Computer Science and Information Systems 11(2):525–548 DOI: 10.2298/CSIS130120022D

User-Centric Privacy-Preserving Statistical Analysis of Ubiquitous Health Monitoring Data*

George Drosatos[1,2] and Pavlos S. Efraimidis[1,2]

[1] Dept. of Electrical and Computer Engineering, Democritus University of Thrace
University Campus, 67100 Xanthi, Greece
{gdrosato,pefraimi}@ee.duth.gr
[2] ATHENA, Research & Innovation Center, University Campus, 67100 Xanthi, Greece

Abstract. In this paper, we propose a user-centric software architecture for managing Ubiquitous Health Monitoring Data (UHMD) generated from wearable sensors in a Ubiquitous Health Monitoring System (UHMS), and examine how these data can be used within privacy-preserving distributed statistical analysis. Two are the main goals of our approach. First, to enhance the privacy of patients. Second, to decongest the Health Monitoring Center (HMC) from the enormous amount of biomedical data generated by the users' wearable sensors. In our solution personal software agents are used to receive and manage the personal medical data of their owners. Moreover, the personal agents can support privacy-preserving distributed statistical analysis of the health data. To this end, we present a cryptographic protocol based on secure multi-party computations that accept as input current or archived values of users' wearable sensors. We describe a prototype implementation that performs a statistical analysis on a community of independent personal agents. Finally, experiments with up to several hundred agents confirm the viability and the effectiveness of our approach.

Keywords: privacy, ubiquitous health data, privacy-preserving statistical analysis, personal software agent, secure multi-party computation.

1. Introduction

The requirement to provide health care to special groups of people who have the need of continuous health monitoring is an integral part of today's society. Moreover, the number of people who need such health monitoring services is increasing. An important reason for this is the aging of the populations, which constitutes a social and economical challenge especially for the developed countries [1]. Related researches which have been carried out both in the European Union [2] and the United States [3] indicate that the number of people over the age of 65 is increasing. A similar increase is expected to take place throughout the developed world. Many elderly people suffer from chronic diseases that require health care and frequent visits to hospitals. For people of this category, it is important to continuously monitor the state of their health. Effective monitoring of the health state can improve the quality of the patients' life or even save their life, while simultaneously reducing the cost of health care [4, 5].

* Preliminary parts of this work were presented at the 4rd International Conference on PErvasive Technologies Related to Assistive Environments (PETRA 2011) and the 8th International Conference on Trust, Privacy & Security in Digital Business (TrustBus 2011).

The rapid development of the wearable sensors technology led to the appearance and the implementation of prototype Ubiquitous Health Monitoring Systems (UHMS's) [4–6]. Moreover, there is a plethora of researches in the area of ambient assistive living services [7–9] and controlled access to ubiquitous hospital information [10]. The objective of a UHMS is to provide continuous health monitoring, both at home and outdoors. People need to have their health condition under control not only when at home, but wherever they are. One of the main features of a UHMS is to automatically generate alerts to notify the family or the patient's doctor about a possible health emergency so they should rush to their help to him. Examples of the data used for the detection of a possible health incident, as they are reported in [11], are: heart rate, blood pressure, galvanic skin response, skin temperature, heat flux, subject motion, speed and the covered distance.

Although ubiquitous computing is an opportunity for improving the health sector; however, for ubiquitous health monitoring technology to become feasible, a number of challenges are facing its presence [12]. These challenges are related to the deployment of this technology [13] and to issues such as resource constraints, user mobility, cost, heterogeneity of devices, scalability, security and privacy. While in [14] the author believes that challenges associated to sensor technology features also exist, such as Quality of Service (QoS), low power consumption and security of the wireless devices.

Privacy is an important issue of UHMS and health-related applications in general, since health data are sensitive personal data of patients. Privacy-related legislation like the European Data Protection Directive [15] and the HIPAA (Health Insurance Portability and Accountability Act) [16] explicitly define the rules for protecting the privacy of patients. The so far general architecture of a UHMS requires that all personal medical data (such as those reported above) which are produced by the patients' wearable sensors are collected and stored in a central service, specifically at the Health Monitoring Center (HMC) [4, 5]. The HMC is responsible not only for the collection and storage, but also for the control of these critical personal data. However, this technique runs significant risks for the security of the actual data, for the privacy of the monitored people, and, moreover, has an enormous computational and storage cost for the HMC. The distributed architecture that we propose in this work can offer the required scalability to handle large or even huge amounts of personal data.

At the same time, statistics of personal health data can be of high value for medical purposes. For example, the use of statistical methods is an integral part of medical research. A medical statistic may comprise a wide variety of data types, the most common of which are based on vital records (birth, death, marriage), morbidity (incidence of disease in a population) and mortality (the number of people who die of a certain disease in relation with the total number of people). Additional personal data items may needed for other well-known statistical computations, like the demographic distribution of a disease based on geographic, ethnic, and gender criteria, the socioeconomic status and education of health care professionals, and the costs of health care services.

In this work, we deal with the privacy-enhanced management of ubiquitous health monitoring data. Moreover, we describe how this data can be used within privacy-preserving distributed statistical analysis. Regarding the first deal, we suggest the decentralization of the collection of medical data at the users' side. This is achieved by the use of personal agents that will be continuously online and collect the medical data of their owners. In addition to the data that are obtained by wearable sensors, the agents may also have other data, such as demographic elements about the patient and further information about his health records, as well. The additional data can be used to support filtering of the results within distributed computations. Apart from the management of the personal data, the patient agent's automatically monitors the different changes in medical data with a dedicated health component. As soon as the health component detects aberrations in the raw health data, it informs the HMC by giving it access to the user's data so as to decide itself for the danger of the situation. In our approach, the usage of the agents does not block the remote monitoring of the patient's health by an authorized doctor; it only ensures the controlled, user-aware, access to these sensitive data.

For the statistical analysis, we propose a cryptographic protocol based on secure multiparty computations that accept as input current or archived values of users' wearable sensors. This distributed computation is performed by a community of the patients' personal software agents. We design an algorithm for the distributed computation, present a prototype implementation of the proposed solution, and obtain experimental results that confirm the viability and the effectiveness of our approach.

Main Advantages of Our Solution

The personal data management approach proposed in this work achieves a number of advantages in comparison with the existing architecture of a UHMS, and simultaneously enhances the privacy of the patients in such a system. The main advantages are:

- Only controlled access to the health data is provided. Every data access is logged by keeping who retrieved which data items and when this happened.
- The whole history of medical data, including the raw sensors' data, can be kept in the agent, whereas this might not be possible on the HMC for practical reasons. At the same time, decongestion of the HMC from the large amount of data, is achieved. This can make the computational requirements of the central servers more tolerable.
- Less risk of massive theft of personal data since they are distributed at the users' side.
- Option for usability of these data by authorized third independent services or for performing distributed computations.

On the other hand, important advantages of our statistical analysis approach in comparison to traditional statistical analysis techniques are:

- Utilizing valuable, sensitive, up-to-date personal data while ensuring privacy.
- Simplifying the process and significantly reducing the time and cost for conducting a statistical analysis.

A prerequisite for our approach is that each patient must have a personal software agent at his disposal and permanent access to the Internet. The computational requirements for the personal agent can be fulfilled with commodity hardware and hence its cost is not high. Thus, it is plausible to assume that patients with a UHMS can afford the extra cost for such an agent.

Our Contributions

- We present a user-centric software architecture for managing UHMD generated from wearable sensors in a UHMS, that allows controlled access to the health data, decongests the HMC, and enhances the privacy of users.

- We propose the usage of personal software agents for the management of biomedical data at the user side.
- We implement a prototype of the agents for this work.
- We present a privacy-preserving cryptographic protocol for distributed statistical analysis of the health data within agent communities.
- We validate our approach with a set of experiments on generated biomedical data in a community of real software agents.

Outline. The rest of this paper is organized as follows. In Section 2, we describe related work. In Section 3, we introduce the management architecture of our privacy-enhanced UHMS. In Section 4, we propose a system for performing privacy-preserving distributed statistical analysis on ubiquitous health data. Finally, conclusions of this work are given in Section 5.

