PrivProphet: Probabilistic Routing in Delay Tolerant Networks preserving privacy

Alejandra Peral Mejía

Abstract — Still to this day, many places on our planet do not have access to a reliable Internet connection. Delay Tolerant Networks (DTN) came into use to provide Internet connection to those locations where the environment is characterized by being unstable and by having high bit error rates and frequent network partitions. One of the elements used by the routing protocol in DTN is transitivity, which provides security to these kinds of networks. In this project, we have proved that the security provided by transitivity is not enough. As a solution, we present PrivProphet, a private routing protocol used as a routing protocol of a DTN. PrivProphet uses the history of encounters to judge if the encountered node has a higher probability of coming across the destination node than the node carrying the message. PrivProphet makes use of homomorphic encryption as a cryptographic technique to provide privacy to the routing protocol. To prove and study its functionality, we present a simulation created with The ONE simulator that provides the results of the exchange of messages.

Index Terms— Delay Tolerant Networks (DTN), Routing protocols, Privacy, Homomorphic encryption, Paillier cryptosystem, The ONE simulatior.

Resumen— Hoy en día, hay muchos lugares en nuestro planeta que no tienen acceso a una conexión fiable de Internet. Las redes tolerantes al retardo (DTN) se utilizan para proporcionar una conexión a Internet en aquellos lugares donde el entorno está caracterizado por ser inestable y por tener particiones frecuentes de red. Uno de los elementos que utiliza el protocolo de encaminamiento utilizado, es la transitividad. Este elemento proporciona seguridad a este tipo de redes. En este trabajo hemos comprobado que esta seguridad no es suficiente. Como solución presentamos PrivProphet, un protocolo de encaminamiento privado utilizado como protocolo de encaminamiento de una red DTN. Utiliza el historial de encuentros para juzgar si el nodo encontrado tiene una probabilidad más alta de llegar al nodo destino que el nodo que tiene el mensaje. PrivProphet utiliza encriptación homomorfica como técnica de criptografía para proporcionar privacidad al protocolo de encaminamiento. Para probar y estudiar su funcionalidad, presentamos una simulación hecha con el simulador The ONE, que proporciona los resultado del intercambio de mensajes.

Palabras Clave— Redes tolerantes al retardo (DTN), Protocolos de encaminamiento, Privacidad, Encriptación homomorfica, Criptosistema de Paillier, Simulador The ONE.

1 INTRODUCTION

NOWADAYS the Internet connections have increased significantly around the world because of the need of information and also because of the growth of the applications with the need of connection.

Almost 90% of the population in developed countries have reliable Internet access. But only about 10% of the population in developing countries have Internet access [9]

The Internet is a big network of computers connected to each other. One of the most important characteristics of this network is to have an end-to-end communication from the beginning and until the end of the communication. Because of this charateristics, it is impossible to reach some places with difficult access. The cause can be anywhere from difficult distances or even geographic locations surrounded by nature where it is problematic to install the appropiative infrastructure. In these environments, the actual Internet architecture would have long delays and frequent loss of data. This is when Delay Tolerant Networks (DTN)[1][2] come into use.

DTN is a non-infrastructure network that deals with the problem of the need to have an end-to-end path, by using mobile devices to communicate. They exchange data with each other using radio waves, Bluetooth or Wi-Fi by proximity and using other nodes to get to their final destination.

Figure 1 shows how nodes communicate in a Delay Telerant Networks. In this figure, every car and motorcycle in the picture carries a mobile device. The rural villages and the school on the right of Figure 1 do not have reliable Internet access. There are two parking lots that also have a mobile device. Finally, there is the urban city with reliable Internet connection.

The users driving the motorcycles and the cars pick up the information from the villages and the school and by sending the messages to each other, the information arrives at the urban city. Bringing back the response to the villages works in the same way.

Pretoria
Hammanskraal
school
URBAN CITY WITH CELLULAR COVERAGE
RURAL VILLAGES WITH LIMITED 2G COVERAG
© Disney

Figure 1 DTN form of communication [14].

We consider the routing [13] problem as the decision of selecting the path to send the message.

There are several routing possibilities for DTN; PrivProphet utilizes a probabilistic routing protocol [3] that uses the history of encounters to judge if the encountered node has a higher probability of coming across the destination node than the node carrying the message.

