A privacy-preserving protocol for finding the nearest doctor in an emergency

Georgios Drosatos
Dept. of Electrical and Computer Engineering
Democritus University of Thrace
University Campus, 67100 Xanthi, Greece
gdrosato@ee.duth.gr

Pavlos S. Efraimidis
Dept. of Electrical and Computer Engineering
Democritus University of Thrace
University Campus, 67100 Xanthi, Greece
pefraimi@ee.duth.gr

ABSTRACT

In this work, we define the Nearest Doctor Problem (NDP) for finding the closest doctor in case of an emergency and present a secure multi-party computation for solving it. The solution is based on a privacy-preserving cryptographic protocol and makes use of the current location of each participating doctor. The protocol is efficient and protects the privacy of the location of all doctors. A prototype implementing the proposed solution for a community of doctors that use mobile devices to obtain their current location is presented.

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous—*Personal data*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Privacy-preserving protocol*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multi-agent systems*

General Terms

Location privacy

Keywords

Location privacy, Personal data, Privacy-preserving protocol, Peer-to-Peer network

1. INTRODUCTION

The advances in information and communication technologies (ICT) and the wide acceptance of electronic transactions for everyday tasks of individuals have a strong impact on the use and protection of personal information. Desktop and mobile computing technology, sensors, and the advances in database and storage technologies have increased the amount of personal information that is generated and the potential for this information to be (permanently) stored and processed. Any kind of personal information that results as the main or the secondary outcome of any electronic of activity of individuals, either personal or professional, belongs to the category of personal data. Personal data has to be protected in order to ensure the individual's privacy rights.

The same advances in ICT that cause the generation of more personal information, also provide vast potential for emerging new applications that can use personal data in favor of the individuals' interests. Some examples are personalized web services that automatically adapt to the profile of an individual and location-based services that behave according to the individual's current location or context. Individuals, as well as the society as a whole, can obtain significant benefits if personal data can be used legitimately for rightful purposes. However, the use of personal data should be done in a way that achieves its simultaneous protection. Each individual has the right to protect his privacy by retaining the absolute control over his personal data and knowing who, when and why gets access to this data. Furthermore, every individual should disclose only the minimum possible personal information that is needed each time to complete a transaction. That is, release of personal data should be done in such a way that only the absolutely necessary items are disclosed and only when this is really needed.

A very interesting class of personal data is dynamic personal data, such as the current location of an individual. The recent progress in mobile device technology and general technologies of ubiquitous computing allows individuals to collect dynamic personal data in order to accomplish useful tasks.

In this paper, we focus on dynamic personal data and examine the possibility of the development of innovative applications that exploit this kind of data, while ensuring strong protection for the privacy of individuals. To this end, we propose the following problem, called Nearest Doctor Problem (NDP), which finds the nearest doctor in case of an emergency. In a hypothetic but feasible scenario, each doctor has a personal agent where his current location is always stored. In case of an emergency, the agents of all doctors interact to identify the doctor who happens to be closer than any other to the emergency location. We assume that the doctors may be off duty and thus the current location of each doctor is sensitive personal data that should not be revealed to anyone else, including other doctors.

The NDP problem is an example of a privacy-preserving ap-

PETRA'10, June 23 - 25, 2010, Samos, Greece.

plication based on dynamic personal data; the location of each individual doctor. We propose a privacy preserving solution that solves the problem without revealing the location of any doctor. The individual who is anonymously identified as the closest doctor can then reveal his identity and offer his services for the emergency event.

The solution that we propose for NDP, makes use of cryptographic primitives and decentralized computation technologies. A basic assumption is that all doctors have at their disposal a personal data management agent where their current location is stored. Each agent is under the control of its owner and all personal agents are permanently connected to the Internet. In case of an emergency, the agents of all doctors execute a distributed computation to identify in a cryptographic safe way who is the closest doctor to the incident. For performance, scalability and fault tolerance reasons and additionally for enhancing privacy, the computation is executed in a fully decentralized way. The agents/nodes are (self)organized in a distributed topology. To achieve this we employ techniques from the field of Peer-to-Peer (P2P) networks. The use of P2P techniques allows us to satisfy the requirements of high scalability of the system and to reduce the risk for privacy breaches. We apply techniques that have been developed in the context of the Quantum P2P network [12] and are based on the well proven Chord [13] architecture for P2P networks.

