Information System for Supporting Location-based Routing Protocols

Gerard Garcia, Sergi Robles, Adrià Sánchez, Carlos Borrego
Departamento de Ingeniería de la Información y de las Comunicaciones
Universidad Autónoma de Barcelona
Email: {*ggarcia, sergi.robles, adria.sanchez, cborrego*}@deic.uab.cat

***Abstract*—This article presents an information system for location-based routing protocols that does not compromise the privacy of the involved nodes. This information system provides a representational model of the most frequented locations of a node, this most frequented zone is called its habitat, and a protocol to compare these habitats among nodes given a target location of interest. Then, the protocol can determine which of the neighbors of a node is nearer or farther from this target location and provides this information to the underlying routing protocol. As it is designed for DTNs, the protocol does not require a trusted third party, instead, it implements a secure multi-party computation based on homomorphic encryption. The protocol is demonstrated to be secure against passive and active adversaries.**

***Index Terms*—Secure multi-party computation, Delay and Disruption Tolerant Network, DTN routing protocol.**

I. Introduction

The Delay and Disruption Tolerant Network (DTN) [7] architecture emerged from the research for developing an interplanetary network. This research was focused in solving the technical difficulties found in out of space networking, p.e. frequent disconnections or slow network links. But as these characteristics could also be found in some scenarios on the Earth, the DTN term was coined in order to include them.

One of the most important challenges that DTNs have to face because of its characteristics is how to perform the routing of messages [16]. The differences between DTNs and traditional networks, such as the lack of an end-to-end circuit between the source and the destination or the fact that nodes can not have a global knowledge of the network due to their disconnected nature, make the routing protocols used in traditional networks ineffective. To overcome this, some routing protocols for DTNs generate metrics that model the behavior of the nodes in the network. Then, with the information provided by the model, the routing protocol can make substantiated decisions on how to forward the messages in order to try to maximize the delivery rate.

This paper presents an information system support for location-based routing protocols. This information system is composed of a representation model of the most frequented locations of a node and a protocol for determining, given n nodes, which of their habitats is nearer or farther of a target location.

This information support would allow a location-based routing protocol the maximization of the delivery rate of the messages by forwarding them to the nodes nearer to the destination location of the given message. Or it would also allow a routing protocol to forward the messages by paths that try to avoid specific zones by sending the messages to the nodes farther of these zones.

To implement the protocol would seem logical to think that the coordinates of the target location could be directly compared to the habitats of the nodes. But the privacy of the involved nodes must be taken into account, since this nodes could identify physical persons and revealing its location could be a threat and an invasion of their privacy. Therefore, the habitats comparisons are treated as a secure multi-party computation [13].

There are several solutions for performing a secure multi-party computation. The ones originally proposed by [9] and [15], and extended by many others, make use of a combinatorial circuit for representing the required computation. Parties execute then a short protocol for every gate of the circuit. The advantage of these approaches is that they are general methods, but the problem is that the protocol depends on the size of the circuit. Therefore, for complex computations these protocols can be inefficient. Other approaches, like the ones proposed by [17] or [2], design specific protocols based on, for example, homomorphic encryption or 1-out-of-N oblivious transfers, to solve specific problems. These solutions are more efficient, but are limited to the solving of these specific problems. The proposal in this paper uses a specific secure multi-party computation based on homomorphic encryption for efficiency reasons.

II. Habitat

This section first describes what is and how is represented an habitat. Then, it shows how the habitats of two nodes con be compared given a target location. And finally, shows how is calculated the distance between two points.

A. Description

The habitat of a node represents its most frequented locations and it is represented by a dynamically created ellipse from the historic of movement of the node. How is the ellipse created is not contemplated in this article. An ellipse can be defined as

ISBN: 978-84-9717-323-0

Definition 1. *The set of points such that the sum of the distances to two fixed points, the foci, is constant. This distance defines the radius of the ellipse.*

Hence, an habitat is defined with two foci points $F1 : (f1_x, f1_y)$ and $F2 : (f2_x, f2_y)$ and a radius r.

B. Comparison

When comparing the habitats of two nodes, three different situation may be found:

1) The target location is outside the two habitats. In this case the one nearer to the target is preferred.
2) The target location is inside the two habitats. In this case the node with the smallest habitat is preferred, as it is more probable that the node pass through this location earlier.
3) The target location is inside one of the habitats but outside the other. In this case the preferred node is the one with the target location inside its habitat.