In this routing protocol, a node keeps the probability to deliver a message to every known node. This means that each node has its contact list with the probability to reach every contact on their list. Thus, the users may know the contacts of other users and even the frequency of seeing them. It is likely for the users to feel unsafe sharing their personal information with the rest of the nodes.

To solve this problem, we present PrivProphet as a private routing protocol. PrivProphet proposes to encrypt the probability with a homomorphic cryptosystem [6][8][10] while the nodes exchange information. This would allow the information to pass from node to node, without having access to other's personal information.

2 State of the Art

2.1 DTN

Delay Tolerant Networks are a none-infrastructure network that allows communication in an environment characterized by being unstable and by having high bit error rates and frequent network partitions.

The DTN architecture [11][7] follows a layer structure, same as the Internet. DTN introduces a new layer called Bundle Layer, located between the application layer and the transportation layer. Therefore, Delay Tolerant Networks can unify different kinds of networks. This layer is in charge of storing the information and redirecting the message to the next node [5].

These networks use the mobility of the users to communicate. Every user in the network has a mobile device. Using radio waves, Bluetooth or Wi-Fi they communicate with each other. One node can use different nodes to get to the final destination.

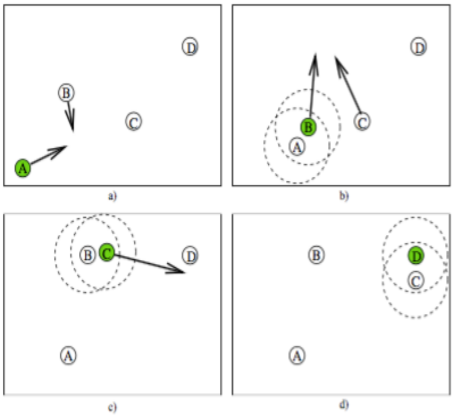

Figure 2 shows a transitive communication between A and D [3].

In order to clarify this concept, we present Figure 2 as an example of how DTN communicates. In the example, node A is the source of a message. Node A would like to communicate with node D, being the final destination of the message. Node A has never met node D, but it can send the message using node B and node C as intermediate nodes. As we can see on the top right of Figure 2, node A sends the message to node B. Node B then sends the message to node C, as shown on the bottom left of Figure 2. Finally, on the bottom right of Figure 2, node C sends the message to node D.

2.2.1 PROPHET

PROPHET is a routing algorithm. We understand a routing algorithm as the one responsible for choosing the next node the message should be forwarded to.

PROPHET is a probabilistic routing protocol that uses the history of encounters and transitivity. It uses $P_{(A,B)}$, this indicates how likely it is for A to encounter B. When two nodes establish a connection they exchange a list with the delivery predictability information for every known node.
The node with the highest probability gets the message.

The delivery predictability update has three parts.

1. Every time there is a connection between two nodes, both of them update their delivery predictability, so nodes that are often encountered have a high delivery probability. This is shown in equation A, where $P_{init} \in (0,1]$ is an initialization constant.

$$P_{(A,B)} = P_{(A,B)old} + \left(1 - P_{(A,B)old}\right) \times P_{init} \qquad [A]$$

2. If two nodes do not establish a connection for a long time, their delivery predictability must decrease considering that they are less likely to forward the message between them. This is shown by equation B, being $\gamma \in (0,1)$ the deacreasing constant and κ the number of time units.

$$P_{(A,B)} = P_{(A,B)old} \times \gamma^{\kappa} \qquad [B]$$

3. The last step is to update the transitivity property. First we are going to explain what transitivity is.
   We have nodes A, B, and C. Nodes A and B and nodes B and C know each other but nodes A and C have never met.

   Transitivity takes into account that the probability of A meeting C increases by A knowing B and, eventually, B knowing C.

   The formula to update the probability by using transitivity is the one in equation C. $\beta \in [0,1]$ value determinate how much importance we want to give to transitivity.

$$P_{(A,C)} = P_{(A,C)old} + \left(1 - P_{(A,C)old}\right) \times P_{(A,B)} \times P_{(B,C)} \times \beta \; [C]$$

The discussion exists about whether it is worth applying cryptography security to DTN because it is already secure by itself by using transitivity.

