  $j$ , every strategy set contains two actions, one cooperative and one selfish, that will be explained in next subsections.
- $\pi_i$  is the payoff of the  $i$ th player  $N_i$ , and it is measured in CP.
- $p_i = \{p_{i0}, p_{i1}, \dots, p_{i4}\}$  is a mixed strategy for player  $i$ , where  $p_{ij} = \{p_{ij}^i, 1 - p_{ij}^i\}$  is a mixed strategy for player  $i$  for subgame  $j$ , and  $p_{ij}^i$  denotes the probability of player  $i$  of acting cooperatively in subgame  $j$ .

5.3.1. Subgame 0. The dilemma of participating

A node that chooses to not participate (NP) never accepts messages not addressed to him. Therefore, it will never obtain CP for forwarding or for delivering proofs of forwarding because it does not generate any. A node that

**Table 3**  
The payoff matrix of subgame 0.

	Any other node $j$		
	P( $q_{c0}^j$ )	NP( $1 - q_{c0}^j$ )	
Node $i$	P( $p_{c0}^i$ ) NP( $1 - p_{c0}^i$ )	$\pi_i(P), \pi_j(P)$ 0, $\pi_j(P)$	$\pi_i(P), 0$ 0, 0

behaves this way will not increase its Level of Cooperation, so, its benefit will be 0 CP. On the other hand, a node that decides to participate (P) will obtain a payoff  $\pi_i(P)$  that depends on the outcomes of subgames 1–4. Table 3 provides the payoff matrix for this subgame.

**Theorem 1.** All nodes choosing to participate (P) with probability  $p_{c0}^i = 1$  form a unique Nash equilibrium for subgame 0 if and only if they behave in a way that allow them to obtain more than 0 CP.

**Proof.** Node  $i$  obtains  $\pi_i(P) \cdot q_{c0}^j + \pi_i(P) \cdot (1 - q_{c0}^j) = \pi_i(P)$  when chooses to participate (P), and  $0 \cdot q_{c0}^j + 0 \cdot (1 - q_{c0}^j) = 0$  when chooses not to participate (NP).

$$\pi_i(P) > \pi_i(NP);$$

$$\pi_i(P) > 0$$

Therefore, if  $\pi_i(P) > 0$ , then  $i$  will always obtain a higher payoff by playing P. As subgame 0 is symmetric, the same applies to any other node. □

Throughout the current section, we will demonstrate that there are profiles that grant nodes  $\pi_i > 0$  CP no matter what other nodes do.

5.3.2. Subgame 1. The dilemma of forwarding

A node that accepts a message to forward but does not forward (NF) it will be punished with  $-\alpha$  CP when any proof of forwarding that marks  $i$  as a suspicious node is delivered to the IM. On the other hand, a node that decides to forward (F) a message will be eventually rewarded with  $+\beta$  if the next node forwards the message too. But it will be punished with  $-\alpha$  if the next node decides to not forward the message because it will be considered by the IM as a suspicious node.

This subgame is sequential and asymmetric because the decision of the second node does not matter unless the first one chooses to forward (F) the message. Then, if the second node plays NF, both nodes are punished with  $-\alpha$ . But if it plays F, his payoff is determined by playing the same subgame once again, with the second node playing first and a third node (the next hop) playing second. Table 4 provides the payoff matrix for this subgame.

**Theorem 2.** All nodes choosing to forward messages (F) with  $p_{c1}^i = 1$  form a unique Nash equilibrium for subgame 1.

**Proof.** Node  $i$  obtains  $\beta \cdot q_{c1}^j - \alpha \cdot (1 - q_{c1}^j)$  when chooses to forward (F), and  $-\alpha \cdot q_{c1}^j - \alpha \cdot (1 - q_{c1}^j) = -\alpha$  when chooses NF. Therefore, his payoff is higher when it plays F than

**Table 4**  
The payoff matrix of subgame 1.

	Other node $j$ , playing second		
	F( $q_{c1}^j$ )	NF( $1 - q_{c1}^j$ )	
Node $i$	F( $p_{c1}^i$ ) Plays first NF( $1 - p_{c1}^i$ )	$\beta$ , play again as first player $-\alpha, 0$	$-\alpha, -\alpha$ $-\alpha, 0$

when it plays NF because both  $\alpha$ ,  $\beta$  and  $q_{c1}^i$  are defined to be positive.

$$\begin{aligned} \pi_i(F) &> \pi_i(NF); \\ \beta \cdot q_{c1}^i - \alpha \cdot (1 - q_{c1}^i) &> -\alpha; \\ q_{c1}^i \cdot (\beta + \alpha) &> 0 \end{aligned}$$

Therefore, for the first player, playing F is a dominant strategy.

By backward induction, the second player will play F, because it is the only way to obtain a positive payoff, because playing NF will grant him  $p_{c1}^i \cdot -\alpha + 0 \cdot (1 - p_{c1}^i) = -\alpha \cdot p_{c1}^i$ , which is negative in any case.  $\square$

5.3.3. Subgame 2. The dilemma of cheating

When forwarding a message, an honest node (H) finishes the receipt exchange protocol by sending the message. On the other hand, a node can try to cheat (NH) by starting the exchange receipt protocol to obtain a proof of forwarding but does not sending the last message. This way the transaction remains unfinished because the receiver has not received the message. A node that behaves this way could deliver the gathered proofs to obtain  $\epsilon$  CP for each one. An honest (H) node could deliver the gathered proofs to obtain not only  $\epsilon$  CP, but also  $\alpha$  CP for no longer being suspicious and  $\beta$  CP for becoming a confirmed relay. This subgame is a one-player game because only the part that starts the transmission can cheat the other part.

Table 5 shows all possible states of the chain of custody, how it is updated, and the reward or punishment given by the IM to each participant when node B delivers a proof of forwarding  $A \xrightarrow{B} C$ . Last two columns correspond to  $\rho$ , the probability of the situation if B is a node that plays NH, and  $\phi$ , the probability of the situation if B is a node that plays H. Notice that there are situations with  $\rho = 0$ , these are situations that can only exist if B forwards messages in an honest way (H). Taking this into account, Table 6 provides the payoff matrix for this subgame.

**Theorem 3.** Any node that is honest (H) with  $p_{c2}^i = 1$  form a unique Nash equilibrium for subgame 2.

**Proof.** Node  $i$  obtains  $\epsilon + \frac{1}{5}\beta - \frac{2}{5}\alpha$  when chooses to be honest (H), and  $\epsilon - \frac{2}{5}\alpha$  when chooses NH. Therefore, his payoff is higher when it plays H than when it plays NH because both  $\alpha$  and  $\beta$  are defined to be positive.

Table 5

Possible situations when delivering  $A \xrightarrow{B} C$ . A cheater node B would not obtain the reward  $+\beta$  CP for forwarding the message, and will always be punished with  $-\alpha$  CP for being suspected of having lost the message.

Last state	New state	A's reward	B's reward	C's reward	Y's reward	$\rho$	$\phi$
...	A...BC	$+\beta$	$+\epsilon - \alpha$	$-\alpha$		1/3	1/5
C...XY	A...XY	$+\beta$	$+\epsilon + \beta$			0	1/5
X...YA	X...BC	$+\alpha + \beta$	$+\epsilon - \alpha$	$-\alpha$	$+\alpha$	1/3	1/5
B...XY	A...XY	$+\beta$	$+\epsilon$			0	1/5
X...AB	X...BC	$+\beta + \alpha$	$+\epsilon$	$-\alpha$		1/3	1/5

Table 6

The payoff matrix of subgame 2.

Node $i$	Other node receiving the message	
	H( $p_{c2}^i$ )	NH( $1 - p_{c2}^i$ )
	$(\epsilon + \frac{1}{5}\beta - \frac{2}{5}\alpha), -\alpha$	$(\epsilon - \frac{2}{5}\alpha), \text{play as first player}$

$$\begin{aligned} \pi_i(H) &> \pi_i(NH); \\ \epsilon + \frac{1}{5}\beta - \frac{2}{5}\alpha &> \epsilon - \frac{2}{5}\alpha; \\ \beta &> -\frac{4}{3}\alpha \end{aligned}$$

Therefore, playing H is a dominant strategy.  $\square$

5.3.4. Subgame 3. The dilemma of delivering proofs of forwarding

A node can choose to deliver proofs of forwarding (PF) in order to be rewarded by the IM. Besides, a node can choose to not deliver them (NPF) and wait until the next node delivers a proof of forwarding that provides him a reward.

This way, a node that plays NPF avoids punishment  $-\alpha$  CP and will eventually be rewarded with  $\beta$  CP for forwarding the message, but it never obtains  $+\epsilon$  CP for delivering proofs. On the other hand, a node that plays PF will be considered suspicious and be punished with  $-\alpha$  CP sometimes, but it will obtain  $+\epsilon$  CP every time that delivers a proof.

Table 7 has the same structure as Table 5. The last columns correspond to  $\mu$ , the probability of the situation if A is a node that never delivers proofs of forwarding and  $\lambda$ , the probability of the situation if B is an honest node that always delivers the proofs. Notice that there is only one situation with  $\mu = 0$ , this is a situation that can only exist if A has delivered a proof previously. Taking this into account, Table 8 provides the payoff matrix for this subgame.

Table 7

Possible situations when node B delivers a proof of forwarding  $A \xrightarrow{B} C$  to the IM. Node B will obtain a higher gain than a node A that does not deliver any proofs and trusts that others will deliver proofs that increase A's Level of Cooperation.

Last state	New state	A's gain	B's gain	C's gain	Y's gain	$\mu$	$\lambda$
...	A...BC	$+\beta$	$+\epsilon - \alpha$	$-\alpha$		1/4	1/5
C...XY	A...XY	$+\beta$	$+\epsilon + \beta$			1/4	1/5
X...YA	X...BC	$+\alpha + \beta$	$+\epsilon - \alpha$	$-\alpha$	$+\alpha$	1/4	1/5
B...XY	A...XY	$+\beta$	$+\epsilon$			1/4	1/5
X...AB	X...BC	$+\beta + \alpha$	$+\epsilon$	$-\alpha$		0	1/5

Table 8

The payoff matrix of subgame 3.

Node $i$	Next node $j$ , playing second	
	PF( $q_{c3}^j$ )	NPF( $1 - q_{c3}^j$ )
Plays first	$(\epsilon + \frac{\beta}{5} - \frac{2}{5}\alpha), (\epsilon + \frac{\beta}{5} - \frac{2}{5}\alpha)$	$(\epsilon + \frac{\beta}{5} - \frac{2}{5}\alpha), (\beta + \frac{\alpha}{4})$
	$(\beta + \frac{\alpha}{4}), (\epsilon + \frac{\beta}{5} - \frac{2}{5}\alpha)$	$0, (\beta + \frac{\alpha}{4})$



**Theorem 4.** All nodes choosing to deliver proofs of forwarding (PF) to the IM with  $p_{c3}^i = 1$  form a unique Nash equilibrium for subgame 3 if  $\epsilon > \frac{4}{5}\beta + \frac{13}{20}\alpha$ .

**Proof.** The first player obtains  $(\epsilon + \frac{\beta}{5} - \frac{2}{5}\alpha) \cdot q_{c3}^j + (\epsilon + \frac{\beta}{5} - \frac{2}{5}\alpha) \cdot (1 - q_{c3}^j) = (\epsilon + \frac{\beta}{5} - \frac{2}{5}\alpha)$  when plays PF, and  $(\beta + \frac{\alpha}{4}) \cdot q_{c3}^j$  when chooses NPF. Therefore, his payoff is higher when it plays PF than when it plays NPF if and only if  $\pi_i(\text{PF}) > \pi_i(\text{NPF})$ ;

$$\epsilon + \frac{\beta}{5} - \frac{2}{5}\alpha > (\beta + \frac{\alpha}{4}) \cdot q_{c3}^j;$$

$$\frac{\epsilon + \frac{\beta}{5} - \frac{2}{5}\alpha}{\beta + \frac{\alpha}{4}} > q_{c3}^j$$

Being  $0 \leq q_{c3}^j \leq 1$ , if the numerator  $(\epsilon + \frac{\beta}{5} - \frac{2}{5}\alpha)$  is higher than the denominator  $(\beta + \frac{\alpha}{4})$ , then this equation holds.

$$\epsilon + \frac{\beta}{5} - \frac{2}{5}\alpha > \beta + \frac{\alpha}{4};$$

$$\epsilon > \frac{4}{5}\beta + \frac{13}{20}\alpha$$

This subgame is asymmetric, so, the second player obtains  $(\epsilon + \frac{\beta}{5} - \frac{2}{5}\alpha) \cdot p_{c3}^j + (\epsilon + \frac{\beta}{5} - \frac{2}{5}\alpha) \cdot (1 - p_{c3}^j) = (\epsilon + \frac{\beta}{5} - \frac{2}{5}\alpha)$  when plays PF, and  $(\beta + \frac{\alpha}{4}) \cdot p_{c3}^j + (\beta + \frac{\alpha}{4}) \cdot (1 - p_{c3}^j) = (\beta + \frac{\alpha}{4})$  when chooses NPF.

As we have demonstrated  $\epsilon + \frac{\beta}{5} - \frac{2}{5}\alpha > \beta + \frac{\alpha}{4}$  above, then we can state than playing PF is a dominant strategy for both players.  $\square$

**Corollary 4.1.** The incentive scheme also grants that an honest (H) node will obtain a positive payoff while delivering a proof of forwarding (PF).

**Proof.** The payoff obtained by a node that plays H and PF is positive if the next equation holds.

$$\pi_i(\text{H}, \text{PF}) > 0;$$

$$\epsilon + \frac{1}{5}\beta - \frac{2}{5}\alpha > 0;$$

$$\epsilon + \frac{1}{5}\beta > \frac{2}{5}\alpha$$

Given that  $\epsilon$  is defined to be higher than  $\frac{13}{20}\alpha$ , then  $\epsilon > \frac{2}{5}\alpha$  also holds because  $\frac{13}{20}\alpha > \frac{2}{5}\alpha$ , so, this equation holds for any value of  $\alpha$  and  $\beta$  even when  $\beta \approx 0$ .  $\square$

### 5.3.5. Subgame 4. The dilemma of delivering proofs of delivery

To obtain proofs of forwarding from the origin or proofs of delivery is important for the IM. When a chain of custody is completed, the IM can delete it and free resources. To a node, there are no differences between forwarding a message from its origin than forwarding it from any other node. The same way, there are no differences between the reward obtained by delivering any proof of forwarding. Therefore, decisions involving proofs of forwarding from the origin are covered by subgames 1 and 3.

A node that has received a message can deliver (PD) the proof of delivery in order to be rewarded by the IM. On the

**Table 9**

Possible situations when delivering  $A \rightarrow B$ . The message has arrived at its destination and all implied nodes are rewarded.

Last state	New state	A's gain	Previous B's reward	B's gain	Y's gain	$\varphi$
...	A..AB}	$+\beta$		$+\epsilon$		1/3
X..YA	X..AB}	$+\alpha + \beta$		$+\epsilon$	$+\alpha + \beta$	1/3
X..AB	X..AB}	$+\alpha + \beta$	$-\alpha$	$+\epsilon + \alpha$		1/3

**Table 10**

The payoff matrix of subgame 4.

	Previous node ( $i-1$ ) on the chain of custody	
Node $i$	PD( $p_{c4}^i$ )	$(\epsilon + \frac{1}{3}\alpha), (\frac{2}{3}\alpha + \beta)$
	NPD( $1 - p_{c4}^i$ )	$(-\frac{1}{3}\alpha), -\alpha$

other hand, a node that has received a message can chose not to deliver (NPD) this proof. This way it will never recover from the punishment  $-\alpha$  for being considered suspicious, and the same happens to its previous node on the chain of custody. This subgame is a one-player game because only the destination node has the proofs of delivery and can choose to deliver them or not.

Table 9 shows all possible situations that can occur when node B delivers a proof of delivery  $A \rightarrow B$  unknown2015. Let  $\varphi$  be the probability of every possible state of the chain of custody at the moment of delivery. Taking this into account, Table 10 provides the payoff matrix for this subgame.

**Theorem 5.** Any node that delivers proofs of delivery (PD) with  $p_{c4}^i = 1$  form a unique Nash equilibrium for subgame 4.

**Proof.** Node  $i$  obtains  $\epsilon + \frac{1}{3}\alpha$  when plays PD, and  $-\frac{1}{3}\alpha$  when chooses NPD. Therefore, his payoff is higher when it plays PD than when it plays NPD because both  $\alpha$  and  $\epsilon$  are defined to be positive.

$$\pi_i(\text{PD}) > \pi_i(\text{NPD});$$

$$\epsilon + \frac{1}{3}\alpha > -\frac{1}{3}\alpha$$

Therefore, playing PD is a dominant strategy.  $\square$

**Corollary 5.1.** The incentive scheme also grants that a node that delivers a proof of delivery (PD) will obtain a positive payoff, and that this will cause another node which has forwarded (F) the message to obtain a positive payoff.

**Proof.** The payoff obtained by a node that plays PD is positive because both  $\alpha$  and  $\epsilon$  are positive.

$$\pi_i(\text{PD}) > 0;$$

$$\epsilon + \frac{1}{3}\alpha > 0$$

The payoff obtained by the previous node, which has played F, when node  $i$  plays PD, is positive because both  $\alpha$  and  $\beta$  are also positive.

$$\pi_{i-1}(F) > 0;$$

$$\frac{2}{3}\alpha + \beta > 0. \square$$

**Table 11**

Average values and standard deviation ( $\sigma$ ) of the time needed to complete a transaction with and without the receipt exchange protocol. The absolute overhead is almost constant.

Message size (MB)	Transmission time without the receipt exchange protocol (s)		Transmission time with the receipt exchange protocol (s)		Absolute overhead (s)		Relative overhead (%)
	Average	$\sigma$	Average	$\sigma$	Average	$\sigma$	
	1	1.07	0.14	4.23	0.26	3.16	
2	2.11	0.21	5.40	0.21	3.29	0.16	155
3	4.46	0.26	7.78	0.38	3.31	0.36	74
4	7.24	0.22	10.62	0.25	3.37	0.17	46
5	10.95	0.57	14.31	0.61	3.35	0.26	30

**Table 12**

Detail of the different parts of the receipt exchange protocol and the time consumed by each one. To sign the receipts is the most costly operation.

Operation	Executions per transaction	Time consumed per execution		Time consumed per transaction	
		Average	$\sigma$ (s)	Average	$\sigma$ (s)
		FSign	2	0.90	0.10
SVerify	2	0.71	0.07	1.42	0.14
Send receipt	2	0.03	0.002	0.06	0.004
Total				3.30	0.25

#### 5.4. Nash equilibrium

From a game theory perspective, every node of the network has to choose a global strategy, which consists of picking a strategy for each of the subgames.

As we have proven in this section, for any node, choosing a strategy that consists of accepting messages addressed to others (P), forwarding these messages (F), being honest during the receipt exchange protocol (H) and delivering to the IM all kind of proofs (PF and PD) is a strictly dominant strategy. This behaviour grants the node higher benefits than any other possible strategy, and it does not matter what the other nodes of the network do.

Consequently, a profile of strategies where all nodes choose to behave this way form a unique Nash equilibrium because it is impossible for any node to increase its profits by deviating from this strategy.

## 6. Performance evaluation

In this section we present some details about the proof-of-concept we have implemented. Then, we provide measurements of the computational overhead introduced by the receipt exchange protocol. Finally, we study if the computational overhead introduced by this protocol is affordable for the network.

### 6.1. Overhead calculation

As a proof-of-concept and in order to obtain a measure of the overhead that the receipt exchange protocol adds to every transaction, we have deployed an implementation of

the receipt exchange protocol on two Raspberry Pi devices.<sup>5</sup> We have used this implementation to send 250 messages of sizes between 1 MB and 5 MB and measured the performance of the protocol and the time needed to compute and exchange the receipts.

The obtained results are shown in Table 11, where the introduced overhead is shown, and Table 12, where the detail of the overhead introduced by each operation is shown. This results have been incorporated to the simulations via a parameter called *overhead time*.

### 6.2. Impact of the overhead

We have used simulations<sup>6</sup> to measure the impact of the overhead caused by the receipt exchange protocol. For this, we have compared a scenario where the fully cooperative nodes does not need to use the protocol (modelled by an *overhead time* of 0 s) with a scenario where the selfish behaviours of the nodes urges us to use the receipt exchange protocol to enable the incentive system.

The experiments show that, as can be seen in Table 13, without the overhead introduced by our protocol, the ratio of aborted transactions is 2.4%, corresponding to the transactions of messages that cannot be finished before the involved nodes move out of reach one from another. When we take into account the overhead introduced by the receipt exchange protocol, the ratio of aborted transactions is 4.5 times higher, because there are more transactions than cannot be successfully finished before the end of an opportunistic contact.

Despite the abort rate, the overall performance of the network is not injured. The design of the receipt exchange protocol, where the transmission of the message is the last step of the protocol, causes that when an exchange finishes abruptly the transaction is considered as not done, and the message is not removed from the source. Even with this increased rate of aborted messages, the impact on the network in terms of delivery ratio is negligible, and the impact in terms of latency is, as will be seen in Section 7, very positive.

<sup>5</sup> Raspberry Pi Broadcom BCM2835 SoC full HD, 700 MHz Low Power ARM1176JZ-F, 512 MB SDRAM, 4 GB SD with Raspbian, equipped with a Wireless Edimax EW-7811Un (802.11b/g/n up to 150 Mbps), a GPS receiver NL-302U (baud rate: 4,800 bauds) and a dual output 5,000 mA h battery.

<sup>6</sup> In order to avoid redundancy, all details about the methodology, the parameters used, the simulated scenario and the simulator will be found on Section 7.

**Table 13**

Average values and standard deviation ( $\sigma$ ) of the results. Although the aborted ratio grows when using the receipt exchange protocol, delivery ratio remains almost unaffected, and latency depends on other parameters.

Scenario	Overhead (s)	Aborted ratio (%)		Delivery ratio (%)		Latency
		Average	$\sigma$	Average	$\sigma$	
Cooperative	0	2.43	0.65	94.85	0.03	$\approx 21,600$ s
Selfish	3.3	11.02	0.57	93.74	0.05	N/A <sup>a</sup>

<sup>a</sup> The latency of the network for the selfish case is heavily dependent on the behaviour of the nodes and the parameters of the incentive system. For this reason, the results obtained cannot be condensed in a single latency value. See Section 7 for more details.

## 7. Simulations and results

In this section, we define how we expect rational nodes to behave when they obtain keys depending on their Level of Cooperation. Afterwards, we present a scenario where our proposal can be applied, and we explain and justify the simulation parameters we have used. Finally, we show and explain the results obtained through simulations.

### 7.1. Re-defining rational behaviour and the utility function

As we proved in the previous section, for all nodes in the network, to behave in a fully cooperative way is a strictly dominant strategy. This apply if they try to maximise their Level of Cooperation, thus, if the utility function of a node is its LC. However, in practical scenarios, nodes would probably be more interested in the benefit they obtain from their LC than in the LC itself.

In order to model this kind of behaviour, we have defined the concept of selfishness and the selfish utility function, formalised in Algorithm 5. The selfishness of a node measures the percentage of times (selfishness  $\in [0, 100]$ ) that a node behaves in a non-cooperative way.

The selfish utility function operates with two variables: the received amount of keys ( $rK$ ), and an indicator of the desired amount of keys that the node would like to receive when it contacts with the IM ( $dK \in [0, 1]$ ). When a node receives an amount of keys lower than the amount established by  $dK$ , the utility it obtains is 0, and when it receives an amount of keys higher or equal than the desired amount, the utility it obtains is its selfishness. This function models the interest of a node that wants to be as selfish as possible, to save resources, but also wants to keep its LC high enough to obtain a certain amount of keys.

#### Algorithm 5. Selfish utility function

---

**Input:**  $rK$ : Amount of received IBC keys.  
 $dK$ : Desired amount of IBC keys.  
**Output:** 0 or 1  
1: **if**  $rK > min_K + dK(max_K - min_K)$  **then**  
2:   **return** selfishness  
3: **else**  
4:   **return** 0  
5: **end if**

---

We have also defined an algorithm that models the strategy used by nodes to maximise their utility; it is shown in Algorithm 6. This algorithm updates nodes'

selfishness with the goal of obtaining the maximum utility possible. The update is performed every time they get new IBC keys from the IM. Essentially, nodes quickly reduce their selfishness when receive fewer keys than the desired amount, and they slowly increase their selfishness while the amount of obtained keys satisfies them.

#### Algorithm 6. Node strategy

---

**Input:**  $rK$ : Amount of received IBC keys.  
 $dK$ : Desired amount of IBC keys.  
1: **if**  $K > min_K + dK(max_K - min_K)$  **then**  
2:   selfishness = max (100, selfishness + 1);  
3: **else**  
4:   selfishness = min (0, selfishness - 2);  
5: **end if**

---

### 7.2. Scenario

The scenario we have used in all the simulations is based on the scenario presented by Borrego and Robles in [35]. This scenario consists of a mobile robot sensor network where messages use the time they are being carried by nodes to execute some tasks. These tasks are called sensing jobs and are injected into the network by the heterogeneous applications that coexist in the network. The multi-purpose approach of the network allows applications to deploy their own nodes. These nodes serve the sensing jobs of some applications without restrictions, forwarding their messages, prioritising their jobs, etc. These nodes could cooperate with all other applications in order to improve the performance of the whole network, but they have no incentive to do that.

We have chosen this scenario because it has particular characteristics that fit well with our incentive schema: (1) nodes have to return periodically to a base station to recharge their batteries or to deliver some data to the sink node, (2) it is a multi-purpose sensor network that works with different kinds of applications, and (3) nodes are as heterogeneous as the applications and can be deployed by different operators with different goals, so the idea of rational nodes that do not always act in a fully cooperative way makes sense.

According to this scenario, we have modelled the operation of the *sensing jobs* that travel among the network of the original publication using messages. A message that travels from one concrete node to another represents a job that has been executed in a node and wants to travel



**Fig. 5.** A snapshot of a simulation. On the upper side, a node carries two messages and waits for an opportunity to forward them. On the left side, one node tries to forward the only message it carries, but the other acts selfishly and rejects the transmission. On the right side, two cooperative nodes just forwarded messages between them, generating some proofs of forwarding.

to another one to continue its work, or a new job recently injected into the network which travels to its first destination. These messages are created with a frequency of one message every 40–80 s and their sizes are between 500 KB and 3 MB.

### 7.3. Simulation parameters

All simulations have been performed using *The Opportunistic Network Simulator* (The ONE) [36] (Fig. 5). We have developed or customised a set of classes that model the behaviour and the movement models of all participants.

We ran all simulations in a field of  $1500 \times 1500$  meters with 100 nodes implementing a custom random walk movement with random speeds between 0.5 m/s and 1.5 m/s that returns to the base station, located at (0, 0), when their IBC keys expire. The Incentive Manager is placed at the base station, without any movement model. Proofs are sent to the IM using direct delivery routing. As in [35], the communication range has been adjusted to 15 meters to all nodes.

Nodes may cooperate and forward messages, may not forward messages and save their own resources, or may drop messages. Dropping messages is a way to reduce the load of the network and assure that messages of their application will be slightly better treated (because there would be more space in the buffers, less congestion in nodes, etc.). Nodes decide how to behave with each message depending on their selfishness and the application that owns the message. Nodes model this behaviour using Algorithm 6 to update their selfishness. Then, when a node is requested to receive a message, it uses its selfishness value to decide probabilistically to behave selfishly or cooperatively. When acting selfishly, nodes randomly

choose either not accepting the message (50%) or dropping it (50%). When a node decides to be cooperative with one message, it uses Spray-and-Wait [37] to route it, using  $L = 3$ , where  $L$  is the maximum number of message copies present in the network. The initial selfishness of each node is randomly chosen at the beginning of the simulation.

Table 14 shows the chosen values of all parameters needed to run the simulations. Note that the amount of desired IBC keys of the nodes,  $dK$ , and the time constant  $T$  are not fixed, we have defined their values at every simulation in order to study their impact on the system. For the sake of simplicity and to avoid interferences with the simulations that make use of different values of  $dK$ , we disabled the waiting time of the nodes when  $rK < \min_K$ .

In order to obtain conclusive results, every simulation has been run five times with the same parameters but different random seeds, and all results have been calculated

**Table 14**  
Parameters used in the simulations.

Parameter	Value
Punishment $\alpha$	49 CP
Reward $\beta$	1 CP
Reward $\epsilon$	30 CP
Duration of keys	5 min
$\min_K$	3
$\max_K$	30
Interval of selfishness	[0%, 100%] <sup>a</sup>
Simulation time	1,000,000 s
Update interval $t$	500 s
Overhead time	3.3 s
Desired amount of keys $dK$	Defined for each simulation
Time constant $T$	Defined for each simulation

<sup>a</sup> The interval of selfishness, [Min selfishness, Max selfishness], has been fixed to [0%, 100%] for each simulation except those necessary to obtain results for the best case ([0%, 0%]) and for the worst case ([100%, 100%]).

as the average of the five runs. In total, we have executed 150 runs.

7.4. Results. Study of the time constant

This set of tests has been designed to identify how risky and likely selfish bursts are in relation to the time constant  $T$  and to study the impact of  $T$  regarding the fairness of the incentive schema. We ran simulations with different  $T$  values between 5000 and 35000 s and we disabled the update behaviour of all nodes. This models a scenario where nodes ignore the incentives and do not care about their Level of Cooperation. Nodes randomly chose their selfishness at the beginning of the simulation and held it until the end.

To know if this incentive schema allows nodes to save high amounts of CP and then behave selfishly without being punished, we have developed the metric *messages to loose (mtl)*, that is calculated using the following equation.

$$mtl = \frac{\max_{balance} - \min_{balance}}{\alpha} \tag{3}$$

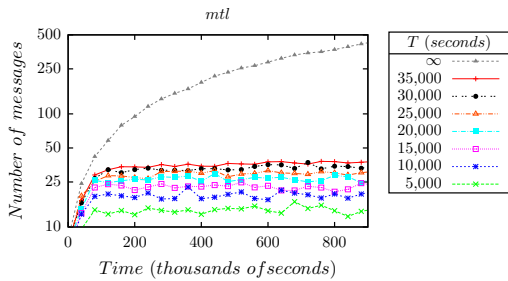


Fig. 6. Number of messages that the node with the highest LC can lose before becoming the node with the lowest balance, note the logarithmic scale. Higher values of  $T$  imply higher values of *mtl* because the LC of the nodes decrease slowly and present behaviour are less important compared with past behaviours. When no decay is applied ( $T = \infty$ ), *mtl* increases linearly.

What *mtl* defines is the amount of messages that the node with the highest Level of Cooperation of the system can lose before its LC becomes the lowest, assuming all other nodes' LC will stay frozen. We have studied this metric for a set of different  $T$  values.

As can be seen in Fig. 6, after the transient period, the number of messages that can be lost becomes stable at a higher or smaller value with small peaks, depending on the  $T$  used. There is an exception when  $T = \infty$ , which means that there is no decay function applied, and nodes can infinitely save CP from their past actions, increasing the risk of selfish bursts and the damage they can do.

Note that highly increasing  $T$  lead to very small increases of *mtl*. Besides, it is important to note that the average number of messages flowing through the network is 1945.56. Therefore, *mtl*, that oscillates between 14.2 and 36.5, suppose a percentage between the 0.73% and the 1.89% of the total messages of the network.

Fig. 7 shows the average Level of Cooperation of all nodes inside every selfishness interval. We only show the results of three of the  $T$  simulated values, 25,000 and 20,000 and 15,000 s because they are enough to understand what we address here. All graphics, the three included here, and all others, show the same pattern: after a transient state, all nodes stabilize their LC and hold it without major changes until the end of the simulation. As the time constant  $T$  becomes higher, the values where the nodes stabilize their LC are higher too. In all cases, nodes with lesser selfishness have a higher LC almost all the time, except during occasional, little time intervals. In these intervals, it is possible that a category (e.g. 20–30%) has a higher LC than the immediately less selfish category (e.g. 10–20%). These intervals become a little more frequent as the time constant  $T$  becomes lower.

On the basis of the previous results, we can conclude that a higher  $T$  is better for the fairness of the system because it increases the differences between the LC obtained by a selfish node and a cooperative one. But an extreme  $T = \infty$ , meaning no decay is applied, allow nodes

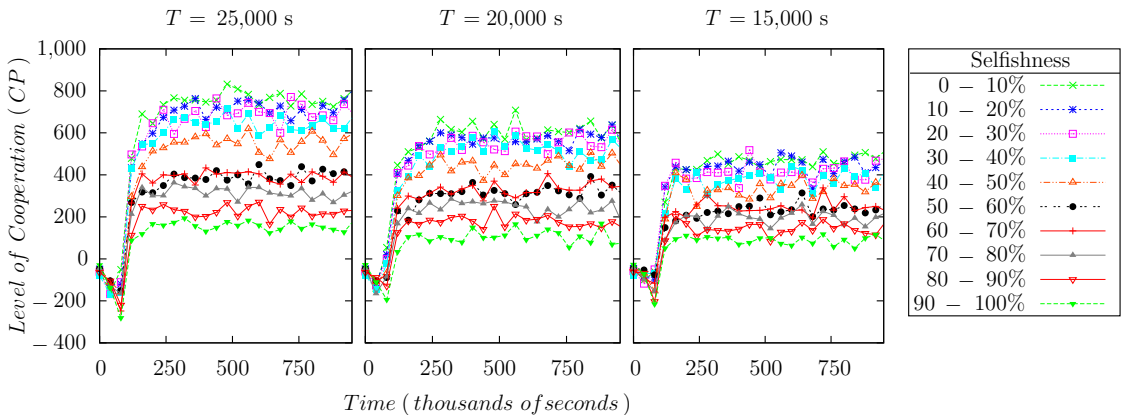


Fig. 7. Average Level of Cooperation of all nodes inside every selfishness interval using different values for the time constant  $T$ . The three graphics share the Y axle. After a transient period, nodes stabilize their LC. As  $T$  becomes smaller, the value where the LC stabilizes and the differences between categories become lesser too, and moments where nodes of one selfish category have a higher LC than nodes of a more cooperative category become slightly more frequent.

to perform selfish bursts without prejudice, so the time constant has to be chosen high enough to assure fairness, but not too high to avoid this.

### 7.5. Results. Impact of message expiration

In DTN, messages usually have a time-to-live (*tll*) and they are discarded when their expiration time is reached. The presented Incentive Scheme punishes the node that discards a message and the previous one. Taking into account that the reason why a node has discarded a message cannot be known, it is mandatory to study the fairness of the system when dealing with this. We need to be sure that nodes will not refuse to accept messages because they are afraid of being punished if the messages expire.

We have measured the *mean absolute error (mae)*, this measures how close is the ranking of nodes elaborated by the IM to an ideal ranking of cooperative nodes elaborated by an omniscient entity that knows all about nodes' actual and past behaviours. We have calculated *mae* using Eq. (4).

$$mae = \frac{1}{|N|} \sum_{i=0}^{|N|-1} |lcr(i) - rbr(i)| \quad (4)$$

where

- *lcr(i)* is the *level of cooperation rank*, the position of node *i* on a list of all nodes of the network, ordered by level of cooperation.
- *rbr(i)* is the *real behaviour rank*, the position of node *i* on a list of all nodes of the network, ordered by their real behaviour, elaborated by an omniscient entity.
- *|N|* is the amount of nodes of the network.

On this set of simulations we measured the fairness of the presented Incentive Scheme to handle message expiration. We have fixed *T* to 20,000 s. We have ran the experiments without message expiration to obtain the lower bound of the *mae* and the average latency. Then, the *tll* of each message was set to 10,000, 20,000 or 30,000 s (one third, two thirds, and one time the average latency without message expiration).

Fig. 8 shows the average *mae* obtained when different *tll* are used. As can be seen, each high decrease of the *tll* leads to a small increase of the mean absolute error. The reason is that the punishment applied to nodes when the messages expire is distributed among all nodes. Besides, cooperative nodes are more exposed to message expiration, but they also have a higher LC, meaning they are more resistant to the punishment. As a result, when a message expires, the LC of its custody node is reduced, but its position on the overall ranking remains almost unchanged. This means that the Incentive Scheme is fair with the nodes regarding message expiration.

### 7.6. Results. Performance of the network

To perform this set of simulations, we have chosen *T* = 20,000 s. Besides which, we have enabled the update

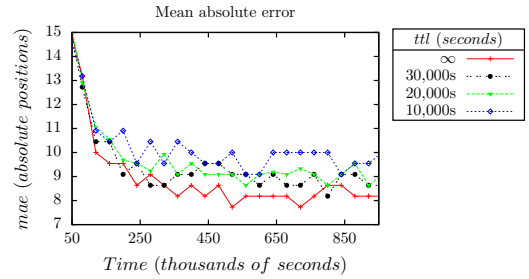


Fig. 8. Mean absolute error depending on the *tll*. The obtained results are very similar. High decreases of the *tll* lead to small increases of the *mae*.