2. Related Work

Personal data of users are commonly stored in central databases at the service provider's side. In this way, the users have essentially no control over the use of their personal data. The idea that individuals should own their personal information themselves and decide how this information is used, is discussed in [17]. A point made in [18] is that, although considering personal data the owner's private property is a very appealing idea, it would be rather difficult to practically apply it and legally enforce it. The argument that personal data would be safer at the user's side is also examined in [19].

To address privacy concerns, different kinds of frameworks that are related to personal data have recently been proposed. In particular, privacy sensitive management of personal data in ubiquitous computing is discussed in [20], and storing personal data in an individual's mobile device is examined in [21]. Of particular importance for the management of health data in this work is Polis [22], a framework for managing personal data at the owner's side. Polis offers privacy-enhanced management of personal data based on the principle that each individual has absolute control on his personal data, which remain permanently at the side of their owner and only there. Each user of Polis is a unique entity which is represented by a corresponding Polis agent. The Polis agents constitute the backbone of the Polis architecture; they are used to manage the personal data of an entity and provide controlled access at the entity's data. The service providers request personal data items of users from their personal agents. The agents provide the requested data if there is a corresponding policy and/or license agreement. In this work, we extend Polis agents with additional features and adapt the decentralized, agent-based approach of Polis for the management of the patients' personal data. Some work related to Polis has been done within the DISCREET project where a rich but also complicated framework for privacy protection has been proposed [23]. This framework is built on the principle that personal data is kept inside a "Discreet Box", located at the service provider's side. An agent-based solution to address usability issues related to P3P (Platform for Privacy Preferences Project) is presented in [24]. General surveys on privacy enhancing technologies are given in [25, 26].

In the second and main part of this work, we present a solution for distributed privacy-preserving statistical analysis of personal health data. Our approach is based on secure multi-party computations (MPCs). The general model of a MPC was firstly proposed by Yao [27] and later was followed by many others [28, 29]. In general, a MPC problem concerns the calculation of a function with inputs from many parties, where the input of each participant is not disclosed to anyone. The only information that should be disclosed is the output of the computation. The general solution for MPC presented in [27] is powerful but commonly leads to impractical implementations.

A secure two-party computation (S2C) for the calculation of statistics from two separate data sets is presented in [30]. Each data set is owned by a company and is not disclosed during the computation. Similar results are shown in [31], this time focusing on linear regression and classification and without using cryptographic techniques. Some indicative works from the related field of privacy-preserving data mining are [32–34]. A major difference of our work from the above is that in our approach every participant is in control of his health data and that the distributed computation is performed by the community of the personal software agents. Using software agents as building blocks for software systems is an established practice; see for example [35] and for a recent survey [36].

Another approach for statistics on personal data is anonymization, i.e., the sanitization of a data collection by removing identifying information. The data anonymization approach and some of its limitations are discussed for example in [37–39]. Data anonymization applies to data collections in central databases and is not directly comparable to our decentralized approach. Finally, an example of an efficient privacy-preserving distributed computation is given in [40], where personal agents of doctors execute a distributed protocol to identify the nearest doctor to an emergency. The focus of the present work is on privacy-preserving distributed statistical analysis using a massive number of participants.

3. Privacy-Enhanced Management of UHMD

In this section, we describe the proposed architecture for privacy-enhanced management of UHMD and show how it fulfills the goal of protecting the personal data and enhancing the privacy of patients.

3.1. Management Architecture

An overview of the proposed architecture for a UHMS is presented in Figure 1. The emphasis of the description is on the part of personal agents. The biomedical data that are produced by the patients' wearable sensors are wirelessly collected through a local wireless network in the patient's body into a personal mobile device, such as a smart phone. Afterwards, the measured biomedical data are transmitted via multiple complementary wireless networks (GPRS, 3G, Wi-Fi), through the Internet, towards the patient's personal agent. The personal agents that are used for this task are the Polis agents and have been suitably modified for this purpose. The features which have been added to the Polis agents so as to be used in a UHMS are:

1. Ability to collect dynamic personal data, such as the biomedical data of the patients' wearable sensors.
2. Ability to control the values of the biomedical data for the detection of some indicative cases of emergency.

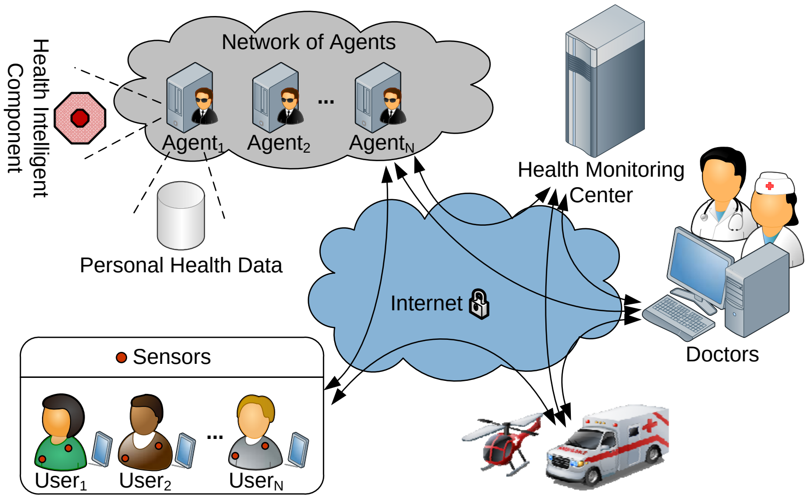


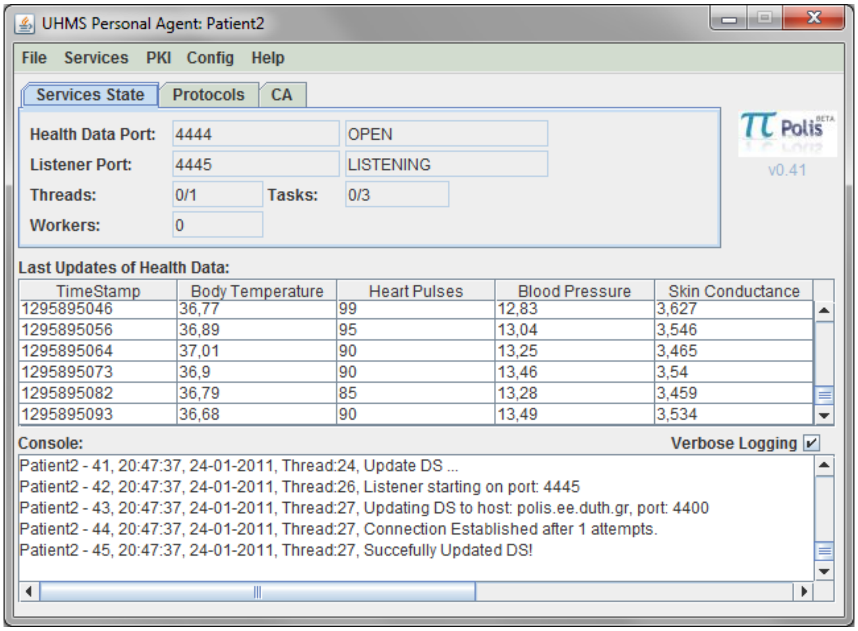
Fig. 1. The proposed management architecture for a UHMS.

A snapshot of a patients' personal agent is shown in Figure 2. On the other hand, the patients' personal agents are self-organized into an appropriate virtual network topology that can provide easy organization and identification of the agents. This network topology can be used as a tool to conduct privacy-preserving distributed computations.

Our architecture can support an intelligent health component which can make a first check of the health data in real time. We provide an overview of the functionality of such a component; a real implementation of such a tool is outside of the scope of this work. The health component of the personal agent checks automatically the incoming vital signs with the purpose to address for further thorough check in HMC if there are indications of an emergency (see Figure 3). An example of rules/decisions that a health component can apply in order to decide about an emergency can be found in [7]. If necessary, the HMC can be consulted by the personal doctor of the patient. The personal doctors are shown as "Doctors" in Figure 1. Depending on the situation, the HMC can coordinate the immediate medical service at the closest or most appropriate local medical facility using the best available transportation service (e.g.: ambulance). Finally, an additional responsibility of the HMC is to inform the family of the patient about his condition so that they could rush to provide their help.