Taking the same example as before, adding a fourth node D, if D and A meet, it is impossible for D to know if the probability of A to meet C is because A has reached C or because of the transitivity. Therefore, there is no way for D to know exactly A's contacts exaclty.

2.2 Homomorphic encryption

Homomorphic encryption [8] is a form of encryption, which makes it possible to operate with two ciphertexts without having to decrypt the message first. With this information no longer being vulnerable, it would be possible to exchange information without exposing data. To decrypt the result of the operation, would give the same result like when operating with the plaintext [6], same as in equation D.

$$a \cdot b = c \qquad E(a) \cdot E(b) = E(c) \qquad [D]$$

2.2.1 Paillier

Paillier [2][1][10] is a probabilistic asymmetric algorithm with homomorphic properties.

The Key generation of Pailliers algorithm starts by A choosing two prime numbers of the same length p and q. Then, A compute n = p·q and λ = lcm (p-1,q-1), and also selects a random integer g where $g \in \mathbb{Z}_{n^2}^*$. Following this action, A would need to make sure that n divides the order of g by checking the existence of $\mu = \left(L(g^{\lambda} \, mod \, n^2)\right)^{-1} mod \, n$, where $L(u) = \frac{u-1}{n}$.
The public key would be (n, g) and the private key would be (λ, μ).
The encryption equation is $c = g^m \cdot r^n \, mod \, n^2$ where m is the message to be sent, and r is a random number, $r \in \mathbb{Z}_n^*$ and $m \in \mathbb{Z}_n$.
The decryption equation is m = L $(c^{\lambda} \, mod \, n^2) \cdot \mu \, mod \, n$, where c is the ciphertext to decrypt $c \in \mathbb{Z}_{n^2}^*$.

Paillier's homomorphic properties are addition E and multiplication F. But it does not accept subtraction or division

$$D\,(E\,(m1) \cdot E\,(m2)\; mod\; n^2) = m1 + m2 \; mod \; n \qquad [E]$$

$$D\,(E\,(m1)^{m2}\; mod\; n^2) = m1 \cdot m2 \; mod \; n \qquad [F]$$

2.3 The ONE

The Opportunistic Network Environment (ONE) [4] is a simulator which simulates different DTN routing algorithms, node movements, and message passing. Nodes send messages to each other using various types of movement models generated by the simulator. It also allows the user to visualize the exchange of messages and the mobility of the nodes in real time. The ONE also produces statistics and reports about the results obtained from the simulation.

3 Objectives

The aim of this final degree thesis is to develop a private routing protocol, which preserves the personal information of the user. In order to achieve this, we use the encryption of the probability coming from the routing protocol explained before. This way both nodes exchanging information can keep their data with privacy. We use Paillier cryptosystem to operate with the data without the need to decrypt the messages.

The second objective of this project is to observe the performance of the algorithm using The ONE simulator explained before.

4 Methodology

The methodology used in this project is the waterfall method [12]. In the Waterfall, the whole project is divided into separate phases. Each phase must be accomplished before the next phase can begin and there is no overlapping in the phases.

We have chosen this methodology because every task of this project is dependent on the previous one. Therefore, we cannot start the implementation without the analysis and the design, and we cannot do the testing without the results of the implementation.

Figure 3 shows how the waterfall method works. Starting from the planning, then the implementation, then, testing, and finally the evaluation.

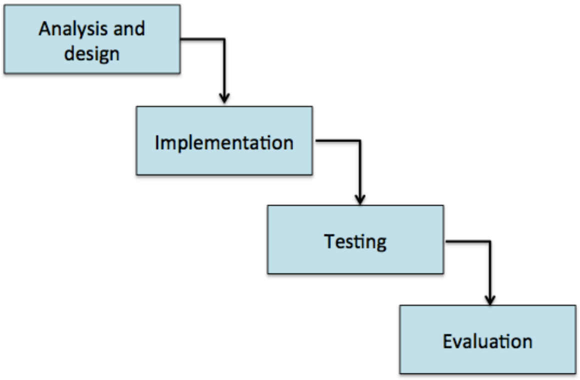

Figure 3 Waterfall stages of the project development.