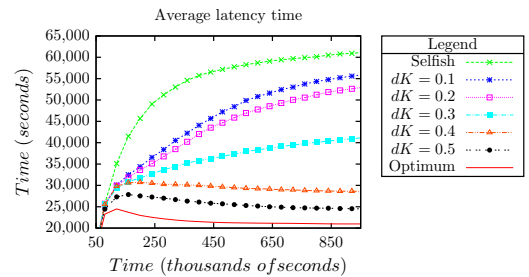


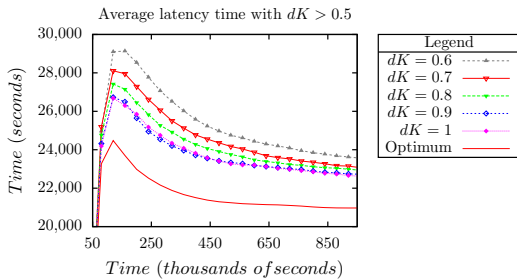
Fig. 9. Latency average depending on the *dk* of nodes. If nodes conform with obtaining 20% or less of the keys, the latency decreases a little. The latency is reduced a 30% if nodes want to obtain at least the 30% of the keys. When nodes want to obtain the 40% of the keys or more, the latency is reduced more than 50%.

behaviour of all nodes and we have run simulations with different values of *dk*.

Fig. 9 shows the average latency (defined as the time required by messages to travel from their source to their destination) of a network where nodes share the *dk* parameter. The selfish case has been measured with all nodes behaving with a 100% of selfishness, meaning the only way a message can arrive at its destination is using direct delivery, so it can be considered the worst case, the upper bound. The optimum case has been measured with all nodes behaving with a 0% of selfishness, so it can be considered the best case, the ideal network where all nodes are fully cooperative: the lower bound.

The obtained results are extremely successful because they show that the incentive system improves the performance of the network significantly. Even when *dk* is as low as 0.3, the latency of the network is reduced by about 30%. When *dk* > 0.3 which definitely is not a strong requirement, the latency of the network is reduced by more than 50% and approaches a lot the lower bound.

Fig. 10 shows the average latencies obtained with *dk* > 0.5. As can be seen, the differences are not significant. This is a good point because it means that the incentive system does not need nodes to care a lot about the amount of IBC keys they receive to improve the overall performance of the network. Even in networks where nodes are easily satisfied when they obtain half of the keys they could obtain, or



**Fig. 10.** Latency average for values of  $dK$  greater than 0.5. Differences between results obtained with values of  $dK$  greater than 0.5 are very small. In all cases, the obtained performance of the network is similar to the performance obtained in the Optimum case.

even little less than half, the incentive scheme provides an important increase in network performance.

We can conclude that the presented incentive scheme, that punishes and rewards nodes on the basis of their behaviour by giving them a higher or lesser amount of IBC keys, forces nodes to be more cooperative, and improves the performance of the network by reducing the latency of the messages by more than 50%.

## 8. Conclusions and future work

We have presented an asynchronous incentive scheme for DTNs. This scheme is based on a receipt exchange protocol designed to overcome the inherent limitations of DNTs and uses the policy “guilty until proven innocent” to punish suspicious nodes and reward cooperative nodes.

Moreover, we have developed a new way of tracking the actions of nodes that allow us to treat nodes in a new, fair way: they will be rewarded for the actions they perform, without depending on other elements like the actions performed by others, the delivery ratio of the routing algorithm used, etc.

The game theory analysis of the incentive scheme has proven that, for each node, accepting and relaying messages, delivering receipts to the Incentive Manager and avoiding cheating is a dominant strategy and the best response to any other node behaviour. This way, we have proved that all nodes behaving this way form a Nash equilibrium.

The results of the simulations show that the usage of the incentive scheme improves the performance of a network, even if nodes try to behave in a selfish way, ignoring their balance. In the concrete scenario of a wireless robot sensor grid network with heterogeneous nodes and applications, latency is reduced by more than 50%. And this improvement is obtained with the only requirement that nodes has to want to obtain at least 40% of the keys they could obtain.

As a future line of research, we plan to modify the system by using different types of IBC keys that only allow nodes to perform a subset of all the possible actions, this way we plan to improve the enforcing mechanism by increasing the punishment to uncooperative nodes, but at the same time giving them more options to redeem and recover. We also plan to be specially focused on adapting

this scheme to incentive nodes to behave in a certain way not only in terms of routing and cooperation but also in terms of movement and location, improving the coverage of the network and the quality of the opportunistic contacts. Finally, we think that re-designing the incentive scheme, respecting the main principles, to charge nodes for every message sent will allow the scheme to be useful to a wider range of networks.

## Acknowledgements

This work has been partially funded by the Ministry of Science and Innovation of Spain, under the reference Project TIN2010-15764 and by the Catalan Government under the reference Project 2014SGR691.

## References

- [1] S. Farrell, V. Cahill, *Delay- and Disruption-Tolerant Networking*, Artech House, Inc., Norwood, MA, USA, 2006.
- [2] K. Scott, S. Burleigh, Bundle Protocol Specification, RFC 5050 (Experimental), 2007.
- [3] M. Joye, G. Neven, *Identity-Based Cryptography*, IOS Press, 1013 BG, Amsterdam, The Netherlands, 2008.
- [4] S. Giordano et al., *Mobile ad hoc networks, Handbook Wirel. Netw. Mob. Comput.* (2002) 325–346.
- [5] S.-B. Lee, G. Pan, J.-S. Park, M. Gerla, S. Lu, Secure incentives for commercial ad dissemination in vehicular networks, in: Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '07, ACM, New York, NY, USA, 2007, pp. 150–159. <http://dx.doi.org/10.1145/1288107.1288128>.
- [6] A. Garyfalos, K. Almeroth, Coupons: a multilevel incentive scheme for information dissemination in mobile networks, *IEEE Trans. Mob. Comput.* 7 (6) (2008) 792–804. <http://dx.doi.org/10.1109/TMC.2008.37>.
- [7] H. Zhu, X. Lin, R. Lu, Y. Fan, X. Shen, Smart: a secure multilayer credit-based incentive scheme for delay-tolerant networks, *IEEE Trans. Veh. Technol.* 58 (8) (2009) 4628–4639. <http://dx.doi.org/10.1109/TVT.2009.2020105>.
- [8] R. Lu, X. Lin, H. Zhu, X. Shen, B. Preiss, Pi: a practical incentive protocol for delay tolerant networks, *IEEE Trans. Wirel. Commun.* 9 (4) (2010) 1483–1493. <http://dx.doi.org/10.1109/TWC.2010.04.090557>.
- [9] L. Buttyán, J.-P. Hubaux, Nuglets: A Virtual Currency to Stimulate Cooperation in Self-organized Mobile Ad Hoc Networks, Tech. Rep., Swiss Federal Institute of Technology, 2001. doi:DSC/2001/001.
- [10] L. Buttyán, L. Dora, M. Félegyházi, I. Vajda, Barter-based cooperation in delay-tolerant personal wireless networks, in: 2013 IEEE 14th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), vol. 0, 2007, pp. 1–6.
- [11] L. Buttyán, L. Dóra, M. Félegyházi, I. Vajda, Barter trade improves message delivery in opportunistic networks, *Ad Hoc Netw.* 8 (1) (2010) 1–14. <http://dx.doi.org/10.1016/j.adhoc.2009.02.005>.
- [12] N. Nisan, Algorithms for selfish agents, in: C. Meinel, S. Tison (Eds.), STACS 99, Lecture Notes in Computer Science, Symposium on Theoretical Aspects of Computer Science, vol. 1563, Springer Berlin Heidelberg, 1999, pp. 1–15. [http://dx.doi.org/10.1007/3-540-49116-3\\_1](http://dx.doi.org/10.1007/3-540-49116-3_1).
- [13] C. Papadimitriou, Algorithms, games, and the internet, in: Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing, STOC '01, ACM, New York, NY, USA, 2001, pp. 749–753. <http://dx.doi.org/10.1145/380752.380883>.
- [14] T. Roughgarden, E. Tardos, How bad is selfish routing?, *J ACM* 49 (2) (2002) 236–259. <http://dx.doi.org/10.1145/506147.506153>.
- [15] S. Zhong, J. Chen, Y. Yang, Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks, *INFOCOM 2003, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*, vol. 3, IEEE Societies, 2003, pp. 1987–1997. <http://dx.doi.org/10.1109/INFOCOM.2003.1209220>.
- [16] A. Balasubramanian, B. Levine, A. Venkataramani, Dtn routing as a resource allocation problem, *SIGCOMM Comput. Commun. Rev.* 37 (4) (2007) 373–384. <http://dx.doi.org/10.1145/1282427.1282422>.
- [17] U. Shevade, H.H. Song, L. Qiu, Y. Zhang, Incentive-aware routing in DTNs, in: IEEE International Conference on Network Protocols, 2008.

- ICNP 2008, 2008, pp. 238–247. <http://dx.doi.org/10.1109/ICNP.2008.4697042>.
- [18] D. Levin, K. Lacurts, N. Spring, B. Bhattacharjee, Bittorrent is an auction: analyzing and improving bittorrent's incentives, in: ACM SIGCOMM Conference, 2008, pp. 243–254. <http://dx.doi.org/10.1145/1402958.1402987>.
- [19] A. Seth, D. Kroeker, M. Zaharia, S. Guo, S. Keshav, Low-cost communication for rural internet kiosks using mechanical backhaul, in: Proceedings of the 12th Annual International Conference on Mobile Computing and Networking, MobiCom '06, ACM, New York, NY, USA, 2006, pp. 334–345. <http://dx.doi.org/10.1145/1161089.1161127>.
- [20] F. Milan, J.J. Jaramillo, R. Srikant, Achieving cooperation in multihop wireless networks of selfish nodes, in: Proceeding from the 2006 Workshop on Game Theory for Communications and Networks, GameNets '06, ACM, New York, NY, USA, 2006. <http://dx.doi.org/10.1145/1190195.1190197>.
- [21] J.J. Jaramillo, R. Srikant, Darwin: distributed and adaptive reputation mechanism for wireless ad-hoc networks, in: Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking, MobiCom '07, ACM, New York, NY, USA, 2007, pp. 87–98. <http://dx.doi.org/10.1145/1287853.1287865>.
- [22] D. Irwin, J. Chase, L. Grit, A. Yumerefendi, Self-recharging virtual currency, in: Proceedings of the 2005 ACM SIGCOMM Workshop on Economics of Peer-to-peer Systems, P2PECON '05, ACM, New York, NY, USA, 2005, pp. 93–98. <http://dx.doi.org/10.1145/1080192.1080194>.
- [23] H. Zhu, S. Du, Z. Gao, M. Dong, Z. Cao, A probabilistic misbehavior detection scheme toward efficient trust establishment in delay-tolerant networks, *IEEE Trans. Parall. Distrib. Syst.* 25 (1) (2014) 22–32. <http://dx.doi.org/10.1109/TPDS.2013.36>.
- [24] B.N. Chun, P. Buonadonna, A. Auyoung, C. Ng, D.C. Parkes, J. Sheidman, A.C. Snoeren, A. Vahdat, Mirage: a microeconomic resource allocation system for sensornet testbeds, in: Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors, 2005.
- [25] D.G. Sullivan, M.I. Seltzer, Isolation with flexibility: a resource management framework for central servers, in: Proceedings of the Annual Conference on USENIX Annual Technical Conference, ATEC '00, USENIX Association, Berkeley, CA, USA, 2000, pp. 27–27.
- [26] Q. Li, G. Cao, Mitigating routing misbehavior in disruption tolerant networks, *IEEE Trans. Inform. Forensics Secur.* 7 (2) (2012) 664–675. <http://dx.doi.org/10.1109/TIFS.2011.2173195>.
- [27] S. Marti, T.J. Giuli, K. Lai, M. Baker, Mitigating routing misbehavior in mobile ad hoc networks, in: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, MobiCom '00, ACM, New York, NY, USA, 2000, pp. 255–265. <http://dx.doi.org/10.1145/345910.345955>.
- [28] S. Kremer, O. Markowitch, J. Zhou, An intensive survey of fair non-repudiation protocols, *Comput. Commun.* 25 (17) (2002) 1606–1621.
- [29] O. Markowitch, Y. Roggeman, Probabilistic non-repudiation without trusted third party, in: 2nd Conference on Security in Communication Networks, Amalfi, Italy, 1999.
- [30] J. Mitsianis, A new approach to enforcing non-repudiation of receipt, manuscript, 2001.
- [31] J. Liu, R. Sun, W. Ma, Y. Li, X. Wang, Fair exchange signature schemes, in: 22nd International Conference on Advanced Information Networking and Applications – Workshops, 2008, AINAW 2008, 2008, pp. 422–427.
- [32] N. Asokan, K. Kostianinen, P. Ginzboorg, J. Ott, C. Luo, Applicability of identity-based cryptography for disruption-tolerant networking, in: MobiOpp '07: Proceedings of the 1st International MobiSys Workshop on Mobile Opportunistic Networking, ACM, New York, NY, USA, 2007, pp. 52–56. <http://dx.doi.org/10.1145/1247694.1247705>.
- [33] Y. Cao, Z. Sun, Routing in delay/disruption tolerant networks: a taxonomy, survey and challenges, *IEEE Commun. Surv. Tutorials* 15 (2) (2013) 654–677. <http://dx.doi.org/10.1109/SURV.2012.042512.00053>.
- [34] J.C. Harsanyi, *Rational Behaviour and Bargaining Equilibrium in Games and Social Situations*, Press Syndicate of the University of Cambridge, New York, USA, 1977.
- [35] C. Borrego, S. Robles, A store-carry-process-and-forward paradigm for intelligent sensor grids, *Inform. Sci.* 222 (2013) 113–125.
- [36] A. Keränen, J. Ott, T. Kärkkäinen, The ONE simulator for DTN protocol evaluation, in: SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, ICST, New York, NY, USA, 2009.
- [37] T. Spyropoulos, K. Psounis, C. Raghavendra, Spray and wait: an efficient routing scheme for intermittently connected mobile networks, in: Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking, ACM, 2005, p. 259.



**Adrián Sánchez-Carmona** Born in Terrassa, Barcelona. He received his degree in Computer Science (5 year programme) at the Universitat Autònoma de Barcelona (UAB). In 2013 he obtained the Master Degree on Security on Information Technology and Communications (UOC-UAB-URV). After finishing his studies he started his PhD. He is actually a PhD student at the Department of Information and Communications Engineering (dEIC).



**Sergi Robles** received his PhD in Computer Science from Universitat Autònoma de Barcelona. He is an associate professor in the Department of Information and Communications Engineering at the Universitat Autònoma de Barcelona, where he leads the Security of Networks and Distributed Applications (SeNDA) research group. His latest research interests include mobile agents and security, and routing in Delay Tolerant Networks.



**Carlos Borrego** Born in Madrid. He received his degree in Computer Science (6 year programme) at the Faculty of Computer Science at the Polytechnic University of Madrid. After finishing his studies he moved to work for CERN (Geneva, Switzerland). In 2001 moved to CASPUR, University La Sapienza (Rome, Italy) and stayed there for four years. In 2005 moved to the Autonomous University of Barcelona (Barcelona, Spain) where he finished his PhD and worked for Pic and Ifae research centers. He is actually researcher and adjunct professor at the Department of Information and Communications Engineering dEIC. He gives lectures on computer networks and cryptography.





“These military devices, leading to victory, must not be divulged beforehand.”

*The Art of War*, SUN TZU

# 5

## PrivHab: A privacy preserving georouting protocol based on a multiagent system for podcast distribution on disconnected areas

Next, we reproduce the following article, which has been published on the international peer-reviewed journal *Ad Hoc Networks*, a second quartile JCR journal with an impact factor of 1.660.

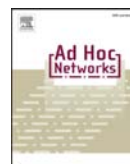
*A. Sanchez-Carmona, S. Robles, C. Borrego. PrivHab: A privacy preserving georouting protocol based on a multiagent system for podcast distribution on disconnected areas. Ad Hoc Networks (2016) ISSN: 1570-8705. DOI: 10.1016/j.adhoc.2016.09.019*

This work was submitted for the first time on 03-Dec-2015, it was revised twice, on 22-Jun-2016 and on 26-Jul-2016, and it was finally accepted on 28-Sep-2016.



Contents lists available at ScienceDirect

## Ad Hoc Networks

journal homepage: [www.elsevier.com/locate/adhoc](http://www.elsevier.com/locate/adhoc)

# PrivHab: A privacy preserving georouting protocol based on a multiagent system for podcast distribution on disconnected areas



Adrián Sánchez-Carmona\*, Sergi Robles, Carlos Borrego

Department of Information and Communications Engineering (dEIC), Universitat Autònoma de Barcelona (UAB), Spain

## ARTICLE INFO

## Article history:

Received 3 December 2015

Revised 26 July 2016

Accepted 22 September 2016

Available online 28 September 2016

## Keywords:

Routing protocols

Mobility-tolerant communication

Privacy

Location tracking

Delay and disruption tolerant networks

## ABSTRACT

We present PrivHab, a privacy preserving georouting protocol that improves multiagent decision-making. PrivHab learns the mobility habits of the nodes of the network. Then, it uses this information to dynamically select to route an agent carrying a piece of data to reach its destination. PrivHab makes use of cryptographic techniques from secure multi-party computation to make the decisions while preserving nodes' privacy. PrivHab uses a waypoint-based routing that achieves a high performance and low overhead in rugged terrain areas that are plenty of physical obstacles. The store-carry-and-forward approach used is combined with mobile agents that provide intelligence, and it is designed to operate in areas that lack network infrastructure. We have evaluated PrivHab under the scope of a realistic podcast distribution application in remote rural areas, where these programs have to be recorded into a physical format and distributed to the local radio stations. The usage of PrivHab aims to reduce this spending of resources. The PrivHab protocol is compared with a set of well-known delay-tolerant routing algorithms and shown to outperform them.

© 2016 Elsevier B.V. All rights reserved.

## 1. Introduction and motivation

In 2003, the Food and Agriculture Organization of the United Nations (FAO<sup>1</sup>) implemented a strategic Programme entitled "Bridging the Rural Digital Divide". The programme highlighted innovative approaches to knowledge exchange that were taking advantage of new digital technologies, and that were based on synergies between information management and communication for development.

Thenceforth, many initiatives have been implemented in fields as e-health, e-government, e-education, e-commerce and e-agriculture. The common goal of these initiatives is to universalize the access to knowledge and information in order to improve the life conditions of people living in developing countries. These applications have to overcome barriers like illiteracy, low cultural level of the population, censorship, etc. E-agriculture services, e.g. Agriwatch [1], use all the technologies at their reach: web, email, telephone, SMS, videos, printers, mail, etc. but even this way, they are constrained by the need of infrastructure and cannot operate in regions lacking it. It happens that regions where the com-

munication networks are unavailable or spotty, where these services can not be implemented, are usually the ones where these e-agriculture services would be more needed and valuable. Unfortunately, this situation is not likely to change because the low-population density and low-income level make economically infeasible or uninteresting to extend the operators' networks into these regions.

We propose to use PrivHab to reduce the digital divide in developing countries by distributing podcast radio programs among local radio stations or other places of interest using Mobile Agent based Delay Tolerant Networking (MADTN) [2]. MADTN, as DTN [3,4] uses the store-carry-and-forward strategy to operate in challenged scenarios where there are no simultaneous end-to-end paths, but it substitutes DTN's bundle (just a container of data) by a Mobile Agent, a software entity that carries the data and makes their own intelligent decisions.

Our proposal consists in creating a network of handheld devices carried by persons, and to use mobile agents that will move through this network to transport the data. Thanks to PrivHab, these agents will be able to make their own routing decisions based on the usual whereabouts of the people carrying the devices, while preserving their privacy.

Our main contributions are summarized below:

- We present an e-agriculture application, based on a real need, that improves the podcast distribution in rural areas where we

\* Corresponding author.

E-mail addresses: [adria.sanchez@deic.uab.cat](mailto:adria.sanchez@deic.uab.cat) (A. Sánchez-Carmona), [sergi.robles@deic.uab.cat](mailto:sergi.robles@deic.uab.cat) (S. Robles), [carlos.borrego@deic.uab.cat](mailto:carlos.borrego@deic.uab.cat) (C. Borrego).<sup>1</sup> More information can be found on <http://www.e-agriculture.org/bridging-rural-digital-divide-programme-overview>.

cannot rely on conventional communication networks to distribute them.

- We lay the foundations of a multi-agent intelligent system that helps the decision-making of the agents that carry the messages, while providing the enough flexibility to let them make their own decisions.
- We define the habitat, the area where a node is more likely to be found, we explain how to exploit the existence of life-cycles of the network users to define it and we model it in a simple way to allow operating it under the scope of an additive homomorphic cryptosystem.
- We define PrivHab, the first geographical routing protocol that uses the habitat to route the agents based on long-term predictions. To protect this information and to avoid its disclosure, PrivHab cryptographically protects it to ensure the habitat become hidden to the other nodes of the network.

To our knowledge, PrivHab is the first privacy preserving routing protocol that uses a geographical routing based in long-term predictions. For this reason, this is also the first work that considers the privacy of a routing information other than the historic of contacts with the other nodes of the network and that provides the tools that make this possible.

The rest of this article is organized as follows. In Section 2, we present an e-agriculture application of podcast distribution that can be enhanced through the usage of PrivHab. Section 3, summarizes the related work. In Section 4, we present the architecture of the multiagent system. In Section 5, we present the habitat, the core concept of PrivHab. In Section 6, we present PrivHab, a protocol that use the habitats of the nodes to route the messages towards its destination while preserving the privacy of the nodes of the network. In Section 7, we expose the results of the experiments made to measure PrivHab's performance. Finally, Section 8 concludes this paper.

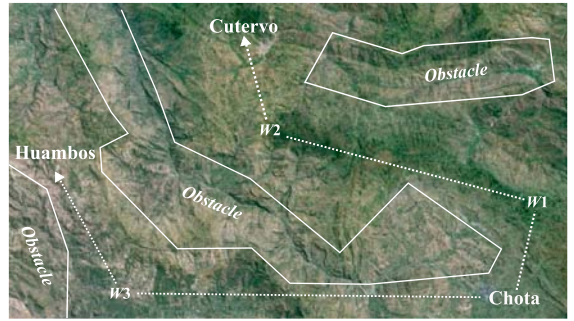
## 2. Scenario of application

In this section, we present a practical example of an e-agriculture application podcast distribution on disconnected areas. This application could be greatly enhanced by using Mobile Agent based Delay Tolerant Networking, the concept of habitat and PrivHab.

### 2.1. Podcast distribution

In some places, due to the region's dialect preference and the illiteracy ratios, radio broadcasting is the most important information source for farmers. It plays a key role in the economy development of the region by disseminating important agricultural information. This is the main way these farmers can obtain information as valuable as what are the most appropriate crops for each season, or the most efficient processing techniques of raw materials, among others.

In the Cajamarca region, in Perú, the Non-Governmental Organization (NGO) *Practical Action*<sup>2</sup> records podcast radio programmes targeted to farmers in Compact Discs and physically distributes them to the local radio stations. The podcasts contain small how-to explanations, newsletters, information about prices, etc. This slow distribution method requires the NGO to spend monetary or personnel resources to bring a copy to every small local station. We aim to replace this physical distribution by a digital and automated one.



**Fig. 1.** Map of a scenario of application located in a rural area of Cajamarca (Perú). White lines are natural obstacles approximate limits. Podcasts sent from the village of Chota to Cutervo have to be routed through waypoints W1 and W2 while messages sent from Chota to Huambos have to be routed through waypoint W3.

We propose to create a Delay Tolerant Network using a set of small devices that can be carried by the members of the NGO's staff or by some local villagers that collaborate with them. If it is needed, some devices can also be deployed on strategic locations. We propose to implement an automatic distribution of podcasts using this network. The deployment's cost of the nodes should be low<sup>3</sup>, and can be considered as an investment, since the NGO will not need to spend more resources on the podcast distribution.

Between the NGO and the local radio stations there could be barriers that nodes carrying the data can not cross, like a cliff, one river without a bridge or a mine field. Moreover, there could be areas that nodes can only cross slowly, like a mountain or forest region, or a river with the nearest bridge located a few kilometres away. Besides, there are some interest locations like markets, churches, or the NGO's offices, that are very likely to have a higher density of nodes. Moreover, there are some zones where nodes can move quickly, due to the quality of tracks or roads, the existence of bridges or the usage of alternative means of transport. Therefore, data should try to avoid the problematic areas and follow paths that take advantage of these interesting zones or locations. This can even imply temporarily moving away the data from its destination. Fig. 1 provides an example: when data from Chota is first routed to the waypoint W1 instead of directly towards Cutervo, the destination. This is a constraint that make georouting protocols that assume a plain world without obstacles, like LAROD [5], unusable. For these reasons, PrivHab allows the sender to define a list of locations (called waypoints) where the data has to pass by in order to reach its destination.

Finally, there are two requirements that can be of great value to the NGO: 1) It has to respect the privacy of its users; and 2) It has to be able to achieve a good performance occupying a small buffer and using fewer resources. A cooperator that has received a device from the NGO in order to distribute the podcasts in an area may not be very concerned about the privacy of its habitat or the amount of buffer occupied by the podcasts. However, if the NGO wants to extend the network cheaply by adding other types of nodes, e.g. volunteers that want to help the NGO by becoming part of the network, it is desirable to reduce as much as possible the impact on the users' devices and lives. Note that lack of privacy has been identified as one of the main reasons for the unwillingness of users to participate in DTN [6].

<sup>2</sup> More information about this programme at <http://practicalaction.org/podcasting-3>

<sup>3</sup> Small devices like Raspberry Pi can be acquired by less than 30\$/unit.

### 3. Related work

In this Section, we first explain the reason why we take a reactive approach instead of a planning one. Then we provide the reader with a review of the related work. We present the state of the art of Geographical Routing Protocols. Later, we analyse the different proposals of Privacy Preserving Routing Protocols in Delay Tolerant Networks. Then, we review some Social-based Routing Protocols that are related, somehow, to our proposal.

Although we realize that the presented problem, distributing podcasts through an opportunistic network, is similar to those solved by multi-agent planning, given the characteristics of the scenario (a DTN), a reactive approach may fit better than a planning approach. As said in [7], an efficient and fast algorithm for selecting candidates on-the-fly is required when the mobility of the nodes produces a changing topology. This is exactly what DTN routing algorithms do. Besides, due to the absence of simultaneous end-to-end paths and network infrastructure, it may take long to obtain the habitats of the nodes in order to plan an itinerary “a priori” because there are nodes that will never establish a direct communication with the sender. Moreover, given the evolving nature of the habitats, they may change before the agent’s arrival, making useless most of the planning effort. Therefore, situations where an agent meet a node with an unexpected habitat that it is useful to bring the message towards its destination, can only be exploited if the decisions are made locally, when this information is still in force.

#### 3.1. Geographical routing protocols

Geographical Routing Protocols have been studied both in Ad-hoc Networks and Delay Tolerant Networks. Most protocols only take into account the position of the nodes at the moment of the transmission, but not their movement pattern. LAROD [5] forwards packets to neighbours inside a certain area located between the forwarder and the destination, without taking into account the mobility patterns of these nodes. In [8], a Location Service called LoDIS is presented to improve LAROD by using gossip-based techniques to update the location of the destination at each hop. LoDIS improves the performance of the routing at the cost of the privacy of all nodes, because it periodically broadcasts their locations and speed vectors. GeoDTN+Nav [9] is designed for routing in a network of streets, and it has three forwarding modes. In the DTN mode, it requires the nodes to know where they are heading. This requirement can be easily met by certain types of vehicles, like buses or taxis, but it is an important restriction in scenarios where nodes are carried by people. LSGO [10] is a georouting protocol designed to work in Vehicular Networks where nodes forward messages to a neighbour based on its location and the link’s quality. LSGO’s main objective is to avoid retransmissions, but its geographic component, that takes into account only the actual location of the involved nodes, is poor. GSPI [11] is a geographic routing protocol for vehicular networks that uses greedy mode on straight roads and to use predictive mode at the intersections, but its predictive mode is short-termed, as it uses the current position and the speed vector of the nodes. GPRP [12] improves this approach by dividing roads into two-dimensional road grids and considering every possible node movement while predicting. This restricts the position prediction in the road grid sequence and improves the performance of the network, but makes this proposal hardly applicable to other kinds of scenarios and difficult its deployment.

As it can be seen, almost all proposals use contemporaneous information and short-term predictions, so they fail to take into account long-term trends of nodes’ mobility. However, in scenarios where the distances to travel are big, and the density of nodes is

low, it is more valuable to know where a node will go in the next hours than where it is currently headed.

#### 3.2. Privacy preserving routing protocols

Privacy Preserving Routing Protocols are based on the assumption that nodes are not willing to voluntarily share any information for the good of the network, and that nodes’ privacy should be preserved in order to stimulate them to become members of the network. ALAR [13] allows a source to send a message through a DTN without revealing its physical location and proposes an anti-localization routing protocol. However, the only information that ALAR protects is the location where the source was when the message was sent. An anonymous communication solution for DTN has been presented in [14], but this one is designed to hide the identity of the nodes, not to protect the private information that these nodes use to make routing decisions. SPRING [15] is a routing protocol designed to vehicular DTN that bases its operation on the deployment of Roadside Units (RSUs) and on the usage of a group signature technique called CPPA [16]. Although its approach is similar to the one of our proposal, SPRING routes messages using a variation of Epidemic [17], and what it hides is the identity of the source node and its location at the moment when the message was sent. In [18], the authors present a generic routing protocol that preserves the privacy of the *routing metric* through the usage of the cryptographic tools derived from the “Yao’s millionaire problem” [19]. This proposal requires both parts of the transaction to be able to calculate their *routing metric* on their own, so it can not be used when the parts need to collaborate to calculate it. A prediction-based privacy preserving routing algorithm is presented in [20]. Hasan et al. provide a way to calculate the maximum probability of delivery within a community without disclosing node’s private information, then, messages that have been disseminated through the community in an epidemic way are routed to other communities if their maximum probabilities of delivery are better. This protocol is designed to work in scenarios where the connectivity, at least inside the communities, is relatively high. SimBet-BF [21] protect the nodes’ contacts information by blurring them using Bloom Filters at the beginning of every contact. Then, it uses two metrics, the *ego betweenness centrality* and the *similarity* to make the routing decisions. In [22] the privacy is also preserved by obfuscating the social network graph announced to the neighbours to make routing decisions. Finally, PRISM [23] routes messages towards a location while preserving the privacy of the nodes, but does not allow the source to decide the identity of the message’s destination.

Unfortunately, most Privacy Preserving Routing Protocols aim to protect the nodes’ contacts information, and their routing usually uses the past contacts of a node to try to predict probability of a new contact in the future. Other informations, as the identity or the locations of the nodes may be protected as well, but to our knowledge, there are no other proposals that preserve more complex informations used to make georouting decisions.

#### 3.3. Social-based routing protocols

There are some Social-based routing protocols that are related, somehow, to the present work. Social-based routing protocols are based on the idea of using the recent past to model the behaviour of a node to predict how it will behave in the near future. BUBBLE RAP [24] classifies nodes using their popularity inside their community. Then, messages are forwarded to more popular nodes until they reach the community of the destination. Its design does not consider hop-distant destinations nor geographic restrictions. So, during the first hops messages can be moved into the opposite direction of their destination while they are forwarded to more pop-

ular nodes. MobySpace [25] leverages the life-cycles of the nodes to track the most visited by every node points of interest. These life-cycles are modelled this using a multi-dimensional probability vector, and messages are forwarded to nodes with a vector that it is closer to the one of the destination. This is a very interesting approach to our concept of habitat, but lacks adaptability. In MobySpace, the points of interest have to be defined *a priori*, and some infrastructure is needed to allow nodes to detect if they are close to these points. Besides, MobySpace may lead to situations where a node that spends most of the time at point A, very close to B, is considered a bad choice because the destination is expected to be on B, without taking into account that A is geographically close to B. SANE [26] uses the same principles but defines the points of interest in a very broad sense, allowing the usage of more abstract concepts, and compares nodes using a metric called “cosine similarity”. HiBoP [27] extends this approach using any contextual information about nodes to make routing decisions. One of its drawbacks is the big amount of memory needed to store information about every other node. Besides, the authors do not explain how this contextual information can be updated as the behaviours of the nodes evolve and change, but they recognize that privacy is an important issue to consider and that more work is needed to solve it. CSI [28] is a social-based routing protocol that models the spatio-temporal behaviours of the nodes using *behavioral profiles*, and forwards one-to-many messages through the nodes that are more similar to the destinations. Besides, the authors realize the importance of the privacy of the nodes and present a privacy-preserving mode of operation. This way the protocol can operate in scenarios where nodes are not willing to send its behavioural profiles to other nodes when needed.

To our knowledge, CSI is the only one proposal that takes into account the privacy of the nodes. Unfortunately, in all other cases, social-based routing protocols expect nodes to broadcast their information about the locations they visit or the details about their interests to the neighbours.

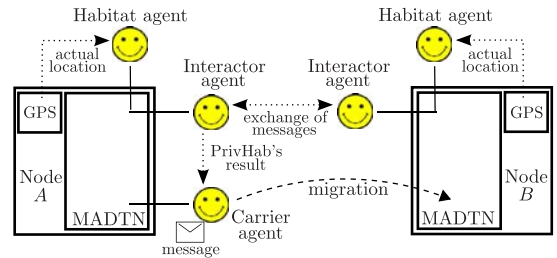
#### 4. A multiagent system

In this section, we first justify the decision of using Mobile agents to solve a network problem. Then, we describe the multiagent system needed to execute PrivHab. Finally, we list and define the different agents and entities involved.

##### 4.1. Usage of Mobile agents' technology

Due to the challenging characteristics of the scenario, to deploy a DTN it is not enough to achieve a fast and reliable podcast distribution. There are long distances between the senders and the receivers of the messages, so each one has to be carried by several nodes to reach its destination. Besides, most of the nodes near the source are likely to never meet with the nodes near the destination, making very difficult to obtain information about how to reach them. MADTN, using Mobile agents, brings us a set of characteristics that PrivHab could benefit in order to deal with these challenges.

A Mobile agent is a software entity that it is autonomous, intelligent, mobile, proactive, and represents a third part. To our consideration, all of these characteristics are beneficial to PrivHab. Agents need autonomy because they have to find their way to its destination in a changing and partially unknown environment; agents also need to be intelligent enough to make decisions that lead them towards their goal; mobility is capital because agents cannot control nodes' movement, so they need to migrate when finding a more useful one; proactivity allows agents to not only react to changes, but also to initiate context-aware actions (e.g. to start the delivery phase when the agent is near the destination); and representativity



**Fig. 2.** Schema of the multiagent system. Dotted lines depict the main interactions between entities, while slashed lines depict the movement of the agents. The Habitat agent updates the habitat using information from the GPS receiver. The Interactor agent exchanges PrivHab's messages with the other nodes and informs the Carrier agent of the result of the execution. The Carrier agent carries the message and makes the decision of migrating, staying or being cloned.

is the characteristic that allows applications with different needs to use the same network in a different way, with the agents making decisions on their behalf.

##### 4.2. Entities involved

PrivHab's goal is to improve the routing of the MADTN agents that carry the messages. The agents involved in this multiagent system are listed and explained below.

- **Habitat agent:** This agent calculates and periodically updates the habitat of the node (more details in Section 5). This agent also informs the Carrier agent of the current location, this way the Carrier agent can track if the node had approached enough the current waypoint and has to start considering the next one.
- **Interactor agent:** Every time a node meets a neighbour, this agent performs the PrivHab's exchange of messages to compare the habitats of the two nodes and decide who is the best choice to carry the message (more details in Section 6). When the exchange of messages has finished, this agent informs the Carrier agent of the result obtained, whether the neighbour is considered a worse or a better choice to carry the data.
- **Carrier agent:** This agent carries the message, and his goal is to deliver it to its destination. In order to achieve this, the Carrier agent moves through the network and makes decisions concerning the best way to reach a location. It uses the result of PrivHab's execution, along with other contextual information, to make a routing decision. The three decisions that the Carrier agent can make are: a) staying at the current node and waiting for other neighbours; b) migrating to the neighbour; and c) being cloned, so one agent remains at the node and the other one migrates to the neighbour.

Fig. 2 depicts the agents and entities that form the system. Apart from the three agents, there are two more concepts that need to be defined here: 1) the message contains the data (e.g. the podcast) that a node has sent to a receiver, it also contains the identifier of both the sender and the receiver, and a list of locations (waypoints) that the message has to pass by in order to reach its destination; and 2) the node is a location-aware mobile device (e.g. a Raspberry Pi or a smartphone), usually carried by a person or placed in a vehicle or in a certain strategic location.

#### 5. A habitat-based routing

In this section, we present the cornerstone of our novel georouting protocol: the habitat of a node. We define the concept and show how we model it using a circle, how it is automatically



**Fig. 3.** Heatmap of a node that spends much time in the south of the state of California. The dark red area corresponds to the area that is usually visited, and the intense yellow spot corresponds to the region where the node spends most of his time. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

calculated and the parameters involved in the calculations. Then, we explain the characteristics of the circular model. Finally, we provide some examples of automatically calculated habitats.