The NDP problem is an illustrative example of an application where personal data can be used for a common good (public health) whereas at the same time the privacy of all involved individuals is preserved. We believe that many new applications can emerge from the same principle of simultaneously using and protecting personal data. For example:

- First aid in case of a car collision/emergency. The European Union has launched the eCall project [7] for dealing with the ability of providing assistance in case of car emergencies. The project's goal is to deploy a hardware black box installed in vehicles that will send an emergency request in case of an accident on the road. The request will be transmitted over wireless communication technologies like GSM and will include information like the GPS coordinates of the emergency location and airbag deployment and impact sensor information. An additional action could be to search if anyone in the nearby cars could offer first aid (who would be entitled to offer help in such cases is an issue that is out of the scope of this work). However, the location of a vehicle is private information and so the search for nearby cars has to be done in privacy-preserving way. A protocol/solution like the NDP solution presented in this work could be used to identity a nearby car. A different problem/application, probably easier than NDP, would be to warn all nearby cars to slow down.
- Police or fire emergency. In case of a police or fire department emergency, a policeman or a fireguard who is not on duty and happens to be near the event location, might be able to provide critical services if he is informed about the emergency. At the same time, since the individual (policeman or fireguard) is not on duty, the exact location of a person is sensitive personal data and nobody has the right to know it. A solution like NDP could identify such an individual (with his consent). The individual would be contacted by its own agent only if he is the closest person and if he is close enough to be able to help in such an emergency.

Related Work. The Active Badge Location System [14] was the first indoor location system for contacting people in an office environment. The system raised issues on location privacy at work. Extensions of the initial system and follow-up projects like [15] offered enhanced features to the users for controlling the way their location data is accessed. However, all these systems assume a trusted server that manages the location data. A system that assumes a decentralized control of personal is the Cricket Location-Support System [10]. Cricket describes an approach that offers an individual the option to learn its physical location within a building (that offer the Cricket service). The user can then decide to whom he discloses his location. This approach offers a better control over who obtains the location information of the individual. However, if the user wants to actively use his location information to perform some task, he has to disclose it. An approach like Cricket could be used to allow individuals to learn their location when they are within buildings where GPS cannot be used. All the above location systems are for indoor applications.

The privacy concerns for applications like NDP are even more critical since they apply to individuals who may be in their private time and not only at their office but at any location. A computation that requires input from two or more parties and calculates the output without revealing the input of any participant is a secure multi-party computation (MPC). A general model for MPC has been proposed in [3] and follow-up works. However, the general model is practically inefficient. More efficient approaches are being developed for specific applications, like for example [17, 2]. The NDP solution presented in this work is an efficient MPC scheme for the NDP problem.

2. THE NDP PROBLEM

In this section we define the Nearest Doctor Problem (NDP). The main goal of NDP is to find the nearest doctor without violating the privacy of doctors. The personal data which are needed for the NDP computation are the exact locations of all doctors. An instance of the NDP problem consists of:

- **N doctors** $D_1, D_2, ..., D_N$.
- For $i = 1, 2, \ldots, N$, let L_i be the current location of doctor D_i. For instance, the location L_i may be the exact GPS location of the doctor, obtained from a portable GPS device.
- **The NDP lookup function:** In case of an emergency, (the agents of) all doctors perform a distributed privacy-preserving computation.
  - **Input:** The location L_{em} of an emergency.
  - **Output:** At the end of the computation, the doctor who is the nearest one to the location of the emergency becomes aware of this fact and can offer his services.

3. A SOLUTION FOR NDP

We describe a distributed privacy-preserving solution for solving the NDP problem. An overview of the architecture of the NDP solution is presented in Figure 1. The core of the solution is a cryptographic protocol for a secure distributed computation. We show that the protocol is safe in the security model of Honest-But-Curious (HBC) users, i.e., the doctors follow the protocol steps but also may try to extract additional information (see Definition 4). The HBC model is commonly used in cryptographic protocols and is well suited for the NDP problem, since the participants are certified doctors.

We make the following plausible assumptions:

- Every doctor has a personal data management agent with permanent access to the Internet.
- The current location of every doctor is stored at its personal agent.

3.1 The NDP Service

We first describe the operation of a service for the NDP. In case of an emergency, the following steps take place:

- The individual who is in an emergency, submits a request to the NDP Service Gateway (NSG). The request contains the current location of the individual (for example, exact geographic longitude and latitude) and possibly additional information about his identity, his current condition etc.
- The NSG is an access point that accepts the request and forwards it to an agent of the doctor's community. The agent who receives the request from the NSG, takes the role of the root-node for the particular computation.
- The root-node coordinates a distributed computation that calculates the distance of the nearest doctor.
- At the end of the distributed computation, the agent of the doctor who is the nearest to the location of the emergency becomes aware of this fact and contacts the NSG to declare his readiness to offer help.

