PrivHab+: a Secure Geographic Routing Protocol for DTN

Adrián Sánchez-Carmona[a,*], Sergi Robles[a], Carlos Borrego[a]

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

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.

1. Introduction and Motivation

Many initiatives have been implemented to improve the life conditions of people living in developing countries by universalizing 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 abscence of simultaneous end-to-end paths [**?**] 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 behaviors, as the historic of contacts established with each other node [**?**]. Then, they share this information with neighbours in order to improve the decision making [**?**]. Moreover, in some cases, a node is linked to a person, e. g. because it is carried in a pocket or backpack [**?**], or because they travel in the same vehicle. Therefore, the information that routing protocols use and share can be seen as private information about people's whereabouts or frequent behaviors. 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 [**?**]. Accordingly, a protocol that protects the privacy of this information expands the amount of scenarios where it can be used [**?**].

Our main contributions are summarized 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 organized 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,

*Corresponding author

Email addresses: adria.sanchez@deic.uab.cat (Adrián Sánchez-Carmona), sergi.robles@deic.uab.cat (Sergi Robles), carlos.borrego@deic.uab.cat (Carlos Borrego)

we introduce the concepts of homomorphic cryptography and Taxicab geometry, both needed to preserve nodes' privacy while routing using the habitat. In Section 4, 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 [**?**] 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 [**?**], 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 scenarios where the topology of the network changes quickly. Using LAROD [**?**], 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 [**?**], 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 [**?**] 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 [**?**] 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 [**?**]. 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 [**?**], 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 [**?**], 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. 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 [**?** **?**] provide anonymity to the nodes of the network by breaking the relation between nodes and identifiers. This way, an observer can not 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 [**?**], 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 [**?**] and

[?], 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 [?], 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 [?] 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 [?] 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 [?] 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 [?] 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 [?], 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 [?] 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.

Unfortunately, although at [?] the authors recognize that privacy is an important issue to consider and that more work is needed to solve it, [?] 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 [? ?].

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 Haggle [?], 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 [?] 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 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[1]. 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. Figure 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,

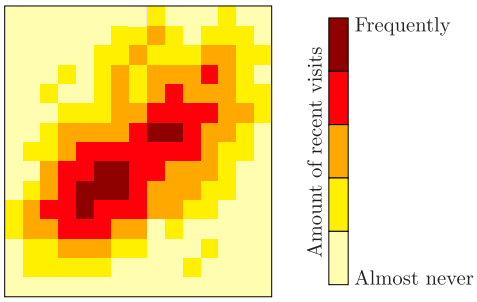

Figure 1: Example of habitat represented with a heatmap. The darker the colour used to depict an area, the more frequently visited it is.

eventually, in a location where he has not been never before. However, it will be far more likely to found him in the darker areas, where he has been repeatedly in the recent past. PrivHab+ makes use of this 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[2]. 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[3]. 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[4] to cartesian coordinates (x, y) and that this mapping is known by all the

[1] 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.

[2] 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.

[3] 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

[4] Any cartographic projection can be used.

nodes of the network. With a frequency of ω 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' behaviors.

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 $r = 0$.

$$H_0 = (L_0, L_0, 0) \tag{1}$$

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 Figure 2.

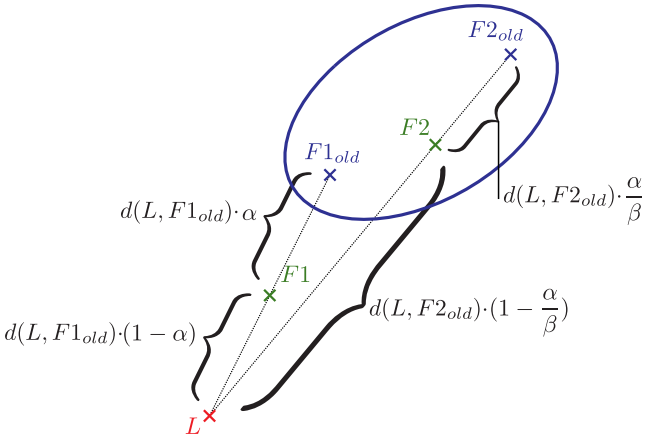

Figure 2: Evolution of the focal points $F1_{old}$ and $F2_{old}$ when the new location L is used to update the habitat. Function $d(L, F)$ denotes distance between L and a focal point F. Note that $F1$ has been attracted by L using an α factor while $F2$ has been attracted using a lesser $\frac{\alpha}{\beta}$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

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

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

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 more the nearest focal point, modifying the habitat's eccentricity depending on the relative position of L and H_{old}. The higher the β used, the more will change the form factor of the habitat when new distant samples are taken[5].

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 Figure 3.