To solve each one of the previous situations, it is necessary to solve the following three problems:

1) How to calculate the distance from an habitat to a target location.
2) Given two habitats, how to determine which one is smaller.
3) How to determine if the target location is inside an habitat.

1) Distance from a target location to an habitat: It is necessary to calculate the distance from the target location defined by a point $P : (x, y)$ to an habitat H, determined by the ellipse with foci points $F1 : (f1_x, f1_y)$ and $F2 : (f2_x, f2_y)$ and radius r.

First, it is defined the point $X : (a, b)$ as the nearest point of the habitat H to the point P, so it would need to comply the next equation

$$|a - f1_x| + |b - f1_y| + |a - f2_x| + |b - f2_y| = r \tag{1}$$

Then, it is defined the function distance as follows

$$d(X, P) = |a - x| + |y - b| \tag{2}$$

and it is minimized while restricted by equation 1, for example with the method of Lagrange multipliers, to get the point X. Finally, to obtain the distance is applied the function distance 2 with the point X and P.

2) Which habitat is smaller: To know which habitat is smaller, are compared the radius of the two habitats. Given two habitats H_1 and H_2, with radius r_1 and r_2 respectively, the one with the smallest radius is the smallest habitat

$$\begin{aligned} r_1 < r_2 &\implies H_1 \\ r_2 < r_1 &\implies H_2 \end{aligned} \tag{3}$$

3) Point inside an habitat: This problem can be resolved as the first one. If the distance obtained is negative or 0, then the point is inside the habitat.

C. Manhattan Geometry

To simplify the previous calculations it is used the definition of distance that Manhattan Geometry [5] provides. In Manhattan Geometry the function distance of two points is defined as the sum of the absolute differences of their Cartesian coordinates. Formally

$$d(p, q) = \sum_{i=1}^{n} |p_i - q_i| \tag{4}$$

where p and q are two points. This way it is simpler to calculate the distance while still can be compared, as it maintains the proportions. This simplification will be of interest when calculating the distance in the secure multi-party computation as the operations that can be performed will be limited by the use of homomorphic encryption,

III. Cryptographic protocol for habitat comparisons

In this section first will be described how the problems for comparing two habitats, described in section II, are solved such that the privacy of the involved nodes remains unaffected. Then, it will be showed how the protocol works for determining, given n nodes, which one, or ones, are nearer or farther to the target location. The protocol will implement a secure multi-party computation based on homomorphic encryption to perform the calculations needed to compare the habitats of the nodes.

A. Homomorphic encryption

An encryption scheme is considered homomorphic if given the set of plain-texts $\mathcal{M}$, the set of the cypher-texts $\mathcal{C}$ and the encryption function $\mathcal{E}$, it satisfies

$$\forall m_1, m_2 \in \mathcal{M}, \mathcal{E}(m_1 \odot_{\mathcal{M}} m_2)) \leftarrow \mathcal{E}(m_1) \odot_{\mathcal{C}} \mathcal{E}(m_2) \tag{5}$$

for some operators $\odot_{\mathcal{M}}$ in $\mathcal{M}$ and $\odot_{\mathcal{C}}$ in $\mathcal{C}$ [8]. If the encryption scheme only satisfies this property for one operation, e.g. multiplication or addition, it is considered partially homomorphic.

For this protocol are used the homomorphic properties of the cryptosystem proposed by P. Paillier in [11], known as the Paillier cryptosystem. This cryptosystem is additively homomorphic, computationally efficient and it allows the multiplication of cyphered-texts by unencrypted constants without the need of decrypting the operands. Therefore, the set of operations cryptographically protected that can be performed are: sum, subtraction and multiplication of an encrypted value by a non-encrypted constant.

B. Secure comparison

The use of homomorphic encryption limits the operations that can be performed over the encrypted operands, therefore, the previous comparison process needs to be adapted to overcome these limitations.

The problem appears when the distance from an habitat to a target location is calculated. To calculate the distance it is necessary to first find the nearest point of the habitat to the target location and then use the found point to calculate the distance, as described in II-B1. But if the operands are encrypted under homomorphic encryption it is not possible to calculate their absolute value, therefore this operation needs to be disposed.