5 Planning

The planning of this project can be divided into four different phases: analysis and design, implementation, testing, and evaluation. Following the methodology argument from the previous section. We have done the following process for the two objectives described in section 3.

- **Analysis and design:** This first phase is focused on the planning of the project and the determination of the objectives to achieve. Also, to study and familiarize ourselves with the concepts we work with on this project: DTN, homomorphic encryption and Paillier cryptosystem. This phase corresponds to tasks 1,2,3 and 4 of the tasks listed at the end of this section.

- **Implementation:** In this phase, we focus on the development of the secure routing protocol. Starting by modifying the algorithm of the routing protocol. This probabilistic routing protocol uses the history of encounters to judge if the encountered node has a higher probability of coming across the destination node than the node carrying the message. Then the next step is to make this routing protocol secure, by adding subtraction to Paillier. This phase corresponds to tasks 5,6 and 7 of the tasks list at the end of this section.

- **Testing:** We used the testing phase to observe the performance of the algorithm using The ONE simulator. First of all, we had to familiarize ourselfs with the simulator to be able to adapt it to our needs. These simulations allow us to have the results when using and not using security and also when using and not using transitivity. This phase corresponds to tasks 8 and 9 of the tasks list at the end of this section.

- **Evaluation:** Finally in this phase, we make an evaluation of the results, and a comparison of the overhead of using privacy. We have done the same with the results of transitivity. This phase corresponds to tasks 10 and 11 of the tasks list at the end of this section.

The following list shows the tasks planned for the project.

1. Familiarize ourselfs with the concepts of the project
2. First interview with the professor
3. Compile and familiarize ourselves with Paillier code
4. First partial submission
5. Paillier code analysis
6. Paillier implementation to PrivProphet
7. Second partial submission
8. Getting started with the simulation program
9. Paillier simulation with The One
10. Study of transitivity
11. Third partial submission
12. Write article
13. Prepare final presentation

6 Development

In this section, we will describe the PrivProphet protocol, how to use subtraction in Paillier cryptosystem and the exchange of messages utilized by the protocol.

6.1 Homomorphic Subtraction using Paillier

Let A and B be two potential intermediate nodes of an end-to-end communication. In this context, the routing protocol must decide which node it is better to send the data to, so the decision is made by comparing the contact lists and the associated probability of each node.

To compare which probability is higher between node A and node B, PrivProphet requires subtraction between encrypted values. If the value of the subtraction [1][2] is positive means that A has more probability of reaching the final node. On the other hand, if the result is negative it means that B has a higher chance of reaching the final destination.

The problem encountered is that the Paillier cryptosystem does not accept subtraction [1][2] operations. PrivProphet, therefore, uses the addition between a positive and a negative number, mapping the negative number, since there are no negative numbers in $\mathbb{Z}_n$.

To do the mapping, we take B's positive probability $P_{(B,C)}$ and mapped it to an integer that remains between $n/2$ *and* n in $\mathbb{Z}_n$. It is crucial that B's probability never exceeds ${}^{n}/_{2}$, in order to guarantee that the mapping exists. Otherwise, it would be impossible to use the addition of a positive and a negative number. Figure 4 provides a scheme of this mapping.

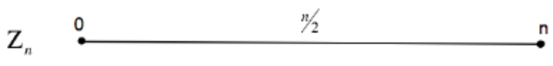

Figure 4: Positive and negative integers are separate in $\mathbb{Z}_n$ by ${}^{n}/_{2}$.

We denote M(m) as the mapping of m. Where m is the probability of one node reaching another node.

6.2 PrivProphet protocol

Assuming that node A is the one carrying the message, node B is the candidate neighbor, and node C is the destination node, A has to decide which node has a higher probability of encountering C. We denote $P_{(X,Z)}$ as the probability of X to encounter Z, and K_Y as Y's public key, where X, Z, and Y are nodes.

To make the exchange of information private and secure, we encrypt it all using Paillier's encryption explained before. We use $E_X(Z)$ to express the Paillier encryption of Z using X's public key.

Figure 5 shows the exchange of messages between node A and node B.