### 5.1. An approach towards the heatmap

In the described scenario, each node is a small device that may be carried by a person, placed in any vehicle or located in a static known place. Therefore, the movements of every node will be strongly related to their carrier. A static node will obviously remain immobile. A node carried by a person will probably spend much time in the vicinity of the carrier's home or workplace. A node placed in a vehicle will often pass by the same points if it is a regular-itinerary vehicle like a bus, or it will be inside a particular area if it is a taxi or similar. In any case, to know the places where a node has been in the past is useful to infer if a node will visit these places again in the future<sup>4</sup>.

To have a heatmap of a node and its neighbours to route an agent would be ideal. For example, a Carrier agent would want to migrate to a node with a heatmap like the one shown in Fig. 3 if it is carrying a message destined to the south of California, but would not if the message is destined to Utah or Wyoming. The heatmap is an extremely accurate, perhaps the most accurate, habitat (the area where someone is more likely to be found) representation. However, creating and maintaining this data is a resource consuming task that does not fit well with the small devices of the presented network.

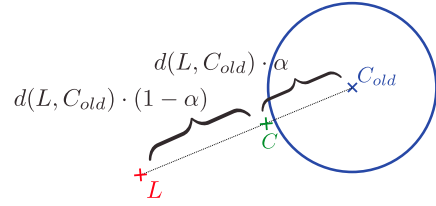
Therefore, we propose to model each nodes' habitat using the simplest geometric shape: the circle. This way, nodes can automatically calculate and store their habitat consuming the minimum computational resources by using a mobile average, and they can use it to make routing decisions quickly.

### 5.2. Definition of the habitat

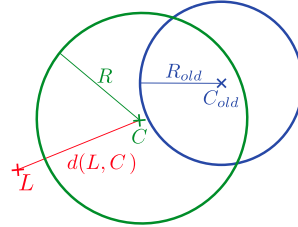
We model each habitat using a circle. Each habitat  $H$  is characterized by two elements: a centre point and a radius. From now on, we will refer as  $C = (x, y)$  to the centre point of the current habitat, and we will use  $R$  to denote their radius. A habitat is defined by the tuple  $H = (C, R)$ .

### 5.3. Calculation of the habitat

The Habitat agent updates the node's habitat in order to capture the trend of the node's mobility pattern. The update process of a habitat consists in obtaining the location of a node and adding it to



**Fig. 4.** The new centre point  $C$  is calculated averaging the old centre  $C_{old}$  and the new location  $L$ . Note that the centre point  $C$  has moved towards  $L$  using an  $\alpha$  factor.



**Fig. 5.** The old radius  $R_{old}$  is used together with the distance  $d(L, C)$  that separates the new location  $L$  and the centre point  $C$  to calculate the radius  $R$  of the habitat.

his habitat's model. Nodes use the Exponentially Weighted Moving Average (EWMA) to update their previous version of the habitat, named  $H_{old}$ , with a frequency of  $\omega$  updates/hour. The Global Positioning System (GPS) can be used to obtain their location, from now on, we will refer as  $L = (x_s, y_s)$  to the location of a node at the moment of the update. We assume that every geographic coordinate (a pair latitude - longitude) can be mapped<sup>5</sup> to cartesian coordinates and that this mapping is known by all the nodes of the network.

#### Step zero. Initialization of the habitat

At the initialization step,  $H_0$  is initialized with the centre point at the same coordinates of the location  $L_0$  (node's location when the calculation starts) and  $R = 0$ .

#### First step. Update of the centre

The first step to update a habitat is to update the centre. The centre point of the current habitat  $H$  is calculated by averaging using EWMA the centre point  $C_{old}$  and the current location  $L$ . The only parameter involved is  $\alpha$  (more details about  $\alpha$  can be found in Section 5.4). This first step is depicted in Fig. 4, where  $C$  is calculated averaging  $C_{old}$  and  $L$  using EWMA.

$$C = L * \alpha + C_{old} * (1 - \alpha) \quad (1)$$

#### Second step. Update of the radius

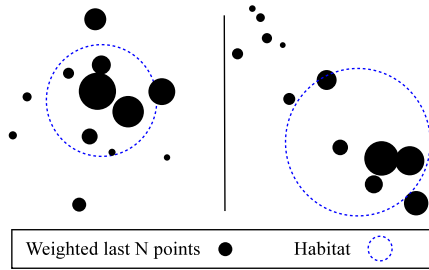
After  $C$  has been calculated, the radius  $R$  is updated by averaging using EWMA the radius  $R_{old}$  of the previous habitat and  $d(L, C)$ , the distance between  $L$  and the centre point  $C$ . This second step is depicted in Fig. 5.

$$R = d(L, C) * \alpha + R_{old} * (1 - \alpha) \quad (2)$$

As  $d(L, C)$  is the radius of a hypothetical circle with centre point  $C$  that contains  $L$ . Then, it will be greater than  $R_{old}$  if  $L$  is outside the circle with centre point  $C$  and radius  $R_{old}$  and it will be smaller than  $R_{old}$  if  $L$  is contained inside this circle. Therefore, the radius  $R$  of the current habitat, which is calculated using  $R_{old}$  and  $D$ , will increase if  $L$  is out of  $H$  and will decrease if  $L$  is contained by  $H$ .

<sup>4</sup> The similarity of the movements patterns of a node to its future movements is above 0.8 for two days, and 0.75 for a week, and remains 0.6 for five weeks [28].

<sup>5</sup> Any cartographic projection can be used.



**Fig. 6.** Two examples of habitats (dotted circles) calculated with  $T\omega = 12$  ( $\alpha = 0.1538$ ), black bubbles depict the last 12 locations, sized according to their relative EWMA weights.

#### 5.4. Characteristics and examples of habitats

A habitat calculated using  $\alpha = \frac{2}{T\omega+1}$  models the mobility habits of a node during the last  $T$  hours. The amount of hours  $T$  a habitat models is called the habitat's time span, it is the span of time modelled by the habitat, and it has to be known and shared by all nodes of the network. In a mobile average, each time a location is used to update the habitat, previous locations lose weight. Concretely, in EWMA, the last  $T\omega$  locations weight the 86% of the total, while previous locations weight the remaining 14%.

Fig. 6 shows two examples<sup>6</sup> of habitats and the locations used to update them. The bubbles representing locations are sized according to their relative weights, so the bigger they are, the more recent they are. Note that the circular habitat model is not designed to contain all the sampled locations. Its purpose is to achieve a compromise between containing all, giving more importance to the last ones, and considering the trend (the more recently sampled locations are more important than the older ones) of the node's movements.

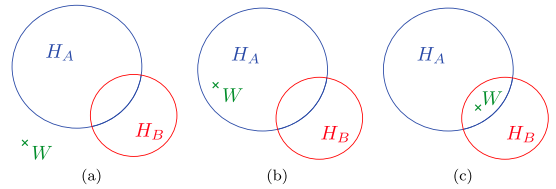
### 6. The PrivHab protocol

In this section, we first describe the PrivHab routing algorithm and its previous assumptions. Then, we introduce some important background concepts that are crucial for PrivHab to protect nodes' privacy. We explain how to use homomorphic encryption to solve two geometric problems: point inclusion and distance between a circle and a point. Following, the details about every message that has to be exchanged by the Interactor agents during the execution of PrivHab are presented. Finally, we provide some discussion about the *secure* nature of the protocol and the privacy of the participants.

#### 6.1. Previous assumptions

PrivHab is designed to operate in scenarios where the approximate locations the message has to pass to reach the destination can be known or guessed by the sender. They may be known beforehand, may be inferred from the knowledge about the terrain, may be discovered via the usage of a distributed secure position service like [29], or via the usage of an alternate communication channel.

This assumption is hard to accomplish in scenarios where the distances and latencies are small, because the nodes can move through all the scenario and it is hard to predict where a node



**Fig. 7.** Three possible situations when comparing two habitats to select the best choice: (a) The next waypoint is located outside the two habitats; (b) Only one of the habitats encloses the location of the next waypoint; (c) The two habitats enclose the location of the next waypoint.

will be in the next few moments. However, it is reasonable in big scale scenarios like the one presented in Section 2 (105 Km<sup>2</sup>), where the distances to travel and the latencies are big, because the movement of the nodes will usually be confined in one concrete part of the scenario, with only few and short occasional trips out of their usual surroundings. If the scenario has these features, it should be easy for the users to know some things like where are the bridges to cross a certain river, what mountainous terrain has to be avoided or what valley leads to the desired location. This is the knowledge needed to set the waypoints. These waypoints travel together with the message<sup>7</sup>.

The reader should note that, even if it is impossible for the sender to set the waypoints, the message can be sent using an approximate destination's location as the only waypoint, and PrivHab will try to route the message directly towards it.

#### 6.2. The routing algorithm

Given the definition of habitat, we assume that nodes spent most of the time inside the area defined by their habitats. For this reason, when two nodes' habitats do not enclose the next waypoint  $W$ , the node with the closest habitat is expected to bring the message nearer the waypoint than the other one. On the same line, when both habitats enclose  $W$ , the node with the smallest habitat is expected to remain closer, and to be more likely to pass by the waypoint.

The routing algorithm uses this reasoning to compare two nodes and to decide who is the best choice to carry the message towards its destination. The algorithm chooses the nodes whose habitat's enclose the destination, prioritizing those nodes whose habitat is the smallest. If a waypoint is contained outside two habitats, then the algorithm chooses the node whose border is the closest to the next waypoint. Fig. 7 show the different situations that can be faced. In (a) and (b) node  $A$  is chosen as the best option, because the waypoint  $W$  is closer to  $H_A$  or inside it. In (c) the best choice is  $B$ , because both habitats contain  $W$ , but  $H_B$  is smaller than  $H_A$ .

#### 6.3. Nodes' privacy

At [27,28], the authors recognize that privacy is an important issue in a routing protocol. In PrivHab, the habitat is used by the Carrier agent to select the next node of its itinerary, the best node to carry the message towards its destination. However, it can not be made public, since this will hurt the privacy of nodes. For this reason, nodes need PrivHab to be secure and do not reveal information about their habitats. On the other hand, waypoints are routing information that has to be known by the nodes that take

<sup>6</sup> The examples have been obtained directly from simulations, and the snapshots have been post-processed for the sake of the readability and the clarity of the figures.

<sup>7</sup> Note that it is much easier to know the approximate physical path that the message has to travel to reach its destination, than to know what nodes have to carry it through this path.



custody of the message. Moreover, although they are not a private information, they must remain hidden to the nodes that do not need this information. Besides, the presented protocol is fully compatible<sup>8</sup> with pseudonym generator mechanisms as [30] that generate pseudonyms of the nodes using its public key, or [31] that uses a secret shared between the nodes and hashing functions. These mechanisms can be used in scenarios where the destination does not want the forwarders of the messages to associate its identity with a set of waypoints.

PrivHab uses techniques of secure multi-party computations to protect nodes' privacy. This way, the habitats and the waypoints are operated and compared while cryptographically protected in order to avoid revealing this private information to the other parts.

#### 6.4. Background: homomorphic encryption

PrivHab requires the cryptosystem used to have a concrete property: to be additive homomorphic. An additive homomorphic cryptosystem is one in which, given two encrypted operands  $E(a)$  and  $E(b)$ ,  $E(a + b)$  can be computed without separately decrypting each one.

The cryptosystem used by PrivHab is the Paillier [32]. In a communication between Alice and Bob, Alice selects two random primes  $p$  and  $q$  and computes  $n = pq$ ; plaintext messages are elements of  $\mathbb{Z}_n$ ; however, ciphertext messages are elements of  $\mathbb{Z}_{n^2}$ . Then Alice picks a random  $g \in \mathbb{Z}_{n^2}^*$  such that  $\gcd((L(g^\lambda \bmod n^2)), n) = 1$ , where  $\lambda = \text{lcm}(p-1, q-1)$  and  $L(x) = (x-1)/n$ . Alice's public key<sup>9</sup> is  $pk_A: (n, g)$  and her private key is  $pk_A: (\lambda, p, q)$ .

To encrypt a message  $m$ , Bob picks a random  $r \in \mathbb{Z}_n^*$  and computes  $c = E(m) = g^m \cdot r^n \bmod n^2$ , the ciphertext of  $m$ . Then, Bob can easily compute  $E(a+b) = E(a) \cdot E(b) \bmod n^2 = g^{a+b} \cdot (r_1 \cdot r_2)^n \bmod n^2$  and  $E(a \cdot s) = E(a)^s \bmod n^2 = g^{a \cdot s} \cdot (r_1^s)^n \bmod n^2$ .

Finally, to decrypt a ciphertext  $c$ , Alice computes  $D(c) = L(c^\lambda \bmod n^2) = m$ .

#### 6.5. Background: point inclusion

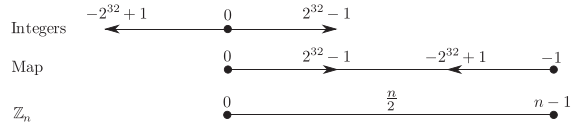
A point  $P: (x_p, y_p)$  is contained inside a circular habitat with centre  $C: (x_c, y_c)$  and radius  $R$  if and only if the distance  $\sqrt{(x_c - x_p)^2 + (y_c - y_p)^2}$  between  $C$  and  $P$  is lesser than  $R$ . Equivalently, we can check the sign of  $d = R^2 - ((x_c - x_p)^2 + (y_c - y_p)^2)$ ,  $P$  is contained inside the circle if  $d > 0$ . This way PrivHab can know if a waypoint is contained inside the habitat using only operations allowed by the Paillier cryptosystem.

#### 6.6. Background: distance between a circle and a point

The distance between a point  $P: (x_p, y_p)$  and a habitat  $H$  with centre  $C: (x_c, y_c)$  and radius  $R$  is  $d(H, P) = \sqrt{(x_c - x_p)^2 + (y_c - y_p)^2} - R$ . Equivalently, we can compute  $X: (a, b)$ , the nearest point of  $H$  to  $P$ , with  $a = x_c - R \cdot \cos \beta$  and  $b = y_c - R \cdot \sin \beta$  being  $\beta = \tan^{-1}(\frac{y_c - y_p}{x_c - x_p})$  the angle between the  $x$  axle and the segment joining  $P$  and  $C$ . Then, we calculate  $d(H, P) = d(X, P) = \sqrt{(a - x_p)^2 + (b - y_p)^2}$ . This way PrivHab can compare one node's distance with another's using only operations allowed by the

<sup>8</sup> A tuple with three values greater or equal than 0, sent in the third step of the protocol, does not reveals if the data has to be sent to  $B$  because it is a better carrier than  $A$  or because  $B$  is the destination.

<sup>9</sup> If Bob does not trust Alice when she generates her Paillier modulus, he can ask to prove it is the product of exactly two nearly equal primes [33].



**Fig. 8.** Positive integers are mapped to  $\mathbb{Z}_n$  using the identity function. Negative integers are mapped to the higher part of  $\mathbb{Z}_n$  using its representation modulo  $n$ . Positives and negatives are separated in  $\mathbb{Z}_n$  by  $n/2$ .

Paillier cryptosystem<sup>10</sup>: by checking the sign of  $d = d_1(X_1, P_1)^2 - d_2(X_2, P_2)^2$ .

#### 6.7. Background: mapping negatives

In order to calculate both the point inclusion and the distance between a circle and a point, PrivHab requires subtraction between encrypted values. To allow us to work with Paillier operations over encrypted data, we substitute the subtraction by the addition of a negative value. However, as there are no negative values in  $\mathbb{Z}_n$ , we map them in a way that they could still be added to other cyphered operands or multiplied by a plain operand.

We map positive integers lower than  $n/2$  using the identity function and negative integers greater than  $-n/2$  with its representation modulo  $n$ , as shown in Eq. (3).

$$\text{Map}(x) = \begin{cases} x & x \in [0, n/2) \\ x + n & x \in (-n/2, 0) \end{cases} \quad (3)$$

This way, we use Paillier addition between a positive integer  $a$  and a negative integer  $-b$  (mapped as  $-b + n$ ) to obtain  $(a - b) + n \bmod n$ . Note that if  $a > b$  then  $(a - b) + n \bmod n = (a - b)$ , and that if  $a < b$  then  $(a - b) + n \bmod n = (a - b) + n$ . The same way, we can use the Paillier multiplication between a negative integer  $-b$  (mapped as  $-b + n$ ) and a plain operand  $s$  to obtain  $(-b + n) \cdot s \bmod n = -b \cdot s + n \cdot s \bmod n = -b \cdot s + n$ . Then, the result of the operation can be recovered using the inverse mapping function shown in Eq. (4).

$$\text{Inverse Map}(x) = \begin{cases} x & x \in [0, n/2) \\ x - n & x \in (n/2, n - 1] \end{cases} \quad (4)$$

In order to use this mapping, we have to ensure that the operations used in our system never exceed the boundary of  $n/2$ , which means that encrypted computation results should never be a positive integer higher than  $n/2$  nor a negative number lower than  $-n/2$ . For this reason, since PrivHab works with 32 bit GPS precision coordinates<sup>11</sup>, the minimum key length ( $n$  value) allowed in PrivHab is 128 bits, since 32 bits are for positive integers, other 32 bits are for the results of multiplications between positive integers, 32 bits more allow the results of multiplications of a negative and a positive integer, and 32 bits more are accounted for negative integers. Finally, Fig. 8 provides a scheme of this mapping.

#### 6.8. Exchanged messages

We assume that every location can be mapped to two-dimensional coordinates with a mapping known to both  $A$ , the node that carries the data, and  $B$ , a candidate neighbour. Let  $A$ 's habitat be  $H_A: (C_A, R_A)$ . Let  $W[i]: (x_{W[i]}, y_{W[i]})$  be the next waypoint

<sup>10</sup> Note that  $d(H, P)^2 = (\sqrt{(x_c - x_p)^2 + (y_c - y_p)^2} - R)^2$  cannot be computed without computing first the square root. While  $d(X, P)^2 = (a - x_p)^2 + (b - y_p)^2$  can be computed without computing any square root.

<sup>11</sup> Note that latitude-longitude pairs have first to be converted into  $(x, y)$  coordinates using any cartographic projection, then these coordinates have to be converted into integers to operate with them. Finally, if needed, the resulting distances or radius must be mapped into negatives to allow subtractions.

where the data has to be carried to. Let  $B$ 's habitat be  $H_B: (C_B, R_B)$ . We denote  $E_Y(m)$  as the Paillier additive homomorphic encryption of  $m$  using  $Y$ 's public key. We denote a message sent by  $A$  to  $B$  with  $A \rightarrow B$ : message.

The PrivHAB protocol, described below, requires the Interactor agents of the two nodes to exchange three messages.

1. Node  $A$  calculates  $d_A = d(H_A, W[i])^2$ , the square of the distance between its habitat and  $W[i]$ ;  $d_A = 0$  if  $W[i] \in H_A$  and  $d_A \geq 1$  otherwise.  $A$  knows both  $H_A$  and  $W[i]$ , so the calculation of  $d_A$  is very easy and can be performed quickly, without using homomorphic encryption.
2. Node  $B$  announces<sup>12</sup> to  $A$  the centre  $C_B: (x_{C_B}, y_{C_B})$  of its habitat.  $B \rightarrow A$ :  $E_B(x_{C_B}), E_B(y_{C_B})$
3. Node  $A$ , using Eqs. (5) and (6), subtracts the coordinates of  $W[i]$  to the coordinates of  $C$ . Then,  $A$  multiplies both results by the same *nonce* (a random one-use value).

$$E_B((x_{C_B} + (-x_{W[i]})) \cdot \text{nonce}) = (E_B(x_{C_B}) \cdot E_B(-x_{W[i]}))^{\text{nonce}} \quad (5)$$

$$E_B((y_{C_B} + (-y_{W[i]})) \cdot \text{nonce}) = (E_B(y_{C_B}) \cdot E_B(-y_{W[i]}))^{\text{nonce}} \quad (6)$$

Following,  $A$  sends to  $B$  the results and the coordinates of  $W[i]$ , the distance  $d_A$  and the radius  $R_A$ .

$$A \rightarrow B: \begin{aligned} &E_B((x_{C_B} + (-x_{W[i]})) \cdot \text{nonce}), E_A(-x_{W[i]}^2), \\ &E_B((y_{C_B} + (-y_{W[i]})) \cdot \text{nonce}), E_A(-y_{W[i]}^2), \\ &E_A(-R_A), E_A(-d_A), E_A(-2x_{W[i]}), E_A(-2y_{W[i]}), \\ &E_A(-x_{W[i]}), E_A(-y_{W[i]}), E_A(-2y_{W[i]}), \end{aligned}$$

4.  $B$  decrypts the received subtractions and uses the decrypted values to compute  $\beta$  using Eq. (7).

$$\beta = \tan^{-1} \left( \frac{(y_{C_B} + (-y_{W[i]})) \cdot \text{nonce}}{(x_{C_B} + (-x_{W[i]})) \cdot \text{nonce}} \right) \quad (7)$$

Node  $B$  uses  $\beta$  to calculate  $X$ , the nearest point of  $H_B$  to  $W[i]$ ,  $X: (a = x_{C_B} - R_B \cdot \cos \beta, b = y_{C_B} - R_B \cdot \sin \beta)$ . Then,  $B$  calculates the square of the distance  $d(H_B, W[i])^2 = d(X, W[i])^2 = d_B$  using Eq. (8).

$$E_A(d_B) = E_A((a - x_{W[i]})^2 + (b - y_{W[i]})^2) =$$

$$E_A(a^2 - 2ax_{W[i]} - x_{W[i]}^2 + b^2 - 2by_{W[i]} - y_{W[i]}^2) =$$

$$\begin{aligned} &E_A(a^2) \cdot E_A(-2x_{W[i]}a) \cdot E_A(-x_{W[i]}^2) \cdot E_A(b^2) \\ &\cdot E_A(-2y_{W[i]}b) \cdot E_A(-y_{W[i]}^2) = \end{aligned} \quad (8)$$

Following,  $B$  calculates the point inclusion of  $W[i]$  in  $H_B$  using Eq. (9), the comparison of distances using Eq. (10), and the comparison of radius using Eq. (11). This time, three different *nonce* values are used to randomize the results. The  $d_A$  factor is used to blur<sup>13</sup> the point inclusion test and the comparison of radius.

$$E_A((R_B^2 + d_B + (-d_A)) \cdot \text{nonce}) = (E_A(R_B^2) \cdot E_A(d_B) \cdot E_A(-d_A))^{\text{nonce}} \quad (9)$$

$$E_A((d_B + (-d_A)) \cdot \text{nonce}) = (E_A(d_B) \cdot E_A(-d_A))^{\text{nonce}} \quad (10)$$

<sup>12</sup> This announcement can be made by adding this information to the messages exchanged during the neighbour discovery process.

<sup>13</sup> If  $d_A > d_B$ , then the best choice is  $B$ , and the result of the point inclusion test and the comparison of radius are not needed.

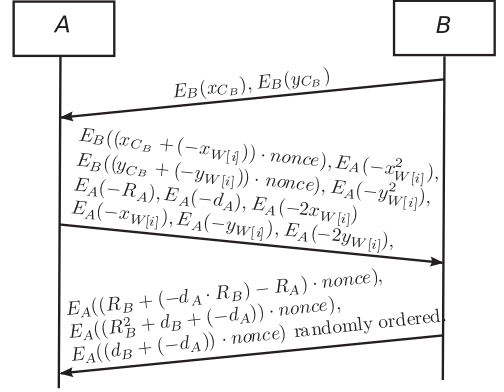


Fig. 9. Sequence of the messages exchanged by the Interactor agents during the execution of PrivHAB.

$$E_A((R_B + (-d_A \cdot R_B) - R_A) \cdot \text{nonce}) =$$

$$(E_A(R_B) \cdot E_A(-d_A)^{R_B} \cdot E_A(-R_A))^{\text{nonce}} \quad (11)$$

Finally,  $B$  orders the results of the two comparisons and the point inclusion test in a random way and sends it to  $A$ .

$$B \rightarrow A: \begin{aligned} &E_A((R_B + (-d_A \cdot R_B) - R_A) \cdot \text{nonce}), \\ &E_A((R_B^2 + d_B + (-d_A)) \cdot \text{nonce}), \\ &E_A((d_B + (-d_A)) \cdot \text{nonce}) \text{ randomly ordered.} \end{aligned}$$

5. Node  $A$  decrypts the three received values.  $B$  is considered a better choice if and only if the three decrypted values are negative or 0.

Fig. 9 depicts the exchange of messages.

## 6.9. Security evaluation

In secure multi-party computations [34], a protocol is considered secure if it reveals only the result of the function and the inferences that can be deduced from this output with one or more input values. The presented protocol has been designed following these principles. On one hand, node  $A$  only knows if  $H_B$  is better or worse than  $H_A$ . Then,  $A$  can use this knowledge to infer about the relation between  $d_A$  and  $d_B$ , the relation between  $R_A$  and  $R_B$ , or to deduce if  $W[i] \in H_B$ . On the other hand, node  $B$  cannot even know the result of the execution, so it cannot learn anything about  $H_A$ . Maintaining  $W[i]$  hidden to  $B$  (only  $\beta$  is revealed) when the data is not forwarded is crucial to avoid that  $B$  can calculate  $d_B$  and use it to infer information about  $H_A$ .

Anything learned by  $A$  about  $H_B$  is also learnable from the result alone. Moreover, when the Carrier agent migrates to  $B$  the waypoints are revealed to it, because waypoints will be needed in next steps of the routing. Otherwise, the only thing  $B$  learns about  $W[i]$  is the angle<sup>14</sup>  $\beta$  where it is located in relation with  $H_B$ .

On the other hand, an active attacker can try to learn things about the other part's habitat by producing chosen-destination arbitrary messages and repeatedly executing PrivHAB. In any case,

<sup>14</sup> The angle  $\beta$  is a less accurate information than the coordinates of  $W[i]$  or the distance between  $W[i]$  and  $H_B$ . Moreover,  $B$  does not even know who is the destination, and the protocol will not be executed again between the same participants. Therefore,  $B$  can not relate  $W[i]$  with any node neither triangulate its location.

the information obtained by the attacker is the same information that he can infer from a truthful execution of the protocol. As  $A$  is the node that starts the transaction and the only one that knows the number of messages he carries, he can determine how many times to execute PrivHab+. If  $A$  executes PrivHab enough times, he can try to uncover the area covered by  $H_B$ . Given that nodes always operate with encrypted data, there is no way for one part to tell apart a truthful execution of PrivHab from an untruthful one. However,  $B$  can decrease the effectiveness of these attacks by limiting the amount of interactions per unit of time with every other node and forcing  $A$  to send him at once the information needed to perform all the executions before sending any response. Besides, the information protected by PrivHab, the habitat, changes periodically. For this reason, slowing enough an attack is equivalent to avoiding it, because when time passes the habitats change and the first things learned by the attacker become obsolete.

## 7. Experiments and results

In this section, we study the computational and communication overhead introduced by PrivHab. Then, we explain the scenario we have chosen to evaluate PrivHab's and other well known DTN routing protocols, and how we have modelled and simulated it. Finally, we provide the obtained results, and we compare PrivHab with a set of popular DTN routing algorithms.

### 7.1. Physical implementation

As a proof-of-concept we have deployed an implementation of the presented protocol on three Raspberry Pi boards<sup>15</sup>. These are very cheap low-end devices that fit very well with the characteristics of the proposed application, and they are ideals to deploy a prototype network that will allow us to run field experiments in the near future. We have used them to measure the overhead that PrivHab adds to every transaction.

We have used our proof-of-concept implementation, using Pailier's length keys of 512, 1024 and 2048 bits, to forward 600 podcasts of sizes between 10MB and 20MB<sup>16</sup>. We have repeated the tests twenty times. We have measured the average time needed by the Interactor agent to make the calculations and to exchange all the messages. The obtained results are shown in Table 1 and have been incorporated to the simulations.

As can be seen in Table 1, PrivHab execution time depends heavily on the key length used. When using keys of 512 bits, PrivHab can be executed by a low-end device in less than half a second. Meaning an overhead of less than 3% when sending messages larger than 10MB. The execution time increases to 2.5 s when using keys of 1024 bits. Given the average length of connectivity windows in remote village scenarios presented in [35], this overhead is acceptable. When using keys of 2048 bits, the execution time is high. The key length should be chosen keeping in mind the duration of the connectivity windows and the security requirements of the scenario. In the presented application, the overhead of 2.5 s using a 1024 bits key is efficient and secure enough<sup>17</sup>.

**Table 1**

Average execution time of PrivHab using different key lengths. The overhead is the extra amount of time needed to send a message of 10MB or 20MB.

Key length	Time (ms)	Overhead 10MB (%)	Overhead 20MB (%)
512 bits	401.94 ± 0.5	2.44	1.22
1024 bits	2, 585.05 ± 23.1	15.69	7.84
2048 bits	15, 018.9 ± 38.8	91.13	45.57

**Table 2**

Parameters used at the simulations.

Parameter	Value
Total nodes	95
Source nodes	1 static
Destination nodes	2 static
Other nodes	92 mobile
Message size	10 – 20 MB
Buffer size	200 MB
Scenario size	15 × 7 Km
Simulated time	2.5 weeks
PrivHab's overhead	2.5 s
Habitat update frequency ( $\omega$ )	2 updates / h
Message generation ratio (messages / hour)	<b>High:</b> 0.5 – 1 <b>Medium:</b> 0.25 – 0.5 <b>Low:</b> 0.125 – 0.25

### 7.2. Modelling and simulations

The scenario we have used in all the simulations is the one presented in Section 2. Nodes implement a mobility pattern that takes into account their *hotspots* [36] (home's and work's location). Agents carrying podcasts are injected in the network by the NGO office, who knows the exact location and the necessary waypoints to reach every destination. Nodes use PrivHab to make routing decisions. Carrier agents always chose to migrate to nodes that are considered better choices by the PrivHab algorithm. Table 2 provides the simulation parameters that have been used.

We have compared the performance of PrivHab with a benchmark of well-known DTN routing protocols used in [37]: Prophet [38], Binary Spray & Wait ( $L = 40$ ) [39], Epidemic [17] and Random [40]. We have added two routing protocols to this set: MaxProp [41] and First Contact<sup>18</sup>. Random and First Contact are traditionally considered to achieve the lower bound of single-copy routing performance. Prophet and MaxProp are representatives in contacts-based prediction routing algorithms, the most common type of routing in privacy preserving protocols. Finally, BS&W and Epidemic are representatives of flooding-based algorithms. All simulations have been performed using *The Opportunistic Network Simulator* (The ONE) [42], and have been repeated twenty times using different random seeds.

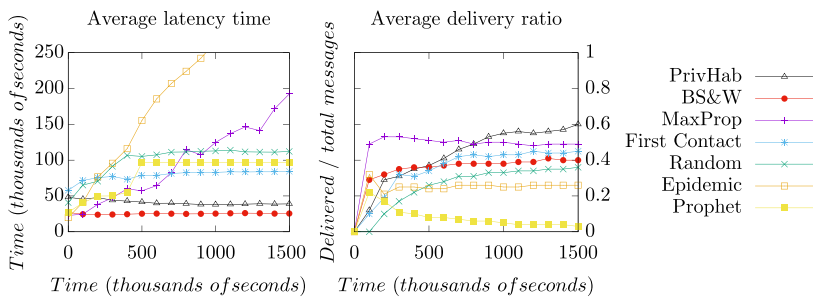
The performance of all the compared protocols in terms of delivery ratio and latency while using different message generation ratios is depicted in Fig. 10. Flooding-based protocols, as Epidemic and Prophet, fill the buffers early and perform badly with a high or medium message generation ratio. Therefore, they obtain high latencies and low delivery ratios because nodes are forced to drop podcasts. When the message generation ratio is low, their latencies improve, but as most of the opportunistic contacts end before nodes had been able to forward all the carried messages, their delivery ratio continues to be low because podcasts lose opportunities to advance through their destination. MaxProp performs better because of his dropping policy based on probabilities of deliv-

<sup>15</sup> Raspberry Pi Broadcom BCM2835 SoC full HD, 700MHz Low Power ARM1176JZ-F, 512MB SDRAM, 256MB SD with Raspbian, Wi-Pi Wireless Adapter (802.11n up to 150Mbps), GPS receiver NL-302U (baud rate: 4800 bauds) and a dual output 5000mAh battery.

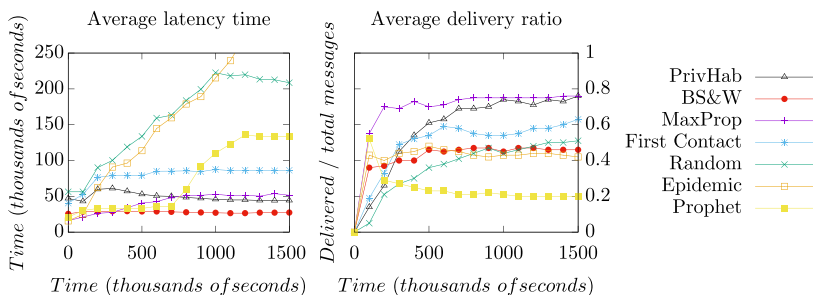
<sup>16</sup> This is the size of an audio file with ID3 version 2.4.0, extended header, containing: MPEG ADTS, layer III, v1, 128 kbps, 44.1 kHz, stereo, with a duration between 10 and 20 minutes.

<sup>17</sup> The effort needed to break the provided security is equivalent to the effort needed to factor a 1024 bits RSA key.

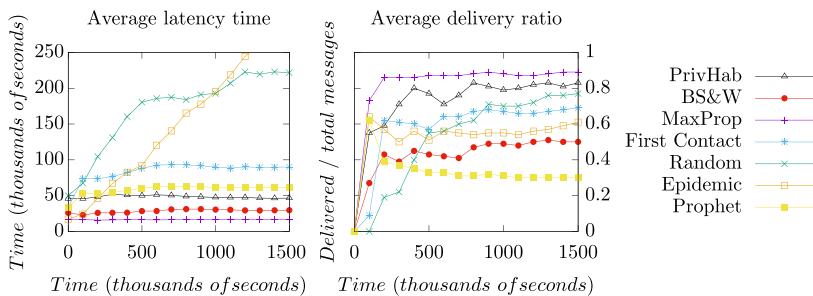
<sup>18</sup> When a neighbour is met, each podcast that has not been carried previously by the new neighbour is forwarded to him.



(a) Latency and delivery ratio obtained using a message generation ratio of: 0.5 – 1 messages/hour.



(b) Latency and delivery ratio obtained using a message generation ratio of: 0.25 – 0.5 messages/hour.



(c) Latency and delivery ratio obtained using a message generation ratio of: 0.125 – 0.25 messages/hour.

**Fig. 10.** Performance's comparative using different message generation ratio. MaxProp equals PrivHab's performance with a medium ratio, and outperforms it with a low one. PrivHab also benefits from a lower ratio to increase the amount of messages it delivers.

ery, and improves vastly as the message generation ratio decreases. This way, MaxProp performs badly in terms of latency with a high message generation ratio, but achieves a good performance with a medium one, and outperforms the other protocols with the low one. Binary Spray & Wait performs well both in terms of latency and delivery ratio in all cases, but does not improve much its performance when the message generation ratio changes because of his depth-style spread. Therefore, it is a good choice (it obtains the lowest latency) with a high message generation ratio, but a bad one with a medium or low ratio. Besides, even in its best case scenario, its delivery ratio is not as good as PrivHab's because in BS&W the spread is not directed towards the destination. Finally, First Contact, Random and specially PrivHab obtain a high delivery ratio in all cases because they do not face the problems related to the size of the buffers and the connectivity windows. However, the performance of these protocols in terms of latency depends on the quality of their decision-making protocol. Random is the worst because it is equally likely to make a bad or a good choice. First

Contact performs better because it forces podcasts to move away from their origin. These two protocols become worse by comparison as the message generation ratio decreases and the flooding protocols improve their results. Finally, PrivHab, that takes the best decisions because it takes into account both the pathway to the destination and the mobility patterns of the neighbours, obtains the best performance with a high message generation ratio, and performs slightly better, or worse, than MaxProp with a medium and low one.

Table 3 shows the average number of dropped messages and the network overhead, calculated as the relation between the number of the relays done and the number of delivered podcasts. Both low and high message generation ratio cases have been considered. Low network overhead is desirable because reducing relays saves battery and increases the amount of time nodes are operational. Epidemic, Prophet, MaxProp and Random generate an enormous overhead of several thousand percent when the message generation ratio is high. This means that almost all nodes effort

**Table 3**

Obtained results in terms of network overhead and number of dropped messages. PrivHab and First Contact waste fewer network resources.

Protocol	Dropped messages		Network overhead (%)	
	High	Low	High	Low
Epidemic	1,041,105.3	742,610.4	86,636.4	6,157.2
Prophet	628,897.4	329,756.3	89,705.5	35,357.7
Maxprop	206,372.7	8,145.9	7,682.2	162.1
BS&W	2,105.1	4,645.0	86.8	64.6
Random	86.9	7.3	40,582.5	2,557.6
First Contact	75.8	5.2	137.9	85.5
PrivHab	75.5	19.1	20.3	12.7

**Table 4**

Feature comparison of the protocols. Contacts-based routing algorithm tend to violate nodes privacy and to have at least a linear complexity.