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

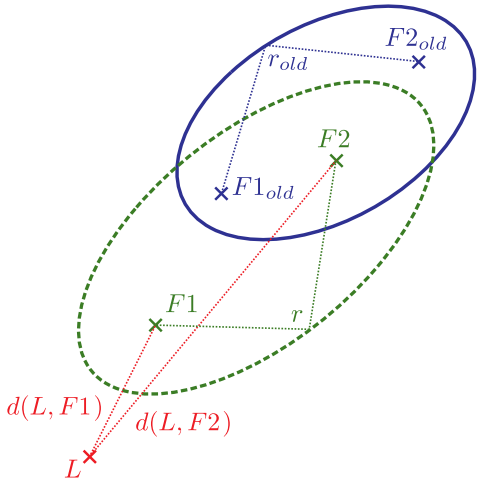

Figure 3: Evolution of the radius. Distances and radius are depicted with dotted lines. The old radius r_{old} is used together with the distances $d(L, F1)$ and $d(L, F2)$ that separate the updated focal points $F1$, $F2$ and the new location L to update the radius r. The radius of the habitat will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

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 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[6] 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.

[5] 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 $\beta > 50$.

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

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 α using Equation 5. Let ω 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 α 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 α 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[7]. A fully homomorphic cryptosystem, like [**?**], 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 [**?**], 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[8] 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 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 the 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 [**?**].

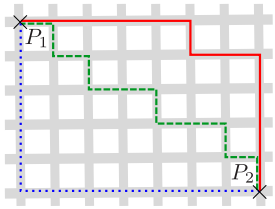

Figure 4: Taxicab geometry distances. All three pictured lines have the same length for the route between P_1 and P_2.

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[9] and is depicted in Figure 4. Manhattan distances can be calculated without computing any square root[10], 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. Figure 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

[7] Sections 4 and 5 will provide more details about this process.

[8] 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 [**?**].

[9] 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.

[10] 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, we explain how to calculate it benefiting from Taxicab geometry properties.

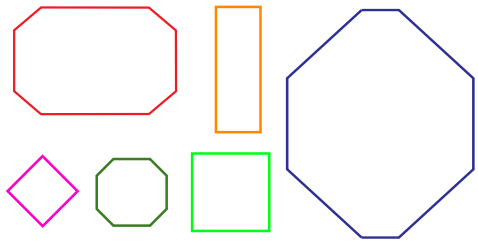

Figure 5: Examples of ellipses in Taxicab geometry. The circle (down, at the left), the square (the third figure at the down row) and the rectangle (above the square) are specific types of ellipses.

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

Finally, Figure 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.

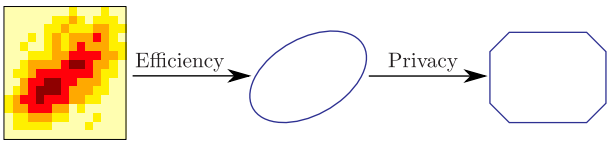

Figure 6: The real habitat is modelled using a simple shape as the ellipse due to efficiency reasons. Then, the Euclidean geometry is substituted by the Taxicab geometry in order to protect nodes' privacy.

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.

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.
$H : (F1, F2, r)$	A habitat.
H_i	The habitat of node i.
r_i	Radius of the habitat of node i.
$F1 : (f1_x, f1_y)$	One of the focal points of a habitat or ellipse.
$F2 : (f2_x, f2_y)$	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.
$E_Y(\cdot)$	Paillier additive homomorphic encryption function using Y's public key.
$D_Y(\cdot)$	Paillier additive homomorphic decryption function using Y's private key.

Table 1: Notation of all elements used in this Section.

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 [**?**]). 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 [**?**] and [**?**], or via the usage of an alternate communication channel. There are three different situations as depicted in Figure 7, where our routing algorithm has to decide who is the best option:

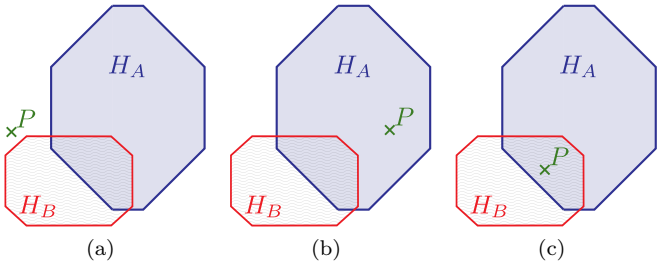

Figure 7: The three possible situations in habitat-based routing: (a) The next waypoint is located outside the two habitats; (b) Only one of the two habitats encloses the location of the next waypoint; (c) The two habitats enclose the location of the next waypoint.

(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 Figure 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 Figure 7).

(c) If P is located inside both habitats, then the best choice will be the node whose habitat is smaller (H_B in Figure 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[11] with pseudonym generator mechanisms as [**?**] or [**?**] that generate pseudonyms of the destination or the forwarders using its public key, or [**?**] 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.