The PrivProphet protocol works as following:

1. Node A calculates its probability of encountering node C; let it be $P_{(A,C)}$. Furthermore A encrypts $P_{(A,C)}$ to be $E_A(P_{(A,C)})$.

2. Node A sends its public key K_A, the destination node, C, and its encrypted probability $E_A(P_{(A,C)})$ to B.

3. Node B receives A's information and maps B's probability to encounter C, $P_{(B,C)}$, as explained in section 6.1. The result of this mapping is $M\ (P_{(B,C)})$. Then node B encrypts the mapping of B's probability $E_A(M\ (P_{(B,C)}))$.

4. Once B has its probability encrypted, the next step is to add A's encrypted probability $E_A(P_{(A,C)})$ and B's encrypted mapped probability $E_A(M\ (P_{(B,C)}))$. Then B generates a random number, to multiply with the result of the addition made before. This is the result of the operation. This last step of multiplication is necessary because otherwise A's could have guessed B's probability, decrypting and subtracting $P_{(A,C)}$ from the decrypted result.

5. Node B sends the result to node A.

6. Node A receives the result of the subtraction from B, decrypts the message and makes a comparison. If the result is higher than 0 and lower than $n/_2$, that means that A's probability is bigger. On the other hand, if it is higher than $n/_2$ and lower than n, B's probability is bigger.

The node that has a higher probability of encountering the final destination is the one that has to have the message. Depending on the result of the subtraction A sends the message to B or does not.

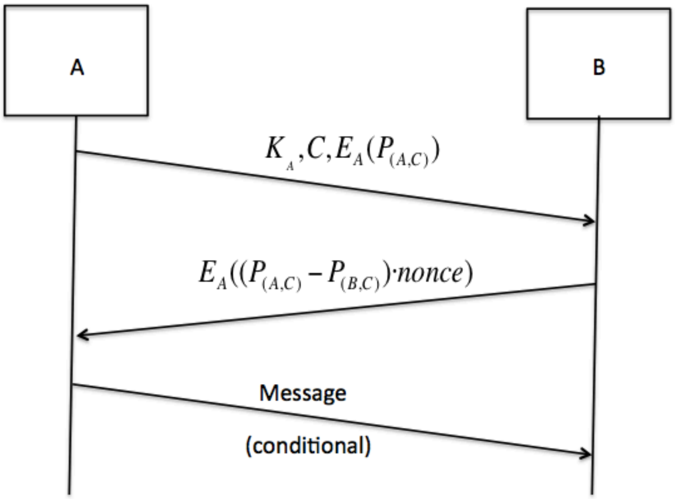

Figure 5 Exchange of messages between node A and node B.

7 Experimentation

In this section, we explain the experimentation using the simulation we have made to evaluate PrivProphet cryptosystem. We have chosen The ONE as a tool to realize the simulation. We mention The ONE in section 2.

7.2 Scenario

To work in a familiar environment, we have chosen the Universitat Autònoma de Barcelona Campus as a scenario to execute the simulation. We use five different groups of nodes, four groups of pedestrians, and one of cars. Each group of pedestrians are students from a determinate bachelor. There are engineers, doctors, lawyers, and philosophers students. Each group has five nodes. Accordingly, in total, there are 25 nodes on our scenario. We have configured each group of nodes to move mainly around their own faculties area. At the beginning the nodes start in a random position, then while the simulation is running the nodes start organizing. Figure 6 shows the area where every kind of node moves.

All nodes use Bluetooth interface. Cars and pedestrians use differenet speeds. Pedestrians move from 0.5 to 1.5 kilometers per hour, on the other hand, cars move from 2.7 to 13.9 kilometers per hour.

The simulator offers different movement models. ShortestPathMapBasedMovement is the most sophisticated one. It uses an algorithm (Dijkstra) to find the shortest path through the map.

Once the node arrives at its destination and waits the established pause time, it generates a new random destination. The node goes to the new destination trough the shortest path found. This is the movement model all the nodes on our scenario uses. As a routing protocol, we use PrivProphet. We describe this protocol in section 6.