Protocol	Type of routing	Nodes' privacy	Protocol's Complexity
PrivHab	Geographic	Preserved	Constant
MaxProp	Contacts-based	Violated	Linear
Prophet	Contacts-based	Violated	Linear
BS&W	Flooding	Not considered	Constant
Epidemic	Flooding	Not considered	Constant
First Contact	One-copy	Not considered	Constant
Random	One-copy	Not considered	Constant

while forwarding podcasts is wasted, either because the podcasts are dropped or because the majority of the relays are bad choices. However, MaxProp improves its results and obtains a lower network overhead when the message generation ratio is low. This means that MaxProp generates copies that fill the buffers and consume energy because multiplies the number of relays done, but it makes use of this effort to deliver the podcasts to their destination. BS&W has a small amount of dropped podcasts, in comparison with the other multi-copy protocols, and a low network overhead. BS&W tries to limit the amount of resources used and obtains the second lowest network overhead with a low message generation ratio, but its performance in terms of latency and delivery ratio is not as good as others'. First Contact and PrivHab have generated a small amount of dropped messages, but the lowest network overhead of PrivHab means that his routing decisions are much better. The small network overhead produced by PrivHab could even allow users to use the same devices to run other applications because the main application does not congests either the device or the network.

Following, Table 4 finishes the comparison, regarding the Cajamarca scenario. In addition to those metrics that had been studied in previous paragraphs, delivery ratio, latency and network overhead; we also take into consideration the type of routing used, the nodes' privacy and the protocol's complexity.

Nodes' privacy is preserved by PrivHab, which is the only one that uses private information in a secure manner. Privacy is obviously not considered by the protocols that do not use node-related information to make choices, but it is heavily violated by Prophet and MaxProp while nodes exchange their likelihood to contact others. However, their privacy preserving counterparts do not have this limitation, but they route the messages similarly, using a contacts-based prediction, so they perform similarly in this scenario. PrivHab, BS&W, Epidemic, First Contact and Random need a constant number of operations to make a routing decision. Contacts-based algorithms need to update and compare an amount of probabilities that grow linear with the number of nodes of the network. When operating in networks with lots of nodes, probabilistic protocols have to limit the amount of encounter probabili-

ties they store. This limitation decreases their performance because this reduces the value of their heuristics.

PrivHab delivers more messages to its destination. Besides, it does it faster than all other protocols except MaxProp with a low message generatio ratio, and it consumes fewer network resources to do so. Moreover, it preserves nodes' privacy and performs well in scenarios where the number of nodes is high and the destinations of the messages are hop-distant. Taking into account all these aspects, we can state that PrivHab is the protocol that suits better to any scenario with characteristics like the presented one.

## 8. Conclusions

The habitat models node's whereabouts during the habitat's time span. It is useful to compare nodes to decide who is a better choice to carry the data towards its destination. In this paper, we present PrivHab, a privacy preserving multiagent geographical routing protocol based on MADTN that uses the habitats to make decisions. PrivHab also makes use of homomorphic cryptography techniques to preserve nodes' privacy. We have presented a podcasts distribution application in rural areas lacking communication networks that could benefit from the characteristics and the performance of PrivHab.

PrivHab's characteristics make him ideal to operate not only in this concrete scenario of application, but also in any other DTN scenario with similar characteristics: scenarios where nodes mobility patterns are complex, but routinary, where lots of hops are needed to reach the destination of the messages from their source, and where nodes are so related, directly or indirectly, to a person that their privacy needs to be protected.

As future lines of research, we plan to study different behaviours for the Carrier agent, to improve the circular model of habitat using a more complex representation, and to develop an enhanced version of PrivHab that compares simultaneously three or more habitats. We also plan to study the performance of PrivHab in different scenarios based on real applications that could benefit from a geographic routing approach.

## Acknowledgment

This work has been partially funded by the Ministry of Economy and Competitivity of Spain, under the reference project TIN2014-55243-P and by the Catalan Government under the reference project 2014-SGR-691, and by the Autonomous University of Barcelona under the reference number 472-03-01/2012.

## References

- [1] A. Singh, *Bridging the Rural Digital Divide. Case Study: Institution Based Information Systems, India.*, Technical Report, Indian Agribusiness Systems Pvt. Ltd., 2005.
- [2] R. Martínez, S. Castillo, S. Robles, A. Sánchez, J. Borrell, M. Cordero, A. Viguria, N. Giuditta, Mobile-agent based delay-tolerant network architecture for non-critical aeronautical data communications, in: Springer (Ed.), *In 10th International Symposium on Distributed Computing and Artificial Intelligence*, 2013.
- [3] C. Borrego, S. Castillo, S. Robles, Striving for sensing: taming your mobile code to share a robot sensor network, *Inf. Sci. (O)* (2014). <http://dx.doi.org/10.1016/j.ins.2014.02.072>.
- [4] A.P. Silva, S. Burleigh, C.M. Hirata, K. Obraczka, A survey on congestion control for delay and disruption tolerant networks, *Ad Hoc Netw.* 25, Part B (2015) 4 804–94. New Research Challenges in Mobile, Opportunistic and Delay-Tolerant Networks Energy-Aware Data Centers: Architecture, Infrastructure, and Communication. <http://dx.doi.org/10.1016/j.adhoc.2014.07.032>.
- [5] E. Kuiper, S. Nadim-Tehrani, Geographical routing with location service in intermittently connected manets, *Vehic. Technol. IEEE Trans.* 60 (2) (2011) 592–604, doi:10.1109/TVT.2010.2091658.
- [6] J. Miao, O. Hasan, S.B. Mokhtar, L. Brunie, K. Yim, An investigation on the unwillingness of nodes to participate in mobile delay tolerant network routing, *Int. J. Inf. Manage.* 33 (2) (2013) 252–262. <http://dx.doi.org/10.1016/j.ijinfomgt.2012.11.001>.
- [7] A. Boukerche, A. Darehshoorzadeh, Opportunistic routing in wireless networks: Models, algorithms, and classifications, *ACM Comput. Surv.* 47 (2) (2014) 22:1–22:36, doi:10.1145/2635675.

- [8] E. Kuiper, S. Nadjm-Tehrani, Geographical routing with location service in intermittently connected manets, *Vehic. Technol. IEEE Trans.* 60 (2) (2011) 592–604, doi:[10.1109/TVT.2010.2091658](https://doi.org/10.1109/TVT.2010.2091658).
- [9] P.-C. Cheng, K. Lee, M. Gerla, J. Hrii, Geodtn+nav: geographic dtn routing with navigator prediction for urban vehicular environments, *Mob. Netw. Appl.* 15 (1) (2010) 61–82, doi:[10.1007/s11036-009-0181-6](https://doi.org/10.1007/s11036-009-0181-6).
- [10] X. Cai, Y. He, C. Zhao, L. Zhu, C. Li, Lsgo: link state aware geographic opportunistic routing protocol for vanets, *EURASIP J. Wireless Commun. Netw.* 2014 (1) (2014) 1–10, doi:[10.1186/1687-1499-2014-96](https://doi.org/10.1186/1687-1499-2014-96).
- [11] T.-Y. Wu, Y.-B. Wang, W.-T. Lee, Mixing greedy and predictive approaches to improve geographic routing for vanet, *Wireless Commun. Mob. Comput.* 12 (4) (2012) 367–378, doi:[10.1002/wcm.1033](https://doi.org/10.1002/wcm.1033).
- [12] C. Si-Ho, L. Keun-Wang, C. Hyun-Seob, Grid-based predictive geographical routing for inter-vehicle communication in urban areas, *International Journal of Distributed Sensor Networks* 2012 (2012), doi:[10.1155/2012/819497](https://doi.org/10.1155/2012/819497).
- [13] X. Lu, P. Hui, D. Towsley, J. Pu, Z. Xiong, Anti-localization anonymous routing for delay tolerant network, *Comput. Netw.* 54 (11) (2010) 1899–1910, <http://dx.doi.org/10.1016/j.comnet.2010.03.002>.
- [14] A. Kate, G. Zaverucha, U. Hengartner, Anonymity and security in delay tolerant networks, in: *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on, 2007*, pp. 504–513, doi:[10.1109/SECCOM.2007.4550373](https://doi.org/10.1109/SECCOM.2007.4550373).
- [15] R. Lu, X. Lin, X. Shen, Spring: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks, in: *INFOCOM, 2010 Proceedings IEEE, 2010*, pp. 1–9, doi:[10.1109/INFCOM.2010.5462161](https://doi.org/10.1109/INFCOM.2010.5462161).
- [16] X. Lin, X. Sun, P.H. Ho, X. Shen, Gsis: A secure and privacy-preserving protocol for vehicular communications, *IEEE Trans. Vehic. Technol.* 56 (6) (2007) 3442–3456, doi:[10.1109/JVT.2007.906878](https://doi.org/10.1109/JVT.2007.906878).
- [17] A. Vahdat, D. Becker, et al., *Epidemic routing for partially connected ad hoc networks*, Technical Report, Technical Report CS-200006, Duke University, 2000.
- [18] L. Zhang, J. Song, J. Pan, Towards privacy-preserving and secure opportunistic routings in vanets, in: *2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), 2014*, pp. 627–635, doi:[10.1109/SAHCN.2014.6990403](https://doi.org/10.1109/SAHCN.2014.6990403).
- [19] A.C. Yao, Protocols for secure computations, in: *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science, in: SFCS '82, IEEE Computer Society, Washington, DC, USA, 1982*, pp. 160–164, doi:[10.1109/SFCS.1982.88](https://doi.org/10.1109/SFCS.1982.88).
- [20] O. Hasan, J. Miao, S.B. Mokhtar, L. Brunie, A privacy preserving prediction-based routing protocol for mobile delay tolerant networks, in: *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on, 2013*, pp. 546–553, doi:[10.1109/AINA.2013.6](https://doi.org/10.1109/AINA.2013.6).
- [21] E. Papapetrou, V.F. Bourgos, A.G. Voyiatzis, Privacy-preserving routing in delay tolerant networks based on bloom filters, in: *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a, 2015*, pp. 1–9, doi:[10.1109/WoWMoM.2015.7158148](https://doi.org/10.1109/WoWMoM.2015.7158148).
- [22] I. Parris, G. Bigwood, T. Henderson, Privacy-enhanced social network routing in opportunistic networks, in: *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on, 2010*, pp. 624–629, doi:[10.1109/PERCOMW.2010.5470511](https://doi.org/10.1109/PERCOMW.2010.5470511).
- [23] K.E. Defrawy, G. Tsudik, Privacy-preserving location-based on-demand routing in manets, *IEEE J. Selected Areas Commun.* 29 (10) (2011) 1926–1934, doi:[10.1109/JSAC.2011.111203](https://doi.org/10.1109/JSAC.2011.111203).
- [24] P. Hui, J. Crowcroft, E. Yoneki, Bubble rap: Social-based forwarding in delay-tolerant networks, *Mob. Comput. IEEE Trans.* 10 (11) (2011) 1576–1589, doi:[10.1109/TMC.2010.246](https://doi.org/10.1109/TMC.2010.246).
- [25] J. Leguay, T. Friedman, V. Conan, Evaluating mobyspace-based routing strategies in delay-tolerant networks, *Wireless Commun. Mob. Comput.* 7 (10) (2007) 1171–1182, doi:[10.1002/wcm.520](https://doi.org/10.1002/wcm.520).
- [26] A. Mei, G. Morabito, P. Santi, J. Stefa, Social-aware stateless forwarding in pocket switched networks, in: *INFOCOM, 2011 Proceedings IEEE, 2011*, pp. 251–255, doi:[10.1109/INFCOM.2011.5935076](https://doi.org/10.1109/INFCOM.2011.5935076).
- [27] C. Boldrini, M. Conti, J. Jacopini, A. Passarella, Hibop: a history based routing protocol for opportunistic networks, in: *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a, 2007*, pp. 1–12, doi:[10.1109/WoWMoM.2007.4351716](https://doi.org/10.1109/WoWMoM.2007.4351716).
- [28] W. Hsu, D. Dutta, A. Helmy, CSI: A paradigm for behavior-oriented delivery services in mobile human networks, *CoRR* (2008). 0807.1153.
- [29] J.-H. Song, V.W. Wong, V.C. Leung, Secure position-based routing protocol for mobile ad hoc networks, *Ad Hoc Netw.* 5 (1) (2007) 7685. Security Issues in Sensor and Ad Hoc Networks <http://dx.doi.org/10.1016/j.adhoc.2006.05.010>.
- [30] R. Jiang, Y. Xing, Anonymous on-demand routing and secure checking of traffic forwarding for mobile ad hoc networks, in: *Reliable Distributed Systems (SRDS), 2012 IEEE 31st Symposium, 2012*, pp. 406–411, doi:[10.1109/SRDS.2012.6](https://doi.org/10.1109/SRDS.2012.6).
- [31] M. Mahmoud, X. Shen, Lightweight privacy-preserving routing and incentive protocol for hybrid ad hoc wireless network, in: *Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2011*, pp. 1006–1011, doi:[10.1109/INFCOMW.2011.5928774](https://doi.org/10.1109/INFCOMW.2011.5928774).
- [32] G. Zhong, I. Goldberg, U. Hengartner, Louis, lester and pierre: Three protocols for location privacy, in: N. Borisov, P. Golle (Eds.), *Privacy Enhancing Technologies, Lecture Notes in Computer Science*, vol. 4776, 2007, pp. 62–76, doi:[10.1007/978-3-540-75551-7\\_5](https://doi.org/10.1007/978-3-540-75551-7_5).
- [33] M. Liskov, R. Silverman, *A Statistical Limited-Knowledge Proof for Secure RSA Keys*, Technical Report, IEE P1363 working group, 1998.
- [34] O. Goldreich, *Secure multi-party computation*, 1998.
- [35] S. Grasic, A. Lindgren, Revisiting a remote village scenario and its dtn routing objective, *Comput. Commun.* 48 (2014) 133–140, doi:[10.1016/j.comcom.2014.04.003](https://doi.org/10.1016/j.comcom.2014.04.003).
- [36] H. Ma, D. Zhao, P. Yuan, Opportunities in mobile crowd sensing, *Commun. Mag. IEEE* 52 (8) (2014) 29–35, doi:[10.1109/MCOM.2014.6871666](https://doi.org/10.1109/MCOM.2014.6871666).
- [37] M. Musolesi, C. Mascolo, Car: Context-aware adaptive routing for delay-tolerant mobile networks, *Mob. Comput. IEEE Trans.* 8 (2) (2009) 246–260, doi:[10.1109/TMC.2008.107](https://doi.org/10.1109/TMC.2008.107).
- [38] A. Lindgren, A. Doria, O. Schelén, Probabilistic routing in intermittently connected networks, *SIGMOBILE Mob. Comput. Commun. Rev.* 7 (3) (2003) 19–20, doi:[10.1145/961268.961272](https://doi.org/10.1145/961268.961272).
- [39] T. Spyropoulos, K. Psounis, C. Raghavendra, Spray and wait: an efficient routing scheme for intermittently connected mobile networks, in: *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, ACM, 2005*, p. 259.
- [40] T. Spyropoulos, R.N. Rais, T. Turletti, K. Obraczka, A. Vasilakos, Routing for disruption tolerant networks: taxonomy and design, *Wireless Netw.* 16 (8) (2010) 2349–2370, doi:[10.1007/s11276-010-0276-9](https://doi.org/10.1007/s11276-010-0276-9).
- [41] J. Burgess, B. Gallagher, D. Jensen, B. Levine, Maxprop: routing for vehicle-based disruption-tolerant networks, in: *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, 2006*, pp. 1–11, doi:[10.1109/INFCOM.2006.228](https://doi.org/10.1109/INFCOM.2006.228).
- [42] A. Keränen, J. Ott, T. Kärkkäinen, *The ONE simulator for DTN protocol evaluation*, in: *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, ICST, New York, NY, USA, 2009*.



“Thus, though we have heard of stupid haste in war, cleverness has never been seen associated with long delays.”

*The Art of War*, SUN TZU

# 6

## PrivHab+: A secure geographic routing protocol for DTN

Next, we reproduce the following article, which has been published on the international peer-reviewed journal *Computer Communications*, a first quartile JCR journal with an impact factor of 2.099.

*A. Sanchez-Carmona, C. Borrego, S. Robles. PrivHab+: A secure geographic routing protocol for DTN. Journal of Computer Communications (February 2016) vol. 78, pp: 56-73. ISSN: 0140-3664. DOI: doi:10.1016/j.comcom.2015.10.002*

This work was submitted for the first time on 19-Nov-2014, it was revised on 28-Sept-2015, and it was finally accepted on 02-Oct-2015.





Contents lists available at ScienceDirect

# Computer Communications

journal homepage: [www.elsevier.com/locate/comcom](http://www.elsevier.com/locate/comcom)

## PrivHab+: A secure geographic routing protocol for DTN



Adrián Sánchez-Carmona\*, Sergi Robles, Carlos Borrego

Department of Information and Communications Engineering (dEIC), Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain

### ARTICLE INFO

#### Article history:

Received 19 November 2014

Revised 28 September 2015

Accepted 2 October 2015

Available online 10 November 2015

#### Keywords:

Opportunistic networking

Delay-Tolerant networking

DTN routing

Georouting

Privacy

### ABSTRACT

We present PrivHab+, a secure geographic routing protocol that learns about the mobility habits of the nodes of the network and uses this information in a secure manner. PrivHab+ is designed to operate in areas that lack of network, using the store-carry-and-forward approach. PrivHab+ compares nodes and chooses the best choice to carry messages towards a known geographical location. To achieve a high performance and low overhead, PrivHab+ uses information about the usual whereabouts of the nodes to make optimal routing decisions. PrivHab+ makes use of cryptographic techniques from secure multi-party computation to preserve nodes' privacy while taking routing decisions. The overhead introduced by PrivHab+ is evaluated using a proof-of-concept implementation, and its performance is studied under the scope of a realistic application of podcast distribution. PrivHab+ is compared, through simulation, with a set of well-known delay-tolerant routing algorithms in two different scenarios of remote rural areas.

© 2015 Elsevier B.V. All rights reserved.

### 1. Introduction and motivation

Many initiatives have been implemented to improve the life conditions of people living in developing countries by universalising the access to knowledge and information. These applications usually target rural areas and are very likely to deal with challenges like a sparse population, and a lack of data communication networks.

The need of infrastructure constrains the reach of these applications, because they cannot operate in regions lacking it. It happens that regions where the communication networks are unavailable or spotty, are usually the ones where these services would be more needed and valuable. Delay Tolerant Networking (DTN), based on the store-carry-and-forward strategy, is designed to operate in these challenged scenarios. DTN deals with the absence of simultaneous end-to-end paths [3] through the usage of mobile devices that opportunistically establish contact and exchange messages between them.

Routing protocols designed to operate in DTN scenarios usually generate and use information about node behaviours, as the historic of contacts established with each other node [27]. Then, they share this information with neighbours in order to improve the decision making [26]. Moreover, in some cases, a node is linked to a person, e. g. because it is carried in a pocket or backpack [31], or because they travel in the same vehicle. Therefore, the information that routing protocols use and share can be seen as private in-

formation about people's whereabouts or frequent behaviours. The more accurate and sensitive this information is, the more useful it is for the routing protocol, the more important is to protect its privacy [2]. Accordingly, a protocol that protects the privacy of this information expands the amount of scenarios where it can be used [13].

Our main contributions are summarised below:

- We introduce the concept of node's habitat, the area where a node is more likely to be found. The habitat is built by exploiting the life-cycles of the network users. It is a very useful tool for making routing decisions by comparing two nodes' habitats and selecting the best choice to deliver a message to its destination. We use an elliptic model of habitat to allow devices of small capabilities to work and to operate with it.
- We define PrivHab+, a novel DTN secure geographical routing protocol designed to operate in areas without network infrastructure. PrivHab+ uses the learnt information about the usual whereabouts of the nodes to find the best neighbour to carry the messages. PrivHab+ protects node's privacy by cryptographically protecting this information to avoid its disclosure.

The rest of this article is organised as follows. In Section 2, reviews the state of the art and provides a description about some related work of Geographical Routing Protocols, Secure Routing Protocols and Social-based Routing Protocols. In Section 3, we present the habitat, a useful information to compare nodes while routing messages. We explain how it is modelled and updated. Later, we introduce the concepts of homomorphic cryptography and Taxicab geometry, both needed to preserve nodes' privacy while routing using the habitat.

\* Corresponding author. Tel.: +34935813577.

E-mail addresses: [adria.sanchez@deic.uab.cat](mailto:adria.sanchez@deic.uab.cat) (A. Sánchez-Carmona), [sergi.robles@deic.uab.cat](mailto:sergi.robles@deic.uab.cat) (S. Robles), [carlos.borrego@deic.uab.cat](mailto:carlos.borrego@deic.uab.cat) (C. Borrego).

<http://dx.doi.org/10.1016/j.comcom.2015.10.002>

0140-3664/© 2015 Elsevier B.V. All rights reserved.

In Section 4.5, we present PrivHab+, a routing protocol that uses the habitats of the nodes to route messages while preserving the privacy of the nodes of the network. In Section 5, we analyse the knowledge obtained by each participant of the protocol and we reason about the privacy that PrivHab+ provides. In Section 6, we present the proof-of-concept we have implemented, and we use it to measure the performance of PrivHab+. In Section 7, we expose the results of the simulations that compare PrivHab+ with a set of well-known DTN routing protocols. Finally, Section 8 concludes this paper.

## 2. Related work

In this section, we provide the reader with a review of the related work. First, we present the state of the art of Geographical Routing Protocols. Later, we analyse the different proposals of Secure Routing Protocols in Delay Tolerant Networks. Then, we review some Social-based Routing Protocols that are related, somehow, to our proposal. Finally, we provide some conclusions about the study of the state of the art.

### 2.1. Geographical Routing Protocols

Geographical Routing Protocols have been studied both in Ad-hoc Networks and Delay Tolerant Networks. Most protocols, like GPSR [19] a protocol with support to Wireless Sensor Networks (WSN), always forward packets to the next hop that is geographically closest to the destination at the moment of the transmission. This approach becomes non useful when nodes cannot form a simultaneous path towards the destination and have to carry the packet until the next encounter. Besides, GPSR only takes into account the position of the nodes at the moment of the transmission, but not their movement. In [1], GPSR is modified to adapt it to DTN by being energy-efficient. However, messages are routed in the basis of a neighbourhood table that does not adapt well to a scenario where the topology of the network changes quickly. Using LAROD [23], nodes forward packets to neighbours inside a certain area located between the forwarder and the destination, without taking into account the mobility patterns of these nodes. In [24], a Location Service called LoDIS is presented to improve LAROD by using gossip-based techniques to update the location of the destination at each hop. Using LoDIS, the performance of the routing is greatly improved, but the privacy of all nodes results heavily damaged because their locations and speed vectors are periodically broadcasted. Moreover, LoDIS uses the speed vector of the nodes to predict their short-term future locations. This model loses precision in networks where the latencies are big due to a low level of connectivity, or because the packets travel big distances before reaching their destination. MoVe [25] is a routing protocol designed to work in Vehicular Networks where nodes forward messages to a neighbour if the neighbour is expected to come closer to the destination. In MoVe, nodes exchange information to determine whether the message shall be forwarded. Nodes use the speed vectors to make routing decisions. This information is not protected and does not take into account the recent past to infer routines or typical movement patterns. GeoDTN+Nav [6] is designed for routing in a network of streets, and it has three forwarding modes. In the DTN mode, it requires the nodes to know where they are heading. This requirement can be easily met by certain types of vehicles, like buses or taxis, but it is not reasonable with other types of nodes (e.g. nodes carried by walking people).

### 2.2. Secure Routing Protocols

Most Secure Routing Protocols aim to protect the routing algorithm's performance against malicious behaviours [18]. By design, it supposes that nodes voluntarily share any intimate information (battery level, state of the buffer, current location, speed vector, most

visited places, past encounters with neighbours, etc.) for the good of the network. These protocols usually consider that the only thing that has to be protected is the performance of the network. Besides, some Secure Routing Protocols, as SEAD [15], provide end-to-end security services to the contents of the messages, such as integrity, authentication, non-repudiation or confidentiality. Unfortunately, there are little proposals of routing algorithms that respect and protect the privacy of all the nodes that form the network. A system called ALAR, presented in [29], allows a source to send a message through a DTN without revealing its physical location and proposes an anti-localisation routing protocol. However, the only information that ALAR protects is the location where the source was when the message was sent. This proposal is incomplete because it only protects one concrete information. However, it proves that, in certain scenarios, nodes are unwilling to share all their information for the good of the network. For this reason, nodes privacy has to be protected. In Ad-hoc Networks, there is a mechanism designed to protect the privacy of the nodes. Pseudonym generators such as [17,4] provide anonymity to the nodes of the network by breaking the relation between nodes and identifiers. This way, an observer cannot gather enough information to learn the behaviour of a node. Pseudonyms change over time, and it is difficult to relate the new ones with the past ones. However, these mechanisms are not compatible with routing protocols where nodes need to share information with their neighbourhood. Hence, the usage of one of these mechanisms indirectly decreases the performance of the network, because they restrict the routing protocols that can be used. Some mechanisms, as the one presented in [43], only protect, by design, the identities of the sender and the receiver of the message. Other Secure Routing Protocols for Ad-hoc Networks, as the one presented in [7] and [33], are based on symmetric key cryptography or hash functions, and on source routing or distance vector protocols. This approach is unsuitable for DTN. An anonymous communication solution for DTN has been presented in [20], but it is designed to hide the identity of the nodes, not to protect the private information that these nodes use to make routing decisions.

### 2.3. Social-based Routing Protocols

There are some Social-based Routing Protocols that are related, somehow, to the present work. Social-based routing protocols are based on the idea of using the recent past to model the behaviour of a node and predict how it will behave in the near future. BUBBLE RAP [16] classifies nodes using their popularity inside their community. Then, messages are forwarded to more popular nodes until they reach the community of the destination. Its design is not good to send messages to hop-distant destinations because locations are not considered. So, during the first hops messages can be carried into the opposite direction of their destination while they are forwarded to more popular nodes. MobySpace [26] leverages the life-cycles of the nodes to track what points of interest are more visited by every node. These life-cycles are modelled this using a multi-dimensional probability vector, and messages are forwarded to nodes with a vector closer to the one of the destination. The classic Euclidean distance is used to measure the distance between vectors. This is a very interesting approach to our concept of habitat, but lacks adaptability. In MobySpace, the points of interest have to be defined *a priori* by an external agent, and some infrastructure is needed to allow nodes to detect if they are close or not to one of these points. Besides, MobySpace may lead to situations where a node that spends most of the time at point *A*, very close to *B*, is considered a bad choice because the destination is expected to be on *B*, without taking into account that *A* is near *B*. SANE [31] uses these same principles but defines the points of interest in a very broad sense, allowing the usage of more abstract concepts, and substitutes the Euclidean distance by a metric called "cosine similarity". HiBop [2] extends this approach using any

contextual information about nodes to make routing decisions. One of its drawbacks is the big amount of memory needed to store information about every contact. Besides, the authors do not explain how this contextual information can be updated as the behaviours of the nodes change. In [32], a general framework called CAR is presented. CAR goes one step further and not only uses the recent past to model the behaviour of a node, but it also tries to predict the future values of the attributes that define the context. However, all predictions are finally condensed in a single value, the probability of delivery. This probability is used to decide the node where every message is forwarded. This system is only useful to calculate the probability of delivery to known nodes. But it has limitations in scenarios with hop-distant destinations, where the first forwarders do not know almost anything about the destination because they never met before. CSI [14] models the spatio-temporal behaviours of the nodes using *behavioural profiles*, and forwards one-to-many messages through the nodes that are more similar to the destinations. Besides, the authors realise the importance of the privacy of the nodes and present a privacy-preserving mode of operation. This way the protocol can operate in scenarios where nodes are not willing to send its behavioural profiles to other nodes when needed.

Unfortunately, although at [2] the authors recognise that privacy is an important issue to consider and that more work is needed to solve it, [14] is the only one proposal that takes into account the privacy of the nodes. In all other cases, nodes are expected to broadcast their information about the locations they visit or the details about their interests to the neighbours.

#### 2.4. Summary

Geographical Routing Protocols are a common routing solution to Delay Tolerant Networks, but almost all proposals use contemporaneous information and short-term predictions, so they fail to take into account long-term trends of nodes' mobility. However, in scenarios where the distances to travel are big, and the density of nodes is low, it is more valuable to know where a node will go in the next hours than where it is currently headed [34,35].

The existence of several Secure Routing Protocols that protect the privacy of the nodes, even if they are limited, proves that in DTN we cannot assume that nodes are willing to share any information for the good of the network. Given the impact of routing protocols on the performance of the network, and taking into account the sensitivity of the information they use, the fact that there are no routing protocols that protect this information is a surprise.

To our knowledge, this work is the very first proposal that combines these two fields in a Secure Geographical Routing Protocol for DTN that uses and at the same time protects participants' private information.

Finally, our contributions, both the habitat as a model of nodes' behaviours and the protocol used to compare it, could fit, after some adaptation, in a variety of frameworks. For example, in some of the Social-based protocols reviewed, or in Haggel [40], a more general one. Note that this only refers to a lower level, to the way nodes store and exchange information. For the sake of simplicity, we will consider a Bundle-based DTN [36] during the rest of this article.

### 3. A habitat-based routing protocol

In this section, we explain how routing protocols need to compare nodes to make decisions, and we present the tools that PrivHab+ will use. We introduce the habitat concept. Then, we show how we model it using an ellipse, how we automatically calculate it and the parameters involved in the calculations. We explain the meaning of the different parameters and how to use them. Then we analyse how we can use additive homomorphic cryptography to compare habitats while preserving the privacy of their owners, and the drawbacks

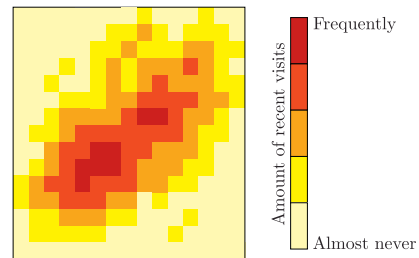


Fig. 1. Example of habitat represented with a heatmap. The darker the colour used to depict an area, the more frequently visited it is.

of this approach. Finally, we explain how to solve these drawbacks by simply changing the usual Euclidean geometry by the Taxicab geometry.

#### 3.1. Comparing nodes to route messages

DTN operation is based on opportunistic, usually unpredictable, contacts between pairs. Each time two or more nodes come close enough to be within communication range, an opportunity arises: messages can be forwarded between them in order to improve their probabilities of reaching their destination. At this moment, the routing protocol has to decide what messages must be relayed to what nodes. In fact, the quality of routing protocols depends on the decision they make<sup>1</sup>. The core of this decision-making process is an elemental operation, a comparison: given a node carrying a message and one neighbour, compare the two nodes to decide who is a better choice to carry the message towards its destination. Each time a routing protocol performs a comparison whose result is mistaken, a message will be relayed to a node that is less likely to deliver it to its destination than the previous one. This leads to a decrease of the performance of the network.

Our proposal solves the routing problem by comparing nodes using their habitat, a novel concept that takes advantage of the routine and the life-cycles of the nodes, to make routing decisions.

#### 3.2. A model of habitat

In a DTN, nodes may be carried by people, placed on any form of vehicle, located in a static known place, etc. Regardless of the type of the carrier, it is very likely that their mobility pattern becomes routine. For example, a static node will obviously remain immobile; a node carried by a person will probably spend a lot of time in the vicinity of the carrier's home or workplace; a node placed on a bus will pass over and over by the same points of their route; and a node placed on a taxi will usually be inside a certain area. We can benefit on this to predict the areas they will visit on the future based on the areas they visited on the past.

This implies that every node has an *habitat*, the area where the node is more likely to be found. Fig. 1 shows a heatmap, the most usual representation of a habitat. The heatmap contains the information of the areas where a node spends more time. It is obvious that a being with a habitat like the one presented in the figure can be found, eventually, in a location where he has not been never before. However, it will be far more likely to find him in the darker areas, where he has been repeatedly in the recent past. PrivHab+ makes use of this

<sup>1</sup> The quality of a routing protocol also depends on the forwarding policy. This policy is used to decide if multiple copies of a single message are created, and if the nodes keep a message after they forwarded it. We provide more discussion about this topic at the end of Section 4.5.

logic. This proposal is the very first approach that makes use of this concept to design a Geographical Routing Algorithm.

Therefore, we propose a system for location-aware nodes equipped with a navigation system to periodically obtain and use their location to update their habitat. For example, Global Positioning System (GPS) receivers are relatively inexpensive and lightweight, so it is reasonable to assume that all devices in the network could be equipped with one. We propose to use a relatively simple model of habitat to allow nodes to calculate it consuming the minimum energy and computational resources, and to operate quickly with it to make routing decisions. We model each habitat using an ellipse because it is simple enough to achieve an efficient protocol. Moreover, the ellipse can represent with precision far more shapes than other considered models, as the circle, the square or the rectangle<sup>2</sup>. Additionally, the usage of a simple geometric shape allows nodes to calculate their habitat using a mobile average, this way we avoid the need for maintaining a historic of past locations.

### 3.3. Definition and update of the elliptic habitat

We model each habitat  $H$  using an ellipse<sup>3</sup>. Therefore, each habitat is defined by three characteristics: two focal points and a radius. From now on, we will refer as  $F1 = (x_1, y_1)$  and  $F2 = (x_2, y_2)$  to the two focal points of the habitat and we will use  $r$  to denote their radius.

We assume that every geographic coordinate (a pair latitude–longitude) can be mapped<sup>4</sup> to Cartesian coordinates  $(x, y)$  and that this mapping is known by all the nodes of the network. With a frequency of  $\omega$  updates/hour, all nodes obtain their location  $L = (x, y)$ , and use an exponentially weighted moving average (EWMA) to update their habitat. The habitat  $H = (F1, F2, r)$  is updated using the previous version of the habitat  $H_{old} = (F1_{old}, F2_{old}, r_{old})$  and the current location  $L$ . The same process is used to build the habitat for the first time at system start-up and to adapt it to any changes in nodes' behaviours.

#### 3.3.1. Initialisation of the elliptic habitat

To initialise the system, the first known location  $L_0$  is used to initialise the habitat with the two focal points at the same coordinates of  $L_0$  and  $r = 0$ .

$$H_0 = (L_0, L_0, 0) \quad (1)$$

#### 3.3.2. Updating the focal points

Let  $F1_{old}$  be the nearest focal point to  $L$  and  $F2_{old}$  be the farthest to  $L$  focal point. The focal points of the habitat  $H$  are calculated by using EWMA to average the focal points of the previous version of the habitat  $H_{old}$  and the current location  $L$ . This first step is depicted in Fig. 2.

$$F1 = L * \alpha + F1_{old} * (1 - \alpha) \quad (2)$$

$$F2 = L * \frac{\alpha}{\beta} + F2_{old} * \left(1 - \frac{\alpha}{\beta}\right) \quad (3)$$

By using  $\beta > 1$ , the current location  $L$  weights more when calculating the new position of the nearest focal point than when calculating the new position of the farthest focal point. This means that  $L$  attracts

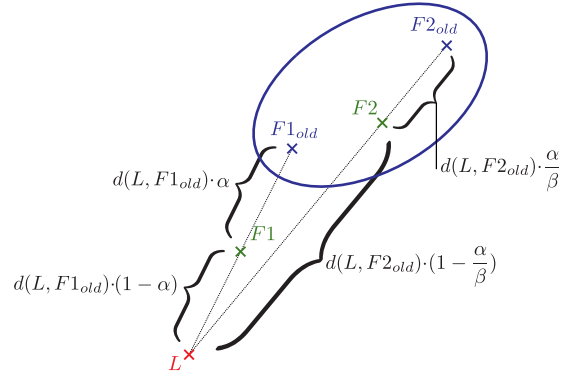


Fig. 2. Evolution of the focal points  $F1_{old}$  and  $F2_{old}$  when the new location  $L$  is used to update the habitat. Function  $d(L, F)$  denotes distance between  $L$  and a focal point  $F$ . Note that  $F1$  has been attracted by  $L$  using an  $\alpha$  factor while  $F2$  has been attracted using a lesser  $\frac{\alpha}{\beta}$  factor.

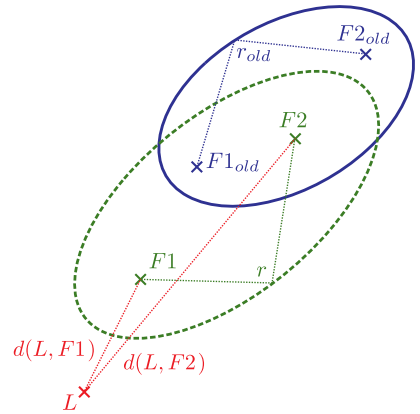


Fig. 3. Evolution of the radius. Distances and radius are depicted with dotted lines. The old radius  $r_{old}$  is used together with the distances  $d(L, F1)$  and  $d(L, F2)$  that separate the updated focal points  $F1, F2$  and the new location  $L$  to update the radius  $r$ . The radius of the habitat will increase if  $L$  is out of  $H_{old}$  and will decrease if  $L$  is contained by  $H_{old}$ .

more the nearest focal point, modifying the habitat's eccentricity depending on the relative position of  $L$  and  $H_{old}$ . The higher the  $\beta$  used, the more will change the form factor of the habitat when new distant samples are taken<sup>5</sup>.