To do so, it is determined where is the target location situated in relation of the habitat to make sure that all the subtractions encode an absolute value.

If the space is divided into 9 regions, it can be determined in which region is the target location calculating the maximum and minimum values of the foci points $(Fx_{min}, Fx_{max}, Fy_{min}, Fy_{max})$ and comparing them with the coordinates of P. Once it is known in which region is situated the target location, the equations 1 and 2 can be redefined without the need of calculating absolute values and then minimized for each of the cases.

Note that the corner regions define b in terms of a. This is the line where X is located, but not any point of this line is the nearest to P. To know exactly where is the nearest point each one of these regions is divided in two subregions. To do so is calculated the left end LE and the right end RE of the habitat and then compared with the target location. These two ends are calculated with the following equations

$$\begin{aligned} LE &= Fx_{max} + Fx_{min} + Fy_{max} - Fy_{min} - r \\ RE &= Fx_{max} + Fx_{min} - Fy_{max} + Fy_{min} + r \end{aligned} \tag{6}$$

If the point is located between LE and RE, X and P share the x coordinate, so $a = x$, otherwise, $b = Fy_{max}$ in the two superior corners and $b = Fy_{min}$ in the other two.

C. Protocol

To describe the protocol it is imagined an scenario where A has multiple neighbors but only A can see all of them. Given that this situation can be quite common, the protocol has been designed such that A will coordinate all the messages. Therefore, the protocol is based in letting A have the information that other nodes need to compare between them so A can distribute it between the other nodes as it requires. Obviously, as the privacy needs to be preserved, this information is encrypted and only accessible by the node referred.

The protocol is divided in two phases, which are represented in figure 1. After the first phase, A has compared its own habitat to the habitats of its neighbors. And during the second phase the surviving nodes (the nodes that still satisfy the requisites of A) are compared among them, under the commands of A, until A decides that it has enough information for its purposes.

The neighbor discovery is conducted with the transmission and detection of periodically transmitted beacons. When a node detects a beacon, if it has messages to forward, it sends a beacon asking all the receiving nodes to announce its presence so it can detect all the neighbor nodes at once. After a time t_1 has elapsed, the protocol continues.

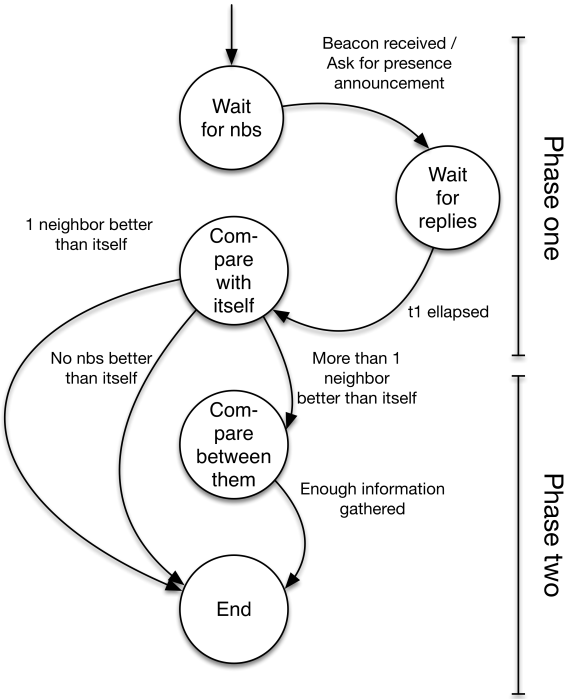
Wait for nbs
Beacon received / Ask for presence announcement
Wait for replies
t1 ellapsed
Com-pare with itself
1 neighbor better than itself
No nbs better than itself
More than 1 neighbor better than itself
Com-pare between them
Enough information gathered
End
Phase one
Phase two

Figure 1. State diagram of the proposed protocol

1) Phase one: Node A compare its habitat with the habitat of another node following the exchange of messages described in figure 2. A is the node that coordinates the protocol and B any other neighbor.