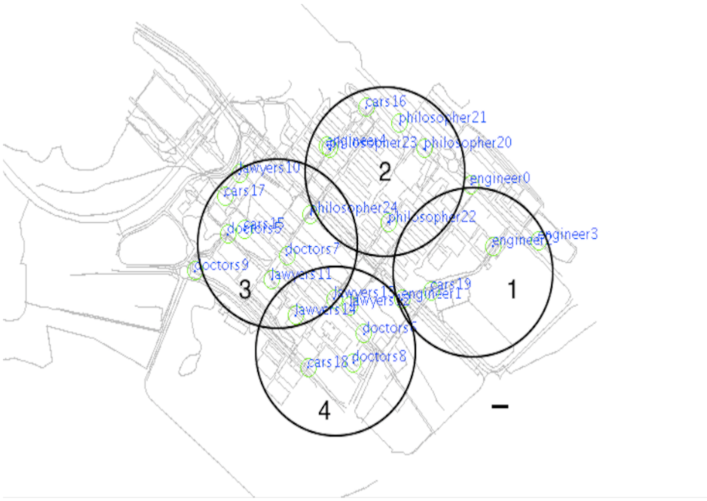

Figure 6 The ONE software, with the area of every pedestrian group.

The nodes create the messages randomly. The simulator generates a message keeping in mind the configuration of the scenario. For example, the number of nodes that create messages, the size of the message, the interval between the creations of two different messages, among others.

7.2 Simulations model

We have done different simulations, with various configurations so we can compare the results.

First of all, as we have said before in this section, we used PrivProphet as the routing protocol. PrivPophet is a private routing protocol. To make sure the simulation is as accurate as possible and fulfills the private promise, we have added a delay corresponding to the encryption time of the message. Then, for the second simulation, we have used the same routing algorithm but not applying security. This routing algorithm is PROPHET.

We have done both of the simulation with diferent transitivity values. There are 11 different executions with transitivity values from 0 to 1, for each simulaion.

The ONE allows us to generate several kinds of reports containing information about the simulation. We have used the following:

1. MessageStateReport: It generates different kinds of statistics about message relay performance.

2. CreatedMessageReport: Reports information about all created messages. For each message, it shows the identifier of the message, the size of the message, the source, and the destination host. There are 340 messages created. As we have configured the source is always an engineer and the destination is always a philosopher.

3. DeliveredMessagesReport: Reports information about all the delivered messages. It shows the messages sent, the delivery time, the source and the path to get to the destination.

We have generated all of the reports for each β value. β is explained in section 2.1.1 .

8 Results

In this section, we are going to do an analysis of the results extracted from the simulation.

We have done a comparison between the results obtained from the simulation using the private routing algorithm PrivProphet and the routing algorithm without security/encryption, PROPHET. We have compared the latency and the delivery probability that the message has to arrive at its final destination.

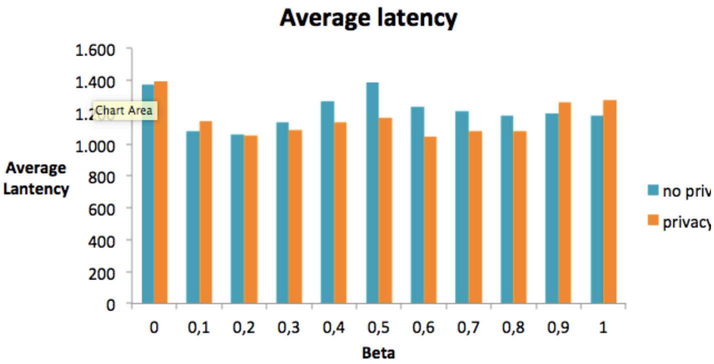

Figure 7 Latency comparison between the algorithm with and without security

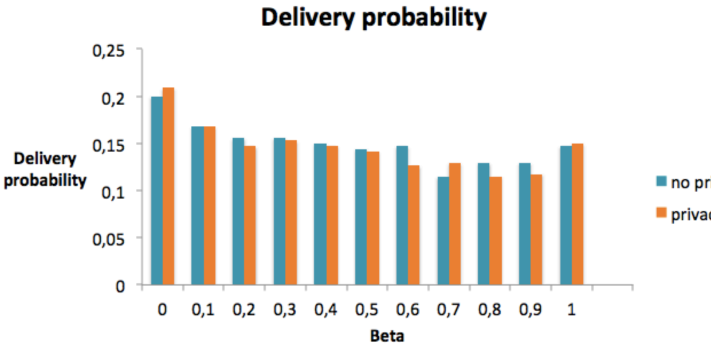

Figure 8 Delivery probability comparison between the algorithm with and without security