3.2. Benefits of the Architecture

The idea of a decentralized architecture for storage and control of the patients' medical data into their personal agents, as it has already been mentioned provides the advantage of enhanced control on the user's personal data. Moreover, this decentralized approach can also contribute to improved data security, since invaders find large collections of personal data much more inviting than an individual's personal data [19]. The decentralized approach grants to the patient the right to control the disclosure of his health data and mitigates its feeling of being under permanent surveillance. In addition to enhancing privacy,



Health Data Port: 4444 OPEN

Listener Port: 4445 LISTENING

Threads: 0/1 Tasks: 0/3

Workers: 0

Last Updates of Health Data:

TimeStamp	Body Temperature	Heart Pulses	Blood Pressure	Skin Conductance
1295895046	36,77	99	12,83	3,627
1295895056	36,89	95	13,04	3,546
1295895064	37,01	90	13,25	3,465
1295895073	36,9	90	13,46	3,54
1295895082	36,79	85	13,28	3,459
1295895093	36,68	90	13,49	3,534

Console: Verbose Logging ☑

```
Patient2 - 41, 20:47:37, 24-01-2011, Thread:24, Update DS ...
Patient2 - 42, 20:47:37, 24-01-2011, Thread:26, Listener starting on port: 4445
Patient2 - 43, 20:47:37, 24-01-2011, Thread:27, Updating DS to host: polis.ee.duth.gr, port: 4400
Patient2 - 44, 20:47:37, 24-01-2011, Thread:27, Connection Established after 1 attempts.
Patient2 - 45, 20:47:37, 24-01-2011, Thread:27, Succefully Updated DS!
```

Fig. 2. A snapshot of UHMS personal agent. On the top, network configuration parameters of the agent can be seen. In the middle, the latest health data received by the agent are shown. Finally, at the bottom, logging information about the operation of the agents is presented.

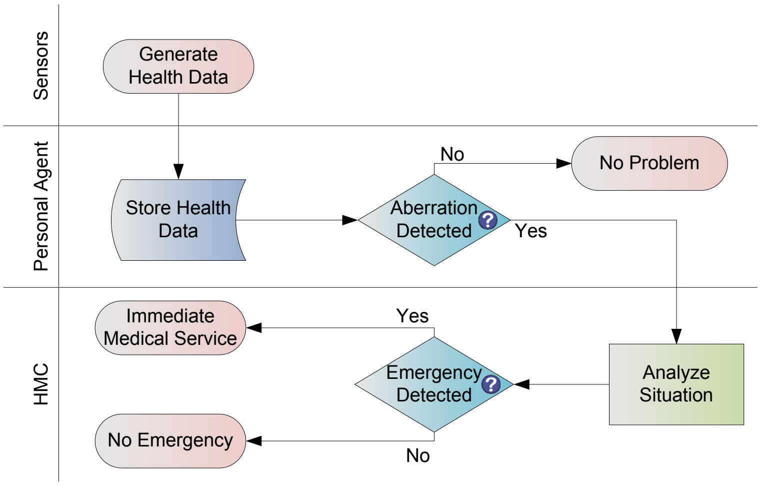
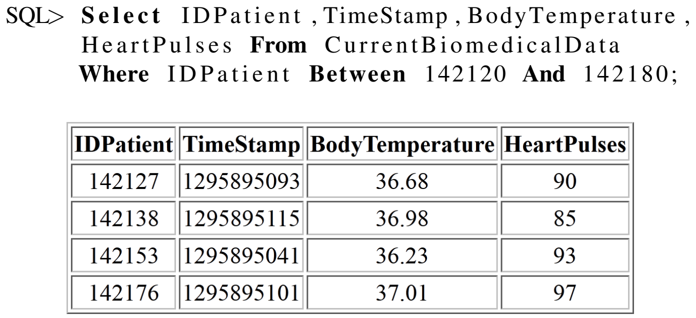


Fig. 3. A system flowchart of the biomedical information.

the decongestion of HMC from the huge amount of data, including raw sensors' data, that would be accepted if the patients sent their data directly to it, is achieved. Even in the case that the data would be collected at the HMC, these would be much less in volume than those that would actually be produced by the sensors, thus the analysis would not be as effective as the one that would be made by the agents themselves by having the complete data. With the proposed health data management approach of this work, the HMC has now to handle only those cases which may be at a certain risk.

Of course, the decentralized architecture holds challenges and issues too. The managing of a personal agent is by definition a critical task, prone to errors and omissions by the user. However, it is possible to mitigate these risks by standarizing or even automating the corresponding procedures of the agent. Furthermore, there are issues about the agent's security; a production-ready agent should satisfy high security levels. We believe that this is a viable task, since the agent has a precise, well-defined functionality and can be operated behind firewalls on a user-controlled computing platform. Also, another issue is what will happen if temporarily the patient's agent has no network connectivity (offline). In this case, the patient's data which are collected by his mobile device could be kept there and later be transmitted to the personal agent as soon as the failure is restored. Moreover, during a failure of the personal agent, health data could also be transmitted directly to the HMC for storage and control. Measures such as the above can ensure fault-tolerance against possible agent failures.

It is noteworthy that storing health data at the patients' side does not exclude the possibility to access the data from a central database as long as the database is entitled to do so. As shown in [22], the personal agents of Polis can be interconnected with mainstream database servers to provide transparent access to the personal data fields. The basic idea is that personal data fields in the central database do not contain the actual data; instead, a ticket represented by an appropriate data object is used to retrieve the data value on the fly. With this approach, which has been tested with an Oracle database server, a query submitted to the database may transparently retrieve – on the fly – personal data items from the associated personal software agents and present the personal data within the recordset (the answer of the database) of the query. An example query and the corresponding recordset are given in Figure 4. The data fields TimeStamp, BodyTemperature and HeartPulses are personal data fields and their content are – transparently for the database user – dynamically retrieved from the corresponding personal agents.



```
SQL> Select IDPatient ,TimeStamp ,BodyTemperature ,
     HeartPulses From CurrentBiomedicalData
     Where IDPatient Between 142120 And 142180;
```

IDPatient	TimeStamp	BodyTemperature	HeartPulses
142127	1295895093	36.68	90
142138	1295895115	36.98	85
142153	1295895041	36.23	93
142176	1295895101	37.01	97

Fig. 4. SQL access to remote health data.

The choice to store the patient's data in an agent enables the possibility to utilize these data for the common wealth. The Nearest Doctor Problem (NDP) [40] is mentioned as a typical example. The NDP is a privacy-preserving protocol, which uses a network of the doctors' agents aiming to find the nearest doctor in case of an emergency, by using dynamic data such as their location. In our case, the data of the patients could be used for a similar distributed computation. Such an example is the monitoring progress/spread of a pandemic in a region. Data such as the location and the body temperature of the patients would be required for this example. Another example is a medical statistical research on the biomedical data of the wearable sensors as well as on the medical records of patients. In the following section we use our distributed data management approach for a distributed statistical analysis application.

4. Privacy-Preserving Statistical Analysis on UHMD

In this section, we present a method for statistical analysis of ubiquitous health monitoring data (UHMD). For the statistical analysis we propose a privacy-preserving distributed computation that is collaboratively executed by the participating personal software agents. We first define the kind of privacy that is achieved and then proceed with the description of the distributed computation.

4.1. Privacy-Preserving Computation

There are two distinct problems that arise in the setting of privacy-preserving statistics/-data mining [41]:

(a) The first is to decide which functions can be safely computed, where safety means that the privacy of individuals is preserved if the result of the computation is disclosed. We will assume that the outcomes of the statistics computations do not violate the privacy of the participating patients and will not further consider this problem in this work.

(b) The second is how, meaning with which algorithms and protocols, to compute the results while minimizing the damage to privacy. For example, it is always possible to pool all of the data in one place and run the computation algorithm on the pooled data. However, this is exactly what we don't want to do (hospitals are not allowed to hand their raw data out, security agencies cannot afford the risk, and governments risk citizen outcry if they do). The focus of our work is on this problem.

Thus, the question we will address is how to achieve privacy of type (b), that is, how to compute the statistic results without pooling the data, and in a way that reveals nothing but the final results of the distributed computation.