3.2 Outline of the Distributed Computation

We present a protocol for a secure distributed computation that solves the NDP problem. The protocol does not disclose the location of any doctor; only a small number of distances are anonymously revealed to the NSG of the computation. The computation consists of three main phases. In Phase 1, the closest interval containing at least one doctor is found. In Phase 2, the distance of the nearest doctor and an associated random ID are found. Finally, in Phase 3, the doctor who owns the random ID realizes that he is the nearest doctor and contacts the NSG to offer his help.

- **Phase 1**
  - **Input:** The location L_{em} of the emergency.
  - **Output:** An interval I containing the minimum distances in which there is at least 1 doctor and at most K doctors, where K is a given constant (e.g. $K = 5$).
  - **Description:** The NSG chooses a node as the root-node for the particular computation and sends the location L_{em} of the emergency to this root-node. The root-node sends a broadcast message that starts the distributed protocol and initiates Phase 1. The protocol is executed on a logical binary tree topology that contains all doctor agents (nodes) as leafs (Figure 2). Phase 1 may last for several rounds. In each round, the root-node collects the (intermediate) result of the computation as an encrypted message and sends this message to the NSG. The message is encrypted with the public key of the NSG, which has to be known to all nodes. Let $Count_D$ be the number of distances that belong to an interval of minimum distances. The NSG decrypts the result and obtains $Count_D$. If $Count_D > K$, then the computation is repeated in a new round, this time within the interval that has been found. This procedure continues until an interval that contains the closest $Count_D < K$ doctors is found. An example of this procedure where the appropriate interval is found in two rounds, is shown in Figure 3. In subsection 3.3 we describe a protocol for Phase 1 that ensures k-anonymity (see Definition 3), where $k = N$ and N is the number of all nodes in the network, for the participants of the protocol.
- **Phase 2**
  - **Input:** The interval I from Phase 1.
  - **Output:** The - anonymously collected - exact distances of the $Count_D$ nearest doctors and the associated random ID.
  - **Description:** In this phase, the NSG sends the interval I of Phase 1 to the root-node which in turn sends a broadcast message to announce the interval I to the agents. Each agent whose distance is in interval I responds by anonymously sending a message to the NSG. The message is encrypted with the public key of the NSG and contains the exact distance of the agent and a random ID (a nonce, i.e., a number used once). The anonymous transmission is achieved by Onion Routing [11] techniques. More information about the onion routing are given in subsection 3.4. The NSG collects all anonymous messages and finds the distance of the nearest doctor and the associated random ID. Since, the messages are anonymously sent, the privacy of the doctors is preserved.
- **Phase 3**
  - **Input:** The random ID associated with the distance of the nearest doctor.
  - **Output:** The owner of the random ID realizes that he is the nearest doctor $D_{nearest}$ and can contact the NSG.