#### 3.3.3. Updating the radius

Let  $d(L, F)$  be the distance between  $L$  and a focal point  $F$ . Once  $F1$  and  $F2$  have been updated. The radius  $r$  of the habitat is updated by averaging using EWMA the old radius  $r_{old}$  and the added distances  $d(L, F1)$  and  $d(L, F2)$  between each focal point of  $H$  and  $L$ . This second step is depicted in Fig. 3.

$$r = (d(L, F1) + d(L, F2)) * \alpha + r_{old} * (1 - \alpha) \quad (4)$$

#### 3.3.4. The habitat's time span

The time span that a habitat considers is a very important parameter. For example, a reader's habitat that considers only the last 2 hours is very likely to be a small circle around its current location. But if the

<sup>2</sup> Besides, in Taxicab geometry (it will be explained below), both the circle, the square and the rectangle are specific types of ellipses. So using the generalisation, the ellipse, we provide the tools needed to use any of these models.

<sup>3</sup> Definition: the set of points such that the distance from any point in that set to a given point called focus plus the distance from that point to the other focus is equal to the ellipse's radius

<sup>4</sup> Any cartographic projection can be used.

<sup>5</sup> Experiments using  $\beta < 50$  have shown that the form factor of the habitats hardly changes and the elliptic habitats usually tend to be quasi-circular habitats. Therefore, we recommend to use  $\beta > 50$ .

habitat considers the last 24 hours, it will probably be a bigger ellipse containing both the reader's home and the reader's place of work. If the considered time span is one week, the reader's habitat will also take into account the places where he or she spends the weekends, and so on.

When the time span of a habitat matches the life-cycle<sup>6</sup> of the nodes of the network, then it will become very useful to predict the areas that the nodes will visit again in the near future.

In order to perform meaningful comparisons between habitats that consider the same time span, PrivHab+ requires the nodes of the network to know it and to calculate the parameter  $\alpha$  using Eq. (5). Let  $\omega$  be the frequency of update of the habitat in updates/hour, and let  $T$  be the time span that a habitat has to consider in hours.

$$\alpha = \frac{2}{T\omega + 1} \tag{5}$$

Using a parameter  $\alpha$  calculated this way, due to the characteristics of EWMA, the last  $T\omega$  locations added to the average tend to weight the 86, 47% of the total. During the rest of the article, we will assume that a habitat considers a time span of  $T$  hours if its parameter  $\alpha$  has been calculated this way.

### 3.4. Homomorphic encryption: Paillier

When two nodes come close enough to establish a communication, their habitats have to be compared in order to choose the best choice for every message. But the habitat is a sensitive information about the recent movements of a node, when a node is carried by an animal or a vehicle, or placed somewhere, this is not a problem. However, When the node is linked to a person, its habitat is a private information of this person. In fact, we cannot expect nodes to harm their own privacy by sharing sensitive information with their neighbours. For this reason, nodes' privacy has to be preserved during the routing process. Our protocol has to allow a node to compare its habitat with the one of its neighbour at the same time that avoids the disclosure of information about any habitat to the other part.

Our protocol uses techniques of public-key cryptography, but we require the cryptosystem used to have a concrete property: to be homomorphic. An homomorphic cryptosystem is one in which, given two encrypted operands  $E(a)$  and  $E(b)$ , one can operate them and compute  $E(a + b)$  or  $E(a \cdot b)$  without separately decrypting each one. This way, a node can cypher and send information about its habitat to a neighbour, and the neighbour can operate it without violating the privacy of the first node<sup>7</sup>. A fully homomorphic cryptosystem, like [10], capable of performing both the addition and the multiplication, would be ideal, but this system is not viable nowadays because of the computational power it requires.

The presented protocol uses the additive homomorphic Paillier cryptosystem [42], capable of performing the addition and the subtraction of two cyphered operands and the multiplication by a unencrypted scalar. This cryptosystem is briefly described next.

In a communication between Alice and Bob, Alice starts by selecting two random primes  $p$  and  $q$  and computes  $n = pq$ ; plaintext messages are elements of  $\mathbb{Z}_n$ ; however, ciphertext messages are elements of  $\mathbb{Z}_{n^2}$ . Then Alice picks a random  $g \in \mathbb{Z}_{n^2}^*$  such that  $\gcd((L(g^\lambda \bmod n^2)), n) = 1$ , where  $\lambda = \text{lcm}(p - 1, q - 1)$  and  $L(x) = (x - 1)/n$ . Alice's public key<sup>8</sup> is  $Pk_A: (n, g)$  and her private key is  $pK_A: (\lambda, p, q)$ .

<sup>6</sup> Usual life-cycles of people are a day or a week. People usually move very similarly to how they moved in the previous cycle.

<sup>7</sup> Sections 4.5 and 5 will provide more details about this process.

<sup>8</sup> Note that if Bob does not trust Alice when she generates her Paillier modulus, he can insist she proves its validity, that it is the product of exactly two nearly equal primes [28].

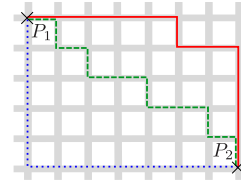


Fig. 4. Taxicab geometry distances. All three pictured lines have the same length for the route between  $P_1$  and  $P_2$ .

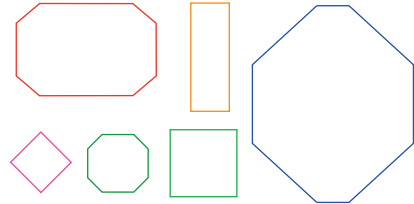


Fig. 5. Examples of ellipses in Taxicab geometry. The circle (down, at the left), the square (the third figure at the down row) and the rectangle (above the square) are specific types of ellipses.

To encrypt a message  $m$ , Bob picks a random  $r \in \mathbb{Z}_n^*$  and computes  $c = E(m) = g^m \cdot r^n \bmod n^2$ , the cyphertext of  $m$ . Finally, Bob can easily compute  $E(a + b) = E(a) \cdot E(b) \bmod n^2 = g^{a+b} \cdot (r_1 \cdot r_2)^n \bmod n^2$ ,  $E(a - b) = E(a)/E(b) \bmod n^2 = g^{a-b} \cdot (r_1/r_2)^n \bmod n^2$ , and  $E(a \cdot s) = E(a)^s \bmod n^2 = g^{a \cdot s} \cdot (r_1^s)^n \bmod n^2$  without decrypting the operands.

Finally, to decrypt a ciphertext  $c$ , Alice computes  $D(c) = L(c^\lambda \bmod n^2) = m$ .

### 3.5. Taxicab geometry

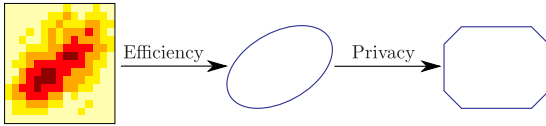
The usage of Paillier's cryptosystem restricts the operations we can use to compare habitats. Concretely, distances cannot be calculated because there is no way to calculate a square root. For this reason, we move from the usual Euclidean geometry to Taxicab geometry [22].

Taxicab is a geometry in which the distance between two points is the sum of the absolute differences of their Cartesian coordinates, instead of being the usual Euclidean distance. This distance function is usually called Manhattan distance<sup>9</sup> and is depicted in Fig. 4. Manhattan distances can be calculated without computing any square root<sup>10</sup>, an operation that is not supported by any homomorphic cryptosystem.

Throughout the entire article, all geometric calculations will be operated in Taxicab geometry, and all references to distances will refer to Manhattan distances. Fig. 5 provides some examples of the aspect of different ellipses in Taxicab geometry. Note that in Taxicab geometry, the ellipse is a generalisation of the circle (an ellipse with the two focal points located at the same place, this also applies in Euclidean geometry); the rectangle (an ellipse with a radius equal to the distance between the two focal points); and the square (an ellipse with a radius equal to the distance between the two focal points, and

<sup>9</sup> This name alludes to the grid layout of most streets on the island of Manhattan. The shortest path a car could take between two intersections in the borough have length equal to the intersections' distance in taxicab geometry.

<sup>10</sup> In order to calculate a Manhattan distance, the absolute value of a subtraction has to be computed. This operation is also not supported by any homomorphic cryptosystem, but, in Section 4.5, we explain how to calculate it benefiting from Taxicab geometry properties.



**Fig. 6.** The real habitat is modelled using a simple shape as the ellipse due to efficiency reasons. Then, the Euclidean geometry is substituted by the Taxicab geometry in order to protect nodes' privacy.

the two focal points placed diagonally between them). In this article we provide the tools to operate with the general case, the ellipse, optimisations and simplifications to operate with specific types of ellipses can be easily inferred.

Finally, Fig. 6 concludes this section with a visual summary of how we adapt the habitat concept to use it as the basis of a Secure Geographical Routing Protocol. First, the real habitat (represented by the heatmap) is modelled using an ellipse due to efficiency reasons, then, the ellipse is considered under Taxicab geometry in order to protect nodes' privacy.

#### 4. PrivHab+

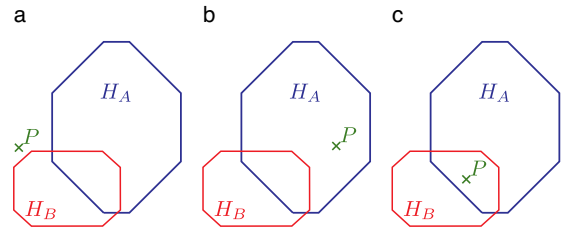
In this section, we present PrivHab+, the very first habitat-based geographical routing protocol that protects the privacy of the participants. Firstly, we introduce the notation needed during the rest of the section and explain the routing algorithm from a high-level point of view. Then, we take some considerations about the privacy of all participants and how the operands coming from others have to be treated. Later, we explain the method to solve the three geometric problems our routing algorithm needs to solve. Following, we provide a method to solve the three geometric problems without hurting the privacy of any participant. Then, we present the messages that has to be exchanged during the execution of the protocol and we explain how PrivHab+ can be implemented using any forwarding policy, and we provide some examples. Finally, we reason about the two-party design of PrivHab+.

##### 4.1. Notation

For the sake of clarity, we provide Table 1, which contains the notation used to refer to each one of the different elements that will appear in this section and a brief description of its meaning. From now on, we will use this notation.

**Table 1**  
Notation of all elements used in this section.

Notation	Meaning
$A$	The node that carries the message and performs the routing.
$B$	The other node involved in the transaction, it is a candidate to carry the message.
$P: (P_x, P_y)$	The point where the message has to be carried to.
$H: (F_1, F_2, r)$	A habitat.
$H_i$	The habitat of node $i$ .
$r_i$	Radius of the habitat of node $i$ .
$F_1: (f_{1x}, f_{1y})$	One of the focal points of a habitat or ellipse.
$F_2: (f_{2x}, f_{2y})$	The other focal point of a habitat or ellipse.
$E$	An ellipse.
$d(Z, W)$	Taxicab distance function between two elements. Let $Z$ be a point and let $W$ be another point, a habitat or an ellipse.
$X: (a, b)$	The nearest point to $P$ that belongs to a habitat.
$nonce$	A positive random value used only once.
$SE, \dots, NW$	Regions of the space relative to a habitat.
$E_V(\cdot)$	Paillier additive homomorphic encryption function using $Y$ 's public key.
$D_V(\cdot)$	Paillier additive homomorphic decryption function using $Y$ 's private key.



**Fig. 7.** The three possible situations in habitat-based routing: (a) the next waypoint is located outside the two habitats; (b) only one of the two habitats encloses the location of the next waypoint; (c) the two habitats enclose the location of the next waypoint.

##### 4.2. A two-phase routing protocol

We propose a routing protocol that operates in two different phases: 1) approximation phase, when messages are routed towards a geographic area using PrivHab+; 2) delivery phase, when messages are delivered to their destination using the classical DTN techniques of routing and delivery (e.g. direct delivery or Spray-and-Wait [38]). In this paper, we focus on the first phase.

During approximation, we use the habitats  $H_A$  and  $H_B$  of nodes  $A$  and  $B$  to decide who is the best choice to carry a message whose destination is located near  $P$ . We assume that an approximate location of the destination can always be known or guessed by the sender of the message, e.g. via the usage of a distributed secure position service like [41] and [37], or via the usage of an alternate communication channel. There are three different situations as depicted in Fig. 7, where our routing algorithm has to decide who is the best option:

- (a) If  $P$  is located outside both habitats, then the best choice will be the node whose habitat is nearest to  $P$  ( $H_B$  in Fig. 7) because it will likely bring the message nearer to its destination.
- (b) If  $P$  is located inside one habitat and outside the other, then the best choice will obviously be the node with the habitat that contains  $P$  ( $H_A$  in Fig. 7).
- (c) If  $P$  is located inside both habitats, then the best choice will be the node whose habitat is smaller ( $H_B$  in Fig. 7). We consider that it is more likely that this node will pass near  $P$  sooner.

We will use this algorithm during the rest of the article to decide the node that is the best choice to deliver every message to its destination.

##### 4.3. Privacy

On one hand, the location  $P$  is used during routing's first phase to approach the destination of a message. Therefore, this is a routing information, carried by the message, which have to be known by the routers that take custody of the message because they will need it in the next executions of PrivHab+. When the destination does not want the forwarders to associate  $P$  to its identity, a pseudonym mechanism can be used. The presented protocol is fully compatible<sup>11</sup> with pseudonym generator mechanisms as [17] or [4] that generate pseudonyms of the destination or the forwarders using its public key, or [30] that uses a secret shared between the nodes and hashing functions. These mechanisms can also be used by nodes that are very jealous of their privacy to avoid other nodes keeping track of the locations where they have encountered.

<sup>11</sup> When a node  $B$  sends a tuple  $E_A(Z), E_A(W)$  with  $Z, W \geq 0$ , it is indistinguishable to  $A$  if  $B$  is a better carrier than  $A$  or if  $B$  is the destination of the message. See Section 4.5 for more details.

Moreover, although  $P$  could not be linked to a node thanks to the usage of pseudonyms, it must remain hidden to the nodes that do not need this information to perform the routing. This measure is crucial to reduce the amount of information that  $B$  can infer about  $H_A$  (see Section 5 for more details).

On the other hand, the habitat is a private information that every node maintains and updates. It has to be used during the approximation phase to decide who are the best node to carry messages near their destination, but it cannot be made public because this will hurt the privacy of nodes. For this reason, both  $A$  and  $B$  need the protocol to be secure and do not reveal information about their habitats to the other part.

#### 4.4. Geometric problems of PrivHab+'s routing

As we seen in the previous sections, to perform our routing algorithm and compare the two habitats  $H_A$  and  $H_B$ , we need to answer three different questions:

1. How far is  $P$  from habitat  $H$ ?
2. Is  $P$  contained inside habitat  $H$ ?
3. Is  $H_A$  smaller than  $H_B$ ?

However, in order to protect the privacy of the participants, PrivHab+ uses homomorphic cryptography. For this reason, the set of operations we can use to do the calculations becomes heavily restricted when using operands coming from different nodes. In particular, we can only use addition, subtraction and multiplication by a non-cyphered operand.

For the sake of clarity, we will use the next paragraphs to briefly explain two different ways to solve these three problems: 1) from a geometric point of view; and 2) using the homomorphic cryptography's constrained tools. Note that, geometrically, a habitat is equivalent to an ellipse.

##### 4.4.1. Distance from a point to an ellipse: geometrically

The distance from a point  $P$  to an ellipse  $E$  with two focal points  $F1$  and  $F2$  and a radius  $r$  in Taxicab geometry is solved this way:

First, we calculate distances  $d(F1, P)$ , between  $F1$  and  $P$ , and  $d(F2, P)$ , between  $F2$  and  $P$ , using Eq. (6).

$$d(F, P) = |F_x - P_x| + |F_y - P_y| \quad (6)$$

Then, we define  $E'$ , the closest point of the border of  $E$  to  $P$ . We split these two distances into two parts: the part that is contained within ellipse; and the part that is outside the ellipse<sup>12</sup>.

$$d(F1, P) = d(F1, E') + d(E', P)$$

$$d(F2, P) = d(F2, E') + d(E', P) \quad (7)$$

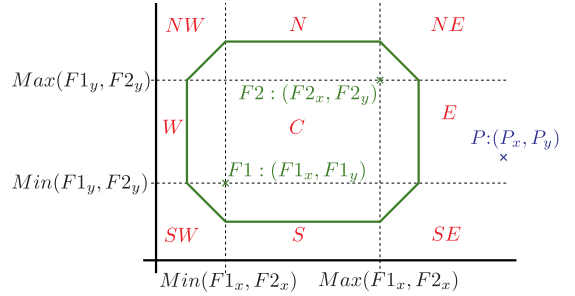
Then, we add these two distances and we subtract the radius  $r = d(F1, E') + d(F2, E')$ . As a result, we obtain the double of the distance between the ellipse and  $P$  without knowing the exact location of  $E'$ .

$$d(F1, P) + d(F2, P) - r = 2 \cdot d(E', P) =$$

$$d(F1, E') + d(F2, E') + 2 \cdot d(E', P) - d(F1, E') - d(F2, E') \quad (8)$$

##### 4.4.2. Distance from a point to a habitat: constrained tools

The absolute value of a cyphered operand cannot be calculated with the constrained tools of homomorphic cryptography. However,



**Fig. 8.** The regions of the space (NW, N, NE, W, C, E, SW, S and SE) are defined in the basis of the coordinates of the focal points  $F1$  and  $F2$ . In the example shown,  $P$  is located in region E, and when we know this we can calculate the distances  $d(F1, P)$  and  $d(F2, P)$ .

we can take advantage of Eq. (9) to walk around this issue and calculate the absolute value of a subtraction if we know beforehand the relation between the two operands.

$$|Z - W| = \begin{cases} Z - W & : Z > W \\ W - Z & : Z < W \end{cases} \quad (9)$$

In order to use Eq. (9) to obtain the absolute value needed to calculate the distance from a point to the habitat (see Eq. (6)), we need to know the relation between the coordinates of  $P: (P_x, P_y)$  and the coordinates of the two focus points  $F1: (F1_x, F1_y)$  and  $F2: (F2_x, F2_y)$ .

So we first divide the space into 9 regions, depending on their relation to the two focus of the habitat, as depicted in Fig. 8. To know the region where  $P$  is located, we calculate the maximum and minimum values of the coordinates of the two focus using Eq. (10). Then we compare them with the coordinates of  $P$ .

$$F_{x_{min}} = \text{Min}(F1_x, F2_x)$$

$$F_{x_{max}} = \text{Max}(F1_x, F2_x)$$

$$F_{y_{min}} = \text{Min}(F1_y, F2_y)$$

$$F_{y_{max}} = \text{Max}(F1_y, F2_y) \quad (10)$$

Once we know the region where  $P$  is located, we can use Eqs. (6) and (9) to calculate the distances between  $F1, F2$  and  $P$ . Table 2 shows how to calculate the added distance between the two focus points and  $P$  depending on the region where  $P$  is located.

**Table 2**

Distance between  $P: (P_x, P_y)$  and the two focus point  $F1: (F1_x, F1_y)$  and  $F2: (F2_x, F2_y)$ , depending on where  $P$  is located.

$d(F1, P) + d(F2, P)$	$P_x \leq F_{x_{min}}$	$F_{x_{min}} < P_x \leq F_{x_{max}}$	$P_x > F_{x_{max}}$
$P_y > F_{y_{max}}$	$(F_{x_{max}} - P_x) +$ $(F_{x_{min}} - P_x) +$ $(P_y - F_{y_{max}}) +$ $(P_y - F_{y_{min}}) +$	$(F_{x_{max}} - P_x) +$ $(P_x - F_{x_{min}}) +$ $(P_y - F_{y_{max}}) +$ $(P_y - F_{y_{min}}) +$	$(P_x - F_{x_{max}}) +$ $(P_x - F_{x_{min}}) +$ $(P_y - F_{y_{max}}) +$ $(P_y - F_{y_{min}}) +$
$y \leq F_{y_{max}}$	$(F_{x_{max}} - P_x) +$ $(F_{x_{max}} - P_x) +$	$(F_{x_{max}} - P_x) +$ $(F_{x_{max}} - P_x) +$	$(P_x - F_{x_{min}}) +$ $(P_x - F_{x_{min}}) +$
$F_{y_{min}} < P_y$	$(F_{y_{max}} - P_y) +$ $(P_y - F_{y_{min}}) +$	0	$(F_{y_{max}} - P_y) +$ $(P_y - F_{y_{min}}) +$
$P_y \leq F_{y_{min}}$	$(F_{x_{max}} - P_x) +$ $(F_{y_{max}} - P_y) +$ $(F_{y_{min}} - P_y) +$	$(F_{x_{max}} - P_x) +$ $(P_x - F_{x_{min}}) +$ $(F_{y_{max}} - P_y) +$ $(F_{y_{min}} - P_y) +$	$(P_x - F_{x_{max}}) +$ $(P_x - F_{x_{min}}) +$ $(F_{y_{max}} - P_y) +$ $(F_{y_{min}} - P_y) +$

After  $d(F1, P) + d(F2, P)$  is obtained from Table 2, the last thing to do is to subtract the radius  $r$ , using Eq. (8) to obtain  $2 \cdot d(H, P)$ , the double of the distance between  $P$  and the habitat  $H$ .

<sup>12</sup> Note that, in the Euclidean geometry, the distance between a point and an ellipse cannot be calculated this way because Eq. (7) only holds in the Taxicab geometry.

Finally, Eq. (11) shows how to use the double of the distance to compare two habitats and decide which one is closer to a certain point  $P$ .

$$2 \cdot d(H_A, P) - 2 \cdot d(H_B, P) < 0 \iff d(H_A, P) < d(H_B, P) \quad (11)$$

Note that a distance between  $P$  and  $H$  calculated this way will be negative if  $P$  is contained inside  $H$ . On the next paragraphs we will explain how benefit from this fact to know if  $P$  is inside or outside the habitat. Note also that the usage of other models of habitat as the square, the circle or the rectangle, that are specific types of ellipses, would simplify the calculations because some regions would disappear and would not need to be considered.

#### 4.4.3. A point contained inside an ellipse: geometrically

Given an ellipse  $E$  characterised by two focal points  $F1: (F1_x, F1_y)$  and  $F2: (F2_x, F2_y)$  and a radius  $r$ , a point  $P: (P_x, P_y)$  is contained inside  $E$  if and only if Eq. (12) holds.

$$|P_x - F1_x| + |P_y - F1_y| + |P_x - F2_x| + |P_y - F2_y| \leq r \quad (12)$$

#### 4.4.4. A point contained inside a habitat: constrained tools

As we have seen, to calculate the distance from a point  $P$  to a habitat  $H$ , what we really calculate is the double of the distance from a point  $P$  located outside the habitat  $H$  to the nearest point of  $H$ . If  $P$  is located inside the habitat, due to the usage of Eq. (9), the absolute value of the distance will be a negative value<sup>13</sup>. Far from being a drawback, we benefit from this property to use the calculated distance to the habitat to know if  $P$  is contained inside it, as shown in Eq. (13).

$$d(H, P) \leq 0 \iff P \in H \quad (13)$$

#### 4.4.5. Comparative of size between ellipses: geometrically

Given two ellipses,  $E_1$  and  $E_2$ , and their respective radius  $r_1$  and  $r_2$ , the smaller ellipse is the one that have the lesser radius. Therefore,  $E_1$  is the smaller ellipse if Eq. (14) holds, otherwise,  $E_2$  is the smaller one.

$$r_1 < r_2 \quad (14)$$

#### 4.4.6. Comparative of size between habitats: constrained tools

To compare the size of habitats  $H_A$  and  $H_B$ , we subtract their radius  $r_A$  and  $r_B$  one from another. Then, we check the sign of the result to decide which habitat is the smallest.

$$(r_A - r_B) * nonce \geq 0 \iff H_A > H_B$$

$$(r_A - r_B) * nonce < 0 \iff H_A < H_B \quad (15)$$

Note on Eq. (15) that we use a positive *nonce*. This value is unknown for the other part of the transaction. It is used to hide the real relation between the radius of the habitats and provide a randomised response. Later, the other part will binarise the result by taking into account only its sign.

### 4.5. Messages exchanged

Let  $A$  be the node that carries a set of messages  $m_i$ , with a habitat  $H_A: (F1_A, F2_A, r_A)$ . Let  $P_i: (P_{xi}, P_{yi})$  be the point where each message  $m_i$  wants to be carried to, and  $B$  be a neighbour with a habitat  $H_B: (F1_B, F2_B, r_B)$ . We denote a message sent by  $A$  to  $B$  with  $A \rightarrow B$ : *message*. By the previous definitions,  $A$  want to know if  $B$  is a better choice to carry each message  $m_i$  towards  $P_i$ .

PrivHab+ consists in five steps, the first of them is totally asynchronous, and requires nodes to exchange three messages. Depending on the result of the execution of the algorithm, an additional last one (the forwarded message) is sent.

0. Node  $A$  calculates  $d_{Ai} = d(H_A, P_i)$ , the distance between its habitat and every  $P_i$ ;  $A$  uses  $d_{Ai} = 0$  if  $P \in H_A$  and  $d_{Ai} \geq 1$  otherwise. As  $A$  knows both  $H_A$  and  $P_i$ , and the operations do not need to be performed using homomorphic encryption.

Besides, node  $B$  calculates the characteristics of its habitat:  $Fx_{max}$ ,  $Fx_{min}$ ,  $Fy_{max}$  and  $Fy_{min}$  using Eq. (10). This calculations can be done asynchronously (e. g. when the habitat is updated).

1. From that moment on, each time  $B$  establishes a contact with and any other node,  $B$  starts by announcing the characteristics of its habitat to its neighbours<sup>14</sup>.

$$B \rightarrow A: \begin{matrix} E_B(Fx_{max}), E_B(Fx_{min}), \\ E_B(Fy_{max}), E_B(Fy_{min}) \end{matrix}$$

2. Node  $A$  compares each received value with the corresponding coordinates of each point  $P_i$ . The comparisons are done by subtracting the corresponding coordinate of  $P_i$  from the characteristics of the habitat and then multiplying the result, to randomise it, with a random one-use value denoted *nonce*.  $A$  compares  $Fx_{max}$  with  $P_{xi}$  using Eq. (16), and calculates the other comparisons the same way. The first two received values are compared with  $P_{xi}$  and the last two with  $P_{yi}$ .

$$\left( \frac{E_B(Fx_{max})}{E_B(P_{xi})} \right)^{nonce} = E_B((Fx_{max} - P_{xi}) \cdot nonce) \quad (16)$$

Then  $A$  sends the comparisons<sup>15</sup> to  $B$  together with the coordinates of each  $P_i$ , the distance  $d_{Ai}$  and the radius  $r_A$  of  $H_A$ .

$$A \rightarrow B: \begin{matrix} E_A(r_A), \{E_B((Fx_{max} - P_{xi}) \cdot nonce), \\ E_A(P_{xi}), E_B((Fx_{min} - P_{xi}) \cdot nonce), \\ E_A(P_{yi}), E_B((Fy_{max} - P_{yi}) \cdot nonce), \\ E_A(2d_{Ai}), E_B((Fy_{min} - P_{yi}) \cdot nonce)\}^i \end{matrix}$$

3. For each  $P_i$ ,  $B$  decrypts all the received comparisons. Node  $B$  knows that each decrypted value greater than zero means that the characteristic of the habitat is greater than the corresponding coordinate of  $P_i$ . This way  $B$  decides the region where  $P_i$  is placed. Then,  $B$  calculates distance  $2d_{Bi}$ . Afterwards,  $B$  computes the comparison between  $2d_{Ai}$  and  $2d_{Bi}$ , using Eq. (17), and the comparison of radius<sup>16</sup>  $r_A$  and  $r_B$  using Eq. (18).

$$\left( \frac{E_B(2d_{Ai})}{E_B(2d_{Bi})} \right)^{nonce} = E_B((2d_{Ai} - 2d_{Bi}) \cdot nonce) \quad (17)$$

$$\left( \frac{E_A(r_A) \cdot E_A(2d_{Ai})^{r_B}}{E_A(r_B)} \right)^{nonce} = E_A((r_A + 2d_{Ai} \cdot r_B - r_B) \cdot nonce) \quad (18)$$

The last step for  $B$  is to send the results, but before that,  $B$  orders each pair of comparisons in a random way unknown to  $A$ .

$$B \rightarrow A: \begin{matrix} \{E_A((2d_{Ai} - 2d_{Bi}) \cdot nonce), \\ E_A((r_A + 2d_{Ai} \cdot r_B - r_B) \cdot nonce) \\ \text{or} \\ E_A((r_A + 2d_{Ai} \cdot r_B - r_B) \cdot nonce), \\ E_A((2d_{Ai} - 2d_{Bi}) \cdot nonce)\}^i \end{matrix}$$

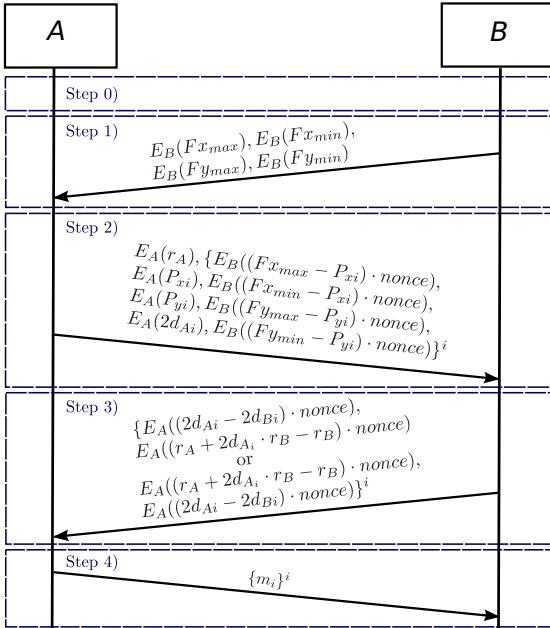
<sup>14</sup> This announcement can be made during the neighbour discovery process, by adding this information to the beacons.

<sup>15</sup> We have used “[{” and “}]” to enclose the part of the information that is repeated one time for each message  $m_i$ .

<sup>16</sup> The added element  $d_{Ai} \cdot r_B$  blurs the comparison. This way  $A$  can only infer information about  $H_B$ 's radius when  $P_i$  is contained both by  $H_A$  and  $H_B$ . See Section 5 for more details.

<sup>13</sup> Note that our protocol checks several times if an operand  $\rho$  is positive or negative. In the Paillier cryptosystem,  $\rho$  will be an element of  $\mathbb{Z}_n$ . To check this condition, if we ensure that  $n$  is sufficiently large and that all values  $\rho$  we will use are  $\rho \leq n/2$ , then we can consider that  $\rho > n/2 \iff \rho < 0$ .





**Fig. 9.** Schema of the messages exchanged during the execution of PrivHab+. At Step 0) the two nodes asynchronously perform calculations that will be used during the protocol. At Step 1) node B uses the neighbour discovery process to send to A the characteristics of the habitat  $H_b$ . At Step 2) node A sends to B the distance  $2d(H_a, P_i)$  and the information B needs to calculate  $2d(H_b, P_i)$ . At Step 3) node B compares both distances, and the radius of the two habitats, randomises the results and sends them to A. Finally, at Step 4) A decrypts the comparisons to know if B is a better choice than A. Finally, A sends, or not, the message  $m_i$  to B according to its forwarding policy.

4. Finally, node A decrypts each pair of comparisons. For every message  $m_i$  for whom the two decrypted values are equal or greater than 0, A learns that B is a better choice. Knowing that, A applies its forwarding policy (more details are provided below) to decide if any message has to be sent to B.

$$A \rightarrow B : \{m_i\}^i$$

Fig. 9 provides a schema of the messages exchanged during each phase of the protocol.

4.6. Forwarding policy

After the execution of PrivHab+, node A carrying message  $m_i$  knows if the execution was successful and if B is a better choice to carry the message towards its destination. Then, A decides if the message has to be forwarded to B, and if a copy of  $m_i$  has to be kept in A. The number of copies of every message flowing through the network will be directly determined by the forwarding policy used. Therefore, this decision, determined by the forwarding policy of A, can have an impact on the performance of PrivHab+.

PrivHab+ is compatible with any forwarding policy. As this paper is essentially focused on the decision making, meaning the comparison of two nodes to decide who is the best choice, the study of the forwarding policy is out of the scope of this paper. However, we provide next a set of examples of different forwarding policies that could be applied. Note that we do not pretend this set to be complete. Further research is planned by the authors to study and analyse all possible options to find the best policy for each scenario.

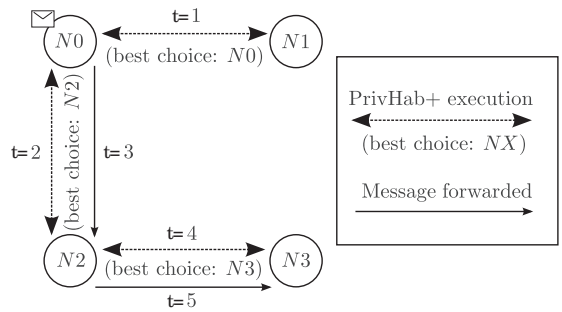
- *Direct single-copy policy:* nodes always forward the message to the node that is a better choice, no copies of the messages are created.
- *Direct multi-copy policy:* nodes always forward the message to the node that is a better choice, but each node that has forwarded a message keeps one copy of it.
- *Limited multi-copy policy:* nodes forward the message to the node that is a better choice and keep a copy a limited amount of times. When a node reaches the threshold for a message, no more copies of this message are created, and it is not forwarded more by this node. Many different strategies can be used to define the threshold of every node and every message.
- *Probabilistic policy:* messages are forwarded to the node that is a better choice a X% of times. They are also forwarded to nodes that are a worse choice a or do not have a habitat to compare a Y% of times. Besides, nodes keep a copy of the forwarded message the Z% of times, where X, Y and Z are parameters of the network.
- *Multi-criteria policy:* nodes execute other routing algorithms and combine their output with PrivHab+'s one to decide if the message has to be forwarded and if a copy has to be kept at the node.

For the sake of simplicity, during the rest of this paper we will assume that PrivHab+ uses, by default, the direct single-copy forwarding policy.

4.7. A two-party protocol

At [9], the authors have studied the enormous complexity of realising a multi-party secure comparison between an indefinite number of nodes. Besides, encounters between two nodes are the most common [12], encounters between three, four or more nodes are so rare that they cannot have a huge impact on the performance of the network. For the sake of simplicity and to maintain the computational overhead as low as possible, we have designed PrivHab+ to operate between two nodes.

PrivHab+ solves the encounters where three or more nodes meet, iterating its execution. PrivHab+ low overhead allow nodes to execute it more than once, and the “winner” of each comparison can be compared again with another neighbour. This process can be repeated until all nodes have been compared and the best has been found, or until the connectivity window ends. Fig. 10 illustrates this process. This way, if the communication ends suddenly before all comparisons are finished, PrivHab+ will find at least a partial “winner”. In the figure, if the communication ends before forwarding the message to the best node (N3), this partial “winner” would be N2, who is better than N0 and N1.



**Fig. 10.** Node N0 carrying a message meets N1, N2 and N3. Numbers denote the order of the operations. N0 compares itself with N1 using PrivHab+ and finds that N1 is a worse choice, so it does not forward the message. Then N0 compares itself with N2, who results to be a better choice, so the message is forwarded to N2. Finally, N2 compares itself with N3 using PrivHab+ and forwards the message to N3 because it is a better choice.

## 5. Security analysis

In this section, we analyse the knowledge obtained by each participant of PrivHab+ under the scope of secure multi-party computations. We first consider the passive adversary mode, where one participant executes the protocol and then makes inferences to obtain knowledge about the other participant's inputs. Then, we consider the active adversary mode, where one participant tampers its messages to try to obtain an advantage. Then, we reason about the security obtained in the two models.

### 5.1. Passive adversary mode

A secure multi-party computation [11] consists in computing a function on any input, on a network where different participants hold each input, and ensuring that no more information is revealed to a participant than what can be inferred from that participant's input and the computed output.

Following, we treat routing as a secure multi-party computation problem where the result of a routing algorithm has to be computed using private data held by the candidate nodes to carry the message. In order to consider PrivHab+ as a secure protocol, we need to prove that it reveals only the result of the function and the inferences that can be deduced from this output with one or more input values [8]. We consider a passive adversary mode where the participants exchange truthful messages and then analyse them trying to obtain information about the other part's habitat.

#### 5.1.1. Knowledge obtained by A

Table 3 summarises all knowledge that can be learned by A, the node that carries the message, about  $H_B$ , the habitat of the candidate node B. In all cases, the obtained knowledge is inferred using the output of the protocol and the inputs provided by A. None information can be learned from the messages exchanged with B, because they are encrypted with B's key, and the ones that A can decrypt are randomised through the usage of random *nonce* values.

**Table 3**

Knowledge obtained by A at the end of the protocol. If B is found to be a better choice, then A infers that B is a better candidate and that  $H_B$  is closer to location P than  $H_A$ . Node A also infers that  $H_B$  is smaller than  $H_A$  in the case that P is contained inside  $H_A$ . If B is found to be a worse choice, then A infers that  $H_B$  is farther to P than  $H_A$ , but cannot know if  $H_B$  is bigger or smaller than  $H_A$ .