4.2. Architecture of the Distributed Computation

Our solution is build on top of the privacy-enhanced UHMS presented earlier in this work. An overview of the architecture of the statistical analysis system including the extra components that are required for a distributed statistical analysis computation, i.e., the Network Community of Personal Agents and the Statistical Analysis Service (SAS), is shown in Figure 5.

The personal agents are organized into a virtual topology, which may be a simple ring topology or a more involved topology for time-critical computations. On the other hand, the SAS is a server that initiates the distributed computation on the users' medical data and collects the aggregate results. Each researcher who wishes to carry out a statistical research and is entitled to do so, can submit his task to the SAS.

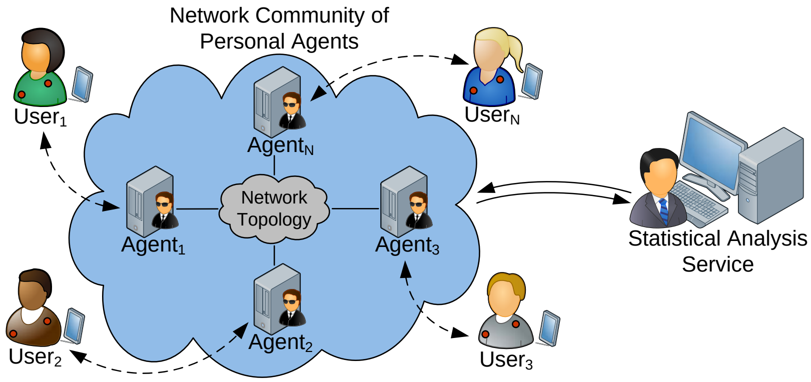


Fig. 5. The architecture for performing privacy-preserving statistical analysis.

4.3. The Main Steps of the Distributed Computation

The main steps of the distributed computation that we propose for the statistics calculation are:

- Initially, the researcher submits the request to conduct a specific statistical analysis to the SAS.
- The SAS accepts the request after verifying the credentials of the researcher.
- The SAS picks one of the personal agents to serve as the root-node for the particular computation and submits the request to it.
- The root-node coordinates a distributed computation that calculates the specified statistical function.
- At the end of the distributed computation, the SAS and the researcher will only learn aggregate results of the computation without any additional information of the personal data of individual participants.

4.4. The Secure Distributed Protocol

In this section, we present the main idea of the cryptographic protocol that is used in the statistical computations. The protocol is secure in the Honest-But-Curious (HBC) model (see Section 4.7), where the users' agents participating in the computation follow the protocol steps but may also try to extract additional information. During the calculation the actual users' personal data are not disclosed in any stage of the process but only the aggregate results are revealed at the end. An instance of a statistical computation problem consists of:

- **N patients** $P_1, P_2, \ldots, P_N$ and their personal data.
- **N personal software agents:** The agents of all patients that will participate in the distributed privacy-preserving computation.
  - **Input:** The type of the statistical function and its parameters. In addition, selectivity constraints for the data set may also be specified. Note that more than one statistical functions on the same dataset can be calculated with a single computation.
  - **Output:** The necessary aggregate values (e.g. w_x, u_x, z_{xy} and n, which are defined later) that are needed to calculate the given statistical function.

Consider the following statistical computation instance: Computing the average of the female patients' age in a city. First, we assume that the results of the specific query are not considered a threat against the users' privacy, that is, privacy type (a) of Section 4.1 is preserved. Then, given the computation instance, the SAS chooses a node from the network of the users' agents as the root-node for the particular computation. The SAS sends the type of the requested computation and its parameters to the root-node. The parameters of the computation, i.e., the female gender and the city name, are used to filter the data set. Each personal agent, decides privately to provide data or not to the statistical research.

A simple topology for the personal agents is a virtual ring topology that contains all agents as nodes (Figure 6.b). For time-critical computations, more complex topologies like a virtual tree can be used (Figure 6.a). The tree topology for example has been used in [40]. At the end of the execution, the root-node collects the results of the calculation as an encrypted message and sends it to the SAS. The message is encrypted with the public key of the SAS, which is assumed to be known to all nodes. In this way, the protocol ensures k-anonymity (see Definition 3), where $k = N$ and N is the number of all the nodes in the network.

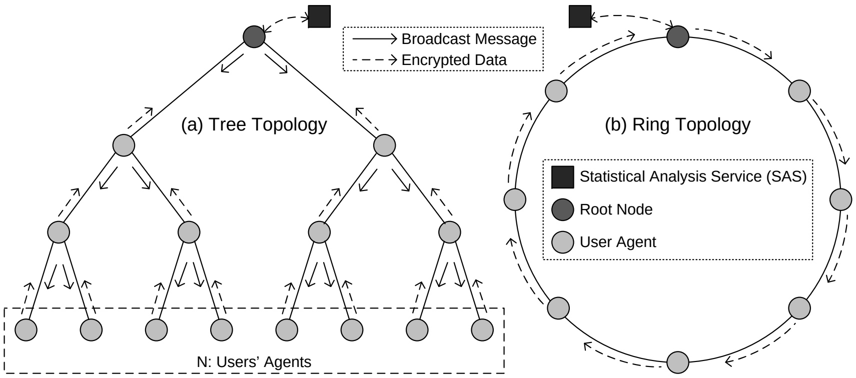


Fig. 6. Possible network topologies.

Cryptographic Tools. The *Paillier cryptosystem* [42] is a probabilistic asymmetric cryptographic algorithm for public key cryptography. The security of Paillier is implied by the Decisional Composite Residuosity Assumption (DCRA). In our cryptographic protocol, we use the additive homomorphic encryption property of the Paillier cryptosystem for calculating aggregate data in a privacy-preserving way.

Definition 1 (Homomorphic Encryption). *Homomorphic encryption [43, 44] 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. Particularly, an encryption algorithm $E()$ is homomorphic if given $E(x_1)$ and $E(x_2)$, one can obtain $E(x_1 \circ x_2)$ without decrypting x_1, x_2, for some operation $\circ$.*

The additive homomorphic encryption property of the Paillier cryptosystem means that multiplication of encrypted values corresponds to addition of decrypted ones, that is,

$$\begin{aligned} E(x_1) \cdot E(x_2) &= (g^{x_1} \cdot r_1^{n_p}) \cdot (g^{x_2} \cdot r_2^{n_p}) \\ &= g^{[x_1 + x_2 \bmod n_p]} \cdot (r_1 r_2)^{n_p} \bmod {n_p}^2 \\ &= E([x_1 + x_2 \bmod n_p]) \ , \end{aligned}$$

where

- x_1 and x_2 are two plain messages such that $x_1, x_2 \in \mathbb{Z}_{n_p}$,
- (n_p, g) is the Paillier public key,
- r_1 and r_2 are two random numbers such that $r_1, r_2 \in \mathbb{Z}^*_{n_p}$, and
- $E(m) = g^m r^{n_p} \bmod {n_p}^2$ is the encryption of message m.

The Paillier cryptosystem is a very popular additively homomorphic cryptosystem. It should be noted, however, that within our proposed distributed computation any other cryptosystem that supports the additive homomorphic property could also be used in place of the Paillier cryptosystem. For example the Benaloh cryptosystem [45] could be used within our solution. Moreover, it would also be possible to use the ElGamal cryptosystem [46], that supports multiplicative homomorphic property, provided that the computations are adapted accordingly. For example in this case one would have to transform the integer x to the group element z^x, for a fixed generator z, before encrypting with ElGamal. Thus, the transformation of multiplicative homomorphic property becomes $E(z^x) \cdot E(z^y) = E(z^x \cdot z^y) = E(z^{x+y})$.

4.5. The Computations

In this section, we use our approach to calculate representative statistical functions with a distributed computation. Wherever it is necessary, the expression of the statistical function is brought to a form that is appropriate for the distributed computation.