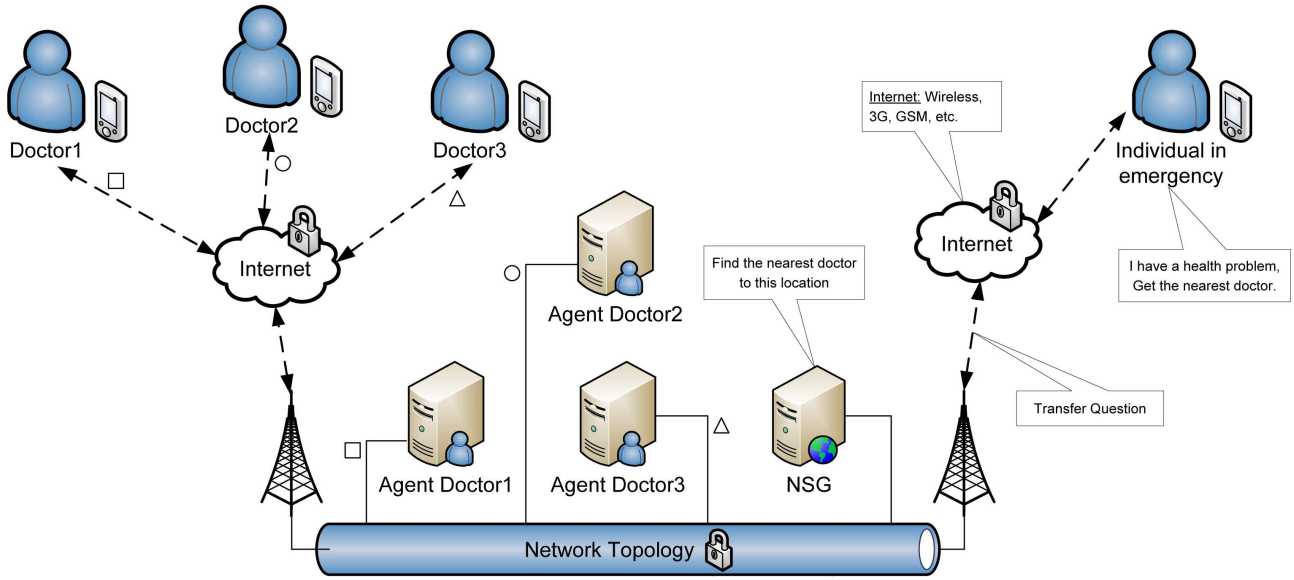

Figure 1: The architecture of NDP solution.

– **Description:** The NSG sends a message containing the random ID of the distance of the nearest doctor to the root-node. The root-node broadcasts the random ID to the agent network. The doctor who generated the random ID becomes aware of the fact that he is the nearest doctor and reveals its identity to the NSG.

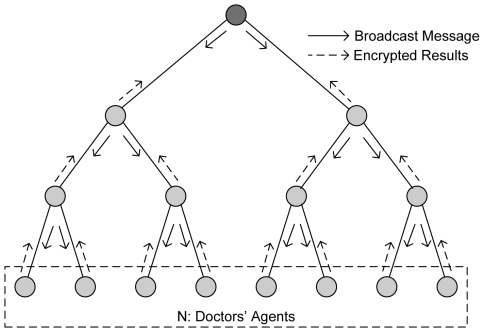

Figure 2: A binary tree topology.

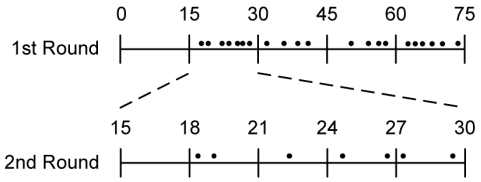

Figure 3: Example of Phase 1.

3.3 A Privacy Preserving Protocol for Phase 1

We present a cryptographic protocol that finds the first interval of distance in which there is at least one doctor. The basic idea of this protocol is based on a related protocol for secure dynamic programming of [17]. The cryptographic protocol uses the ElGamal public key cryptosystem [9]. An important feature of the ElGamal cryptosystem (and other public key cryptosystems) is its homomorphic property.

Definition 1. **ElGamal Cryptosystem:** The ElGamal cryptosystem is an asymmetric key algorithm for public key cryptography which is based on the Diffie-Hellman key agreement.

Definition 2. **Homomorphic Encryption:** The homomorphic encryption is a form of encryption where one can perform a specific algebraic operation on the plaintext by performing a (possibly different) algebraic operation on the ciphertext.

The homomorphic property of ElGamal cryptosystem is shown in the following equation:

$$\mathcal{E}(x_1) \cdot \mathcal{E}(x_2) = (g^{r_1}, x_1 \cdot h^{r_1})(g^{r_2}, x_2 \cdot h^{r_2}) =$$

$$(g^{r_1+r_2}, (x_1 \cdot x_2)h^{r_1+r_2}) = \mathcal{E}(x_1 \cdot x_2)$$

where:

- x_1 and x_2: two plain messages
- (G, q, g, h): the ElGamal public key
- G: is a cyclic group
- q: is a large prime integer
- g: is a generator of the group G
- $h = g^x$: is part of public key and x is the private key
- r_1 and r_2: two random numbers where $r_1, r_2 \in \{0, \ldots, q-1\}$
- $\mathcal{E}(m) = (g^r, m \cdot h^r)$: is the encryption of message m

3.3.1 The Protocol

The protocol accepts three parameters: The minimum distance $minDist$, the maximum distance $maxDist$ and the number n of subintervals. The interval $(minDist, maxDist)$ is partitioned into n subintervals. For simplicity we use subintervals of equal size, but it is straightforward to adapt this for example to geometrically increasing subintervals. The outcome of the protocol is the finding of the first subinterval that contains (the distance of) at least one doctor. To achieve this, each subinterval is represented with a ciphertext and the whole interval is represented with the ordered list (or tuple) of all ciphertexts. Overall, each message has n encrypted numbers, as many as the subintervals into which the initial interval is partitioned. A message containing the ordered list of ciphertexts passes from each agent. Each agent prepares its own ordered list of ciphertexts as follows: For doctor D_i, where $i = 1, 2, \ldots, N$, let $\ell_i \in 1, 2, \ldots, n$ be the number of the subinterval that contains the distance of the doctor. Then, the ciphertext for the ℓ_i first subintervals are encryptions of the number "1". For the rest of the subintervals the ciphertexts are encryptions of a number z, where $z > 1$ is a fixed value known to all agents. An example of a local message is shown in Figure 4.