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 subsections, 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 Equation 6.

$$d(F, P) = |F_x - P_x| + |F_y - P_y| \tag{6}$$

Then, we define E', the closest point of the border of E to P. We split these two distances in two parts: the part that is contained within ellipse; and the part that is outside the ellipse[12].

$$d(F1, P) = d(F1, E') + d(E', P)$$

[11] 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 Subsection 4.5 for more details.

[12] Note that, in Euclidean geometry, the distance between a point and an ellipse can not be calculated this way because Equation 7 only holds in Taxicab geometry.

$$d(F2,P) = d(F2,E') + d(E',P) \quad (7)$$

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, we can take advantage of Equation 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 Equation 9 to obtain the absolute value needed to calculate the distance from a point to the habitat (see Equation 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 $F2 : (F2_x, F2_y)$.

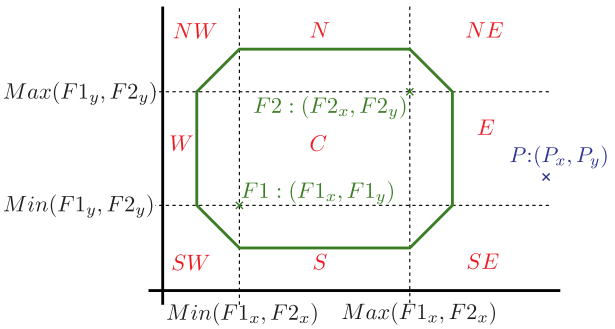

Figure 8: The regions of the space (NW, N, NE, W, C, E, SW, S and SE) are defined in the basis of the coordinates of the focal points $F1$ and $F2$. In the example shown, P is located in region E, and when we know this we can calculate the distances $d(F1,P)$ and $d(F2,P)$.

So we first divide the space into 9 regions, depending on their relation to the two focus of the habitat, as depicted in Figure 8. To know the region where P is located, we calculate the maximum and minimum values of the coordinates of the two focus using Equation 10. Then we compare them with the coordinates of P.

$$\begin{aligned} Fx_{min} &= Min(F1_x, F2_x) \\ Fx_{max} &= Max(F1_x, F2_x) \\ Fy_{min} &= Min(F1_y, F2_y) \\ Fy_{max} &= Max(F1_y, F2_y) \end{aligned} \quad (10)$$

Once we know the region where P is located, we can use Equations 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.

$d(F1,P)+$ $d(F2,P)$	$P_x \leq Fx_{min}$	$Fx_{min} < P_x$ $P_x \leq Fx_{max}$	$P_x > Fx_{max}$
$P_y > Fy_{max}$	$(Fx_{max} - P_x)+$ $(Fx_{min} - P_x)+$ $(P_y - Fy_{max})+$ $(P_y - Fy_{min})$	$(Fx_{max} - P_x)+$ $(P_x - Fx_{min})+$ $(P_y - Fy_{max})+$ $(P_y - Fy_{min})$	$(P_x - Fx_{max})+$ $(P_x - Fx_{min})+$ $(P_y - Fy_{max})+$ $(P_y - Fy_{min})+$
$y \leq Fy_{max}$ $Fy_{min} < P_y$	$(Fx_{max} - P_x)+$ $(Fx_{max} - P_x)+$ $(Fy_{max} - P_y)+$ $(P_y - Fy_{min})$	0	$(P_x - Fx_{max})+$ $(P_x - Fx_{min})+$ $(Fy_{max} - P_y)+$ $(P_y - Fy_{min})$
$P_y \leq Fy_{min}$	$(Fx_{max} - P_x)+$ $(Fx_{max} - P_x)+$ $(Fy_{max} - P_y)+$ $(Fy_{min} - P_y)$	$(Fx_{max} - P_x)+$ $(P_x - Fx_{min})+$ $(Fy_{max} - P_y)+$ $(Fy_{min} - P_y)$	$(P_x - Fx_{max})+$ $(P_x - Fx_{min})+$ $(Fy_{max} - P_y)+$ $(Fy_{min} - P_y)$

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.

After $d(F1,P)+d(F2,P)$ is obtained from Table 2, the last thing to do is to subtract the radius r, using Equation 8 to obtain $2 \cdot d(H,P)$, the double of the distance between P and the habitat H.