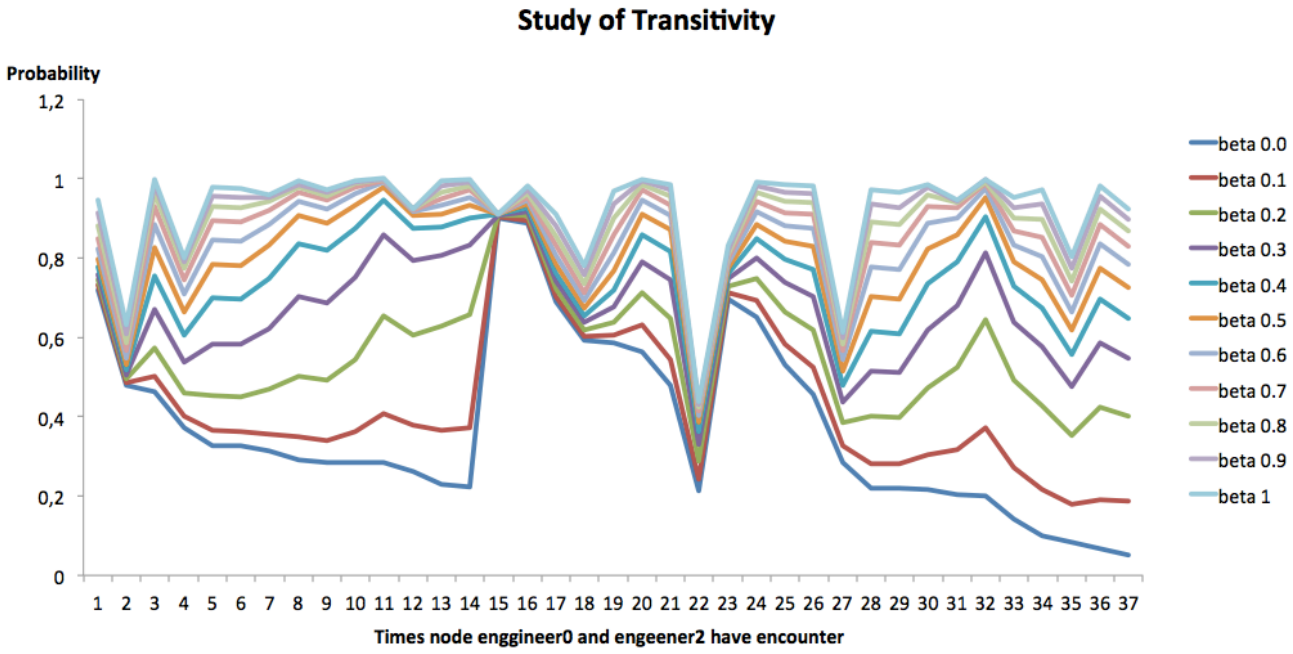

Figure 9 Node engineer2's probability chart to find Node engineer0.

Figure 7 shows the difference of the average latency between the algorithms both when using and when not using privacy. This comparison is shown for every beta value. Also in Figure 8, we make the same comparison but comparing delivery probability. In both cases the difference is minimum.

After analyzing the results, we can tell that it is advantageous to apply security to our algorithm, considering that the amount of latency and deliveries lost are insignificant compared to the non-secure algorithm.

9 Transitivity study

As we mention in section 2.1.1, the discussion existe about whether it is worth applying cryptography security to DTN because it is already secure by itself using transitivity. The aim of this section is to demonstrate that transitivity does not provide enough security to DTN.

Figure 9 shows Node engineer2's probability to meet with Node engineer0 during the execution of the simulation. In the graphic we find the comparison of the same execution, but with different beta values.

We notice from Figure 9 that there is not a significant difference between the probabilities while using beta values of 0.0, 0.1, 0.2. The difference between 0.0 and 0.1 beta values is about 14%. Therefore, it is so similar that it becomes trivial to tell which is the real probability of Node engineer2 to meet with Node engineer0 even though we are using transitivity.