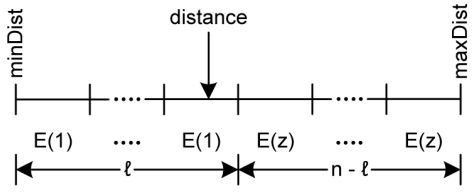

Figure 4: The local message.

When the agent receives the global message, it calculates a new global message in the following way: Each of the first $\ell_i + 1$ ciphertexts of the new global message is the product of the respective ciphertexts of the local message and the global message. The remaining $n - (\ell_i + 1)$ ciphertexts of the new global message are set equal to the respective ciphertexts of the local message. The outcome is the new global message which is then forwarded to the next node or nodes.

The distributed computation is performed on a logical binary tree topology in which the leafs of the tree are the N doctor agents. The depth of the tree is $\lfloor \log_2 N \rfloor + 1$ and so the global outcome is computed after $\lfloor \log_2 N \rfloor + 1$ parallel steps.

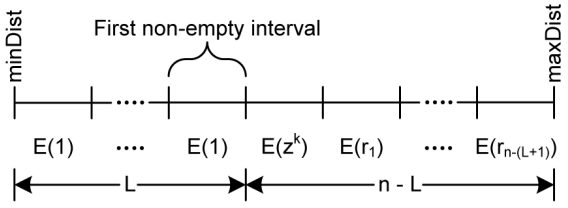

Figure 5: The final global message.

The general form of the final global message is shown in Figure 5. Let L be the number of the last ciphertext that is an encryption of the number "1". Then, the value of L indicates that the first $L - 1$ subintervals are empty (no doctor is located at a distance in these intervals) and subinterval is the first non-empty subinterval. The exponent k of the number in the $(L+1)$-th ciphertext reveals the number of doctors in this subinterval. The ciphertexts of higher subintervals are encryptions of some random powers of z and are ignored. The NSG node decrypts the final-message and obtains the first non-empty interval and the number of doctors in it.

3.4 Onion Routing

In Phase 2 of the distributed computation we use Onion Routing [11, 16], a popular technique for anonymous communication over a network. A simplified description of Onion Routing is: A node that wants to send a message to another node does not send the message directly to its destination. Instead, the sender chooses a random path that passes through intermediate nodes and terminates at the destination node. Moreover, the sender encrypts the message repeatedly with the keys of the intermediate nodes. So the message is packed with multiple layers of encryption and looks like an "onion". Each intermediate node that receives the message, takes away a layer of encryption to reveal routing instructions, and sends the message to the next router where this process is repeated. This prevents intermediary nodes from knowing the origin, the destination and the contents of the message.

The advantage of onion routing is that it is not necessary to trust each cooperating router; if one or more routers are compromised, anonymous communication can still be achieved. This is because each router in an onion routing network accepts messages, re-encrypts them and then forwards them to another onion router. An attacker with the ability to monitor every onion router in a network might be able to trace the path of a message through the network, but an attacker with more limited capabilities will have difficulty even if he controls one or more onion routers on the message's path.

In order to accomplish the anonymity for sending a message like we described in the Phase 2 (subsection 3.2), one option is to use the Tor network [5], a second-generation onion routing platform. The Tor network is a widely used, general purpose platform for onion routing. Another option, is to implement an onion routing approach within the agent community, where the agents send their message through other random agents of the agents community.

3.5 Network Topology

A critical component of the NDP solution is the logical network topology of the agents. To this end we employ networking technologies from the field Peer-to-Peer (P2P) networks. In particular, we apply techniques that have been developed in the context of the Quantum P2P network [12] and are based on the well known Chord [13] topology for P2P networks.

The network topology has the following features: The agents are organized into a logical ring that serves as the backbone of the topology. Each node in the ring, knows its predecessor and its successor. Actually, for increased tolerance to node changes/failures, each node keeps links to a set of successors. In addition, each node maintains a set of links, called fingers, to nodes at geometrically increasing distance

in the ring. These links allow the network to behave as a logical tree topology. The links of existing nodes and the establishment of the links of new nodes are accomplished with stabilization procedures that are similar to the stabilization procedures of the original Chord network. The proposed network architecture provides a fully decentralized and scalable network topology for the doctors' agents. An example of the topology with the logical ring and some of the characteristic fingers or "chords" of the Chord architecture is shown in Figure 6.