Finally, Equation 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 characterized 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 Equation 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 Equation 9, the absolute value

of the distance will be a negative value[13]. 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 Equation 13.

$$d(H, P) \leq 0 \iff P \in H \tag{13}$$

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 Equation 14 holds, otherwise, E_2 is the smaller one.

$$r_1 < r_2 \tag{14}$$

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.

$$\begin{aligned}(r_A - r_B) * nonce \geq 0 &\iff H_A > H_B \\ (r_A - r_B) * nonce < 0 &\iff H_A < H_B\end{aligned} \tag{15}$$

Note on Equation 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 randomized response. Later, the other part will binarize 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 Equation 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[14].

$$B \rightarrow A: \quad \begin{array}{l} E_B(Fx_{max}), E_B(Fx_{min}), \\ E_B(Fy_{max}), E_B(Fy_{min}) \end{array}$$

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 randomize it, with a random one-use value denoted *nonce*. A compares Fx_{max} with P_{xi} using Equation 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}.

$$\left(\frac{E_B(Fx_{max})}{E_B(P_{xi})}\right)^{nonce} = E_B((Fx_{max} - P_{xi}) \cdot nonce) \tag{16}$$

   Then A sends the comparisons[15] to B together with the coordinates of each P_i, the distance d_{Ai} and the radius r_A of H_A.

$$A \rightarrow B: \quad \begin{array}{l} E_A(r_A), \{E_B((Fx_{max} - P_{xi}) \cdot nonce), \\ E_A(P_{xi}), E_B((Fx_{min} - P_{xi}) \cdot nonce), \\ E_A(P_{yi}), E_B((Fy_{max} - P_{yi}) \cdot nonce), \\ E_A(2d_{Ai}), E_B((Fy_{min} - P_{yi}) \cdot nonce)\}^i \end{array}$$

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 Equation 17, and the comparison of radius[16] r_A and r_B using Equation 18.

$$\left(\frac{E_B(2d_{Ai})}{E_B(2d_{Bi})}\right)^{nonce} = E_B((2d_{Ai} - 2d_{Bi}) \cdot nonce) \tag{17}$$

[13] Note that our protocol checks several times if an operand ρ is positive or negative. In the Paillier cryptosystem, ρ will be an element of $\mathbb{Z}_n$. To check this condition, if we ensure that n is sufficiently large and that all values ρ we will use are $\rho \leq n/2$, then we can consider that $\rho > n/2 \iff \rho < 0$.

[14] This announcement can be made during the neighbour discovery process, by adding this information to the beacons.

[15] We have used "{" and "}^{i}" to enclose the part of the information that is repeated one time for each message m_i.

[16] 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.

$$\left(\frac{E_A(r_A)\cdot E_A(2d_{Ai})^{r_B}}{E_A(r_B)}\right)^{nonce}$$
$$= E_A((r_A + 2d_{Ai}\cdot r_B - r_B)\cdot nonce) \quad (18)$$

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.

$$B \rightarrow A: \quad \begin{array}{c} \{E_A((2d_{Ai} - 2d_{Bi})\cdot nonce), \\ E_A((r_A + 2d_{A_i}\cdot r_B - r_B)\cdot nonce) \\ \text{or} \\ E_A((r_A + 2d_{A_i}\cdot r_B - r_B)\cdot nonce), \\ E_A((2d_{Ai} - 2d_{Bi})\cdot nonce)\}^i \end{array}$$

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.

$$A \rightarrow B: \quad \{m_i\}^i$$

Figure 9 provides a schema of the messages exchanged during each phase of the protocol.

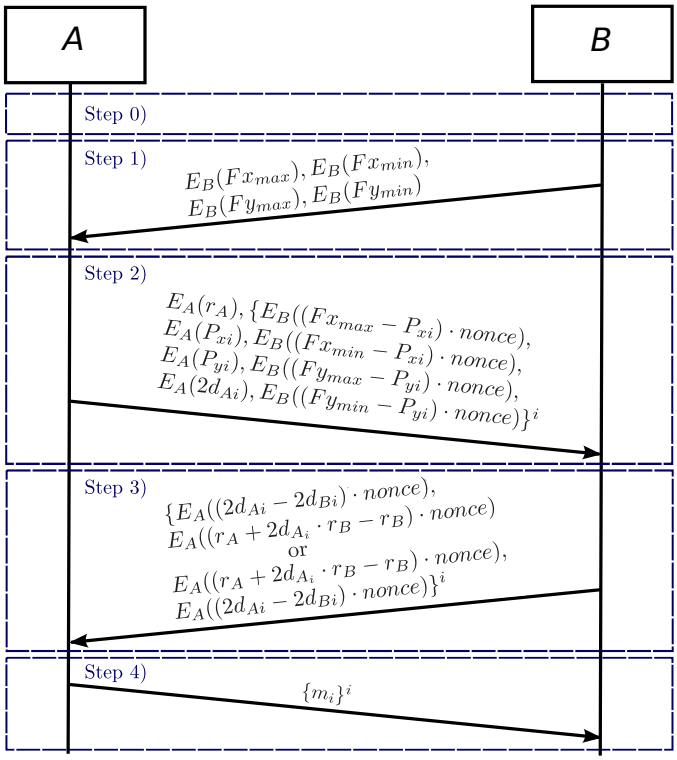

Figure 9: Schema of the messages exchanged during the execution of PrivHab+. At Step 0) the two nodes asynchronously perform calculations that will be used during the protocol. At Step 1) node B uses the neighbour discovery process to send to A the characteristics of the habitat H_B. At Step 2) node 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, randomizes 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.