A knows		A infers		
Input	Output	$d_A \leftrightarrow d_B$	$P \leftrightarrow H_B$	$r_A \leftrightarrow r_B$
$P \in H_A$	B	$d_A = d_B = 0$	$P \in H_B$	$r_A \geq r_B$
	A	$d_A \leq d_B$	$P \notin H_B$ or $r_A < r_B$	
$P \notin H_A$	B	$d_A \geq d_B$	Nothing	Nothing
	A	$d_A < d_B$	$P \notin H_B$	Nothing

#### 5.1.2. Knowledge obtained by B

The knowledge obtained by B depends on the forwarding policy of A. The only thing B knows is not the output of PrivHab+, but the fact that the message has finally been forwarded or not. If the forwarding policy used makes possible to not send the message when B is a better choice, or to send the message even if B is a worse choice, then B cannot infer PrivHab+'s output. Therefore, in this situation B cannot learn anything about  $H_A$ . Assuming that B knows A's forwarding policy, we will analyse the worst-case scenario: a direct (single-copy or multi-copy) forwarding policy that allows B to know the output of PrivHab+ from the forwarding of the message.

Table 4 summarises all knowledge that can be learned by B, the candidate node to take custody of the message. Only one information, P's region, can be learned from the message received from A.  $H_A$

**Table 4**

Knowledge obtained by B at the end of the protocol. If the message is sent B infers that it is a better candidate than A and receives the coordinates of P with the message. If the message is not sent, B learns the region where P is located, but not  $d_B$ . This only applies in the worst-case scenario: when the forwarding policy of A makes the output of PrivHab+ easy to establish for B.

B knows	B learns		B infers
	About P	About $d_B$	
Output			$d_A \leftrightarrow d_B$
Message received	$P: (P_x, P_y)$	$d_B$	$d_A \geq d_B$
Message do not received	Region where P is located	Nothing	$d_A \leq d_B$

characteristics are encrypted with A's key, and the comparisons that B can decrypt are randomised through the usage of random *nonce* values. Only the region where P is located is revealed. This knowledge about P's region is necessary for B to calculate  $d_B$ . Node B can also infer the relation between  $d_A$  and  $d_B$ , even without knowing<sup>17</sup>  $d_B$ , from the forwarding of the message. Note that maintaining P hidden to B (only P's region is revealed) if the message is not forwarded is crucial to avoid that B can calculate  $d_B$  and use it to infer more information about  $H_A$ .

#### 5.1.3. Conclusions

Anything learned by A about  $H_B$ , or by B about  $H_A$ , from the protocol is also learnable from the output alone. The computation made is a routing protocol, so, if  $m$  is forwarded to B, coordinates of P are revealed to B because they will be needed in the next executions. Otherwise, the only thing B learns about P is the region<sup>18</sup> where it is located in relation with  $H_B$ , because this knowledge is necessary for B to compute  $d_B$ .

Therefore, PrivHab+ is secure to A and B because it reveals only the result of the algorithm and inferences derived from this result. Besides, PrivHab+ provides best-effort privacy to P because it hides its location and reveals only the region where P is located. As we have explained in the previous section, this can be easily enhanced by breaking the relation between the destination and P using a pseudonym generator mechanism.

### 5.2. Active adversary mode

In the active adversary mode, we consider an attacker that may use untruthful information about their own habitat, the messages they carry, or the location P where a message is intended, in order to disclose private information about the other part's habitat.

#### 5.2.1. Knowledge obtained by A

A node carrying a message can lie about P,  $d_A$  and  $r_A$  in order to uncover information about  $H_B$ . There are two strategies that an active attacker A can follow: 1) Produce chosen-destination arbitrary messages using a set of false  $P'$  and  $d'_A$  to try to discover the area covered by  $H_B$ ; and 2) tamper  $r'_A$  to learn about  $r_B$ .

1. Discover the area covered by  $H_B$ : every time PrivHab+ is executed, A learns that  $H_B$  is located completely outside a circle with centre at P and radius  $d_A$  if node A is chosen as the best choice. The same way, A learns that at least one part of  $H_B$  is located inside a circle with centre at P and radius  $d_A$  if the best choice is B. Therefore, node A can exploit this by producing arbitrary messages destined to a set of locations  $P'$  and using set of false distances  $d'_A$ , and then repeatedly execute PrivHab+ to try to learn the area covered by

<sup>17</sup> Node B does not even know  $d_B$  until receiving P with the message and computing the distance again. The reason is that  $d_B$  is calculated via homomorphic cryptography and only A can decrypt it.

<sup>18</sup> The region where P is located is far less accurate that the coordinates of P or the distance between P and  $H_B$ . Moreover, B does not even know who is the destination, and therefore, B cannot relate this P's region with any node.

$H_B$ . The knowledge that  $A$  can obtain from this is summarised by Table 5.

**Table 5**

Knowledge obtained by  $A$  at the end of the protocol when  $A$  uses  $d'_A$  and  $P'$  instead of  $d_A$  and  $P$ . If  $A$  is chosen,  $A$  learns where  $H_B$  is not located. If  $B$  is chosen,  $A$  learns that a part of  $H_B$  is inside an area. The third column establishes the situations where it is useful for  $A$  to lie about  $d_A$ .

A knows Output	A learns About $d_B$	Useful iff
$A$	$d_B \geq d'_A$	$d'_A > d_A$
$B$	$d_B \leq d'_A$	$d'_A < d_A$

2. *Discover  $r_B$* : the result of an execution of PrivHab+ consists of a tuple containing two results randomly ordered. Each result can be greater or equal than zero ( $\geq 0$ ), or negative ( $< 0$ ). One of them, the radius comparison, only makes sense if and only if  $d_A = 0$ . In order to know the result of the radius comparison,  $A$  needs to repeatedly execute PrivHab+ using the same values  $d'_A = 0$  and  $r'_A$ , and a different  $P$ , until obtaining a different result in one of the two comparisons. When this happens, node  $A$  learns which result corresponds to each comparison, and learns if  $r_B$  is higher or lesser than  $r'_A$ . Note that the only way to obtain a different result in one comparison using this method is by using two false  $P'_1$  and  $P'_2$  that are located one inside  $H_B$  and the other outside it. Table 6 summarises this process.

**Table 6**

Knowledge obtained by  $A$ . Depending on how the result of the comparison of distances change when using a different  $P'$ , node  $A$  learns the relation between  $r'_A$  and  $r_B$ . If  $A$  has selected  $P'_1$  and  $P'_2$  randomly, then he also learns which of them is located inside  $H_B$  and which is located outside it. The third column establishes the situations where it is useful for  $A$  to lie about  $r_A$ .

A knows		A learns		Useful
Result 1	Result 2	$P_i \in H_B$	$r'_A \leftrightarrow r_B$	iff
$(< 0, < 0)$	$(< 0, \geq 0)$	$P_2$	$r'_A < r_B$	$r'_A > r_A$
$(< 0, \geq 0)$	$(\geq 0, \geq 0)$	$P_2$	$r'_A \geq r_B$	$r'_A < r_A$
$(< 0, \geq 0)$	$(< 0, < 0)$	$P_1$	$r'_A < r_B$	$r'_A > r_A$
$(\geq 0, \geq 0)$	$(< 0, \geq 0)$	$P_1$	$r'_A \geq r_B$	$r'_A < r_A$

### 5.2.2. Knowledge obtained by B

Node  $B$  does not initiate the execution of PrivHab+, nor controls the amount of messages  $m_i$  that will be routed. Then, its only chance to lie is manipulating the results of the comparisons sent in Step 3. The candidate node  $B$  can lie about its habitat, using  $H'_B$  instead of  $H_B$ , or about the distance from its habitat to  $P$ , using  $d'_B$  instead of  $d_B$ . Given that using a tampered habitat  $H'_B$  will lead to the calculation of an untruthful distance  $d'_B$ , both cases can be treated likewise. Table 7 summarises all knowledge learned by  $B$  in these two cases.

Node  $B$  will obtain more information about  $H_A$  lying than being truthful only if  $B$  finally receives the message and  $d'_B > d_B$ , or if  $B$  does not receive the message and  $d'_B < d_B$ . In both cases,  $P$ , and, therefore  $d_B$ , are unknown to  $B$  prior of the exchange. Therefore,  $B$  wants  $d'_B$  to be high to obtain more information if  $B$  will win the comparison, but a higher  $d'_B$  makes  $B$  less likely to win it. Equivalently,  $B$  wants  $d'_B$  to be small if  $B$  will lose the comparison, but a lesser  $d'_B$  makes  $B$  more likely to be selected as the best candidate. Besides,  $B$  will not obtain  $P$  if does not receive the message, and knowing the distance between  $H_A$  and  $P$  is not useful if  $P$  is unknown. For these reasons, there is no a straightforward strategy to select  $H'_B$  or  $d_B$  and guarantee that  $B$  will take an advantage from this.

**Table 7**

Knowledge obtained by  $B$  at the end of the protocol when  $B$  uses  $d'_B$  instead of  $d_B$ . If the message is sent  $B$  infers that it is a better candidate than  $A$ . The third column establishes the situations where it is useful for  $B$  to lie about  $d_B$ . This only applies in the worst-case scenario: when the forwarding policy of  $A$  makes the output of PrivHab+ easy to establish for  $B$ .

$B$ knows Output	$B$ learns About $d_A$	Useful iff
Message received	$d_A \geq d'_B$	$d'_B > d_B$
Message do not received	$d_A \leq d'_B$	$d'_B < d_B$ $B$ knows $P$

### 5.2.3. Conclusions

An active attacker can try to learn things about the other part's habitat by using untruthful information during the execution of PrivHab+.  $A$  can try to learn the area covered by  $H_B$  and its radius  $r_B$ , while  $B$  can try to learn the distance from  $H_A$  to  $P$ . In both cases, the information obtained by the attacker is *the same* information (the result of one or more comparisons) that he can infer from a truthful execution of the protocol. The only thing an attacker can change is the value to compare with the other part's radius or distance. However, the attacker can only benefit from these changes if the change made and the result of PrivHab+ are aligned. And in all cases happens that changing the value to improve its usefulness decreases the probability of obtaining the desired result.

As  $A$  is the node that starts the transaction and the only one that knows the number of messages he carries, he can determine how many times to execute PrivHab+. If  $A$  executes PrivHab+ enough times, using untruthful information and the attacks described in Section 5.2.1, he can completely uncover the area covered by  $H_B$  and its radius. Given that nodes always operate with encrypted data, there is no way for one part to tell apart a truthful execution of PrivHab+ from an untruthful one. However,  $B$  can decrease the effectiveness of these attacks by limiting the amount of interactions per unit of time with every other node.

When  $A$  is performing a series of untruthful executions to discover  $B$ 's habitat,  $A$  wants to know the result of the previous execution to improve the amount of obtained information in the next one. For example,  $A$  can start by selecting an evenly spread set of positions to try to discover the area covered by  $H_B$ . However, when  $A$  has found that there is a part of  $H_B$  inside the circle defined by one of these positions, it is much more useful to  $A$  to investigate this circle and its surroundings than continue with the remaining positions. Therefore,  $B$  can reduce the effectiveness of the attacker by taking the countermeasure of forcing  $A$  to send him at once the information needed to perform all the executions before sending any response.

Finally, when combining the two proposed measures, limiting the amount of executions per unit of time, and requiring all the information at once before sending any results to  $A$ , the effectivity of an active attack becomes greatly reduced, and the attacker ends learning almost the same things that he would learn by being truthful. Besides, the information protected by PrivHab+, the habitat, changes periodically. For this reason, slowing enough an attack can be considered equivalently as avoiding it, because when time passes the habitats change and the first things learned by the attacker become obsolete.

## 6. Experimental results

In this section we present some details about the proof-of-concept we have implemented. Then, we provide measurements of the computational and communication overhead introduced by the presented protocol.

**Table 8**

Percentage of the execution time spent in every operation. The communicational overhead is negligible and almost all the overhead introduced is computational. Note that rows add more than 100% because the computation of steps 0 and 1 is done asynchronously and it is not taken into account to calculate the execution time of PrivHab+.

Device	Key length (bits)	Steps 0 and 1 computation	Step 2 computation	Step 3 computation	Step 4 computation	Sending messages
Raspberry Pi	512	13.54%	27.15%	61.01%	11.33%	0.51%
	1024	13.79%	26.13%	62.07%	11.48%	0.32%
	2048	16.87%	30.56%	56.12%	13.08%	0.24%
i5 Laptop	512	11.06%	35.03%	54.58%	9.18%	1.21%
	1024	12.58%	31.11%	58.89%	9.69%	0.31%
	2048	13.29%	26.71%	61.18%	12.05%	0.06%

### 6.1. Implementation details

We have deployed an implementation of the presented protocol on two different sets of devices: three Raspberry Pi boards<sup>19</sup>, and two i5 laptops<sup>20</sup>. The Raspberry Pi boards are very cheap low-end devices, ideals to deploy a cheap prototype network that will allow us to run field experiments in a near future. The laptops have been chosen as representatives of short-term high-end mobile devices, indeed the i5 processor slightly outperforms the iPhone 6' A8, the most powerful mobile phone processor prior to the writing of this article. The objective of this proof-of-concept implementation is to demonstrate the viability of the proposal, and to obtain a measure of the overhead that PrivHab+ adds to every transaction.

### 6.2. Results obtained

We have established a DTN network using the chosen devices and we have used this implementation to send 500 messages of sizes between 10MB and 20MB. We have repeated the tests five times, using Paillier's length keys of 512, 1024 and 2048 bits. We have measured the average time needed to make the calculations and to exchange all messages of Fig. 9. The obtained results are shown in Table 9, and have been incorporated to the simulations.

**Table 9**

Execution time of PrivHab+ to route one message in both devices, the Raspberry Pi and the i5 Laptop, using different key lengths. The overhead is calculated as the extra amount of time needed to send a message of 10 MB or 20 MB.

Device	Key length (bits)	Time (ms)	Overhead 10 MB (%)	Overhead 20 MB (%)
Raspberry Pi	512	783.95	4.74	2.42
	1024	5,487.94	33.21	16.94
	2048	34,244.12	207.26	105.72
i5 Laptop	512	20.58	0.12	0.06
	1024	118.91	0.71	0.36
	2048	755.54	4.57	2.33

As can be seen in Table 9, PrivHab+ execution time depends heavily on the key length used. When using keys of 512 bits, PrivHab+ can be executed by a low-end device in less than a second, meaning an overhead of less than a 5% when sending messages larger than 10MB. The execution time increases to almost 5.5 s when using keys of 1024 bits. Given the average length of connectivity windows in remote village scenarios presented in [12], this overhead is acceptable.

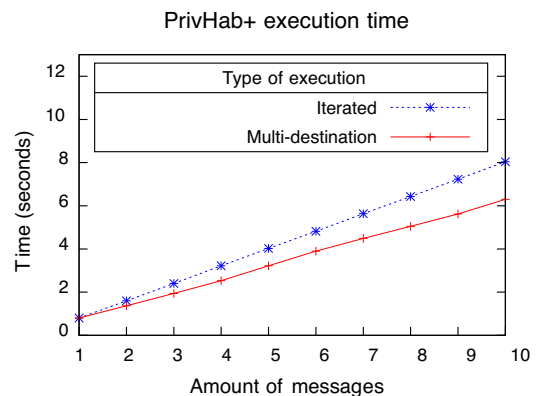
<sup>19</sup> Raspberry Pi Broadcom BCM2835 SoC full HD, 700 MHz Low Power ARM1176JZ-F, 512 MB SDRAM, 512 MB SD with Raspbian, equipped with a Wi-Pi Wireless Adapter (802.11n up to 150 Mbps), a GPS receiver NL-302U (baud rate: 4800 bauds) and a dual output 5000 mAh battery.

<sup>20</sup> Intel Core-i5 (third generation): dual core 3.3 GHz, 4 GB RAM, WiFi 802.11 b/g/n Dual Antenna, with Ubuntu 14.04 LTS, equipped with a GPS receiver NL-302U (baud rate: 4800 bauds).

The usage of keys of 2048 bits or more in low-end devices is discouraged because of the high overhead times they produce. In a high-end processor, PrivHab+ can be executed in less than a second even when using extra-large keys of 2048 bits. Due to this, the key length should be chosen keeping in mind the devices used and the time that can be spent by executing the protocol.

PrivHab+ can be executed once to simultaneously route all messages. This is called a multi-destination execution. This execution is faster but its result is all-or-nothing, meaning that no message can be routed if the connectivity window suddenly ends before finishing the execution of PrivHab+. In contrast, PrivHab+ can be executed to route one message at a time. This is called the iterated execution. This execution is slower, lasts 20% more time than the multi-destination execution, but when the communication suddenly ends, all previously processed messages have been routed. Fig. 11 depicts the time needed by PrivHab+ to execute the protocol when routing messages using both types of execution. The authors suggest to use a mixed strategy: using one multi-destination execution to route the first messages and then iterate each message one by one.

Finally, Table 8 shows the percentage of time consumed by each operation. The time needed to compute and send the first message, during steps 0 and 1, is not counted as a part of PrivHab+'s overhead because this message can be computed and sent asynchronously during the neighbour discovery phase, as explained in Section 4.5. As can be seen, the communicational overhead is quasi negligible, and most of the time is spent to compute the third message, at step 3. In fact, the computation of the third message is the most time-consuming operation because it includes decrypting the second message,



**Fig. 11.** Execution time between the two different strategies to execute PrivHab+ with multiple messages to send, in a Raspberry Pi using keys of 512 bits. Executing the whole protocol one time for each destination lasts around 20% more than performing one multi-destination execution.



**Fig. 12.** Map of a scenario of application located in a rural area of Cajamarca (Perú). White lines are natural obstacles approximate limits. Dotted white lines represent the pathways where messages sent from the village of Chota to Cutervo or to Huambos have to be routed through. The size of the area under consideration is  $30 \times 30$  km.



**Fig. 13.** Map of a scenario of application located in Gwanda (Zimbabwe). White lines are natural obstacles approximate limits. Dotted white lines represent the pathways where messages sent from the InfoCenter of Gwanda to Sablevale and the two farm's zones have to be routed through.

calculating the distance between the habitat and the destination, and calculating the results operating with cyphered operands.

## 7. Simulations

In this section we explain the two scenarios we have chosen to evaluate PrivHab+'s performance, and how we have modelled and simulated it. Afterwards, we provide the obtained results of both scenarios, comparing PrivHab+ performance and characteristics with other popular DTN routing algorithms. Finally, we provide a qualitative comparison with all other evaluated routing protocols.

### 7.1. First scenario: podcasts distribution in Cajamarca

To carry out the first set of simulations, we have chosen a podcasts distribution scenario located in the Cajamarca region, in Perú, where the NGO *Practical Action*<sup>21</sup> records podcast radio programmes targeted to farmers in Compact Discs and physically distributes them to the local radio stations. The scenario consists of an NGO office located in the village of Chota that distributes radio podcast programs to two NGO's local radio stations located in the villages of Huambos and Cutervo. We substituted the physical distribution method by a digital and automated one using DTN networking. The podcasts are distributed through an opportunistic network. This application has to deal with challenges like a sparse population, with the receivers of the information far away from each other, a rugged terrain and a lack of data communication networks.

This scenario has been chosen because its characteristics make it ideal to evaluate the performance of a geographic routing protocol. Firstly, the area, shown in Fig. 12, is full of mountains that restrict the movement of the nodes, so short-term movement information as the speed vector of a node is not useful to route messages. Secondly, due to the movement patterns of nodes there are pairs of nodes whose probability of encounter is almost zero. These nodes are forced to use intermediate nodes to carry their messages towards its destination. Besides, it is based on a real application of DTN networking placed in an environment that lacks network infrastructure, where a solution based in the usage of small and cheap devices would be viable.

### 7.2. Second scenario: podcasts distribution in Gwanda

To carry out the second set of simulations, we have chosen another podcasts distribution scenario located in Gwanda, in Zimbabwe. Due

to the success of their initiative in other rural areas, the NGO *Practical Action*<sup>22</sup> use a manpower of 60 cooperators to bring the podcasts to the villagers. The poor radio signal of the area makes unusable the approach of recording CDs and distributing it to the local radio stations. Therefore, the cooperators, equipped with portable MP3 players and speakers, have to physically travel to the NGO office to obtain new podcasts. The scenario consists of an NGO office located in the village of Gwanda that distributes radio podcast programs to five cooperators that roam around Gwanda, the village of Sablevale and the two main farm's zones near Gwanda. We implemented a digital and automated distribution method that distributes the podcasts through an opportunistic network. This application has to deal with challenges like a sparse population, mobile receivers of the information, and a lack of data communication networks.

This scenario has been chosen to evaluate the performance of PrivHab+ because it has some characteristics different than the previous one. The area is smaller than the Cajamarca's one ( $15 \times 7$  km) and, as shown in Fig. 13, the main physical obstacle that restricts the movement of the nodes is the Mtshabezi River. Besides, the density of nodes is higher, and there are five different mobile destinations, although the NGO knows the approximated zone where they are assigned. As there are more destinations than in the Cajamarca scenario, and nodes are very unlikely to be useful to deliver messages to more than one destination. Therefore, there are more nodes whose usefulness to deliver messages to certain destinations is almost zero, and a good decision making is critical to obtain a good performance.

### 7.3. Characteristics of the application

The application we consider in these two scenarios is a podcast distribution application based on the needs of the NGO *Practical Action*. This NGO already has a manpower of cooperators devoted to distributing the podcasts physically in the two explained scenarios. Therefore, we assume that it could be easy to assign one cheap device to every cooperator. This way, *Practical Action* could transform its manpower of cooperators into a Delay Tolerant Network of mobile nodes.

One can think that a cooperator that has been assigned by the NGO to a certain area, and that has received a device from the NGO in order to distribute the podcasts in that area, may not be very concerned about the privacy of its habitat or the amount of buffer occupied by the podcasts. However, if the NGO wants to extend the network cheaply by adding other types of nodes, e.g. volunteers that want to help the NGO, there are two characteristics of PrivHab+ that can make it more useful than other DTN routing solutions: 1) PrivHab+

<sup>21</sup> More information about this programme at <http://practicalaction.org/podcasting-3>

<sup>22</sup> More information about this programme at <http://practicalaction.org/podcasting-gwanda>

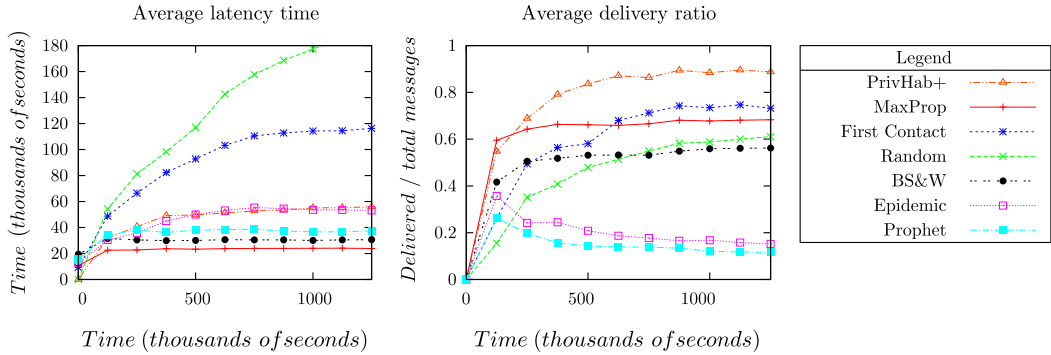


Fig. 14. Obtained results in terms of latency and delivery ratio in the Cajamarca scenario. PrivHab+ and MaxProp perform far better than the rest, obtaining a low latency and a high delivery ratio.

protects the privacy of its users; and 2) PrivHab+ can achieve a good performance occupying a small buffer.

A volunteer could just install an app on his PDA to become part of the network. This way, he could help the podcast distribution by simply carrying his mobile device in the pocket when he performs his daily routine. Given that hurting people's privacy do not seem a good way to incentivise them to install an app, it is important that PrivHab+ guarantees their privacy. The same way, we cannot expect users to renounce to a big part of their storage capacity to carry podcasts because they probably want to continue using their devices normally. As a high usage of resources will give the users reasons for leaving the network, it is desirable to reduce as much as possible the impact on the users' devices. Therefore, it is useful that PrivHab+ is capable of achieving a good performance even using small storage buffers.

#### 7.4. Simulation details

In our interpretation of these scenarios, nodes implement a mobility pattern that takes into account home and work locations. Nodes have a 200 MB buffer and a wireless interface featuring a communication range of 30 m and speed up to 500 kbps. Messages of 10–20 MB<sup>23</sup> are injected periodically in the network by the NGO office, who knows the location, exact on the first scenario, approximated on the second one, of the waypoints and the destinations. The type and the amount of nodes simulated in each scenario are shown in Table 10.

**Table 10**  
Number and type of the nodes involved in the simulations of each scenario.

Number and type of nodes	Scenario	
	Cajamarca	Gwanda
Total	95	66
Source	1 static	1 static
Destination	2 static	5 mobile
Other	92 mobile	60 mobile

During the approximation phase nodes calculate their habitat as explained in Section 3, and the protocol detailed in Section 4.5 is used to make the routing decisions. For the sake of simplicity, nodes implementing PrivHab+ use a direct single-copy forwarding policy. During the delivery phase, nodes use direct delivery to give the messages

to their destination. We have modelled the computational and communication overhead introduced by PrivHab+ considering that nodes need 5.5 additional seconds to perform each transaction. This overhead time is based on real experimentation, it is the average time consumed by a Raspberry Pi board using a 1024 bits key.

In both scenarios, we have compared the performance obtained by PrivHab+ (using  $T = 48$  h on the first scenario,  $T = 24$  on the second one, and  $\beta = 60$  on both) with a bench-mark of well-known routing protocols used in [32]: Prophet [27], Binary Spray & Wait ( $L = 40$ ) [38], Epidemic and Random [39]. We have added two routing protocols to this set: MaxProp [5] and First Contact. All simulations have been performed using *The Opportunistic Network Simulator* (The ONE) [21], and have been repeated twenty times using different random seeds, then, the average results of the twenty repetitions have been calculated.

#### 7.5. Simulation results: Cajamarca

Results obtained on the first scenario are shown in Fig. 14, where the performance of all the compared protocols in terms of delivery ratio and latency is depicted. Single-copy protocols, as First Contact and Random obtain a medium-to-high delivery ratio because they do not face most of the problems related to the size of the buffers and nodes' congestion. In contrast, their latency is high. Random's decision making is equally likely to make a bad or a good choice at every relay, but the latter ones are far more rare and valuable. First Contact performs slightly better because it avoids loops and forces messages to move away from their origin after they have visited all the near neighbours. Flooding-based protocols, as Epidemic and Prophet, obtain low latencies but also low delivery ratios. These protocols fill the buffers early and force nodes to drop messages. Most messages are dropped before reaching to its destination, but the ones that are not dropped arrive fast. MaxProp, also a flooding-based protocol, obtains a low latency and a good delivery ratio because of its better dropping policy based on probabilities of delivery. BS&W has a replication-based approach that limits flooding and performs a sort of depth-spread. BS&W performs similar to MaxProp in terms of latency, but obtain a medium delivery ratio because of its lack of a dropping policy that avoids dropping messages near their destination. Finally, PrivHab+ obtains the highest delivery ratio thanks to the quality of its decision making. PrivHab+ takes the best decisions because it is the only one that takes into account both the location of the destination and the mobility patterns of the neighbours. Even with the drawback of using a single-copy forwarding policy, PrivHab+'s obtains a very low latency that is only slightly improved by flooding-based protocols that obtain lower delivery ratios.

<sup>23</sup> This is the size of an audio file with ID3 version 2.4.0, extended header, contains: MPEG ADTS, layer III, v1, 128 kbps, 44.1 kHz, stereo, with a duration between 10 and 20 min.

**Table 11**

Obtained results in terms of network overhead, amount of dropped messages, aborted relays and hops performed by the delivered messages. Single-copy protocols like PrivHab+ and First Contact are the ones that waste fewer network resources.

Protocol	Dropped messages	Overhead	Aborted relays	Hops
Epidemic	197,030	964.66%	114,380	26.67
Prophet	130,647	855.96%	382,557	13.95
Maxprop	9929	65.91%	252,023	11.21
BS&W	33,373	36.66%	114,380	9.50
Random	396	112.40%	375,200	180.13
First Contact	75	46.73%	217,280	59.54
PrivHab+	128	9.68%	51,343	8.46

Table 11 shows the average number of aborted relays, dropped messages, hops performed by the delivered messages, and the network overhead (calculated as the relation between the number of the relays done and the number of delivered messages). A low network overhead is desirable in scenarios where the resources are constrained. Reducing the number of relays saves battery and increases the amount of time nodes are operational, improving this way the performance of the whole network.

Epidemic and Prophet generate an enormous overhead of around one thousand percent that means that almost all nodes effort while forwarding messages is wasted, because the forwarded messages will probably be dropped before being delivered to their destination. Besides, Epidemic force messages to pass through a high number of intermediate hops after arriving its destination, causing a higher latency. MaxProp and Binary Spray & Wait (BS&W) generate a smaller amount of dropped messages and a lesser network overhead. These two protocols try to compensate their poor decision making by generating copies. Creating copies fills the buffers and consumes a lot of energy, but these two protocols create copies in a clever way than Epidemic and Prophet, consume fewer resources and need a lesser number of hops to obtain better results. Between them, MaxProp better delivery ratio can be explained because it spreads messages in a more equitable way through the network than BS&W. Note that MaxProp manages to drop less than a half of messages than BS&W and needs almost two average hops less to reach each message's destination. Random and First Contact reduce highly the amount of messages dropped because do not flood the network with copies. However, their network overhead is also high because the majority of their relays are bad choices. Note that their number of hops and aborted relays is really high because messages spend a lot of time being relayed to nodes that will not approach them to its destination. Finally, PrivHab+ generates the smallest amount of dropped messages and

the lowest network overhead because PrivHab+'s routing decisions are much better than the decisions taken by all other protocols.

### 7.6. Simulations: Gwanda

Results obtained on the second scenario are shown in Fig. 15, where the performance of all the compared protocols in terms of delivery ratio and latency is depicted. In comparison with the results of Fig. 14 of the previous scenario, we can identify three main differences.

The first difference is that latencies obtained by all protocols are around a 50% lower. The reason is that physical distances in the Gwanda scenario are smaller. As a consequence, messages have to spend less time being carried by a node from one village to another.

The second difference is that two flooding-based protocols as Epidemic and Prophet, that ranked 3rd and 4th in the Cajamarca scenario in terms of latency, perform a little worse in this scenario. Both protocols are unable to tell the not useful relays apart from the useful ones. For this reason, they are harmed by the higher amount of nodes that are not useful to deliver messages to certain destinations. PrivHab+'s ability to identify useful relays through the comparison of habitats has benefited from this circumstance to obtain a lower latency (ranking 3rd).

Finally, the third difference is the lower delivery ratio of First Contact, Random and BS&W. The density of nodes is higher, so First Contact and Random have to make more routing decisions, and they usually make the wrong one. BS&W decreased delivery ratio is a consequence of the big share of created copies that are forwarded to the higher amount of not useful nodes. The rest of the results obtained are similar between the two scenarios. PrivHab+ low latency is only slightly improved by replication-based protocols like BS&W and MaxProp. However, in terms of delivery ratio, PrivHab+ greatly outperforms all other compared protocols, specially Epidemic, BS&W and Prophet.

Table 13 shows the average number of aborted relays, dropped messages, hops performed by the delivered messages, and the network overhead introduced by each protocol. As in the Cajamarca scenario, Epidemic and Prophet generate an enormous overhead. This means that almost all nodes effort while forwarding messages is wasted, because most of the forwarded messages are dropped before being delivered to their destination. The decreased efficiency of BS&W in this scenario is reflected in the introduced network overhead and in the number of hops. In this scenario, both values are higher than MaxProp's. Note that MaxProp's number of hops is the smallest one, but its delivery ratio it's not the best. The reason is that sometimes MaxProp does not forward messages to nodes with low probabilities of encounter (because they never met the

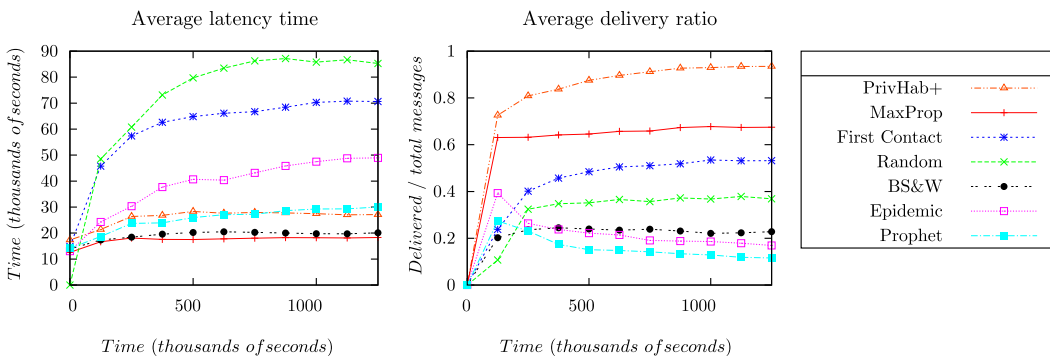


Fig. 15. Obtained results in terms of latency and delivery ratio in the Gwanda scenario. PrivHab+ and MaxProp perform far better than the rest, obtaining a low latency and a high delivery ratio.

**Table 12**

Feature comparison of all the routing protocols. MaxProp and PrivHab+ have the best performance marks, but PrivHab+, with less overhead, privacy respectful and a constant complexity instead of a linear one, has a set of characteristics that make it better in scenarios like the two we have studied.

Protocol	PrivHab+	MaxProp	BS&W	Prophet	Epidemic	First Contact	Random
Delivery ratio	Very high	High	Low	Very low	Very low	Medium	Low
Latency	Low	Very low	Very low	Low	Medium	High	Very high
Network overhead	Very low	Medium	Medium	Very high	Very high	Low	Medium
Nodes' privacy	Protected	Violated	Not considered	Violated	Not considered	Not considered	Not considered
Protocol's complexity	Constant	Linear	Constant	Linear	Constant	Constant	Constant
Suitability to reach hop-distant destinations	High	High	Low	Very low	Very low	Medium	Very low

**Table 13**

Obtained results in terms of network overhead, amount of dropped messages, aborted relays and hops performed by the delivered messages. Single-copy protocols like PrivHab+ and First Contact are the ones that waste fewer network resources.

Protocol	Dropped messages	Overhead	Aborted relays	Hops
Epidemic	249,740	1089.53%	486,253	18.57
Prophet	156,716	957.98%	453,219	9.85
Maxprop	15,910	86.69%	322,832	6.35
BS&W	37,927	101.50%	122,217	13.46
Random	939	191.910%	324,955	149.21
First Contact	692	62.06%	168,085	41.45
PrivHab+	82	8.51%	43,839	7.41

destination before) that are good choices because of their habitats. PrivHab+ recognise this nodes and use them to carry the messages, and this way it achieves a higher delivery ratio. Random and First Contact drop a small amount of messages because they do not flood the network with copies, but their overhead and number of hops are also high because the majority of their relays are bad choices. Finally, PrivHab+ generates the smallest amount of dropped messages and the lowest network overhead because PrivHab+'s routing decisions are much better than the decisions taken by all other protocols.

The small network overhead produced by PrivHab+ could allow users to use the same devices to run other applications like e-mail, voice messaging, blog-style publications, etc. Note that, being PrivHab+ the protocol with the higher computational overhead (5.5 s), it is also the one with the lowest amount of aborted relays. In fact, PrivHab+ takes better routing decisions. This reduces the total number of relays needed to deliver a message to its destination and the time that messages last in the network. As a consequence, nodes carry less messages and can forward all of them before the opportunistic contacts end. Therefore, we can state that the computational and communication overhead introduced by PrivHab+ is perfectly assumable because it is compensated by its better decision making, improving the performance of the network.

### 7.7. Qualitative comparison

Table 12 summarises the whole comparison between all protocols. In addition to those already mentioned, delivery ratio, latency and network overhead; we also take into consideration nodes' privacy, the protocol's complexity, and the suitability to reach hop-distant destinations. Delivery ratio, latency and network overhead are the main performance indicators of a routing protocol. The importance of privacy has been discussed before. The protocol's complexity could be important while using small devices and the number of nodes in the network grows. The suitability to reach hop-distant destinations is a capital aspect in scenarios where messages have to be forwarded many times due to the long distances between the source and the destination.

Nodes' privacy is protected by PrivHab+, which is the only one that uses private information in a secure manner. Privacy is obviously not considered by the protocols that do not use node-related information to make choices. Besides, it is heavily violated by Prophet and MaxProp while nodes exchange their likelihood to contact others

without protecting it. Furthermore, security of these two protocols cannot be easily enhanced, because they need to flood the network with a private information that is the basis of their operation.

The complexity of PrivHab+, BS&W, Epidemic, First Contact and Random is constant. These protocols need to perform always the same amount of operations to make a routing decision. MaxProp and Prophet need to update and compare an amount of probabilities that grow linear with the number of nodes of the network. When operating in networks with lots of nodes, both probabilistic protocols have to limit the amount of encounter probabilities they store, decreasing this way their performance.