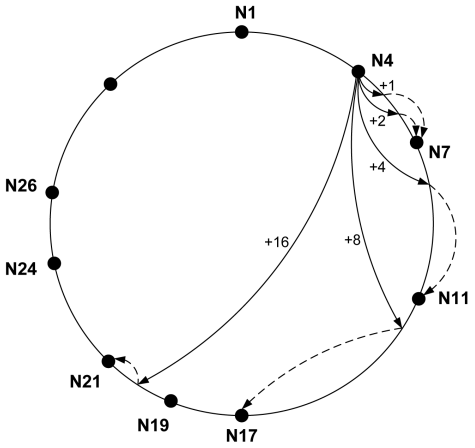

Figure 6: A network topology with the fingers of node N4.

4. THE PROTOCOL'S SECURITY

In this Section, we show that the proposed protocol for NDP does not violate the location privacy of the doctors. The security holds for the model of Honest-But-Curious (HBC) users. We first note that each doctor does not use his location but only his distance to the location of the emergency. The security of the ElGamal cryptosystem and its homomorphic property ensures that the distances cannot be associated with any particular doctor. The security of onion routing protects the anonymity of the nearest doctors that disclose their distance in Phase 2. Below, we discuss the security features of each Phase in detail.

- **Phase 1**
  - Each doctor uses his private location and the location of the emergency to calculate its own distance. Only the distance is used in the distributed computation, not the private location. Moreover, the doctor uses in the computation only the interval to which his distance belongs, and not the exact value of its distance.
  - Each agent that receives a global message cannot obtain information about the contents of the message, because the ciphertexts are encrypted with ElGamal encryption.
  - All ciphertexts of the global message are altered by each node, even the ones that are multiplied with an encrypted number "1".
  - At the end of each round of Phase 1, the global message reveals the number of doctors in the first interval that contains any doctors. No individual doctor can be associated with the doctors in the first interval. Consequently, Phase 1 preserves k-anonymity (see Definition 3), where $k = N$ and N is the number of all agents in the network.
  - In each round, the statistical information on the number of doctors in the first interval is revealed. This information does not violate the individual location privacy of the doctors.
- **Phase 2**
  - The security features of Onion Routing ensure that the exact distances of the doctors in the first interval are anonymously sent to the root-node. Hence, k-anonymity (for k = N) is preserved in this phase too.
  - We assume that Onion Routing works reliably. More details on the security of Onion Routing can be found in [5].
- **Phase 3**
  - In this phase the random ID is announced to the network. The agent that recognizes this ID can now contact the NSG and reveal its identity. The location privacy of all other agents is preserved.

Definition 3. **k-anonymity:** An informal definition of k-anonymity in the context of NDP is that no less than k individual doctors can be associated with a particular distance. For a more general definition of k-anonymity that is also valid in databases see [4].

Definition 4. **Honest-But-Curious (HBC):** An honest-but-curious party (adversary) [1] follows the prescribed protocol properly, but may keep intermediate computation results, e.g. messages exchanged, and try to deduce additional information from them other than the protocol result.

5. EXPERIMENTAL RESULTS

To confirm the feasibility of the NDP solution and examine its practical efficiency we developed an NDP prototype. The application is developed in Java and for the cryptographic primitives the Bouncycastle [8] library is used. The personal agents of the Polis platform developed in [6] are used as the personal data management agents of the doctors. In this approach, the management of the current location of each doctor is assigned to its personal Polis agent.

In the prototype, the cryptographic protocol is fully implemented with the use of production ready libraries. The main compromise at this point of the development of the prototype is that the network topology functionalities are not fully implemented yet. Thus, the network uses only the ring topology and furthermore the topology is not supposed to variate during a distributed computation. These performance and fault tolerance restrictions will be surpassed if the network topology is fully implemented.

Even at this point of development, the number of agents is only restricted by technical issues related to managing

large numbers of agents in an experimental environment. We performed experiments of the NDP solution with up to 30 agents, which took 40 seconds to resolve. Below we describe an experiment with 4 agents and the NSG.

The experiment covers a square area of 10000 km^2. The locations of the doctors and the emergency chosen independently and uniformly at random in the above area. Each agent chooses its random location and the NSG chooses a random location for the emergency event. The NDP solution tries to find within a distance of at most $75km$ the nearest doctor. The values of the internal parameters of the NDP solution are $K = 2$ and $z = 2$.