4.6. 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 [**?**], the authors have studied the enormous complexity of realizing a multi-party secure comparison between an indefinite number of nodes. Besides, encounters between two nodes are the most common [**?**], 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.

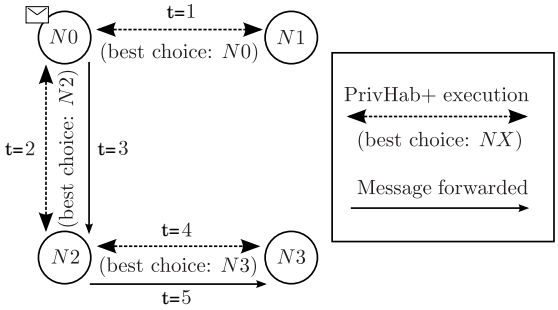

Figure 10: Node $N0$ carrying a message meets $N1$, $N2$ and $N3$. Numbers denote the order of the operations. $N0$ compares itself with $N1$ using PrivHab+ and finds that $N1$ is a worse choice, so it does not forward the message. Then $N0$ compares itself with $N2$, who results to be a better choice, 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.

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. Figure 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 $N1$.

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 adversay mode

A secure multi-party computation [**?**] 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 [**?**]. 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 summarizes 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 randomized through the usage of random *nonce* values.

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

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.

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.

B **knows**	B **learns**		B **infers**
Output	About P	About d_B	$d_A \leftrightarrow d_B$
Message received	$P : (P_x, P_y)$	d_B	$d_A \geq d_B$
Message do not received	Region where P is located	Nothing	$d_A \leq d_B$

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.

Table 4 summarizes 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. Since H_A characteristics are encrypted with A's key, and the comparisons that B can decrypt are randomized 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[17] 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[18] 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.

[17]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.

[18]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.

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 H_B. The knowledge that A can obtain from this is summarized by Table 5.

A **knows**	A **learns**	**Useful**
Output	About d_B	**iff**
A	$d_B \geq d'_A$	$d'_A > d_A$
B	$d_B \leq d'_A$	$d'_A < d_A$

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.

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 (≥ 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 repeteadly 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 summarizes this process.

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	$r'_A \geq r_B$	$r'_A < r_A$

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 $P1'$ and $P2'$ randomnly, 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.

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 summarizes all knowledge learned by B in these two cases.

B **knows**	B **learns**	**Useful**
Output	About d_A	**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

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.

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.

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 subsection 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 serie 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 mea-

surements of the computational and communication overhead introduced by the presented protocol.

6.1. Implementation details

We have deployed an implementation of the presented protocol on two different sets of devices: three Raspberry Pi boards[19], and two i5 laptops[20]. 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 Figure 9. The obtained results are shown in Table 9, and have been incorporated to the simulations.

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 seconds when using keys of 1024 bits. Given the average length of connectivity windows in remote village scenarios presented in [**?**], this overhead is acceptable. 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

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

[20]Intel Core-i5 (third generation): dual core 3,3 GHz, 4GB 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).

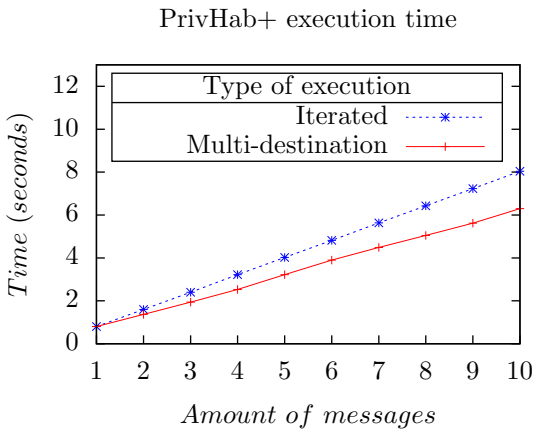

Figure 11: Execution time between the two different strategies to execute PrivHab+ with multiple messages to send, in a Raspberry Pi using keys of 512 bits. Executing the whole protocol one time for each destination lasts around 20% more than performing one multi-destination execution.

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. Figure 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. 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, 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.

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%

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)	Time (ms)	Overhead 10MB (%)	Overhead 20MB (%)
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

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 10MB or 20MB.

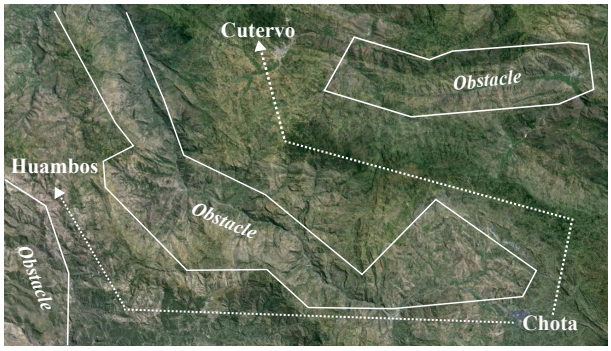

Figure 12: Map of a scenario of application located in a rural area of Cajamarca (Perú). White lines are natural obstacles approximate limits. Dotted white lines represent the pathways where messages sent from the village of Chota to Cutervo or to Huambos have to be routed through. The size of the area under consideration is 30x30Km.