Arithmetic Mean. The arithmetic mean of a variable X (with sample space $\{x_1, \ldots, x_n\}$) is given by the following equation:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

We use the additive homomorphic property of Paillier encryption to calculate the value of the terms $u_x = \sum_{i=1}^{n} x_i$ and n. The calculation is privacy-preserving; no single x_i is disclosed. Once the SAS learns the values of the terms u_x and n, it can compute the arithmetic mean. More precisely, using the homomorphic property of Paillier, the two terms u_x and n can be transformed into the following form:

$$E_{pk}(u_x) = \prod_{i=1}^{n} E_{pk}(x_i) \text{ and } E_{pk}(n) = \prod_{i=1}^{n} E_{pk}(1) \ ,$$

where the E_{pk} indicates that the message is encrypted with the current public key of SAS for the specific statistical analysis. Each agent i that participates in the statistical analysis, prepares its own encryptions $E_{pk}(x_i)$ and $E_{pk}(1)$. These encrypted messages are used to calculate the above two global products. Agents that do not participate in the statistical computation (because for example they do not satisfy some selection criterion) multiply each of the above two products with an independent encryption of zero $E_{pk}(0)$.

Frequency Distribution. The frequency distribution is a tabulation of the values that one or more variables take in a sample. Each entry in the table contains the frequency or count of the occurrences of values within a particular group or interval; in this way, the table summarizes the distribution of values in the sample. The graphical representation of the frequency distribution is the well known histogram. Figure 7 shows how the frequency distribution would become by using ciphertext as counters in each range, where each ciphertext is given by the following equation:

$$E_{pk}(n_v) = \prod_{i=1}^{n} E_{pk}(m), \text{ where } m = \begin{cases} 1, \ x \in [x_{v-1}, x_v) \\ 0, \ x \notin [x_{v-1}, x_v) \end{cases}$$

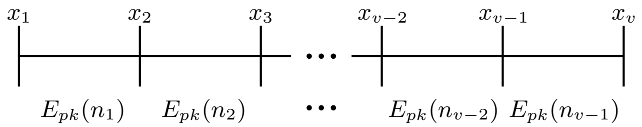


Fig. 7. Representation of a frequency distribution.

Linear Correlation Coefficient. The linear correlation coefficient $corr(X, Y)$ of two random variables X and Y is a measure of the strength and the direction of a linear relationship between two variables and is defined as:

$$corr(X, Y) = \frac{n \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \sqrt{n \sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2}}$$

The unknown terms that are required to calculate the linear correlation coefficient with the help of the homomorphic property of Paillier are $w_x = \sum_{i=1}^{n} x_i^2$, $u_x = \sum_{i=1}^{n} x_i$, $w_y = \sum_{i=1}^{n} y_i^2$, $u_y = \sum_{i=1}^{n} y_i$, $z_{xy} = \sum_{i=1}^{n} x_i y_i$ and n, by taking the following form:

$$E_{pk}(w_x) = \prod_{i=1}^{n} E_{pk}(x_i^2), \;\; E_{pk}(u_x) = \prod_{i=1}^{n} E_{pk}(x_i),$$

$$E_{pk}(w_y) = \prod_{i=1}^{n} E_{pk}(y_i^2), \;\; E_{pk}(u_y) = \prod_{i=1}^{n} E_{pk}(y_i),$$

$$E_{pk}(z_{xy}) = \prod_{i=1}^{n} E_{pk}(x_i y_i) \;\text{ and }\; E_{pk}(n) = \prod_{i=1}^{n} E_{pk}(1)$$

Variance. The variance $var(X)$ of a variable X is used as a measure of how far a set of numbers are spread out from each other. The unknown terms that are required to calculate the equation of variance with the help of the homomorphic property of Paillier are w_x, u_x and n. These terms can be calculated as shown earlier in the computations of the *arthmetic mean* and the *linear correlation coefficient*.

Covariance. The covariance $cov(X, Y)$ of two random variables X and Y is a measure of the strength of the correlation between the two variables. The unknown terms that are required to calculate the equation of covariance with the help of the homomorphic property of Paillier are u_x, u_y, z_{xy} and n.

Linear Regression. The linear regression of a dependent variable Y of the regressors X is given by the equation $y = a + bx$, where a and b are parameters. The unknown terms that are required to calculate the parameters of line y with the help of the homomorphic property of Paillier are w_x, u_x, u_y, z_{xy} and n.

From the analysis of the above statistical functions, we conclude that apart from the frequency distribution, all other function can be simultaneously calculated by computing once the required aggregate terms. Moreover, it is clear that the proposed solution can also be used to calculate other statistical functions, such as the polynomial regression and so on. We discuss such issues in the next section.

4.6. Computations with Homomorphic Cryptosystems

In our algorithm, we exploit the additive homomorphic property of Paillier to calculate additive aggregations which are then used to compute the values of statistical functions. We note that the same method can be used for multiplication-based aggregation if a cryptosystem supporting the multiplicative homomorphic property is used in place of Paillier. For example, the ElGamal and the RSA cryptosystems support multiplicative homomorphic encryption. Moreover, there are recent results on “somewhat” homomorphic cryptosystems, i.e., cryptosystems which support a limited number of homomorphic operations including both additive and multiplicative operations. More importantly, during the last years fully homomorphic cryptosystems supporting any number of additions and multiplications have been published, starting with the seminal work of Gentry [44]. Until now, fully homomorphic cryptosystems are not efficient enough to be used in practical applications like ours, though one could probably use somewhat homomorphic cryptosystems for some appropriate functions. A discussion of the efficiency and the practical relevance of current fully homomorphic and somewhat homomorphic cryptosystems [47].

4.7. The Protocol's Security

In this section, we show that the proposed protocol of a distributed statistical analysis in a UHMS does not violate the privacy of participants. The security holds for the model of Honest-But-Curious (HBC) users.

Definition 2 (Honest-But-Curious). *An honest-but-curious party (adversary) [48] follows the prescribed computation 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.*

In the cryptographic protocol described above, the information exchanged by agents is encrypted with the Paillier cryptosystem [42], which is known to offer Semantic Security [49], that is, it is infeasible for a computationally bounded adversary to derive significant information about a message (plaintext) when given only its ciphertext and the corresponding public encryption key. Consequently, assuming honest-but-curious parties and that users' agents do not collude with the SAS party outside of the protocol, our approach is semantically secure. In Section 4.8, we show that the case where some user agents collude with the SAS outside of the protocol can be handled with a threshold decryption model.

From the above, we conclude that the computation with the homomorphic encryptions does not leak personal information of participating individuals (privacy type (b) in Section 4.1). As noted earlier, the (decrypted) outcomes of the statistic computation are also assumed to preserve privacy of type (a). We can now discuss the privacy guarantee of the whole approach. A common criterion for privacy protection is k-anonymity, which requires that data of the outcome cannot be associated with any particular patient.

Definition 3 (k-anonymity). *A simple definition of k-anonymity [50] in the context of this work is that no less than k individual users can be associated with a particular personal data value.*

The proposed solution offers k-anonymity in the sense that the result computed at the end of the protocol cannot be attributed to any of the N participated agents, i.e., $k = N$ even if the list of participating users is known (assuming no background information on specific users is available). In summary, the key security features of our protocol are:

- Each agent that receives a message from the previous node cannot obtain information about the contents of the message, because the ciphertexts are encrypted with the Paillier cryptosystem.
- Each node alters the ciphertexts of the computation. Even the nodes that do not participate in the statistical function multiply the ciphertexts with an encryption of number

"0", which is the neutral element of the additive homomorphic property of Paillier. Thus, the ciphertext is modified at every node, even if the corresponding node does not give any input to the computation.

- At the end of the protocol, only the variables that are needed for a particular statistical function are revealed. As a result, no individual can be associated with the value that he had used in the computation. Consequently, the proposed protocol preserves k-anonymity for $k = N$, where N is the number of all agents in the network.

Another criterion for evaluating privacy protection is the concept of differential privacy. Loosely speaking, the aim of differential privacy is to ensure that the ability of an adversary to inflict harm (or good, for that matter) – of any sort, to any set of people – should be essentially the same, independent of whether any individual opts in to, opts out of, the dataset [51, 52]. The formal definition of differential privacy follows.

Definition 4 (ϵ-Differential privacy [52]). *A randomized function $\mathcal{K}$ gives ϵ-differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(\mathcal{K})$, the following holds:*

$$Pr[\mathcal{K}(D_1) \in S] \leq exp(\epsilon) \times Pr[\mathcal{K}(D_2) \in S]$$

The probability is taken is over the coin tosses of $\mathcal{K}$.