The NSG node chooses a node, in this case agent 1, as the root-node and forwards the location of the emergency event to this node. The coordinates of the location of the emergency are $L_{em} = [41.140110, 24.913660]$ and the exact distances of the 4 agents from this emergency location are:

$$Agent\_1 \Rightarrow 17.544817km$$
$$Agent\_2 \Rightarrow 53.157742km$$
$$Agent\_3 \Rightarrow 25.797003km$$
$$Agent\_4 \Rightarrow 66.221868km$$

The cryptographic protocol starts with Phase 1. In the first round the interval $[0, 75]$ km is partitioned into 5 equal intervals. As a result, the encrypted representation of agents' distance (in km) is:

	$0-15$	$15-30$	$30-45$	$45-60$	$60-75$
$Agent\_1$	$E(1)$	$E(1)$	$E(2)$	$E(2)$	$E(2)$
$Agent\_2$	$E(1)$	$E(1)$	$E(1)$	$E(1)$	$E(2)$
$Agent\_3$	$E(1)$	$E(1)$	$E(2)$	$E(2)$	$E(2)$
$Agent\_4$	$E(1)$	$E(1)$	$E(1)$	$E(1)$	$E(1)$

The final global message in round 1 is shown below:

	$0-15$	$15-30$	$30-45$	$45-60$	$60-75$
$result$	$E(1)$	$E(1)$	$E(2^2)$	$E(2)$	$E(2)$

The decryption of the ciphertexts of the final message reveals that the first non-empty interval is the interval $[15, 30)$ (in km), which contains two doctors. Since the number of doctors in the first interval is equal or less than K, Phase 1 terminates. In Phase 2, the root-node broadcasts the first interval to all nodes. Each of the two nodes that understand that they are in this interval, sends its exact distance and a random ID nonce to the NSG. The nodes use onion routing to send their message anonymously.

The NSG receives the following two exact distances and the associated random ID numbers:

$$[Dist = 17.544817, ID = 56770656]$$
$$[Dist = 25.797003, ID = 45413392]$$

The NSG finds that the minimum distance is 17.544817 km. In Phase 3, the NSG sends the random ID nonce that is associated with the minimum distance to the root-node, which in turn broadcasts this ID to the doctors' network.

$$ID : 56770656$$

Finally, $Agent\_1$ realizes it is the nearest doctor and directly contacts the NSG to offer its services. A snapshot of the application in the implementation phase of the experiment is shown in Figure 7.

6. DISCUSSION

The experiments with the NDP prototype confirmed the feasibility of the approach in our NDP solution. However, there are still important open questions, technical and non-technical. One issue concerns the possibility for acceptance of a solution like NDP by real doctor communities or other communities that could use such a system. Clearly, many individuals may not be eager to adopt a technology like the NDP solution in their everyday life. However, such difficulties commonly exist for every new technology. We believe that the doctors' community should not feel threatened in any way by an application like the NDP solution, because:

1. The privacy of each doctor's location is preserved and under the absolute control of the doctor.
2. The solution is simple and cheap enough to be feasible even with current information and communication technologies (ICT).
3. The benefits of an application like NDP for the public health is practically immeasurable.

There are of course several technical issues that should be addressed. Even though wireless communication options like 3G, Wi-Fi and Satellite communications are now widely available there are still technical and economic issues. For instance, a personal mobile device will have to regularly update the doctor's location at his personal data management agent. This may cause the energy consumption of the mobile device and the cost for the wireless data transfer to become prohibitive. However, the current momentum of mobile device technology and telecommunications services predispose that these issues will soon be overcome.

Important technical difficulties concern the fault tolerance and the scalability of the network topology. A lookup of the nearest doctor may take too long if the size of the doctors' community is in the order of many hundreds and above. Furthermore, a temporary node failure, which becomes more likely as the network size increases, can disrupt the whole distributed computation. We believe that a fully developed network platform based on Quantum and Chord can address these technical issues of the network topology.

Finally, using a more appropriate distance metric would be a straightforward improvement of the NDP solution. In the prototype, we use the great-circle distance. Navigation software at the agents' side could for example use the GPS location to calculate an estimate of the time that the doctor will need to reach the location of the emergency. Such a distance metric would be much more effective for the NDP problem.