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*[21] 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 Figure 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*[22] 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

[21] More information about this programme at http://practicalaction.org/podcasting-3

[22] More information about this programme at http://practicalaction.org/podcasting-gwanda

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.

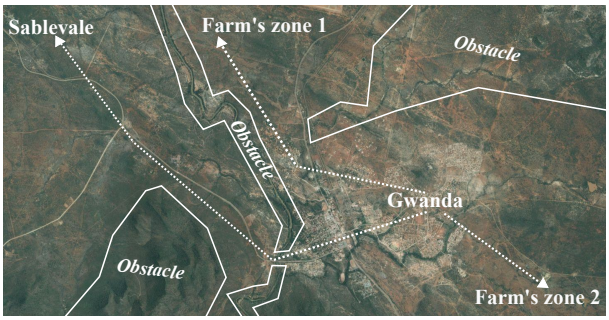

Figure 13: Map of a scenario of application located in Gwanda (Zimbabwe). White lines are natural obstacles approximate limits. Dotted white lines represent the pathways where messages sent from the InfoCenter of Gwanda to Sablevale and the two farm's zones have to be routed through.

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 (15x7Km) and, as shown in Figure 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+ 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 incentivize them to install an app, it is important that PrivHab+ guarantees their privacy. The same way, we can not 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 200MB buffer and a wireless interface featuring a communication range of 30 meters and speed up to 500Kbps. Messages of 10-20MB[23] 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 on Table 10.

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

Table 10: Number and type of the nodes involved in the simulations of each scenario.

During the approximation phase nodes calculate their habitat as explained in Section 3, and the protocol detailed in Section 4 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.

[23]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 minutes.

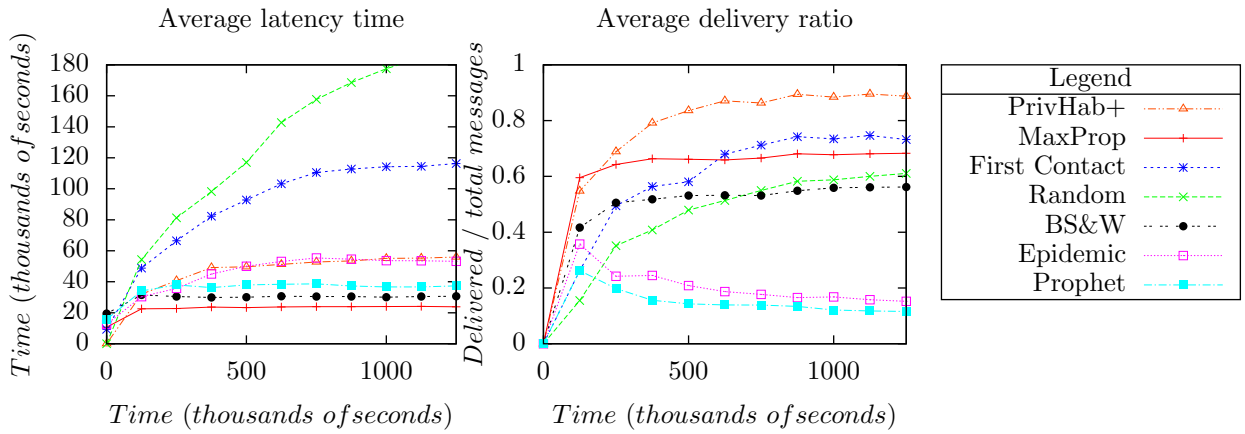

Figure 14: Obtained results in terms of latency and delivery ratio in the Cajamarca scenario. PrivHab+ and MaxProp perform far better than the rest, obtaining a low latency and a high delivery ratio.

In both scenarios, we have compared the performance obtained by PrivHab+ (using $T = 48$ hours 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 [**?**]: Prophet [**?**], Binary Spray & Wait (L=40) [**?**], Epidemic and Random [**?**]. We have added two routing protocols to this set: MaxProp [**?**] and First Contact. All simulations have been performed using *The Opportunistic Network Simulator* (The ONE) [**?**], 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 Figure 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.

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	9,929	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: 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.