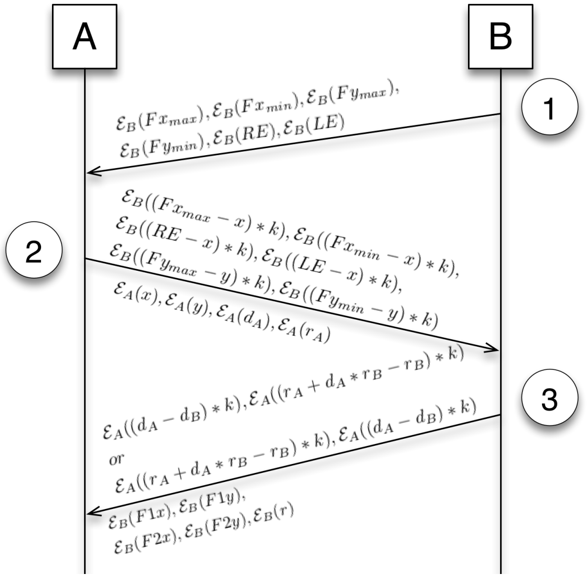
A
B
1
$\mathcal{E}_B(Fx_{max}), \mathcal{E}_B(Fx_{min}), \mathcal{E}_B(Fy_{max}),$
$\mathcal{E}_B(Fy_{min}), \mathcal{E}_B(RE), \mathcal{E}_B(LE)$
2
$\mathcal{E}_B((Fx_{max} - x) * k), \mathcal{E}_B((Fx_{min} - x) * k),$
$\mathcal{E}_B((RE - x) * k), \mathcal{E}_B((LE - x) * k),$
$\mathcal{E}_B((Fy_{max} - y) * k), \mathcal{E}_B((Fy_{min} - y) * k)$
$\mathcal{E}_A(x), \mathcal{E}_A(y), \mathcal{E}_A(d_A), \mathcal{E}_A(r_A)$
3
$\mathcal{E}_A((d_A - d_B) * k), \mathcal{E}_A((r_A + d_A * r_B - r_B) * k)$
or
$\mathcal{E}_A((r_A + d_A * r_B - r_B) * k), \mathcal{E}_A((d_A - d_B) * k)$
$\mathcal{E}_B(F1x), \mathcal{E}_B(F1y),$
$\mathcal{E}_B(F2x), \mathcal{E}_B(F2y), \mathcal{E}_B(r)$

Figure 2. Phase one: Exchange of messages to compare a given habitat and a target location known by A with the habitat of other nodes

① Node B sends the limits of its habitat together with its presence beacon.

- Node A subtracts to the limits the coordinates of the target location so node B can determine in which region is situated the target location respect its habitat. Note that all the results are obfuscated, by multiplying them by k, so node B can only know the situation of the target location respect its habitat and not the exact position of it.

② Node A transmits the subtracted limits together with the target location $P : (x, y)$, its distance d_A (which node A has calculated without any restriction, as he knows the target location an its habitat), and its radius r_A.

- Node B calculates $\mathcal{E}_{\mathcal{A}}(d_B)$ with the supplied target locations, then, it creates the tuple described in the figure. This is done this way to do not let node A know if the distances are equal (or 0) as it will not distinguish between the comparison of the distance and the comparison of the radius.

③ Node B transmits the results to node A, and as node A needs to have the required information so other nodes can compare among them, node B includes with the result its own habitat ($F1$, $F2$ and r), encrypted with its own key.

- To interpret the result node A checks the sign of the two values of the tuple, if the values are 0 or positive, node B is considered to be nearer than A to the target location.
- With the habitat and knowing in which region is the target location, node A can calculate the distance of that node to it, although it is encrypted and unaccessible for itself. But as node A does not know the region where is the target location situated, it calculates the distance for each of the regions, so it ends having 13 different distances, one for each region and subregion.

Once the protocol finishes, node A knows if node B is nearer or farther to the target location than itself, but it has not learned anything about the habitat of node A. And on the other hand, node B has not learned the target location, only the situation of its habitat respect to this target location and has been able to compare its habitat with the habitat of node A without learning it.

If no nodes satisfy the requisites of node A, or only one node satisfies them, the protocol ends. Otherwise, the protocol continues with phase two.

2) Phase two: Phase two starts with node A knowing which of its nodes satisfy its requisites, and having the distance of the habitat to the destination target location of these nodes, although the distance is encrypted and node A does not know which of the 13 distances is the correct one.