If privacy of type (a) (Section 4.1) is preserved, for example, no queries or sequences of queries addressing a very small number of individuals are permitted etc., then it is plausible to assume that our approach achieves a satisfactory level of differential privacy. Note that the outcomes of the statistical computations are sums or aggregate results computed from a large number of sensor measurements and demographic values of a large population. One may also consider of adding Laplace noise [53] to the statistical results in order to further enhance the differential privacy criterion, even though there is some recent criticism of such an approach [54].

4.8. Security Discussion

In this section, we identify some representative threats against our application and discuss how they are or can be addressed within our approach. The threats concern either the correctness of the aggregated results or the privacy of the involved participants.

- *Incorrect sensor measurements.* This case refers to the case where one or more sensors generate erroneous data of values large enough to significantly influence the aggregate result. Such incidents could disrupt a statistical analysis and would be difficult to be noticed in the statistical results. However, such incorrect measurements could be detected by the intelligent health component or some dedicated filter of the patient's agent and excluded from the current statistical analysis. This solution is acceptable in the HBC model. Moreover, even for the case where such incorrect measurements could be maliciously submitted in order to skew the statistical result, we could use more advance techniques of the area of electronic voting [55]. In this case, each node would have to run a zero-knowledge proof with its predecessor/s with purpose to verify that the measurements are within an acceptable range.

- *Dedicated queries with purpose to reveal personal biomedical data of a particular patient.* One query or a set of queries may be chosen and submitted to target specific patients, by using background information on the set of participating individuals. Such dedicated queries may cause leakage of personal data of the selected patients. As noted earlier, such an attack is a threat against privacy of type (a) and the participants have to be protected with respect to such attacks. The problem is well known in the area of statistical databases [56] and it is not something new. A possible solution could be to use a second authority which will check if there are enough patients who cover the query's criteria before the SAS performs the specific statistical analysis.
- *Collusion among some patients and the SAS.* In this case, the SAS will try to collaborate with at least two patients (in the simple ring topology) with purpose to reveal the private values of a patient. These two patients have to be the predecessor and the successor of the particular patient. More specifically, the colluding predecessor creates neutral ciphertexts and forwards them to the intermediate node. This node would then encrypt its private values and forward the result to its colluding successor (according to the topology). The successor would then immediately return the values to the SAS which now gets to decrypt these private values. Such malicious behaviors can be effectively handled by deploying threshold decryption model [57] for the decryption of the encrypted values. Threshold decryption model requires a number of designated parties exceeding an appropriate threshold to cooperate for the decryption to be possible.

4.9. Experimental Results

To evaluate our solution, we developed a prototype that carries out distributed statistical analysis on medical data. The application is implemented in Java and for the cryptographic primitives the Bouncycastle [58] library is used. The personal agents of the Polis platform [22] are used as the personal data management agents of the patients. For this approach, the Polis agents were suitably modified so as to be able to manage both health records and health data that would actually be collected through a secure communication channel by the patients' wearable sensors. The community of the personal agents is organized as a Peer-to-Peer network. At this stage of development of the prototype, the backbone of the topology is a virtual ring topology. The ring offers a simple and reliable solution for the interconnection of the agents. For time-critical calculations of statistics a more involved topology like a virtual tree should be used.

The personal agents use production-ready cryptographic libraries and employ 1024 bits RSA X.509 certificates. The communication between agents is performed over secure sockets (SSL/TLS) with both client and server authentication. Below we describe an experiment of a distributed statistical analysis with 6 agents and the SAS. The requested statistic is:

- *The arithmetic mean of the current body temperature of patients who are aged between 55 and 65 years old and their gender is female.*

For the needs of the experiment, each agent generates random values for the age, the gender, and the current body temperature. We assume that the selectivity of the query criteria is high enough to preserve privacy of type (a). Then, in brief, the statistical computation works as follows. Initially, the SAS randomly chooses a node from the agents' network, in this case agent 'Patient2', as the root-node, and forwards the description of the statistical computation to it. The values of each agent which are related to the computation are shown in Table 1. The last two columns show the aggregate values that are encrypted after the corresponding agent applies its values to the results. Since the homomorphic property of Paillier applies to integers, decimal values like the body temperature have also to be represented with integers. In our example, the temperature is rounded to a number with at most two decimal digits and then multiplied by 100 to become an integer.

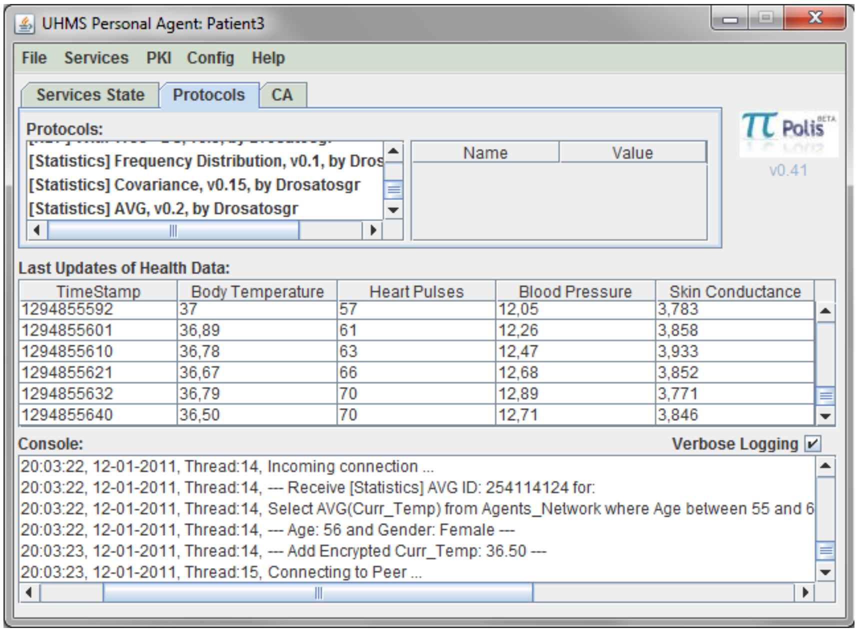
Table 1. Example of computation, where the agents in gray rows did not take part in computation.

Agent	Curr. Temp.	Age	Gender	$E_{pk}(u_x) = \prod_{i=1}^{n} E_{pk}(x_i)$	$E_{pk}(n) = \prod_{i=1}^{n} E_{pk}(1)$
Patient2	36.68 ^oC	51	Female	$E(0)$	$E(0)$
Patient3	36.50 ^oC	56	Female	$E(3650)$	$E(1)$
Patient4	37.70 ^oC	60	Female	$E(7420)$	$E(2)$
Patient5	38.10 ^oC	65	Female	$E(11230)$	$E(3)$
Patient6	37.12 ^oC	59	Male	$E(11230)$	$E(3)$
Patient1	36.20 ^oC	63	Female	$E(14850)$	$E(4)$

At the end of the computation, the agent 'Patient2' as the root-node collects the results and sends them back to the SAS. Finally, the SAS decrypts the results and finds that the average of the question which was submitted is $37.125\ ^oC$. A snapshot of the application during the execution of the experiment is shown in Figure 8.

We also performed a set of large-scale experiments with up to 300 agents. More precisely, we evaluated the efficiency of our solution with a series of experiments on a gradually increasing number of up to 300 agents. For this experiment, a network of 30 computer workstations with Intel Core 2 Quad Q8300 CPU's at 2.5 GHz, 2 GB RAM and a 100 Mbps network, were used. The workstations were running a 32-bit operating system and the agents were executed in 32-bit Java virtual machines. Each computer was shared by at most 10 agents, to ensure an even workload distribution and avoid single overloaded workstations; an overloaded workstation would become a bottleneck that could significantly delay the execution of the whole protocol.

The running times of our experiments are shown in Figure 9. In this figure, we present the execution times for the computation of the arithmetic mean, the variance and the frequency distribution (for 10 subintervals) functions. As expected, the execution times depend practically linearly on the number of agents which take part in the computation and on the number of encryptions and multiplications in every statistical function. The overall running time is more than satisfactory for batch execution of statistical computations. In case of large numbers of statistical computations, the rate of computations can be substantially improved by using a pipeline of independent computations. For cases where the run-time of the computations is important, the distributed computation can be executed on a virtual tree or some other – low depth – topology, instead of the ring topology. In this case one would expect, and we actually have such preliminary measurements albeit within a different context [40], that the total running time will depend only logarithmically on the total number of nodes.