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, Epi-

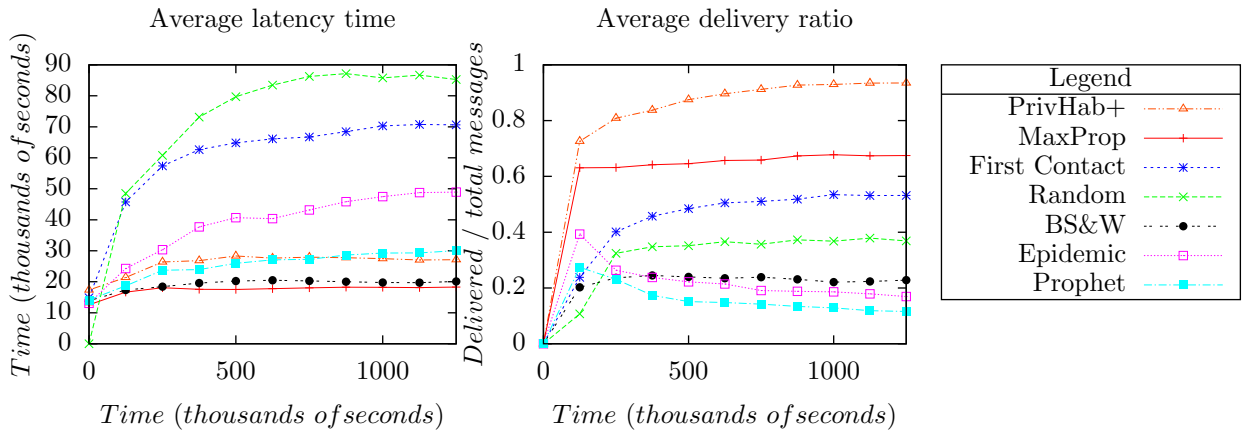

Figure 15: Obtained results in terms of latency and delivery ratio in the Gwanda scenario. PrivHab+ and MaxProp perform far better than the rest, obtaining a low latency and a high delivery ratio.

demic 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 Figure 15, where the performance of all the compared protocols in terms of delivery ratio and latency is depicted. In comparison with the results of Figure 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 3$^{\text{rd}}$ and 4$^{\text{th}}$ 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 3$^{\text{rd}}$).

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

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 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	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

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.

destination before) that are good choices because of their habitats. PrivHab+ recognize 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 (5.5s), 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 summarizes 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 transitiveness 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 analyzed as a secure multi-party computation to proof 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 minimizes 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.

9. 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] Yingjie Li 0003, Ten-Hwang Lai, Ming T. Liu, Min-Te Sun, and Junmo Yang. Dtgr: Disruption-tolerant geographic routing for wireless ad hoc networks. *Simulation*, 82(6):399–411, 2006.

[2] Chiara Boldrini, M. Conti, Jacopo 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 on a*, pages 1–12, June 2007.

[3] Carlos Borrego, Sergio Castillo, and Sergi Robles. Striving for sensing: Taming your mobile code to share a robot sensor network. *Information Sciences*, (0), 2014.

[4] A. Boukerche, K. El-Khatib, Li Xu, and L. Korba. Sdar: a secure distributed anonymous routing protocol for wireless and mobile ad hoc networks. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 618–624, Nov 2004.

[5] J. Burgess, Brian Gallagher, D. Jensen, and B.N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–11, 2006.

[6] Pei-Chun Cheng, KevinC. Lee, Mario Gerla, and Jérôme Härri. Geodtn+nav: Geographic dtn routing with navigator prediction for urban vehicular environments. *Mobile Networks and Applications*, 15(1):61–82, 2010.

[7] Yih chun Hu. Ariadne: A secure on-demand routing protocol for ad hoc networks. pages 12–23, 2002.

[8] Keith B. Frikken. Algorithms and theory of computation handbook. chapter Secure Multiparty Computation, pages 14–14. Chapman & Hall/CRC, 2010.

[9] Gerard Garcia, Sergi Robles, Adria Sánchez, and Carlos Borrego. Information system for supporting location-based routing protocols. 2014.

[10] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.

[11] Oded Goldreich. Secure multi-party computation, 1998.

[12] Samo Grasic and Anders Lindgren. Revisiting a remote village scenario and its dtn routing objective. *Computer Communications*, 48:133–140, 2014.

[13] S. Guo, M. H. Falaki, E. A. Oliver, S. Ur Rahman, A. Seth, M. A. Zaharia, and S. Keshav. Very low-cost internet access using kiosknet. *SIGCOMM Comput. Commun. Rev.*, 37(5):95–100, October 2007.

[14] Wei-jen Hsu, Debojyoti Dutta, and Ahmed Helmy. CSI: A paradigm for behavior-oriented delivery services in mobile human networks. *CoRR*, abs/0807.1153, 2008.

[15] Yih-Chun Hu, David B. Johnson, and Adrian Perrig. Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks, 2003.

[16] Pan Hui, J. Crowcroft, and E. Yoneki. Bubble rap: Social-based forwarding in delay-tolerant networks. *Mobile Computing, IEEE Transactions on*, 10(11):1576–1589, Nov 2011.