In our future work, we plan to examine several possibilities. Completing the implementation of the network topology, in order support the establishment of the logical binary tree topology for computations in P2P network. This will allow us to conduct a larger and more realistic set of experiments. We believe that an optimized network may support efficient distributed computation between hundreds or even thousands of agents. An important direction is the investigation of privacy and security issues of NDP under different

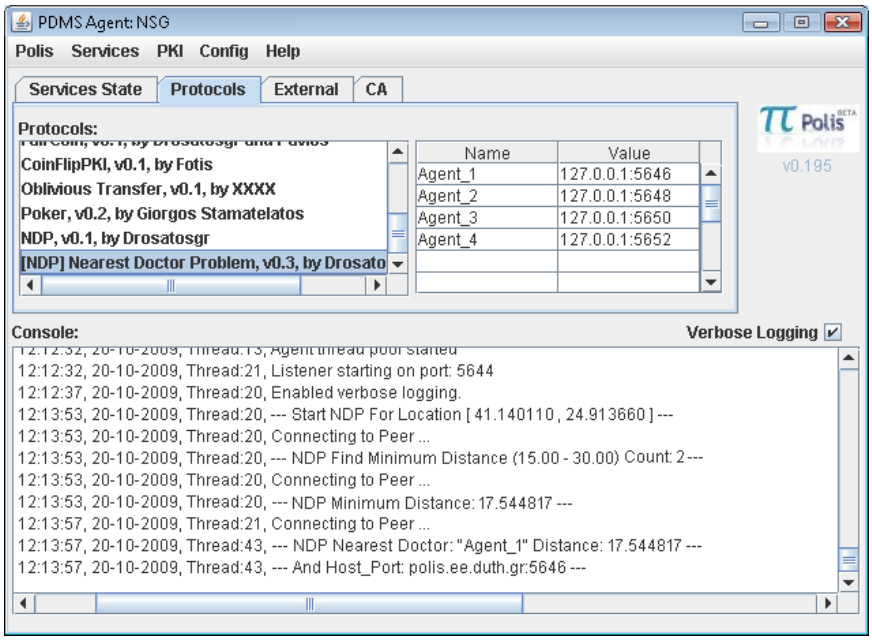

Figure 7: A snapshot of the NSG (NDP Service Gateway).

security assumptions, for example with the inclusion of a number of malicious nodes. Finally, the possibility of privacy leakages through the execution of a large number of consecutive NDP requests must also be examined.

7. REFERENCES

[1] A. Acquisti, S. Gritzalis, C. Lambrinoudakis, and S. De Capitani di Vimercati. *Digital privacy.* Auerbach Publications, Taylor & Francis Group, 6000 Broken Sound ParkWay NW, 2008.

[2] D. Bickson, D. Dolev, G. Bezman, and B. Pinkas. Peer-to-peer secure multi-party numerical computation. *Peer-to-Peer Computing, IEEE International Conference on*, 0:257–266, 2008.

[3] A. Chi-Chih Yao. Protocols for secure computations. In *Proceedings of Twenty-third IEEE Symposium on Foundations of Computer Science*, pages 160–164. Chicago, Illinois, November 1982.

[4] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. *Advances in Information Security*, volume 33. Springer US, 2007.

[5] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, pages 303–320, August 2004.

[6] P. S. Efraimidis, G. Drosatos, F. Nalbadis, and A. Tasidou. Towards privacy in personal data management. *Journal on Information Management & Computer Security*, 17(4), 2009.

[7] Europe's Information Society. eSafety, November 2009. http://ec.europa.eu/esafety.

[8] D. Hook. *Beginning Cryptography with Java.* Wiley Publishing, Inc., Indianapolis, In 46256, USA, 2005.

[9] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, Inc., Boca Raton, FL, USA, 1997.

[10] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *6th ACM MOBICOM*, Boston, MA, August 2000.

[11] M. Reed, P. Syverson, and D. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16(4), 1998.

[12] G. Stamatelatos, G. Drosatos, and P. S. Efraimidis. Quantum: A peer-to-peer network for distributed computations with enhanced privacy. In *EYRHKA 2009 Conference Proceedings*, pages 201–210. 3rd Pan-hellenic Scientific Student Conference on Informatics, September 2009.

[13] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *ACM SIGCOMM'01*, pages 149–160. San Diego, CA, August 2001.

[14] R. Want, A. Hopper, a. Veronica Falc and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102, 1992.

[15] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, October 1997.

[16] Wikipedia. Onion routing, November 2009. http://en.wikipedia.org/wiki/Onion_routing.

[17] M. Yokoo and K. Suzuki. Secure multi-agent dynamic programming based on homomorphic encryption and its application to combinatorial auctions. In *AAMAS'02*. Bologna, Italy, July 2002.