Finally, in big scenarios where destinations are distant and messages have to be carried by many nodes, flooding-based protocols become poor routing protocols because they tend to congest the nodes that are nearest to the origin. This is what happens with Prophet and Epidemic. BS&W is slightly better because it avoids creating all the copies near the source node. First Contact is better than Random because, eventually, the message moves away from the origin, but both does it slowly anyway. The transitivity of probabilities makes MaxProp perform well in this circumstance. However, as nodes that are far away in terms of hops are very likely to be far away too in terms of geographic distance, PrivHab+ is the most suitable routing protocol for delivering messages to hop-distant destinations because it is designed to make messages travel distances towards their destination.

PrivHab+ decision making is based on the comparison of habitats. For this reason, it requires the scenario to be big enough to benefit from a geographic routing approach, and it is only useful when the movement patterns of the nodes constitute some kind of routine. When this happens, this decision making allows PrivHab+ to deliver more messages to their destination, even when using a single-copy forwarding policy. Besides, in these scenarios PrivHab+ performs faster than all other protocols except BS&W and MaxProp and consumes far less network resources. Moreover, it preserves nodes' privacy and performs really well when the number of nodes is high and the destinations of the messages are distant. Finally, PrivHab+ is efficient enough to be executed in small and cheap devices and the overhead that introduces is compensated by the quality of the routing decisions it makes, improving the performance of the network.

## 8. Conclusions

We have defined an elliptic model of habitat. The habitat models node's whereabouts based on the idea of exploiting life-cycles. The habitat is useful to compare the intermediate nodes to decide who is a better choice to carry each message towards its destination. We have presented PrivHab+, a secure geographical DTN routing protocol that uses the habitat to make routing decisions. PrivHab+ takes advantage of TaxiCab geometry and makes use of homomorphic cryptography techniques to preserve the privacy of the participants while comparing the habitats of the candidate nodes.

PrivHab+ has been analysed as a secure multi-party computation to prove that the protocol is secure. The only knowledge that can be learned by each participant about others intimacy is the same that could be inferred from the output of the protocol. This is an important point that makes PrivHab+ recommendable to use in scenarios



where nodes are so related, directly or indirectly, to a person that their privacy needs to be protected.

We have developed a proof-of-concept implementation that demonstrates that the presented protocol is viable and that it can be executed on small devices with a good performance. Both the computation and the communication overhead introduced by PrivHab+ is proven to be affordable and to not degrade the performance of the network. Besides, simulations based on two podcast distribution scenarios have shown that PrivHab+ performs better than a set of well known DTN routing protocols and minimises the network overhead. The qualitative comparison between PrivHab+ and the other routing protocols shows that PrivHab+ is a good choice not only for this two scenarios. In fact, PrivHab+ is a good choice in any DTN scenario where nodes are linked to people, where mobility patterns are routinary, and where the considered distances are high, forcing the need of lots of hops to reach each destination.

As future lines of research, we plan to study the impact of different forwarding policies on the performance of PrivHab+, to improve the elliptic model of habitat using a more complex representation, that does not have to be necessarily a geometric figure, and to develop an enhanced version of PrivHab+ that compares simultaneously three or more habitats. We also plan to study the performance of PrivHab+ in different scenarios and to present more real applications that could benefit from this secure geographic routing protocol.

## Acknowledgments

The authors wish to thank Gerard García Vandellós for his work in the implementation of the proof-of-concept software and the insights he provided that helped to improve the presented protocol.

This work has been partially funded by the Ministry of Science and Innovation of Spain, under the reference project TIN2014-55243-P, by the Catalan Government under the reference project 2014SGR691 and by the Autonomous University of Barcelona under the reference number 472-03-01/2012.

## References

- [1] Y. Li 0003, T.-H. Lai, M.T. Liu, M.-T. Sun, J. Yang, DTGR: Disruption-tolerant geographic routing for wireless ad hoc networks, *Simulation* 82 (6) (2006) 399–411.
- [2] C. Boldrini, M. Conti, J. Jacopini, A. Passarella, Hibop: a history based routing protocol for opportunistic networks, in: Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007, 2007, pp. 1–12, doi:10.1109/WOWMOM.2007.4351716.
- [3] C. Borrego, S. Castillo, S. Robles, Striving for sensing: Taming your mobile code to share a robot sensor network, *Inf. Sci.* (2014), <http://dx.doi.org/10.1016/j.ins.2014.02.072>.
- [4] A. Boukerche, K. El-Khatib, L. Xu, L. Korba, Sdar: a secure distributed anonymous routing protocol for wireless and mobile ad hoc networks, in: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, 2004, pp. 618–624, doi:10.1109/LCN.2004.109.
- [5] J. Burgess, B. Gallagher, D. Jensen, B. Levine, Maxprop: Routing for vehicle-based disruption-tolerant networks, in: Proceedings of the 25th IEEE International Conference on Computer Communications. INFOCOM, 2006, pp. 1–11, doi:10.1109/INFOCOM.2006.228.
- [6] P.-C. Cheng, K. Lee, M. Gerla, J. Härrri, Geodtn+nav: Geographic DTN routing with navigator prediction for urban vehicular environments, *Mobile Netw. Appl.* 15 (1) (2010) 61–82, doi:10.1007/s11036-009-0181-6.
- [7] Y.-C. Hu, A. Perrig, D.B. Johnson, Ariadne: a secure on-demand routing protocol for ad hoc networks, *Wirel. Netw.* 11 (2005) 1–2 (January 2005), 21–38. DOI=<http://dx.doi.org/10.1007/s11276-004-4744-y>.
- [8] K.B. Frikken, *Algorithms and Theory of Computation Handbook*, Chapman & Hall/CRC, 2010.14–14
- [9] G. Garcia, S. Robles, A. Sánchez, C. Borrego, Information system for supporting location-based routing protocols, *Actas de la XIII Reunión Española sobre Criptología y Seguridad de la Información, Universidad de Alicante, Universidad de Alicante*, 2014 September, pp. 203–208. ISBN: 978-84-9717-323-0.
- [10] C. Gentry, A fully homomorphic encryption scheme, Stanford University, 2009 Ph.D. thesis.
- [11] Goldreich, O. (1998). Secure multi-party computation. Manuscript. Preliminary version. <http://www.wisdom.weizmann.ac.il/~oded/PSX/prot.pdf>.
- [12] S. Grasic, A. Lindgren, Revisiting a remote village scenario and its DTN routing objective, *Comput. Commun.* 48 (2014) 133–140, doi:10.1016/j.comcom.2014.04.003.
- [13] S. Guo, M.H. Falaki, E.A. Oliver, S. Ur Rahman, A. Seth, M.A. Zaharia, S. Keshav, Very low-cost internet access using kiosknet, *SIGCOMM Comput. Commun. Rev.* 37 (5) (2007) 95–100, doi:10.1145/1290168.1290181.
- [14] W. Hsu, D. Dutta, A. Helmy, CSI: A paradigm for behavior-oriented delivery services in mobile human networks, *CoRR* (2008).arXiv: abs/0807.1153
- [15] Y.-C. Hu, D.B. Johnson, A. Perrig, SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks, in: Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '02). IEEE Computer Society, Washington, DC, USA, 2002 3-.
- [16] P. Hui, J. Crowcroft, E. Yoneki, Bubble rap: Social-based forwarding in delay-tolerant networks, *IEEE Trans. Mobile Comput.* 10 (11) (2011) 1576–1589, doi:10.1109/TMC.2010.246.
- [17] R. Jiang, Y. Xing, Anonymous on-demand routing and secure checking of traffic forwarding for mobile ad hoc networks, in: Proceedings of the IEEE 31st Symposium on Reliable Distributed Systems (SRDS), 2012, pp. 406–411, doi:10.1109/SRDS.2012.6.
- [18] C. Karlof, D. Wagner, Secure routing in wireless sensor networks: Attacks and countermeasures, *Ad hoc networks* 1 (2) (2003) 293–315.
- [19] B. Karp, H.T. Kung, Gpsr: Greedy perimeter stateless routing for wireless mobile, in: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, in: MobiCom '00, ACM, New York, NY, USA, 2000, pp. 243–254, doi:10.1145/345910.345953.
- [20] A. Kate, G. Zaverucha, U. Hengartner, Anonymity and security in delay tolerant networks, in: Proceedings of the Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on, 2007, pp. 504–513, doi:10.1109/SECCOM.2007.4550373.
- [21] A. Keränen, J. Ott, T. Kärkkäinen, The ONE Simulator for DTN Protocol Evaluation, in: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, SIMUTools '09., ICST, New York, NY, USA, 2009.
- [22] E. Krause, *Taxicab Geometry: An Adventure in Non-Euclidean Geometry*, Dover Publications, New York, 1987.
- [23] E. Kuiper, S. Nadjm-Tehrani, Geographical routing in intermittently connected ad hoc networks, in: Proceedings of the Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008. 22nd International Conference on, 2008, pp. 1690–1695, doi:10.1109/WAINA.2008.132.
- [24] E. Kuiper, S. Nadjm-Tehrani, Geographical routing with location service in intermittently connected manets, *IEEE Trans. Veh. Technol.* 60 (2) (2011) 592–604, doi:10.1109TVT.2010.2091658.
- [25] J. LeBrun, C.-N. Chuah, D. Ghosal, M. Zhang, Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks, in: Proceedings of the IEEE 61st Vehicular Technology Conference, 2005. VTC 2005-Spring, 2005, vol. 4, 2005, pp. 2289–2293 Vol. 4, doi:10.1109/VETEC5.2005.1543743.
- [26] J. Leguay, T. Friedman, V. Conan, Evaluating mobspace-based routing strategies in delay-tolerant networks, *Wirel. Commun. Mobile Comput.* 7 (10) (2007) 1171–1182, doi:10.1002/wcm.520.
- [27] A. Lindgren, A. Doria, O. Schelén, Probabilistic routing in intermittently connected networks, *SIGMOBILE Mob. Comput. Commun. Rev.* 7 (3) (2003) 19–20, doi:10.1145/961268.961272.
- [28] M. Liskov, R. Silverman, *A Statistical Limited-Knowledge Proof for Secure RSA Keys*, Technical Report, IEE P1363 working group, 1998.
- [29] X. Lu, P. Hui, D. Towsley, J. Pu, Z. Xiong, Anti-localization anonymous routing for delay tolerant network, *Comput. Netw.* 54 (11) (2010) 1899–1910. <http://dx.doi.org/10.1016/j.comnet.2010.03.002>.
- [30] M. Mahmoud, X. Shen, Lightweight privacy-preserving routing and incentive protocol for hybrid ad hoc wireless network, in: Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2011, pp. 1006–1011, doi:10.1109/INFOCOM.2011.5928774.
- [31] A. Mei, G. Morabito, P. Santi, J. Stefa, Social-aware stateless forwarding in pocket switched networks, in: Proceedings of the IEEE INFOCOM, 2011, pp. 251–255, doi:10.1109/INFOCOM.2011.5935076.
- [32] M. Musolesi, C. Mascolo, Car: Context-aware adaptive routing for delay-tolerant mobile networks, *IEEE Trans. Mobile Comput.* 8 (2) (2009) 246–260, doi:10.1109/TMC.2008.107.
- [33] H. Rifà-Pous, J. Herrera-Joancomartí, Secure dynamic manet on-demand (SEDYMO) routing protocol, in: Communication Networks and Services Research (CNSR), IEEE Computer Society, Los Alamitos, CA, USA, 2007, pp. 372–380, doi:10.1109/CNSR.2007.57.
- [34] A. Sánchez-Carmona, S. Robles, C. Borrego, Privhab: a multiagent secure georouting protocol for podcast distribution on disconnected areas, in: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4–8, 2015, 2015, pp. 1697–1698.
- [35] A. Sánchez-Carmona, S. Robles, C. Borrego, G. Garcia-Vandellós, Privhab: A multiagent secure georouting protocol for distributing podcasts in disconnected areas, in: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4–8, 2015, 2015b, pp. 1943–1944.
- [36] K. Scott, S. Burleigh, Bundle Protocol Specification, 2007, (RFC 5050 (Experimental)).
- [37] J.-H. Song, V.W. Wong, V.C. Leung, Secure position-based routing protocol for mobile ad hoc networks, *Ad Hoc Netw.* 5 (1) (2007) 76–85, <http://dx.doi.org/10.1016/j.adhoc.2006.05.010>. Security Issues in Sensor and Ad Hoc Networks
- [38] T. Spyropoulos, K. Psounis, C. Raghavendra, Spray and wait: an efficient routing scheme for intermittently connected mobile networks, in: Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking, ACM, 2005, p. 259.

- [39] T. Spyropoulos, R.N. Rais, T. Turletti, K. Obraczka, A. Vasilakos, Routing for disruption tolerant networks: taxonomy and design, *Wirel. Netw.* 16 (8) (2010) 2349–2370, doi:[10.1007/s11276-010-0276-9](https://doi.org/10.1007/s11276-010-0276-9).
- [40] J. Su, J. Scott, P. Hui, J. Crowcroft, E. Lara, C. Diot, A. Goel, M. Lim, E. Upton, Huggle: seamless networking for mobile applications, in: J. Krumm, G. Abowd, A. Seneviratne, T. Strang (Eds.), *UbiComp 2007: Ubiquitous Computing*, Lecture Notes in Computer Science, 4717, Springer Berlin Heidelberg, 2007, pp. 391–408, doi:[10.1007/978-3-540-74853-3\\_23](https://doi.org/10.1007/978-3-540-74853-3_23).
- [41] X. Wu, B. Bhargava, Ao2p: ad hoc on-demand position-based private routing protocol, *IEEE Trans. Mobile Comput.* 4 (4) (2005) 335–348, doi:[10.1109/TMC.2005.50](https://doi.org/10.1109/TMC.2005.50).
- [42] G. Zhong, I. Goldberg, U. Hengartner, Louis, lester and pierre: three protocols for location privacy, in: N. Borisov, P. Golle (Eds.), *Privacy Enhancing Technologies*, Lecture Notes in Computer Science, 4776, Springer Berlin Heidelberg, 2007, pp. 62–76, doi:[10.1007/978-3-540-75551-7\\_5](https://doi.org/10.1007/978-3-540-75551-7_5).
- [43] B. Zhu, Z. Wan, M.S. Kankanhalli, F. Bao, R.H. Deng, *Anonymous secure routing in mobile ad-hoc networks*, in: *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, IEEE, 2004, pp. 102–108.



## Part III

### Discussion

*“The kind of tools that could have saved César’s job...”*



*“If you know the enemy and know yourself you need not fear the results of a hundred battles.”*

*The Art of War, SUN TZU*

# 7

## Results

**I**N this chapter, we provide a summary of the main systems and proposals presented in every article of the compendium. In order to avoid mere repetition from the publications, the systems are not explained with all detail here. Instead, we present and discuss every publications’ contribution, focusing on their relation with the main objective of this thesis.

### 7.1 Identity-based access control for pro-active messages DTN

Pro-active messages change the traditional approach to Opportunistic Network (OppNet) routing. Instead of using routing algorithms deployed in every node of the network, a pro-active message carries its own routing code. Nodes have to provide the necessary infrastructure to let the messages decide their way

towards their destination. This infrastructure consists of three different features: *a)* executing the mobile routing code carried by the messages; *b)* collecting contextual information that could be used by the routing code to make a decision; *c)* allowing the pro-active messages to use this information during routing.

In order to implement this features, it is needed to define a format for the messages to allow them to carry their routing code. Besides, it is necessary an access control mechanism for telling apart the messages that are authorised to access an information from the ones that are not. This access control system requires a way to identify every message and has to operate using only the cryptographic tools that are usable in an OppNet<sup>1</sup>.

### 7.1.1 A network of pro-active messages

In order to allow messages to carry their own routing code, the structure of a pro-active message must contain a minimum of four fields, as depicted in Figure 7.1:

1. **Source address:** the address of the node that sent the message.
2. **Destination address:** the address of the node where the message is intended.
3. **Content:** the data from the application.
4. **Routing Code:** mobile code that has to be executed in every node.

Source address	Destination address
Content	
Routing Code	

**Figure 7.1:** Fields of a pro-active message.

Every node should execute the routing code of a pro-active message. Routing code operates above the list of the current node's neighbours and the contextual information stored on the node, and it returns the subset of neighbours where the message has to be forwarded to.

<sup>1</sup> Due to the low connectivity ratios that are typical in an OppNet, the tools based on the Public Key Infrastructure (PKI) become unusable [Bhu16].

### 7.1.2 Pro-active messages' identity

When a message tries to access a contextual information, it is necessary to identify and authenticate that message to decide if it is authorised to access it. To do this, the first thing needed is to establish *what* defines the identity of a pro-active message.

We consider that the routing code of a pro-active messages defines its identity. As the content of a message is meaningless during routing, and the source and destination addresses are arbitrary, this is the only reasonable option. Besides, this is something that cannot be stolen and used for malicious purposes, because authorised routing codes, by definition, do not perform malicious actions over the information.

Using a simile with common identity-based access control system, the pro-active message's behaviour is used to identify and authenticate it. This way, the message do not need to *know* or *have* or *do* anything; it is authenticated for what it *is*.

In order to decide about access control, it is desirable to manage fixed-size elements, but routing code length is arbitrary. For this reason, two hash functions are applied to the routing code to obtain the binary sequence that identifies the pro-active message and, at the same time, differentiates it from any other. Note that, from this point on, two messages that behave the same way can be considered equivalents from an access control perspective.

### 7.1.3 Identity-based access control

The Identity-based access control system defines two sets of data and how to operate them to grant (or deny) the access to a piece of information.

#### The Authorized Hashes Set and the Entries Set

The two data sets are the Authorized Hashes Set (*AHS*) and the Entries Set (*ES*). The first one defines the identities of the messages that are authorised to access every piece of information, and the second one contains the information itself.



The Authorized Hashes Set contains a collection of triplets according to the structure of Equation 7.1.

$$(j, h'(c_i), E_{h(c_i)}(k_j)) \quad (7.1)$$

Where:

- $j$  identifies the information.
- $h'(c_i)$  identifies the routing code  $c_i$  of a message that can access the information.
- $E_{h(c_i)}(k_j)$  is the symmetric key needed to decrypt the information. It is cyphered with a symmetric key encryption algorithm  $E$  using the result of a hash function  $h$  over the routing code as key.

The Entries Set is a collection of pairs as according to the structure of Equation 7.2.

$$(j, E_{k_j}(I_j)) \quad (7.2)$$

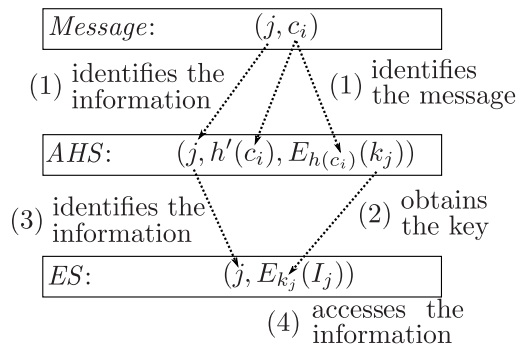
Where:

- $j$  identifies the information.
- $E_{k_j}(I_j)$  is the information itself, cyphered with  $E$  using the key  $k_j$  that is stored in the *AHS*.

### Operation of the Identity-based access control system

The Identity-based access control aims to provide an access control system that fits in a pro-active messages OppNet. It is based on a discretionary policy, and it is designed loosely following the guidelines of an access control list. This approach consists of associating each system's object with all the authorised subjects that can access it and the actions they can perform on the object. Concretely, instead of relating the name of every subject with its permissions, the system associates the identifier of each subject to the key needed to decrypt the information.

The operation's schema of the Identity-based access control is depicted in Figure 7.2. In the first place (1), a hash ( $h'$ ) of the routing code of the message ( $c_i$ ) and the identifier of the information ( $j$ ) it wants to access are used to locate the appropriate entry at the *AHS*. Then (2), another hash function ( $h$ ) is applied to the routing code to decrypt the key ( $k$ ) used to cypher the information. Next (3), the identifier of the information is used to locate the appropriate entry at the *ES*. Finally (4), the key  $k$  is used to decrypt the information ( $I$ ). If the information is successfully decrypted, the routing code can access it and modify it.



**Figure 7.2:** Inside the boxes, the message whose routing code tries to access a piece of information, and the corresponding entries at the *AHS* and the *ES*. The numbers between parenthesis indicate the order of the operations, the arrows indicate which fields are related, and the text explains how the process to access the requested information is conducted.

Using the Identity-based access control system not only provides security against a hypothetical attacker that alters a pro-active message's routing code to access or modify some information, but it also allows nodes to spread updates of routing information securely because the information can be transmitted while cyphered.

#### 7.1.4 Main contributions

Next, we summarise the contributions of *Identity-based access control for pro-active messages DTN* to the research of access control system under the scope of Opportunistic Networking.

- **The Identity-based access control system:** a secure access control

system for messages with routing code designed to protect a particular resource: structured and organised information used by messages during routing.

- **The foundations of a pro-active messages’ network:** we point out the need of tools to build a network where different applications with different characteristics may require different treatments for their messages, because their routing needs may be different. This idea has been one of the motivations that led Borrego *et. al.* to develop the active DTN (aDTN) paradigm in [Bor13, BRF15].
- **Routing code-based identity:** the usage of the routing code as the own identity of the message to make an access control decision is a solution to a problem with no evidence of having being faced before.
- **A methodic security analysis of the system:** the security of the access control system has been methodically analysed against three different scenarios: 1) an attacker that creates a malicious routing code to access an information for which it is unauthorised; 2) an attacker that intercepts an exchange of contextual information and tries to access these entries; 3) an attacker that compromises a node and wants to access to all entries.
- **A proof-of-concept implementation of the system:** based on the real-time mobile agent’s platform MobileC [mob12], we implemented a proof-of-concept version of the whole system that allowed us to measure its performance and proof that it can be useful under the scope of an OppNet application. This process forced us to define and create the Routing Information Database (RIDB) and implement it using a set of files with Resource Description Framework (RDF) statements.

## 7.2 Endeavouring to be in the good books. Awarding DTN network use for acknowledging the reception of bundles

In an OppNet where not all nodes are deployed or managed by a central authority, the users may not be interested in routing and may prefer to save their resources for their own messages. However, if this behaviour is adopted by the majority of the network, the performance collapses, and the network itself becomes unuseful.

Therefore, it is necessary to encourage the users<sup>2</sup> to behave in a cooperative way for the sake of the common good, even if they are only interested in their own good.

To deal with this, the actions that can be performed by every user need to be defined and tracked. Then, it is necessary a fair reward and punishment system that takes into account how these actions are tracked. It is also needed a mechanism to force the users to care about these rewards and punishments. And, last but not least, all these tools must be able to operate without being simultaneously accessing a third party, as they must operate on a disconnected OppNet environment.

### 7.2.1 Receipt exchange protocol

An incentive scheme needs to track the actions performed by every user of the network. This is the only way the rewards can be distributed with fairness. The receipt exchange protocol takes care of it by generating transaction's receipts every time a message is forwarded. The protocol is based on the principles of Fair Exchange Signature Scheme (FESS) [LSM<sup>+</sup>08] and Identity Based Cryptography (IBC) [Sha85], but modifies the nature of the FESS' *keystone*, and therefore, its whole purpose. With the receipt exchange protocol, we obtain a system that keeps track of the actions done by the nodes by exchanging a low number of messages; that does not require the involvement of a third party during the transactions; and that avoids key management issues in OppNet scenarios thanks to the properties of IBC.

#### Definitions

The receipt exchange protocol defines two key elements: the voucher and the receipt.

**voucher** describes a transaction. Four fields form it: the sender, the receiver, the name of the voucher's issuer, and a flag that indicates if the transaction is a relay between intermediary nodes or if one of the nodes involved is either

---

<sup>2</sup> Nodes are not autonomous; they are operated by users. For this reason, along Section 7.2, we will use the terms "node" and "user" indistinctly.

the message's sender or destination. A voucher is correct if it describes the transaction properly.

**receipt** contains a voucher and a signature that binds the voucher to its issuer and to the message transmitted. A receipt is generated by cryptographically signing the voucher, and it is verified by cryptographically verifying the signature. However, a receipt is only valid if it is verified and presented together with the *keystone*.

### Exchange of messages

Next, we present the exchange of messages generated by the receipt exchange protocol every time that a couple of nodes establish contact and one of them wants to send a message to the other one.

1. The node who initiates the transaction (from now on, the *initiator*) creates a voucher describing it, and then signs it to produce the receipt that contains this voucher. This receipt is sent to the other node (from now on, the *receiver*).
2. The receiver checks the voucher and verifies the receipt. If the verification holds, the receiver creates its own voucher and signs it to produce a receipt. Then, it sends this receipt to the initiator.
3. The initiator checks the voucher and verifies the receipt. If the verification holds, the initiator sends the message to the receiver and concludes the transaction. The message itself is the *keystone* that provides validity to the two receipts.

### Created evidences

At the end of the receipt exchange, there are three possible scenarios, depending on how many parts of the protocol have been completed before the transaction's end:

1. **Only one receipt sent:** The receiver has the initiator's receipt, but it does not have validity because it lacks the *keystone*. Therefore, none of the

parts can proof that they have been in contact because of the transfer of the message.

2. **Two receipts sent:** The receiver has the initiator's receipt, but it does not have validity because it lacks the *keystone*. The initiator has the receiver's receipt and the *keystone*, that has not been released. Therefore, the initiator can proof that the receiver has been involved with this message's transfer. However, by doing this, the initiator compromises itself with the transfer by releasing the *keystone*.
3. **Protocol completed:** Both parts have the other's receipt and the message that acts as the *keystone*, so they both have unequivocal evidence that they have been in contact because of the transfer of the message.

From an incentive scheme's point of view, the first scenario lacks any interest: the message has not been forwarded. Therefore, there is no need to punish or reward anyone. However, the two other scenarios are more important: if it has been a message transmission, it means that some node should be rewarded (for forwarding the message, as the initiator on the third scenario) or punished (for dropping, losing or not forwarding it to another node, as the initiator on the second scenario or the receiver on the third one). Unfortunately, using the evidences generated during the receipt exchange protocol, there is no way to tell between the second and the third scenario, meaning that there is no way to decide which node should be rewarded and which node should be punished. This is something that must be taken into account by the incentive scheme to provide fairness to the reward system.

### 7.2.2 Incentive scheme

An incentive scheme is, essentially, a set of rules that increase or decrease the users' balances according to their actions. Obviously, these rules must be closely related to the mechanism used to keep track of the users' actions, to the characteristics of the network, and to the behaviours that we want to encourage. Therefore, the incentive scheme design has to take into account the particularities of the receipt exchange protocol, and the need of operating asynchronously, using only partial knowledge of the actions performed by the nodes.

### Guilty until proven innocent

As explained before, when one of the users involved in a transaction acts dishonestly by not forwarding the message, neither of the two pieces of evidence is sufficient to prove, unequivocally, which node is guilty and which one is innocent. However, we can identify the two nodes that are suspicious of having lost the message. To deal with this, our incentive scheme uses the policy “guilty until proven innocent”, meaning that the two nodes involved with the loss of a message will be marked as suspicious nodes and punished until it is demonstrated that they behaved honestly.

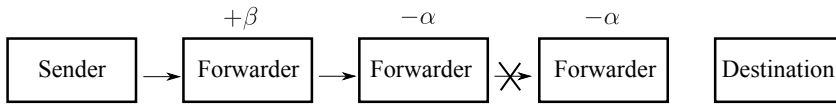
Acting this way, one small act of unfairness is performed by the incentive scheme every time a message is lost. However, when we look at the big picture, we can fairly tell apart the innocent nodes (they have been punished a few times, and so, probably wrongly) from the guilty ones (they have been punished a lot of times). Later, when the innocence of a node is proven, the punishments are removed, and they are properly rewarded for their behaviour.

### Rewards and punishments

As the Incentive Manager (IM) collects the receipts and the *keystones* that proof which nodes have been in contact due to the transfer of every message, it builds the chain of custody of every message and punishes or rewards the nodes with Cooperation Points (CP) on every update of the chain of custody. The application of the “guilty until proven innocent” policy is done this way: the two last nodes of the chain of custody are considered suspicious, so they are punished; the third last node and all previous ones except the first are confirmed relays, so they are rewarded. When a node changes from being suspicious to being a confirmed relay, first its punishment is removed, and then it is rewarded. Figure 7.3 illustrates the meaning of these rewards and punishments.

The amount of CP that it is added to or subtracted to every node is defined by the parameters  $\alpha$ ,  $\beta$  and  $\epsilon$ .

- $\alpha$ : The punishment applied to nodes that are suspicious of not forwarding a message.
- $\beta$ : The reward given to nodes when it is proven that they have forwarded a message.



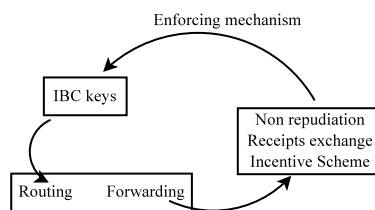
**Figure 7.3:** Illustration of the reward and punishment scheme. The arrows depict the relays of the message. The first node is not rewarded nor punished. The second node is rewarded with  $+\beta$  CP because it is a confirmed relay. The third and the fourth nodes are suspected of having lost the message; therefore, they are punished with  $-\alpha$  CP.

- $\epsilon$ : The reward given to a node for each proof delivered to the manager.

### The enforcing mechanism

IBC-based OppNets are based on the assumption that nodes will, eventually, connect with the Private Key Generator (PKG) to obtain a set of private IBC keys. The IM has to receive the proofs of all transactions, track the chain of custody of every message and reward or punish nodes due to their behaviour. By placing the IM and the PKG together, nodes can use the trip to upload proofs and to obtain new IBC keys.

In general, nodes prefer to obtain more keys when they contact the PKG, because this way they gain more independence and they can operate for more time without asking the PKG again for more keys. For this reason, we relate the amount of CP obtained by the nodes with the amount of IBC keys they obtain from the PKG, Figure 7.4 illustrates this.



**Figure 7.4:** Nodes need IBC keys to send their messages, and they obtain the keys by forwarding other's messages.

When a node asks for keys, we normalise the demanding node CP inside the interval  $[min_{CP}, max_{CP}]$  and map that value inside the interval  $[min_K, max_K]$ ,



defined by the minimum and maximum possible amount of keys to deliver. This way, nodes are not excluded from the network because they obtain, at least,  $min_K$ , and all nodes are treated with fairness because they are compared with their neighbours, who are supposed to have similar forwarding opportunities. This procedure is formalized in Equation 7.3.

$$\text{Amount of keys} = min_K + \frac{CP - min_{CP}}{max_{CP} - min_{CP}}(max_K - min_K) \quad (7.3)$$

$$CP_t = e^{-t/T} CP_{-1} \quad (7.4)$$

Using the exponential decay function of Equation 7.4 and a time constant  $T$  to gradually decrease Nodes' CP over time, *selfish bursts*<sup>3</sup> are prevented. Besides, this way, the nodes with a very low (or negative) CP can recover from their past uncooperative behaviour if they start being cooperative because past actions will weigh less than present actions.

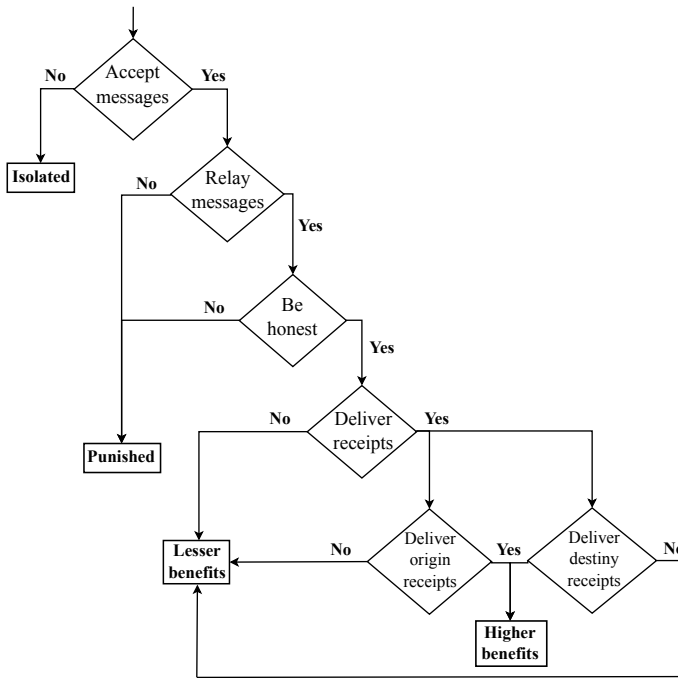
### Game theory analysis

Every participant of the network has to make a decision about the next behaviours: to participate in the network; to accept other's messages to forward them; to cheat or be honest when exchanging receipts; to deliver proofs of forwarding to the IM; and to deliver proofs of delivery to the IM. A strategy of one node consists in his decision do or not every one of this things. Figure 7.5 shows all possible strategies and the expected outcome related to each one.

Considering that the participants are rational, *i.e.* they will never behave in a way that could turn against their interests, the game theory analysis defined the possible values<sup>4</sup> of the parameters  $\alpha$ ,  $\beta$  and  $\epsilon$  to grant that every node of the network will be interested in choosing a concrete strategy, which consists of accepting messages addressed to others, forwarding these messages, being honest during the receipt exchange protocol and delivering to the IM all kind of proofs is a strictly dominant strategy.

<sup>3</sup> The behaviour of a node accumulating CP and then using it to behave in a very uncooperative way without being punished.

<sup>4</sup>The constraints are:  $\alpha > 0$ ,  $\beta > 0$  and  $\epsilon > \frac{13}{20}\alpha + \frac{4}{5}\beta$ . Chapter4 provides the details about how these constraints have been calculated.



**Figure 7.5:** Flow chart of the strategy that every user has to choose. The only way to reach the higher benefits is to participate on the network and to cooperate with the other nodes, to be honest, and to deliver the receipts to the IM.

This behaviour grants the node higher benefits than any other possible strategy, and it does not matter what the other nodes of the network do. Consequently, a profile of strategies where all nodes choose to behave this way forms a unique Nash equilibrium [Nas50] because it is impossible for any node to increase its profits by deviating from this strategy.

### 7.2.3 Main contributions

Next, we summarise the contributions of *Endeavouring to be in the good books. Awarding DTN network use for acknowledging the reception of bundles* to the research of incentive systems under the scope of Opportunistic Networking.

- **A receipt exchange protocol:** designed on the basis of FESS and IBC.

It is the result of transforming a protocol where two nodes sign a document they know beforehand into a protocol where two nodes forward a message and generate pieces of evidence about the transaction done. The last step of the protocol was modified to send the message, instead of a random *keystone*. Besides, we have modified the structure of the receipts, adding the needed fields to make it store unequivocal information about the transaction they are related and denying the reuse of past receipts on future transactions of the same message. Additionally, we have benefited from hash functions properties to optimise the protocol and reduce the amount of space needed by nodes to store the receipts.

- **An asynchronous incentive scheme for OppNet:** the system rewards nodes without waiting to build the whole chain of custody. Every time a node delivers a proof to the IM, it updates the state of the chain of custody, and it distributes rewards and punishments as if the new state will be the last. This is a feature that makes this system way more adapted to the characteristics of OppNets than other state-of-the-art proposals.
- **A detailed game theory analysis:** used to find the restrictions on the parameters' values. We have split the system into a set of subgames, and we have solved each one of them. Besides, we have proof that a profile of strategies where all nodes choose to behave in a fully cooperative way forms a unique Nash equilibrium because it is impossible for any node to increase its profits by deviating from this strategy.
- **A proof-of-concept implementation of the system:** built to obtain a measure of the overhead introduced by the system. We used the obtained measurements to study the ratio of aborted transactions and the impact on the latency and delivery ratio.
- **A simulation-based study:** using The Opportunistic Network Simulator (The ONE) [KOK09], we have examined the incentive scheme under the scope of a wireless robot sensor grid network with heterogeneous nodes and applications. We have reasoned about the fairness of the system and about how to chose the  $T$  (time constant) parameter and about the impact of message expiration. Finally, we have studied the increase of network's performance due to the usage of the incentive scheme in presence of not cooperative nodes.

## 7.3 PrivHab: A privacy preserving georouting protocol based on a multiagent system for podcast distribution on disconnected areas

Facing the challenges of an e-agriculture podcast distribution application on disconnected areas, we proposed to create a network by distributing small devices among the members of a Non-governmental Organization (NGO) and some local villagers. The proposed network should use OppNet networking to work without any infrastructure, allowing a low deployment cost for the NGO.

The network's cost should be as low as possible, for this reason, we want to not only use the NGO's own nodes but also to allow any volunteers to help the NGO by becoming part of the network using their own devices. Obviously, the impact on the volunteers' device and life should be reduced to the minimum. This imposes two requirements over the system used: *a)* the privacy of the nodes has to be preserved; *b)* the system has to achieve a good performance by occupying a small buffer and using few resources.