At this point, node A has to determine which comparisons need to be done to reach a decision. As a method to compare the habitat of any two nodes is provided, any filtering or sorting algorithm can be used. For example, if node A only wants to know which node is the nearest to the target location, A could just make all nodes compare randomly while discards the ones that lose a comparison until only one node is left. Or if A wants to sort all its neighbors from nearer to farther, it can perform a Quicksort ordering. This decision is beyond the scope of this article.

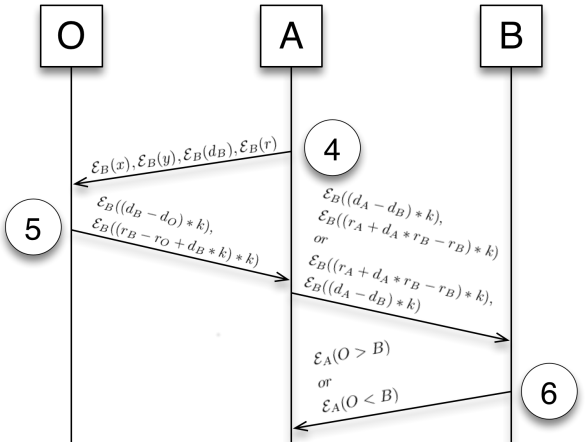
O
A
B
4
$\mathcal{E}_B(x), \mathcal{E}_B(y), \mathcal{E}_B(d_B), \mathcal{E}_B(r)$
5
$\mathcal{E}_B((d_B - d_O) * k),$
$\mathcal{E}_B((r_B - r_O + d_B * k) * k)$
$\mathcal{E}_B((d_A - d_B) * k),$
$\mathcal{E}_B((r_A + d_A * r_B - r_B) * k)$
or
$\mathcal{E}_B((r_A + d_A * r_B - r_B) * k),$
$\mathcal{E}_B((d_A - d_B) * k)$
$\mathcal{E}_A(O > B)$
or
$\mathcal{E}_A(O < B)$
6

Figure 3. Phase two: Exchange of messages to compare the habitat of B with the habitat of other nodes

To perform this comparisons node A starts the exchange of messages shown in figure 3. Being node B the node to be compared an O the others nodes to compare with B:

④ Node A sends to the other nodes the information that they need to compare themselves with node B: the distances of node B to the destination target location d_B (13 of them, one for each region and subregion), the target location $P : (x, y)$ and the radius r_B of the habitat of node B. All this values are encrypted with the key of B.

- This is the same information that they received from node A in the first step of the protocol, with the difference that it is encrypted by node B and that they do not know which one is the correct distance, so they need to compare for each one.
  With this information they calculate their distance to the target location again, but encrypted for node B, and they subtract it to each of the d_B provided.

⑤ The next step requires opening the result of the comparison. Only node B can open the results, therefore, there are two options:

- If they can see each other, they send the result directly to node B,
- If they can not see each other, they send it to node A so node A can forward it to node B.

Once node B has the result, it opens the correct one.

⑥ Finally, node B transmits the interpreted results to node A.

This comparisons can be done in both ways, therefore, if node A does not completely trust node B it can ask node B to compare itself with the other nodes and only accept the results that match.

D. Performance

The information transfered by the protocol can be calculated with the following formula, b is the size of each value, m is the number of messages ready to be forwarded by a given node and c is the number of comparisons performed in the phase two (which number will depend of the chosen sorting algorithm):

$$12 * b + 11 * b * m + (17 * b * m + 2 * b) * c$$

The size of each value will depend on the key used to encrypt them, being them twice the size of the key.

It is safe to assume that most DTNs have a window time of at least a few seconds, for example [4] and [12], and a bandwidth in the order of, at least, the hundreds of kilobytes per second, thus, the overhead of the protocol would not impact on the performance of the network.

Regarding computation performance, it will depend on the hardware of the nodes, but as the operations to encrypt and decrypt of the Paillier cryptosystem have low complexity, it is feasible to implement the protocol in any moderately powerful hardware.

Finally, note that this information system is designed to complement other routing protocols, therefore it should be used when its use is expected to improve the delivery ratio of the messages.