[17] Rui Jiang and Yuan 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 on*, pages 406–411, Oct 2012.

[18] Chris Karlof and David Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures, 2003.

[19] Brad Karp and H. T. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom '00, pages 243–254, New York, NY, USA, 2000. ACM.

[20] A. Kate, G.M. Zaverucha, and 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*, pages 504–513, Sept 2007.

[21] Ari Keränen, Jörg Ott, and Teemu 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.

[22] Eugene Krause. *Taxicab Geometry : an adventure in non-Euclidean geometry*. Dover Publications, New York, 1987.

[23] E. Kuiper and S. Nadjm-Tehrani. Geographical routing in intermittently connected ad hoc networks. In *Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008. 22nd International Conference on*, pages 1690–1695, March 2008.

[24] E. Kuiper and S. Nadjm-Tehrani. Geographical routing with location service in intermittently connected manets. *Vehicular Technology, IEEE Transactions on*, 60(2):592–604, Feb 2011.

[25] J. LeBrun, Chen-Nee Chuah, D. Ghosal, and M. Zhang. Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks. In *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, volume 4, pages 2289–2293 Vol. 4, May 2005.

[26] Jérémie Leguay, Timur Friedman, and Vania Conan. Evaluating mobyspace-based routing strategies in delay-tolerant networks. *Wireless Communications and Mobile Computing*, 7(10):1171–1182, 2007.

[27] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, July 2003.

[28] M. Liskov and R. Silverman. A statistical limited-knowledge proof for secure rsa keys. Technical report, IEE P1363 working group, 1998.

[29] Xiaofeng Lu, Pan Hui, Don Towsley, Juahua Pu, and Zhang Xiong. Anti-localization anonymous routing for delay tolerant network. *Computer Networks*, 54(11):1899 – 1910, 2010.

[30] M.E. Mahmoud and Xuemin Shen. Lightweight privacy-preserving routing and incentive protocol for hybrid ad hoc wireless network. In *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pages 1006–1011, Apr. 2011.

[31] A Mei, G. Morabito, P. Santi, and J. Stefa. Social-aware stateless forwarding in pocket switched networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 251–255, April 2011.

[32] M. Musolesi and C. Mascolo. Car: Context-aware adaptive routing for delay-tolerant mobile networks. *Mobile Computing, IEEE Transactions on*, 8(2):246–260, Feb 2009.

[33] Helena Rifà-Pous and Jordi Herrera-Joancomartí. Secure dynamic manet on-demand (sedymo) routing protocol. In *Communication Networks and Services Research (CNSR)*, pages 372–380, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[34] Adrián Sánchez-Carmona, Sergi Robles, and Carlos 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*, pages 1697–1698, 2015.

[35] Adrián Sánchez-Carmona, Sergi Robles, Carlos Borrego, and Gerard 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*, pages 1943–1944, 2015.

[36] K. Scott and S. Burleigh. Bundle Protocol Specification. RFC 5050 (Experimental), November 2007.

[37] Joo-Han Song, Vincent W.S. Wong, and Victor C.M. Leung. Secure position-based routing protocol for mobile ad hoc networks. *Ad Hoc Networks*, 5(1):76 – 85, 2007. Security Issues in Sensor and Ad Hoc Networks.

[38] T. Spyropoulos, K. Psounis, and C.S. 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*, page 259. ACM, 2005.

[39] Thrasyvoulos Spyropoulos, Rao Naveed Rais, Thierry Turletti, Katia Obraczka, and Athanasios Vasilakos. Routing for disruption tolerant networks: Taxonomy and design. *Wirel. Netw.*, 16(8):2349–2370, November 2010.

[40] Jing Su, James Scott, Pan Hui, Jon Crowcroft, Eyal Lara, Christophe Diot, Ashvin Goel, MengHow Lim, and Eben Upton. Haggle: Seamless networking for mobile applications. In John Krumm, GregoryD. Abowd, Aruna Seneviratne, and Thomas Strang, editors, *UbiComp 2007: Ubiquitous Computing*, volume 4717 of *Lecture Notes in Computer Science*, pages 391–408. Springer Berlin Heidelberg, 2007.

[41] Xiaoxin Wu and B. Bhargava. Ao2p: ad hoc on-demand position-based private routing protocol. *Mobile Computing, IEEE Transactions on*, 4(4):335–348, July 2005.

[42] Ge Zhong, Ian Goldberg, and Urs Hengartner. Louis, lester and pierre: Three protocols for location privacy. In Nikita Borisov and Philippe Golle, editors, *Privacy Enhancing Technologies*, volume 4776 of *Lecture Notes in Computer Science*, pages 62–76. Springer Berlin Heidelberg, 2007.

[43] Bo Zhu, Zhiguo Wan, Mohan S. Kankanhalli, Feng Bao, and Robert H. Deng. Anonymous secure routing in mobile ad-hoc networks, 2004.