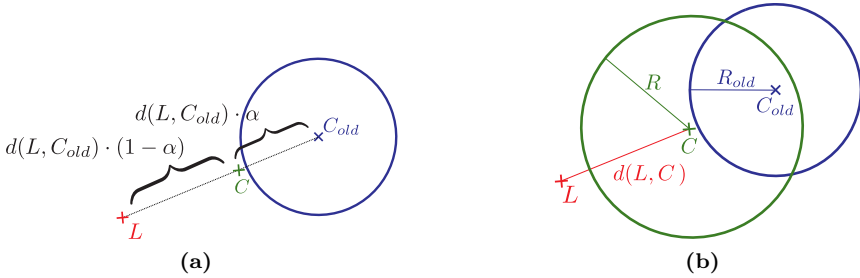
In order to implement a system that matches the needs of this application, we need to design a routing algorithm adapted to the characteristics of the scenario and the nodes, and we also need to be sure that the nodes' privacy is preserved. Finally, as on the previous contributions, the whole system has to be able to operate using only tools that are available in Delay Tolerant Network (DTN).

### 7.3.1 The Habitat

The movements of every node of the network are strongly related to the person carrying it. Therefore, to know the places where a node has been in the past is useful to infer if a node will visit these places again in the future. In order to model the node's usual whereabouts, we have defined the **habitat**: the area where someone is more likely to be found. We propose to model the habitat using the simplest geometric shape: the circle, this way nodes can automatically calculate and store it consuming the minimum computational resources, and they can make quick routing decisions.

### The habitat's circular model

The habitat is regularly calculated and updated by obtaining the location of a node and adding it to his habitat using an Exponentially Weighted Moving Average (EWMA) [Rob00]. The centre point of the habitat is calculated by averaging<sup>5</sup> the centre point ( $C_{old}$ ) and the current location ( $L$ ). This first step is depicted in Figure 7.6 and formalized by Equation 7.5.



**Figure 7.6:** (a) The centre point is updated by averaging the centre  $C_{old}$  and the new location  $L$ . Note that the centre point  $C$  has moved towards  $L$  according to an  $\alpha$  factor; (b) The radius  $R_{old}$  is used together with the distance  $d(L, C)$ , that separates the new location and the centre point  $C$ , to update the radius  $R$  of the habitat.

$$C = L * \alpha + C_{old} * (1 - \alpha) \quad (7.5)$$

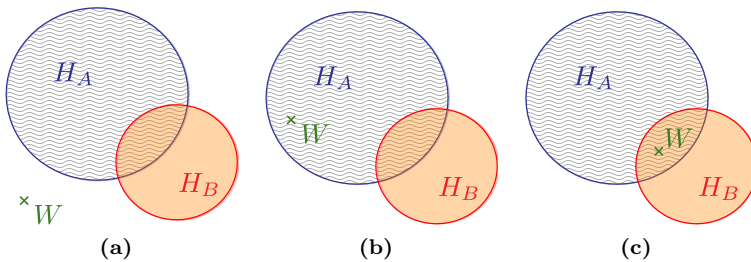
Then, the habitat's radius ( $R$ ) is updated by averaging the radius ( $R_{old}$ ) the distance between the actual location and the centre point. This second step is depicted in Figure 7.6 and formalized by Equation 7.6.

$$R = d(L, C) * \alpha + R_{old} * (1 - \alpha) \quad (7.6)$$

<sup>5</sup>The  $\alpha$  parameter determines the habitat's time span ( $T$ ). When chosen according to  $\alpha = \frac{2}{T\omega+1}$ , being  $\omega$  the update frequency, the moving average weights the last  $T\omega$  locations a 86% of the total. Then, we say that the habitat models the last  $T$  hours.

### 7.3.2 A Habitat-based routing algorithm

Given the definition of habitat, we assume that nodes spend most of the time inside the area defined by their habitats. Therefore, the PrivHab algorithm uses this reasoning to decide which node is better to carry a message towards its destination. The algorithm chooses the nodes whose habitat enclose the destination, prioritising those nodes whose habitat is the smallest. If a waypoint is contained outside two habitats, then the algorithm chooses the node whose border is the closest to the next waypoint. Figure 7.7 show the different situations that can be faced. In (a) and (b) node  $A$  is chosen as the best option, because the target waypoint  $W$  is closer to  $H_A$  or inside it. In (c) the best choice is  $B$ , because both habitats contain  $W$ , but  $H_B$  is smaller than  $H_A$ .



**Figure 7.7:** There are three possible situations when comparing two habitats: (a) The next waypoint ( $W$ ) is located outside the two habitats; (b) Only one of the habitats encloses the location of the next waypoint; (c) The two habitats enclose the location of the next waypoint.

### 7.3.3 Mapping negatives to perform homomorphic subtractions

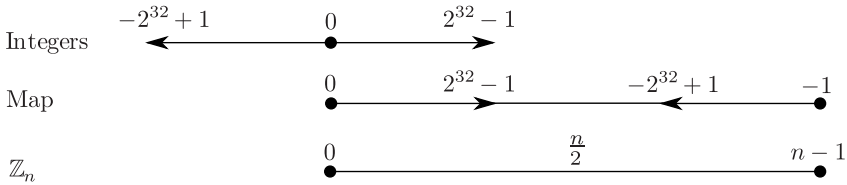
PrivHab uses techniques of secure multi-party computations to protect nodes' privacy. The goal is to operate and compare the habitats and waypoints or location destinations while cryptographically protected to avoid revealing this private information to the other parts. This operation can be done using an additive homomorphic cryptosystem<sup>6</sup>.

<sup>6</sup> An additive homomorphic cryptosystem is one in which, given two encrypted operands  $E(a)$  and  $E(b)$ ,  $E(a + b)$  can be computed without separately decrypting each one.

In order to execute the routing algorithm, PrivHab needs to know if a point is contained inside a circle, and to calculate the distance between a circle and a point, the only way to do this is by performing additions, subtractions and multiplication using encrypted operands. The chosen cryptosystem, Paillier [Pai99], as any other additive homomorphic cryptosystem [Gen09], provides the addition of cyphered values and the multiplication by a plain operand. We managed to perform the subtraction by performing the addition of a negative value. However, as there are no negative values in  $\mathbb{Z}_n$ , we mapped them in a way that they could still be added to other cyphered operands or multiplied by a plain operand.

We map positive integers lower than  $n/2$  using the identity function and negative integers greater than  $-n/2$  with its representation modulo  $n$ , as shown in Equation 7.7. Figure 7.8 provides a scheme of this mapping.

$$\text{Map}(x) = \begin{cases} x & x \in [0, n/2) \\ x + n & x \in (-n/2, 0) \end{cases} \quad (7.7)$$



**Figure 7.8:** Positive integers are mapped to  $\mathbb{Z}_n$  using the identity function. Negative integers are mapped to the higher part of  $\mathbb{Z}_n$  using its representation modulo  $n$ .

This way, we use Paillier addition between a positive integer  $a$  and a negative integer  $-b$  mapped as  $-b + n$  to obtain  $(a - b) + n \bmod n$ . If  $a > b$  then  $(a - b) + n \bmod n = (a - b)$ , and if  $a < b$  then  $(a - b) + n \bmod n = a - b$ .

Besides, multiplication between a negative integer  $-b$  (mapped as  $-b + n$ ) and  $s$  to obtain  $(-b + n) \cdot s \bmod n$  is calculated doing  $-b \cdot s + n \cdot s \bmod n = -b \cdot s$ .

Then, the result of both operations can be recovered by reversing the previous mapping, using Equation 7.8.

$$\text{Inverse Map}(x) = \begin{cases} x & x \in [0, n/2) \\ x - n & x \in (n/2, n - 1] \end{cases} \quad (7.8)$$

### Example of subtraction of cyphered values

Next, we provide one example in order to better illustrate how to perform a subtraction:

1. Let Alice have Public key  $(n, g) = (15, 2)$  and Private key  $(\lambda, p, q) = (4, 3, 5)$ .
2. Then, Bob picks a random  $r = 2$  and cyphers  $m_B = 3$  by doing  $c_B = g^m r^n \bmod n^2 = 19 = E_A(m_B)$
3. Later, Charlie wants to subtract 5 from  $E_A(m_B)$ , so he first maps  $-5$  as  $(-5 + 15 = 10)$  (remember that the boundary that separates positives from negatives is set to  $15/2$ ) and cyphers  $m_C = 10$  by picking a random  $r = 4$  and doing  $c_C = g^m r^n \bmod n^2 = 151 = E_A(m_C)$ .
4. Charlie adds  $E_A(m_B)$  with  $E_A(m_C)$  by doing  $E_A(m_B) \cdot E_A(m_C) = 151 \cdot 19 \bmod 225 = 169 = E_A(m_B + m_C)$ .
5. Then, Alice decrypts  $E_A(m_B + m_C)$  by calculating  $m = L(c^\lambda \bmod n^2) = L(169^4 \bmod 225) = L(196) = 13$ .
6. Finally, as 13 is greater than  $n/2$ , Alice undoes the mapping doing  $13 - 15 = -2$ , which is the result of subtracting (or adding a negative) 5 from 3.

### Example of multiplication by a plain operand

Besides, using the same keys of the previous example, if Charlie wants to multiply the  $-2$  by a plain operand  $s = 3$ :

1. Charlie calculates  $E_A((m_B + m_C) \cdot s) = E_A(m_B + m_C)^s = 169^3 \bmod 225 = 109$ .
2. Alice decrypts  $E_A((m_B + m_C) \cdot s)$  by calculating  $m = L(c^\lambda \bmod n^2) = L(109^4 \bmod 225) = L(136) = 9$ .
3. Finally, as 9 is greater than  $n/2$ , Alice undoes the mapping doing  $9 - 15 = -6$ , which is the result of multiplying a cyphered  $-2$  by a plain 3.



### 7.3.4 The PrivHab's exchange of messages

The PrivHab's exchange of messages goal is to execute the PrivHab routing algorithm to compare two nodes based on their habitats, but without disclosing any private information to the other part. PrivHab achieves this by operating with homomorphically encrypted operands and benefiting from the previously presented negative's mapping to make comparisons.

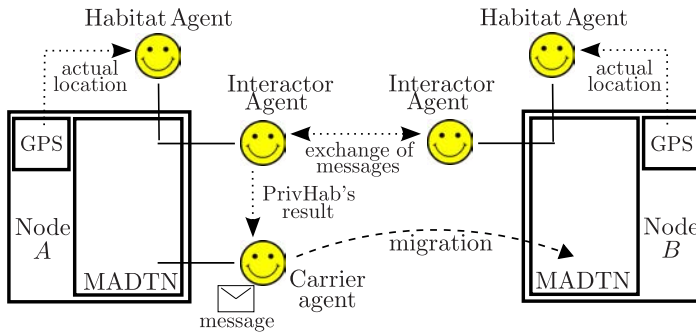
Let  $A$  be the node that carries the data, and  $B$  a candidate neighbour. Let  $W$ , known only by  $A$ , be the next waypoint where the data has to be carried to. The PrivHab protocol, described below, requires the two nodes to perform the following operations:

1. Node  $B$  adds the coordinates of its habitat's centre point ( $C$ ), cyphered, to the beacons sent during the neighbour discovery process.
2. Node  $A$  compares the coordinates of the waypoint  $W$  with  $C$ , by subtracting them and then multiplying both results by the same *nonce* (a random one-use value). Following,  $A$  sends to  $B$  the results, the coordinates of  $W$ , the distance between  $W$  and  $A$ 's habitat and its radius. These three last values are cyphered using  $A$ 's public key, so  $B$  can operate but can not decrypt them.
3. Node  $B$  decrypts the comparison between  $W$  and  $C$ , and uses it to calculate the square of the distance between  $W$  and  $B$ 's habitat. Following,  $B$  calculates if  $W$  is inside  $B$ 's habitat, and compares the distances and radius. This time, three different *nonce* values are used to randomise the results. At this point,  $B$  has all the information needed to execute the routing algorithm, but everything is cyphered with  $A$ 's key, so, node  $B$  sends all the results, randomly ordered, to  $A$ .
4. Node  $A$  decrypts the three received values.  $B$ 's habitat wins the comparison if and only if the three decrypted values are negative or 0. Finally, if  $A$ , according to its forwarding policy decides so, the message is forwarded to  $B$ .

### 7.3.5 A multiagent-based system

PrivHab is designed to operate in a Mobile Agent based DTN (MADTN) [MCR<sup>+</sup>13]. In this kind of networks, the messages are carried by mobile agents. Therefore,

PrivHab’s goal is to help these agents by providing them with routing information to improve their itinerary selection. In MADTN, Mobile agents provide autonomy to find their way to their destination in a partially unknown and changing environment. Agents also have the intelligence to make decisions that lead them towards their goal. The agents cannot control nodes’ movement, so they are mobile to be able to migrate when they find a more useful one. Besides, agents are proactive, so they can initiate context-aware actions as starting the delivery phase when the agent is near the destination; and they represent applications with different needs, allowing them to use the same network in their own way, with the agents making decisions on their behalf. All the agents involved in the multiagent system that enables the operation of PrivHab are listed and explained below and depicted in Figure 7.9.



**Figure 7.9:** Dotted lines depict the main interactions between entities, while slashed lines depict the movement of the agents. The Habitat Agent updates the habitat using information from the GPS receiver. The Interactor Agent exchanges PrivHab’s messages with the other nodes and informs the Carrier agent of the result of the execution. The Carrier Agent carries the message and makes the decision of migrating, staying or being cloned.

- **Habitat Agent:** This agent calculates and periodically updates the habitat of the node.
- **Interactor Agent:** This agent performs the PrivHab’s exchange of messages to compare the habitats of the two nodes. Then, this agent informs the Carrier agent of the result of the comparison.
- **Carrier Agent:** This agent carries the message, and his goal is to deliver it to its destination. It uses the result of PrivHab’s execution, along with other contextual information, to decide its itinerary.

### 7.3.6 Main contributions

Next, we summarise the contributions of *PrivHab: A privacy preserving georouting protocol based on a multiagent system for podcast distribution on disconnected areas* to the research of privacy preserving georouting protocols under the scope of Opportunistic Networking.

- **A scenario of application:** based on a real need, we analysed the work done by the NGO *Practical Action* on a rural and disconnected area in Gwanda and concluded that there is a need of automating the podcast distribution. We proposed a solution based on a low-cost network and designed the appropriate tools to make it work.
- **The habitat:** a tool to model the nodes' whereabouts. As most of the OppNet solutions do not apply well to high-distances scenarios where messages should be routed in a long-term basis, we have developed a simple habitat model that enables the usage of a georouting protocol that benefits from the life-cycles of the people to make predictions.
- **The foundations of a multiagent system:** that benefits from PrivHab to improve the decision-making of the MADTN agents.
- **A mechanism to perform subtractions using homomorphic encryption:** a major feature that increases the utility of additive homomorphic cryptosystems. Concretely, PrivHab benefits from it to execute a routing algorithm that compares two nodes' habitats without disclosing any private information to the other part.
- **A proof-of-concept implementation:** built to obtain a measure of the overhead introduced by the system depending on the key length of the cryptosystem used.

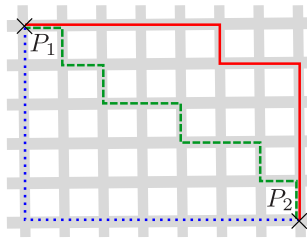
## 7.4 PrivHab+: A secure geographic routing protocol for DTN

PrivHab+ is the result of the work done to overcome PrivHab's limitations and to equip it with new features. On the one hand, changing the type of geometry used has enabled the usage of different geometric shapes (represented by the ellipse)

to model the habitats, but also has forced us to find a new way to calculate distances and to re-design the whole exchange of messages. On the other hand, we optimised the execution time by adding a multi-destination execution mode, and we broadened the applicability of the system by decoupling it from MADTN.

### 7.4.1 Usage of Taxicab Geometry

The usage of an additive homomorphic cryptosystem restricts the operations we can use to compare habitats. For example, Euclidean distances cannot be calculated because there is no way to calculate the square root of an encrypted operand, this imposes a constraint on how we model the habitats. For this reason, PrivHab+ moves from the Euclidean geometry used in PrivHab to Taxicab geometry, in which the distance between two points is the addition of the absolute differences of their Cartesian coordinates. This distance function is usually called Manhattan distance, and it can be calculated without computing any square root. Figure 7.10 shows an example of Manhattan distances.

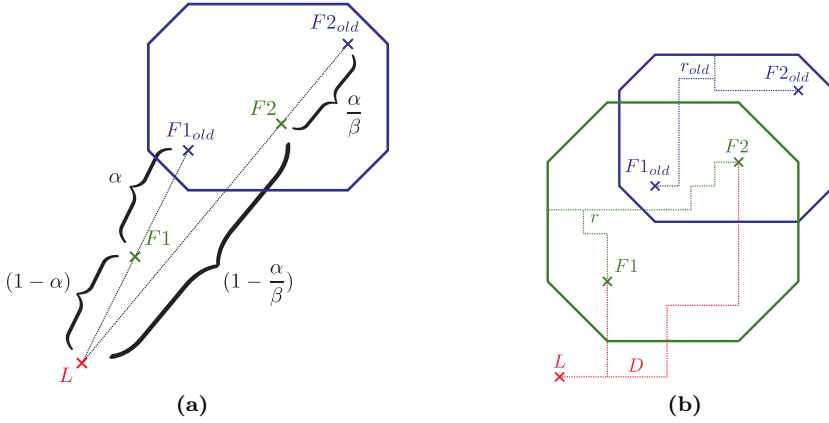


**Figure 7.10:** In Taxicab geometry, all three pictured lines have the same length for the route between  $P_1$  and  $P_2$ .

### 7.4.2 The elliptic habitat

The usage of Taxicab geometry allows PrivHab+ to operate with other models of habitat. Following, we provide the definition of the elliptic habitat, and we define how to update and calculate it.

The first step of the habitat's update is to update the focal points of the habitat are calculated by using EWMA to average the focal points of the habitat and the current location ( $L$ ). Let  $F1_{old}$  be the nearest focal point to the node's



**Figure 7.11:** (a) Evolution of the focal points  $F1_{old}$  and  $F2_{old}$  when the new location  $L$  is used to update the habitat.  $d(L, F)$  denotes distance between  $L$  and  $F$ . Note that  $F1$  has been attracted by  $L$  according to an  $\alpha$  factor while  $F2$  has been attracted using a lesser  $\frac{\alpha}{\beta}$  factor; (b) The radius  $r_{old}$  is used together with the distances  $d(L, F1)$  and  $d(L, F2)$  that separate the focal points  $F1$ ,  $F2$  and the location  $L$  to update the radius  $r$ .

location and  $F2_{old}$  be the farthest one. This process is depicted in Figure 7.11 and formalised in Equations<sup>7</sup> 7.9 and 7.10.

$$F1 = L * \alpha + F1_{old} * (1 - \alpha) \tag{7.9}$$

$$F2 = L * \frac{\alpha}{\beta} + F2_{old} * (1 - \frac{\alpha}{\beta}) \tag{7.10}$$

The second step is to update the habitats radius ( $r$ ), by averaging using EWMA the radius  $r_{old}$  and the added distances  $d(L, F1)$  and  $d(L, F2)$  between each focal point of and  $L$ . This second step is depicted in Figure 7.11 and formalized by Equation 7.11.

$$r = (d(L, F1) + d(L, F2)) * \alpha + r_{old} * (1 - \alpha) \tag{7.11}$$

<sup>7</sup>By using  $\beta > 1$ , the current location  $L$  weights more when calculating the new position of the nearest focal point than when calculating the new position of the farthest focal point. This means that  $L$  attracts more the nearest focal point, modifying the habitat's eccentricity depending on the relative position of  $L$ .

### 7.4.3 PrivHab+

As an evolved version of PrivHab, PrivHab+ benefits from some of the previous contributions, as the idea of habitat or the need of preserving the privacy of the users. However, PrivHab+ uses Taxicab geometry instead of the Euclidean geometry, so distances need to be calculated differently, and it supports some features that provide an increase of applicability. Next, we summarise the main differences between this system and his previous version.

#### Taxicab geometry calculations

The PrivHab routing algorithm, that remained unchanged, requires to calculate how far is a location from a habitat, if a location is inside a habitat, and which habitat has the smaller radius. Moving to Taxicab geometry changes the way this three values have to be calculated.

Essentially, almost all calculations need to compute the absolute value of the result of a subtraction, which cannot be calculated with homomorphic cryptography. However, we can take advantage of Equation 7.12 to walk around this issue and calculate the absolute value if we know beforehand the relation between the two operands.

$$|Z - W| = \begin{cases} Z - W & : Z > W \\ W - Z & : Z < W \end{cases} \quad (7.12)$$

Using this, PrivHab+ calculates the Manhattan distances needed to execute the routing algorithm to compare habitats.

#### Decoupled from MADTN

Given the good performance results obtained by PrivHab when compared with other DTN routing protocols, we decided to decouple PrivHab+ from MADTN. This way, PrivHab+ becomes a standalone routing protocol that does not requires of a particular architecture to operate. This modification was done in order to allow its deployment into a a bundle-based DTN [SB07]. However, PrivHab+

can also be adapted to fit into a variety of OppNet frameworks, such as Haggie [SSH<sup>+</sup>07] or HiBOp [BCJP07].

### Multi-destination execution mode

We added to PrivHab+ a multi-destination execution mode. This type of execution processes simultaneously all messages at once. It is around a 20% faster, but it takes an all-or-nothing approach, meaning that no message could be routed if the connectivity window ends before finishing the execution.

Therefore, we propose to use a mixed strategy: use a multi-destination execution to route faster the first messages and then iterate the remaining of the messages one by one. The amount of messages to execute at the same time should be decided depending on concrete characteristics of the network, as the average connectivity window.

### The PrivHab+' exchange of messages

The PrivHab+'s exchange of messages executes the PrivHab routing algorithm to compare two nodes based on their elliptic habitats, but without disclosing any private information to the other part. PrivHab+ benefits from Equation 7.12 to calculate the absolute values needed to make the comparison.

Let  $A$  be the node that carries a set of messages, and  $B$  a candidate neighbour. By the previous definitions,  $A$  wants to know if  $B$  is a better choice to carry each message towards its destination. The PrivHab protocol, described below, requires the two nodes to perform the following operations:

1. Node  $B$  calculates a summary of the characteristics of its habitat and adds it to the beacons that announce its presence to the neighbours.
2. Node  $A$  compares<sup>8</sup> the received values with the destination of every message. Then  $A$  sends the comparisons to  $B$  together with the coordinates of every destination, the double of the distance from  $A$ 's habitat to this destination and the radius of its habitat.

---

<sup>8</sup>The comparisons are done by subtracting the corresponding coordinates of the destination from the characteristics of the habitat and then multiplying the result with a random one-use value.

3. For every message, *B* decrypts all the received comparisons. Node *B* knows that each decrypted value greater than zero means that the characteristic of the habitat is greater than the corresponding coordinate of the destination. Node *B* uses this information and Equation 7.12 to calculate the double of distance between its habitat and the destination. Afterwards, node *B* sends to *A* the comparison between distances, and the comparison between the radius radius, ordered in a random way.
4. Finally, node *A* decrypts every pair of comparisons. For every message for whom the two decrypted values are equal or greater than 0, *A* learns that *B* is a better choice, and forwards the message.

#### 7.4.4 Main contributions

Next, we summarize the contributions of *PrivHab+*: A secure geographic routing protocol for DTN to the research of secure geographic routing protocols under the scope of Opportunistic Networking.

- **The usage of Taxicab Geometry:** allows *PrivHab+* to operate with different habitat models, because the manhattan distances can be calculated using an additive homomorphic cryptosystem, while the usual Euclidean distances don't.
- **The elliptic habitat:** provides flexibility to how *PrivHab+* models the nodes' usual whereabouts. It demonstrates that the system can operate using a model more complex than the circle.
- **The *PrivHab+* protocol:** a privacy preserving georouting protocol for OppNet that evolves *PrivHab* by adding new features and increasing its performance.
- **A formal security analysis:** the privacy of the routing protocol has been methodically analysed against two different scenarios: 1) a passive adversary that exchanges truthful messages and analyses them to obtain information about the other part; 2) an active adversary that forges messages using untruthful information in order to disclose private information about the other part.
- **A new scenario:** based on the same podcast distribution application, we located another potential scenario at the region of Cajamarca, in Perú.



- **A qualitative comparison:** the main characteristics of PrivHab+ have been studied and compared against a set of well-known protocols that have been chosen as representatives of contact-based prediction routing algorithms, and epidemic-based routing algorithms, the most commonly used routing protocols in DTN.
- **A proof-of-concept implementation:** built to obtain a measure of the overhead introduced by the system depending on the key length of the cryptosystem used.





*“So in war, the way is to avoid what is strong and to strike at what is weak.”*

*The Art of War, SUN TZU*



## Conclusions

**T**HE main objective of this thesis is to develop new Opportunistic Networking tools to deal with heterogeneous environments, and to extend, this way, Opportunistic Network (OppNet)’s applicability.

In the first place, we have considered an environment of heterogeneous applications that share a network, but have different needs and require their messages to be treated in different ways. To deal with this situation, we have defined pro-active messages, the messages that carry their own routing code, and have spotted the most important requirement of this network’s architecture: every application defines the contextual information it needs to perform its own routing. Using the cryptographic tools that are at hand in an OppNet, we have developed an access control system that protected and granted the privacy and the integrity of this application-related contextual information. This way, we have made feasible the whole pro-active message system.

In the second place, we have considered an environment of nodes owned by

heterogeneous users that want to use the network to send their messages, but do not have any interest in using their resources to contribute to the network's operation. In this situation, we first have implemented the receipt exchange protocol, a mechanism to keep track of the users' actions, based on a two-party signature scheme where both signatures become valid at the same moment. Then, we have defined a reward and punishment scheme that takes into account the particularities of OppNet networks and the receipts exchanged by the nodes. Finally, we have designed an enforcing mechanism that forces users to care about the rewards and punishments they obtain when forwarding others' messages. This last piece is the one that ties together the Identity-based asynchronous incentive scheme for OppNet.

In third place, we have considered a network of participants that want to collaborate with the network by making their heterogeneous nodes available, but do not want to renounce to their privacy by sharing routing information with the other participants. In this case, we have elaborated a habitat-based georouting protocol that fits with the characteristics of a scenario of application based on a real podcast distribution application, and we have used homomorphic cryptography, and our mapping-based subtractions, to protect the users' privacy.

Following, we have improved the previous habitat-based georouting protocol by increasing its performance. This way, we have aimed to extend its applicability by reducing its impact on the users' devices. This performance optimisation has relied upon two different aspects. On the one hand, we have moved from Euclidean geometry to Taxicab geometry. This change has allowed improving the habitat's modelling by using more complex geometric shapes, as the ellipse, because it simplified the distances' calculations. On the other hand, we have added a new multi-destination mode of operation that benefits from the first moments of every connectivity window to process and forward a bunch of messages, instead of processing them one by one. Besides, we have decoupled the protocol from Mobile Agent based DTN (MADTN) and have allowed its operation both as a standalone georouting protocol or as an integrated part of an OppNet framework.

The three presented tools, the access control system; the incentive scheme and the georouting protocol; and some of their main principles, as the identification of messages using their identity; the habitat model; the usage of homomorphic cryptography and the mapping; can be combined among them to solve a situation that requires it. They can also be combined with any other tool meant to deal with other types of heterogeneity or be coupled with other existing systems that do not consider it.

Therefore, as a result of this thesis, OppNet applicability has been extended, because we have provided tools to allow applications to use the same network in different ways, to deal with the users' selfishness and to provide privacy to the users that make their devices a node available to the network.

## Other contributions

Due to the format of this thesis, a compendium of publications, we focused on the four main published works. However, the research done during its development has generated, directly or indirectly, a set of other publications that we will briefly present next.

### Identity-based access control for pro-active messages DTN

Previously to the writing the journal article, the authors sketched their ideas on a Spanish national conference, held in San Sebastián. The full bibliographic reference is provided below.

*“A. Sanchez-Carmona, C. Borrego, J. Andújar, S. Robles. Control de Acceso para Mensajes Pro-activos en Redes DTN. Proceedings for the XII Reunión Española sobre Criptología y Seguridad de la Información (RECSI). Mondragón Unibertsitatea.(September 2012)”*

Besides, the concept of pro-active messages evolved into a Bundle Protocol extension to allow messages to carry their own routing code. This work was published in the international journal of the second quartile *Computer Networks*. The full bibliographic reference is provided below.

*“C. Borrego, S. Robles, A. Fabregues, A. Sanchez-Carmona. A mobile code bundle extension for application-defined routing in delay and disruption tolerant networking. International Journal on Computer Networks (July 2015) vol. 87, pp: 59-77. ISSN:13891286. DOI:10.1016/j.comnet.2015.05.017.”*

## A privacy preserving georouting protocol for DTN

The first findings of the model of habitat and the architecture of the multiagent system were presented in a short paper at a CORE-A\* ranked international conference, held in Istanbul. At the same conference, a demonstration of the evolution of the circular habitat was also presented. The two full bibliographic references are provided below.

“A. Sanchez-Carmona, S. Robles, C. Borrego. *PrivHab: A Multiagent Secure Georouting Protocol for Podcast Distribution on Disconnected Areas*. In *14th International Conference on Autonomous Agents and Multiagents Systems (AAMAS 15)*. Istanbul. ACM Press, pp: 1697-1698. (May 2015) ISBN: 978-1-4503-3413-6.”

“A. Sanchez-Carmona, S. Robles, G. Garcia, C. Borrego. *PrivHab: A Multiagent Secure Georouting Protocol for Distributing Podcasts in Disconnected Areas (Demonstration)*. In *14th International Conference on Autonomous Agents and Multiagents Systems (AAMAS 15)*. Istanbul. ACM Press, pp: 1943-1944. (May 2015) ISBN: 978-1-4503-3413-6.”

Later, the authors applied their previous findings to a concrete scenario of application located in Gwanda, Zimbabwe, and we presented our proposal at an international conference, held in Salamanca. The full bibliographic reference is provided below.

“A. Sanchez-Carmona, C. Borrego, S. Robles. *Podcast Distribution on Gwanda using PrivHab: a Multiagent Secure Georouting Protocol*. In *13th Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 15)*. Salamanca. Springer Verlag vol. 372, pp: 29-37. (June 2015) DOI: 10.1007/978-3-319-19629-4\_4.”

The previous article obtained an “Award of Scientific Excellence” by the scientific comitee of the international conference. As a consequence, they invited the authors to expand their work and publish it on a non-indexed journal, and they did it. The full bibliographic reference is provided below.

“A. Sanchez-Carmona, S. Robles, C. Borrego. *Improving Podcast Distribution on Gwanda using PrivHaba Multiagent Secure Georouting Protocol*. *Advances in Distributed Computing and Artificial Intelligence Journal* (November 2015) vol. 4. no. 1. ISSN: 2255-2863.”

## Other related works

At the beginning of this thesis, the author spent some time looking for OppNet scenarios of application. One of the firsts scenarios considered led to a collaboration in an article presented at an international conference, held in Salamanca. The full bibliographic reference is provided below.

*“S. Castillo, R. Martínez, S. Robles, A. Sanchez-Carmona, J. Borrell, M. Cordero, A. Viguria, N. Giuditta. Mobile-Agent Based Delay-Tolerant Network Architecture for Non-Critical Aeronautical Data Communications. Proceedings for the 10th International Symposium on Distributed Computing and Artificial Intelligence. Salamanca. Springer vol. 217, pp: 513-520. (May 2013) DOI: 10.1007/978-3-319-00551-5\_61.”*

Finally, one of the most promising future lines of research has lead to the writing of an article. In this work, we use the model of habitat as a tool to define the characteristics of the nodes that may be interested in receiving a certain message. This way, we build a *profile-cast* [HDH08] habitat-based routing paradigm. This work is entitled *Killing two birds with one stone: using mobility behavioral profiles both as destinations and as a tool to hand the messages*, and, at the moment of writing this thesis, it is currently under review at the first quartile international journal *Computer Communications*.

## Future Research

At this late stage of the thesis, many future lines of research arise. Next, we briefly describe some of them, starting with the general topics that have to do with the whole thesis, and following, the lines that are more specifically related to the main works done.

On the one hand, how to best evaluate systems meant to heterogeneous environments, and to compare them with other proposals is usually an issue. Therefore, finding new ways of modelling, simulating or reproducing heterogeneous environments, could be very useful. Another approach to solve this problem could be to define a corpus of OppNet scenarios that allow researchers to compare their proposals with others' ones with fairness. Apart, we consider that the new trending of collaborative economy applications, which operate in a decentralised way, may provide new interesting scenarios to evaluate our proposals.



On the other hand, there are other heterogeneous environments that have not been treated on this thesis, but that could become interesting lines of research. For example, a network of nodes with very different capabilities, or a network of volunteer users that only carry and forward the messages of the subset of applications in which they participate. Besides, all the environments considered in this thesis have been faced following the approach of “making tools to help César on his job”. However, it could also be very interesting to follow a slightly different approach: “making César’s job unnecessary”.

Regarding the identity-based access control line of research, the system could be improved to allow *a posteriori* modifications to the sets of access rules, so the applications could revoke or add access permissions to previously defined information. Besides, the proposal could be improved by refining the format, structure, semantics and organisation of the contextual information used to make the routing decisions. Aside, the idea of a message carrying its routing code could be furtherly expanded. For example, by defining messages that carry not only their routing code, but a mobile code that decides is the message has expired, if it has to be delivered to a node, how many times should it be replicated, if an acknowledgment message should be created, etc.

About the incentive scheme line of research, the enforcing mechanism could be enhanced by using keys that only allow nodes to perform a subset of all the possible actions. This way, the uncooperative nodes would be more punished, but they would have more options to redeem and recover. Besides, the reward and punishment system could be re-designed to charge nodes for every message they send. Another interesting research line revolves around incentivizing other nodes’ actions that could benefit the network, as travelling to a certain location or contacting a usually isolated node, instead of the forwarding of messages.

With regard to the privacy preserving georouting line of research, the habitat model could be vastly improved by using a more complex shape, that perhaps do not need to be a geometric shape. The way it is updated and calculated could be changed and even a new routing algorithm that takes into account different habitats’ time spans could be designed. Besides, the habitat could be used not only to route the messages, but also to define a profile of target nodes that should receive a particular message, enabling a habitat-based *profile-cast* message sending. Aside, the mapping that allows to perform homomorphic subtractions has enabled a whole new line of research, because there are lots of contacts-based or history-based routing protocols that could be improved and re-designed by using it to preserve the privacy of the users.





# Bibliography

- [BCJP07] C. Boldrini, M. Conti, J. Jacopini, and A. Passarella. Hibop: a history based routing protocol for opportunistic networks. In *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium*, pages 1–12, June 2007.
- [BCR14] C. Borrego, S. Castillo, and S. Robles. Striving for sensing: Taming your mobile code to share a robot sensor network. *Information Sciences*, (0), 2014.
- [Bhu16] M. Nasir Mumtaz Bhutta. Public-key infrastructure validation and revocation mechanism suitable for delay/disruption tolerant networks. *IET Information Security*, March 2016.
- [Bor13] C. Borrego. *A Mobile Code-based Multi-Routing Protocol Architecture for Delay and Disruption Tolerant Networking*. PhD thesis, January 2013.
- [BR13] C. Borrego and S. Robles. A store-carry-process-and-forward paradigm for intelligent sensor grids. *Information Sciences*, 222:113 – 125, February 2013.
- [BRF15] C. Borrego, S. Robles, and A. Fabregues. A mobile code bundle extension for application-defined routing in delay and disruption tolerant networking. July 2015.
- [Gen09] C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.

- [HDH08] W. Hsu, D. Dutta, and A. Helmy. Profile-cast: Behavior-aware mobile networking. In *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, pages 3033–3038. IEEE, 2008.
- [KOK09] A. Keränen, J. Ott, and T. Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, New York, NY, USA, 2009. ICST.
- [LSM<sup>+</sup>08] J. Liu, R. Sun, W. Ma, Y. Li, and X. Wang. Fair exchange signature schemes. In *Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008. 22nd International Conference on*, pages 422–427, Mar. 2008.
- [MCR<sup>+</sup>13] R. Martínez, S. Castillo, S. Robles, A. Sánchez-Carmona, J. Borrell, M. Cordero, A. Viguria, and N. Giuditta. Mobile-agent based delay-tolerant network architecture for non-critical aeronautical data communications. In Springer, editor, *In 10th International Symposium on Distributed Computing and Artificial Intelligence*, May 2013.
- [mob12] Mobile-C: a Multi-Agent Platform for Mobile C/C++ Agents. Apr. 2012.
- [Nas50] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.
- [Pai99] P. Paillier. *Public-Key Cryptosystems Based on Composite Degree Residuosity Classes*, pages 223–238. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.
- [Rob00] S. W. Roberts. Control chart tests based on geometric moving averages. *Technometrics*, 42(1):97–101, 2000.
- [SB07] K. Scott and S. Burleigh. Bundle Protocol Specification. RFC 5050 (Experimental), November 2007.
- [Sha85] A. Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in Cryptology*, pages 47–53, New York, NY, USA, 1985. Springer-Verlag New York, Inc.
- [SSH<sup>+</sup>07] J. Su, J. Scott, P. Hui, J. Crowcroft, E. Lara, C. Diot, A. Goel, M. Lim, and E. Upton. Huggle: Seamless networking for mobile applications. In *UbiComp 2007: Ubiquitous Computing*, volume 4717 of *Lecture Notes in Computer Science*, pages 391–408. Springer Berlin Heidelberg, 2007.