10 Conclusions

In this paper, we have proved that the security provided by transitivity is not enough since it is possible to know which are the contacts of a node. As a solution, we present PrivProphet, a probabilistic routing protocol that uses the history of encounters to judge if the encountered node has a higher probability of coming across the destination node than the node carrying the message. PrivProphet uses homomorphic encryption to preserve nodes privacy. We have created the possibility to operate with subtraction in Paillier's cryptosystem to make security possible. We have implemented and integrated the subtraction method to the PrivProphet routing protocol. We have analyzed the performance of the algorithm by using The ONE simulator. We have arrived at the conclusion that the time difference of applying security is minimal and does not affect the communications made during the execution.

11 Future work

As a future work, we propose to implement privacy to other routing algorithms, for example, Spray and Wait, a replication-based routing algorithm or Bubble Rap Protocol, a social-based forwarding algorithm. After, it will be interesting to compare the performance between all the algorithms.

ACKNOWLEDGMENT

The acknowledgment of this project goes especially to my professor Carlos Borrego who guided me to achieve our objectives. Without him this project would not have been possible. I would also like to thank Sergi Robles for offering me the opportunity to do this project. Finally, I would like to thank my family, my friends and especially my partner for the support during the project and during these last years of university.

BIBLIOGRAPHY

[1] Adrián Sánchez-Carmona, Sergi Robles, Carlos Borrego. "PrivHab: a Privacy Preserving Georouting Protocol based on a Multiagent System for Podcast Distribution on Disconnected Areas ." Barcelona : AAMAS, 2015. 1697-1698.

[2] Adrián Sánchez-Carmona, Sergi Robles, Carlos Borrego. "PrivHab+: A secure geographic routing protocol for DTN." In *Comuter communications* , 56–73. Barclona : ELSEVIER , 2015.

[3] Anders Lindgren, Avri Doria, Olov Schelén. "Probabilistic Routing in Intermittently Connected Networks." In *Service Assurance with Partial and Intermittent Resources*, by Avri Doria, Olov Schelén Anders Lindgren, pp 239-254. Lulea : Springer Berlin Heidelberg, 2004.

[4] Ari Keränen, Jörg Ott, Teemu Kärkkäinen. "The ONE Simulator for DTN Protocol Evaluation." Helsinky: Simutools , 2009.

[5] Fall, Kevin. "A Delay-Tolerant Network Architecture For Chalenged Internets." *SIGCOMM* . New York, 2003. 27-34.

[6] Gentry, Craig. *A Filly Homomorphic encryption sheme.* 2009.

[7] Jhon Eduardo, Bedoya Camacho. "Propuesta y simulación de una solución basada en redes tolerantes al retardo para proporcionar comunicaciones en entornos remotos aislados." Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, Madrid, 2003.

[8] Micciancio, Daniele. "A first glimpse of cryptography's Holy Grail." *Communications of the ACM*, 2010.

[9] Páez, María Irene. "Análisis y evolución de prestaciones de protocolos de encaminamiento en redes tolerantes al retardo." Universidad Politécinca de Madrid, Madrid, 2013.

[10] Paillier, Pascal. "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes." In *Advances in Cryptology — EUROCRYPT '99*. Springer Berlin Heidelberg, 1999.

[11] Romero Amondaray, Lídice Sánchez Paz, Héctor Ramón. "Las redes tolerantes al retardo. Una solución a as comunicaciones reales en cuba." *Rebista Electrónica de Estudios Telemáticos* 2010.

[12] S. Balaji, Dr.M.Sundararajan. "Waterfall vs V-Model vs Agile: A comparative study on sdlc." *International Journal of Information and Buiness MAnagement* , 2012.

[13] Sushant Jain, Kevin Fall, Rabin Patra. "Routing in a Delay Tolernat Network." *SIGCOMM '04.* Portland, 2004.

[14] Adriano Galati, Theodoros Bourchas, Sandra Siby, Seth Frey, Maria Olivares, Stefan Mangold. "Mobile-enabled delay tolerant networking in rural developing regions." *Global Humanitarian Technology Conference (GHTC).* IEEE, 2014.