IV. Security Discussion

The global security of the proposed protocol is not determined by the individual behavior of the nodes, therefore, the security rests in three main points: the Paillier cryptosystem, the key management and the control of the information indirectly disclosed. The security of the first two points can be quickly determined: The Paillier cryptosystem is proved to be robust and secure [11] and the security of the key management is responsibility of the node as the keys are never transmitted during the protocol. More interesting is the analysis of the information disclosed.

To analyze the information disclosed by the execution of the protocol, the framework defined in [6] for the analysis of the security of multi-party computations will be used as starting point. This framework describes different adversary models and gives a generalization of the concept of ideal process, already proposed by other authors in [10] and [3].

In this framework a protocol is considered to securely perform a given task if executing the real protocol amounts to "emulating" the ideal process for that task. In the ideal process there is an incorruptible trusted party who receives the inputs of all the parties, locally computes the desired outputs and transmits them to the required parties. To define what is "emulating", first is necessary to formalize the output of performing a given task.

The output is formalized as the information that the task explicitly outputs in addition of what can be inferred. In other words, the information the task outputs once it has successfully finished and the information that can be deduced from the process of performing the task.

Now, emulating a task is performing it in such a way that its output is exactly the same as the ideal task. Thus, all parties will learn identical information from both the real protocol and the ideal process.

The adversaries covered in this analysis are classified in passive and active adversaries. Passive adversaries, also called, semi-honest, only gather information and do not modify the behavior of the parties. On the other hand active adversaries, also called "Byzantine", modify the outputs of the function so they can corrupt other parties to get more information.

To simplify the study of the security of the proposed protocol, it will be divided in two subtask. The subtask of comparing the habitats of two nodes when one of them knows the target location, and the subtask of comparing a given habitat and a target location to n other nodes. The first subtask corresponds to the first phase of the protocol, and the second subtask to the second phase. These phases are described in section III. Also, it will be distinguished between the security of the nodes (the privacy of their habitats) and the security of the target location (as it can be the location of a node).

The analysis will be performed as follows. For each subtask and adversary, it will be described the output of performing the subtask in the ideal process and then it will be compared with the output of performing the same subtask in the real protocol.

A. One-to-one habitat comparison subtask

In the step one of the protocol it is performed a comparison between the habitats of two nodes, one of them knowing the target destination. Then, the task of this subtask is the determination, between these two habitats, of which one is better suited to forward a message to this destination. The nodes will be called A and B, being A the one with the message to forward.

To perform the ideal process of this task, A would send its habitat and the target location P to the trusted third party using a secure channel and B its own habitat with another secure channel. The output that the trusted third party would transmit would be the communication to the A of which of the habitats is considered better. Therefore, A would not learn any additional information about the habitat of B and B would not learn anything about the habitat of A nor the target location.

Now, will be show how this subtask behaves against passive and security adversaries.

1) Passive adversaries: The execution of the ideal process in presence of passive adversaries would not lead to any opportunity for them to gather additional information. Otherwise, the real protocol reveals to B the region where the target location P is located regarding its habitat.

The revelation of the region is the only difference between the ideal process and the real protocol, but if it points to a node this is not a threat to its privacy as its location can be hidden by breaking the relation between the target location and the node, p.e. with the technique used in [14]. Therefore it can be stated that this part of the protocol does not compromise the security of the forwarding nodes nor the nodes to which the target location can refer in presence of passive adversaries.

2) Active adversaries: In the ideal process, the active adversaries can only modify the inputs sent to the trusted third party, on the other hand, in the real protocol, both A and B

can modify the values in the intermediate steps of the protocol to try to corrupt the other node.

If the modified values were the values operated with the encrypted data, the consequences would be the same than modifying the original inputs. Thus, this modifications would not affect the security of the protocol as it has been defined because this attacks could be also performed in the ideal process. In addition, no other alterations can be done that would give any advantage to the adversary as all the values are encrypted and any modification to these values would produce random uncontrolled outputs when decrypted.

Hence, it is possible to state that this part of the protocol is also secure against active adversaries as it does not compromise the security of the nodes not of the target location.