Protocols:

[Statistics] Frequency Distribution, v0.1, by Dros
[Statistics] Covariance, v0.15, by Drosatosgr
[Statistics] AVG, v0.2, by Drosatosgr

Last Updates of Health Data:

TimeStamp	Body Temperature	Heart Pulses	Blood Pressure	Skin Conductance
1294855592	37	57	12,05	3,783
1294855601	36,89	61	12,26	3,858
1294855610	36,78	63	12,47	3,933
1294855621	36,67	66	12,68	3,852
1294855632	36,79	70	12,89	3,771
1294855640	36,50	70	12,71	3,846

Console:

```
20:03:22, 12-01-2011, Thread:14, Incoming connection ...
20:03:22, 12-01-2011, Thread:14, --- Receive [Statistics] AVG ID: 254114124 for:
20:03:22, 12-01-2011, Thread:14, Select AVG(Curr_Temp) from Agents_Network where Age between 55 and 6
20:03:22, 12-01-2011, Thread:14, --- Age: 56 and Gender: Female ---
20:03:23, 12-01-2011, Thread:14, --- Add Encrypted Curr_Temp: 36.50 ---
20:03:23, 12-01-2011, Thread:15, Connecting to Peer ...
```

Fig. 8. A snapshot of the agent ‘Patient3’.

Finally, the execution times of the computations could be significantly reduced by simply using 64-bit Java virtual machines for running the experiments. This change would greatly improve the execution times especially of the heavy encryption operations which involve BigInteger[3] variables. In a comparable, independent, experiment we noticed an almost four-times improvement of the execution times when 64-bit Java was used in place of 32-bit Java. The use of the 64-bit virtual machine seems to effectively exploit the bigger registers of the AMD64 architecture for the cryptographic operations.

5. Conclusions

The tendency of the society towards increasing numbers of elderly people and generally people who need continuous health monitoring makes the need of Ubiquitous Health Monitoring Systems (UHMS) imperative. At the same time the concerns of the public about privacy are also rising. In this work, we presented a software architecture for privacy-enhanced UHMS and proposed the use of the ubiquitous health data that are obtained by the wearable sensors in a UHMS for caring out statistical researches. The proposed architecture allows the patients to have enhanced control over their personal data, so as not to have the feeling of being continuously under surveillance. The enhanced control on their personal data was achieved by using personal software agents for the management of the patients’ personal data. Putting personal agents in charge of personal health data can open the way for the definition and implementation of new services which utilize

[3] BigInteger is an immutable arbitrary-precision integer.

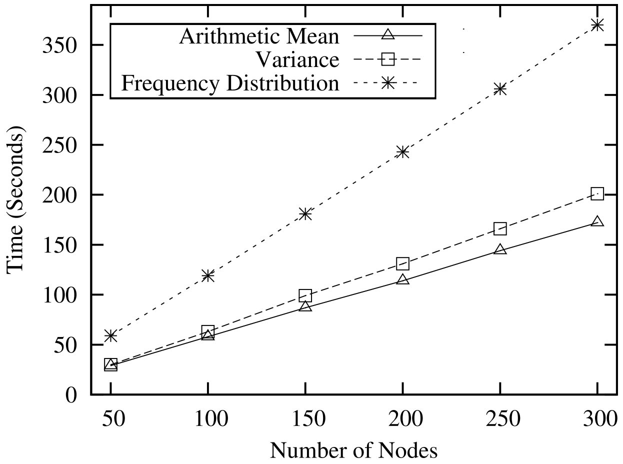


Fig. 9. Computation times of arithmetic mean, variance and frequency distribution (for 10 subintervals) statistical functions with respect to the number of agents.

personal data to contribute to public well being, while at the same ensuring the privacy of the involved individuals.

In this direction, we designed an algorithm for the distributed computation and, based on this algorithm and cryptographic primitives, we presented a solution for privacy-preserving statistical analysis on ubiquitous health data. The protection of privacy is achieved by using cryptographic techniques and performing a distributed computation within a network of patients' personal agents. We described how representative statistical functions can be executed distributedly by using the proposed cryptographic protocol. Finally, we developed a prototype implementation and performed an experimental evaluation that confirmed the viability and the efficiency of our approach.

Overall, our work demonstrates the feasibility of decentralized, scalable, privacy-enhanced management of Ubiquitous Health Monitoring Data (UHMD), and, most importantly, presents how privacy-preserving statistical analysis can be efficiently performed on such an architecture.

Acknowledgments. The present work was partially funded by the project ATLAS (Advanced Tourism Planning), GSRT/CO-OPERATION/11SYN-10-1730.

References

1. Commissioned by Philips. Healthcare strategies for an ageing society. In *The fourth report in a series of four from the Economist Intelligence Unit*, pages 1–31, UK, December 2009. The Economist. `http://graphics.eiu.com/upload/eb/Philips_Healthcare_ageing_3011WEB.pdf`.
2. Asghar Zaidi. Features and challenges of population ageing: The european perspective. In *This Policy Brief is derived from the presentation made at the Social and Economic Council of Spain (CONSEJO ECONOMICOY SOCIAL, CES, Madrid), as their keynote speaker in the*

conference "Ageing of Population". European Centre for Social Welfare Policy and Research, 2008. `http://www.euro.centre.org/data/1204800003_27721.pdf`.

3. G.T. Huang. Monitoring mom: As population matures, so do assisted-living technologies. In *Technical Review 20*, July 2003.
4. Chris Otto, Aleksandar Milenkovic, Corey Sanders, and Emil Jovanov. System Architecture of a Wireless Body Area Sensor Network for Ubiquitous Health Monitoring. *Journal of Mobile Multimedia*, 1:307–326, January 2006.
5. Arjan Durresi, Arben Merkoci, Mimoza Durresi, and Leonard Barolli. Integrated biomedical system for ubiquitous health monitoring. In *Proceedings of the 1st international conference on Network-based information systems*, NBiS'07, pages 397–405, Berlin, Heidelberg, 2007. Springer-Verlag.
6. Akira Yamazaki, Akio Koyama, Junpei Arai, and Leonard Barolli. Design and implementation of a ubiquitous health monitoring system. *Int. J. Web Grid Serv.*, 5:339–355, December 2009.
7. Dimitrios D. Vergados. Service personalization for assistive living in a mobile ambient healthcare-networked environment. *Personal Ubiquitous Comput.*, 14:575–590, September 2010.
8. A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, and J. Stankovic. Alarm-net: Wireless sensor networks for assisted-living and residential monitoring. Technical report, Department of Computer Science, University of Virginia, 2006.
9. L. Atallah, B. Lo, Guang-Zhong Yang, and F. Siegemund. Wirelessly accessible sensor populations (wasp) for elderly care monitoring. In *Proceedings of the 2nd International Conference on Pervasive Computing Technologies for Healthcare*, PervasiveHealth '08, pages 2 –7, 2008.
10. Xuan Hung Le, Sungyoung Lee, Young-Koo Lee, Heejo Lee, Murad Khalid, and Ravi Sankar. Activity-oriented access control to ubiquitous hospital information and services. *Information Sciences*, 180(16):2979 – 2990, 2010.
11. Fabrice Camous, Dónall McCann, and Mark Roantree. Capturing personal health data from wearable sensors. In *Proceedings of the 2008 International Symposium on Applications and the Internet*, pages 153–156, Washington, DC, USA, 2008. IEEE Computer Society.
12. M. Elkhodr, S. Shahrestani, and H. Cheung. Ubiquitous health monitoring systems: Addressing security concerns. *J. Comput. Sci.*, 7(10):1465–1473, 2011.
13. Shinyoung Lim, Tae Hwan Oh, Y.B. Choi, and T. Lakshman. Security issues on wireless body area network for remote healthcare monitoring. In *Proceedings of IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC)*, pages 327–332, 2010.
14. T. Znati. On the challenges and opportunities of pervasive and ubiquitous computing in health care. In *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom '05)*, pages 396–396, 2005.
15. European Parliament. Directive 95/46/EC. In *Official Journal L 281*, pages 0031–0050. 24 October 1995. `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:EN:HTML`.
16. 104th U.S. Congress. Health insurance portability and accountability act. In *Public Law 104-191*. Aug. 21 1996.
17. K.C. Laudon. Markets and privacy. *Commun. ACM*, 39(9):92–104, 1996.
18. P. Samuelson. Privacy as intellectual property? *Stanford Law Review*, 52:1125, 2000.
19. Deirdre Mulligan and Ari Schwartz. Your place or mine?: privacy concerns and solutions for server and client-side storage of personal information. In *Proceedings of the tenth conference on Computers, freedom and privacy: challenging the assumptions (CFP '00)*, pages 81–84, New York, NY, USA, 2000. ACM.
20. J.I. Hong. *An Architecture for Privacy-Sensitive Ubiquitous Computing*. PhD thesis, University of California at Berkeley, Computer Science Division, Berkeley, 2005.
21. P. Jäppinen. *ME - Mobile Electronic Personality*. PhD thesis, Lappeenranta University of Technology, Finland, 2004.