B. One-to-N habitat comparison subtask

The other subtask is the comparison of a given habitat among other n nodes. The task is determining which one or which ones, depending the sorting or filtering process used, fulfill the requirements of the initiator of the protocol. It starts at the last step of the first phase of the protocol, when the node gives, together with the result of the comparison, its encrypted habitat. This subtask performs the part of the protocol used in the previous subtask but with different parties, then, only the differences from the previous subtask need to be discussed.

To perform this subtask in the ideal model, all the nodes would send its habitat to the trusted third party, and the node A, in addition, would send the target location P. Now, the trusted third party would perform the required calculations and would send the results of these comparisons to the node A.

The main difference from the previous subtask is that node A is allowed to have the information regarding the habitat of another node, but it is not able to access to it as it is encrypted. Then, once node A has calculated the distance of the target node using its habitat information, the protocol behaves in the same way than in the first subtask and no additional information is disclosed.

Therefore, as if node A has the information of another node does not affect the security of the protocol it is possible to assume that this subtask is also secure against passive and active adversaries. Thus, the whole protocol is considered secure against active and passive adversaries.

V. Conclusion

In this article we have proposed a support information system for location-based routing protocols that does not compromise the privacy of the involved nodes. This system allows the nodes to use geographical information to make routing decisions. With the proposed system, it is possible to forward the messages in such a way that they take a specific path. For example, the messages can be sent as directly as possible to their destination, avoiding specific areas…

All the required calculations are securely performed on the involved nodes while protected by the homomorphic Paillier cryptosystem. Then, since no trusted third parties are needed, this protocol is suitable for DTNs.

These calculations can be considered a specific case of a secure multi-party computation. Hence, the security of the protocol has been analyzed from this point of view. This analysis has concluded that the protocol is secure against passive and active adversaries, including collusion attacks.

As future work, it would be interesting to implement the proposed information system, e.g. in the aDTN platform currently developed by the SeNDA research group [1]. This platform allows the exchange of messages following the store-carry-process-and-forward paradigm proposed in [4] which allows the messages to provide their own routing code. Then, it would be of interest to develop several routing algorithms that make use of this information system and compare the performance of these routing algorithms to other routing protocols.

Acknowledgment

This work was partly supported by grant TIN2010-15764 from the Ministerio de Ciencia e Innovación of Spain and by grant 2014SGR-691 from Generalitat de Catalunya.

References

[1] Adtn implementation. https://senda.uab.cat/wiki/aDTN.

[2] Mikhail J Atallah and Wenliang Du. Secure multi-party computational geometry. In *Algorithms and Data Structures*, pages 165–179. Springer, 2001.

[3] Donald Beaver. Foundations of secure interactive computing. In *Advances in Cryptology—CRYPTO'91*, pages 377–391. Springer, 1992.

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

[5] Donald R Byrkit. Taxicab geometry—a non-euclidean geometry of lattice points. *The Mathematics Teacher*, pages 418–422, 1971.

[6] Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000.

[7] Kevin Fall and Stephen Farrell. Dtn: an architectural retrospective. *Selected Areas in Communications, IEEE Journal on*, 26(5):828–836, 2008.

[8] Caroline Fontaine and Fabien Galand. A survey of homomorphic encryption for nonspecialists. *EURASIP Journal on Information Security*, 2007, 2007.

[9] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 218–229. ACM, 1987.

[10] Silvio Micali and Phillip Rogaway. Secure computation. In *Advances in Cryptology—CRYPTO'91*, pages 392–404. Springer, 1992.

[11] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in cryptology—EUROCRYPT'99*, pages 223–238. Springer, 1999.

[12] Nagasai Panchakarla, Jörg Ott, and Gesner Junior. Delay-tolerant adaptive real-time communication: a case study for voice. *Proceedings of ExtremeCom'12*, 2012.

[13] Manoj Prabhakaran and Amit Sahai. *Secure Multi-Party Computation*, volume 10. IOS Press, 2013.

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

[15] Andrew Chi-Chih Yao. Protocols for secure computations. In *FOCS*, volume 82, pages 160–164, 1982.

[16] Zhensheng Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. *Communications Surveys & Tutorials, IEEE*, 8(1):24–37, 2006.

[17] Ge Zhong, Ian Goldberg, and Urs Hengartner. Louis, lester and pierre: Three protocols for location privacy. In *Privacy Enhancing Technologies*, pages 62–76. Springer, 2007.