22. Pavlos S. Efraimidis, Georgios Drosatos, Fotis Nalbadis, and Aimilia Tasidou. Towards privacy in personal data management. *Journal on Information Management & Computer Security*, 17(4):311–329, 2009.
23. G.V. Lioudakis, E.A. Koutsoloukas, N.L. Dellas, N. Tselikas, S. Kapellaki, G.N. Prezerakos, D.I. Kaklamani, and I.S. Venieris. A middleware architecture for privacy protection. *Comput. Networks*, 51(16):4679–4696, 2007.
24. Hsu-Hui Lee and Mark Stamp. An agent-based privacy-enhancing model. *Information Management & Computer Security*, 16(3):305–319, 2008.
25. Ian Goldberg. Privacy-enhancing technologies for the internet iii: Ten years later. In A. Acquisti, S. Gritzalis, C. Lambrinoudakis, and S. di Vimercati, editors, *Chapter 1 of Digital Privacy: Theory, Technologies, and Practices*. Auerbach, December 2007.
26. S. Gritzalis. Enhancing web privacy and anonymity in the digital era. *Information Management and Computer Security*, 12(3):255–287, 2004.
27. Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *Proceedings of Twenty-third IEEE Symposium on Foundations of Computer Science*, pages 160–164. Chicago, Illinois, November 1982.
28. Li Shundong, Wang Daoshun, Dai Yiqi, and Luo Ping. Symmetric cryptographic solution to yaos millionaires problem and an evaluation of secure multiparty computations. *Information Sciences*, 178(1):244 – 255, 2008.
29. Kun Peng, Colin Boyd, Ed Dawson, and Byoungcheon Lee. Ciphertext comparison, a new solution to the millionaire problem. In *Proceedings of the 7th International Conference on Information and Communications Security (ICICS 2005)*, volume 3783 of *LNCS*, pages 84–96. Springer, 2005.
30. W. Du and M. Atallah. Privacy-preserving cooperative statistical analysis. In *Proceedings of the 17th Annual Computer Security Applications Conference*, pages 102–112, Washington, DC, USA, 2001. IEEE Computer Society.
31. Wenliang Du, Shigang Chen, and Yunghsiang S. Han. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *Proceedings of the 4th SIAM International Conference on Data Mining*, pages 222–233, 2004.
32. Murat Kantarcioglu and Onur Kardes. Privacy-preserving data mining in the malicious model. *International Journal of Information and Computer Security*, 2:353–375, January 2008.
33. Yitao Duan, NetEase Youdao, John Canny, and Justin Z. Zhan. P4P: practical large-scale privacy-preserving distributed computation robust against malicious users. In *USENIX Security Symposium*, pages 207–222, 2010.
34. Nissim Matatov, Lior Rokach, and Oded Maimon. Privacy-preserving data mining: A feature set partitioning approach. *Information Sciences*, 180(14):2696 – 2720, 2010.
35. Michael R. Genesereth and Steven P. Ketchpel. Software agents. *Commun. ACM*, 37(7):48–ff., July 1994.
36. Costin Badica, Zoran Budimac, Hans-Dieter Burkhard, and Mirjana Ivanovic. Software agents: Languages, tools, platforms. *Comput. Sci. Inf. Syst.*, 8(2):255–298, 2011.
37. Charu C. Aggarwal. On k-anonymity and the curse of dimensionality. In *Proceedings of the 31st international conference on Very large data bases*, VLDB '05, pages 901–909. VLDB Endowment, 2005.
38. Victor Muntés-Mulero and Jordi Nin. Privacy and anonymization for very large datasets. In *Proceeding of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 2117–2118, New York, NY, USA, 2009. ACM.
39. Sheng Zhong, Zhiqiang Yang, and Tingting Chen. k-anonymous data collection. *Information Sciences*, 179(17):2948 – 2963, 2009.
40. George Drosatos and Pavlos S. Efraimidis. An efficient privacy-preserving solution for finding the nearest doctor. *Personal and Ubiquitous Computing*, 18(1):75–90, 2014.
41. Yehuda Lindell and Benny Pinkas. Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, 1:59–98, 2009.

42. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology–EUROCRYPT 99*, pages 223–238. Springer Verlag LNCS 1592, 1999.
43. R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphisms. In *Foundations of Secure Computation*, pages 169–177. Academic Press, 1978.
44. Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st annual ACM symposium on Theory of computing (STOC '09)*, pages 169–178, New York, NY, USA, 2009. ACM.
45. Josh Benaloh. Dense probabilistic encryption. In *Proceedings of the workshop on selected areas of cryptography*, pages 120–128, 1994.
46. T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *Information Theory, IEEE Transactions on*, 31(4):469 – 472, July 1985.
47. Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the ACM workshop on Cloud computing security workshop (CCSW '11)*, pages 113–124, New York, NY, USA, 2011. ACM.
48. 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.
49. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270 – 299, 1984.
50. V. Ciriani, S. Capitani di Vimercati, S. Foresti, and P. Samarati. κ-anonymity. In Ting Yu and Sushil Jajodia, editors, *Secure Data Management in Decentralized Systems*, volume 33 of *Advances in Information Security*, pages 323–353. Springer US, 2007.
51. Cynthia Dwork. A firm foundation for private data analysis. *Commun. ACM*, 54:86–95, January 2011.
52. Cynthia Dwork. Differential privacy: a survey of results. In *Proceedings of the 5th international conference on Theory and applications of models of computation*, TAMC'08, pages 1–19, Berlin, Heidelberg, 2008. Springer-Verlag.
53. Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.
54. Rathindra Sarathy and Krishnamurty Muralidhar. Evaluating laplace noise addition to satisfy differential privacy for numeric data. *Transactions on Data Privacy*, 4(1):1–17, April 2011.
55. Steve Kremer, Mark Ryan, and Ben Smyth. Election verifiability in electronic voting protocols. In *ESORICS 2010*, pages 389–404, Heidelberg, 2010. Springer.
56. Nabil R. Adam and John C. Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Comput. Surv.*, 21:515–556, December 1989.
57. Ivan Damgård and Mats Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography (PKC '01)*, pages 119–136, London, UK, 2001. Springer-Verlag.
58. Bouncycastle. Legion of the bouncy castle, January 2011. `http://www.bouncycastle.org/`.

George Drosatos is a Post-Doctoral Researcher at the Athena Research and Innovation Center, branch of Xanthi (Greece). He received his diploma thesis in Electrical and Computer Engineering from Democritus University of Thrace (Greece) in 2006. In addition, he acquired a Master's degree (March 2010, advised by Prof. Alexandros Karakos) and a PhD degree (December 2013, advised by Assistant Prof. Pavlos S. Efraimidis) both from the Dept. of Electrical and Computer Engineering of the Democritus University of Thrace. His research interests are in the field of privacy and in particular privacy in ubiquitous computing.

Pavlos S. Efraimidis is an Assistant Professor at the Dept. of Electrical and Computer Engineering of the Democritus University of Thrace (Greece). He graduated from the Dept. of Computer Engineering and Informatics of the University of Patras (Greece) in 1996 and obtained a PhD in Informatics in 2000 from the University of Patras under the supervision of Paul Spirakis. His research interests are in the fields of algorithms and privacy. He is a member of ACM, IEEE and EATCS.

Received: January 20, 2013; Accepted: March 20, 2014.