ROBOTS IN ASSISTED
LIVING ENVIRONMENTS

UNOBTRUSIVE, EFFICIENT, RELIABLE AND MODULAR
SOLUTIONS FOR INDEPENDENT AGEING

Research Innovation Action

Project Number: 643892 Start Date of Project: 01/04/2015 Duration: 36 months

DELIVERABLE 5.6

Large-scale and privacy-preserving data fusion and interpretation I

Dissemination Level	**Public**
Due Date of Deliverable	Project Month 24, March 2017
Actual Submission Date	31 March 2017
Work Package	WP5, Overall architecture of the RADIO ecosystem of services for the medical care institutions and informal care-givers
Task	T5.3, RADIO overall data management process T5.4, Development of large-scale data aggregation and interpretation methods
Lead Beneficiary	NCSR-D
Contributing Beneficiaries	TWG
Type	R
Status	Submitted
Version	Final

Project funded by the European Unions Horizon 2020 Research and Innovation Actions

Abstract

This report describes the related data management processes concerning both technical related controls to protect against respective privacy/security issues and attacks as well as data management procedures aiming to defend the RADIO platform against soft issues such as information misuse and unauthorized access.

History

Version	Date	Reason	Revised by
01	13 Jun 2016	First draft, establishing document structure and work allocation. Introduction.	NCSR-D
02	12 Jul 2016	RASSP Protocol (Sections 2 and 3)	NCSR-D
03	24 Oct 2016	R interface (Section 4)	NCSR-D
04	20 Feb 2017	System and network security (Section 5)	TWG
05	27 Mar 2017	Pre final version given for review	NCSR-D
06	29 Mar 2017	Internal review	AVN
Fin	31 Mar 2017	Addressing review comments, final document preparation and submission	NCSR-D

Executive Symmary

RADIO Home deployments interact and exchange data beyond the boundaries of the local network. Actually, it is envisaged that RADIO Home deployments will seamlessly integrate in the RADIO ecosystem as nodes and collectively provide data mining capabilities to medical research institutions. The data exchange and processing between the entities of the RADIO ecosystem should employ techniques and methods for preserving the privacy of the data and protecting the data for misuse and unauthorized access.

This report describes techniques and methods for supporting a privacy preserving data mining system that can, on one hand, collaboratively compute statistical values and on the other hand prevent the leakage of private data outside the boundaries of a RADIO Home. Moreover, this report focuses on all the related data management processes concerning both the technical related controls to project against respective security issues as well as on data management procedures aiming to defend the RADIO platform against information misuse and unauthorized access.

Abbreviations and Acronyms

AAL	Ambient Assisted Living
RASSP	RADIO Secure Summation Protocol
REST API	Representational state transfer applications programming interface
IoT	Internet of Things
IPSec	Internet Protocol Security is a protocol suite for secure IP
VPN	Virtual Private Network
RPC	Remote Procedure Call

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

1.1 Purpose and Scope

The purpose of this deliverable is to provide the methods and techniques for the management of the data exchanged between the RADIO ecosystem entities in a privacy-preserving and secure way.

This deliverable will focus on all related data management processes concerning both technical related controls to protect against respective privacy/security issues and attacks as well as data management procedures aiming to defend the RADIO platform against soft issues such as information misuse, unauthorized access, accidental error etc.

The rest of the document is structured as follows. Section 2 discusses the state of the art on methods for privacy preserving data mining and presents the health data mining system developed in RADIO. Section 3 presents the RADIO privacy-preserving protocol that underlies the data mining system and Section 4 presents the medical researcher's interface to the data mining system. Section 5 discusses techniques used to ensure the network security and methods for preventing unauthorized access to private data.

1.2 Approach

Task 5.3 tackles all aspects of coordination and communication system emphasizing on security and privacy issues. Specifically this task will focus on data management processes concerning both technical related controls to protect against respective privacy/security issues and attacks as well as data management procedures aiming to defend the RADIO platform against soft security issues such as information misuse, unauthorized access, accidental error etc. TWG leads this task, supported by NCSR-D and AVN.

Task 5.4 develops and prototypes the methods needed by medical care institutions in order to aggregate and interpret the detailed ADL and mood recognition results into trends and averages at the right level of abstraction for inspection by medical personnel. The main considerations are scalability, privacy preservation and access to information on a need-to-know basis.

NCSR-D leads this task and contributes with large-scale and privacy-preserving data management at the medical care institutions site.

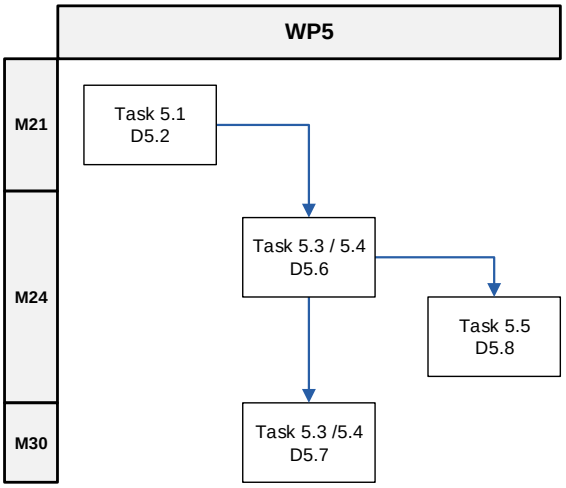

Figure 1: Dependencies between this deliverable and other deliverables.

1.3 Relation to other Work Packages and Deliverables

This deliverable is informed from *D5.2 Architecture of the RADIO Ecosystem II* about the architecture of the RADIO ecosystem.

This deliverable provides to *D5.8 Integrated RADIO prototype* the prototypes of the techniques and algorithms developed for large-scale and privacy-preserving data management. In its turn, D5.3 provides the requirements of the RADIO ecosystem architecture for the next iteration of *D5.7 Large-scale and privacy-preserving data fusion and interpretation II*.

This deliverable will be the basis for *D5.7 Large-scale and privacy-preserving data fusion and interpretation II*.

2 THE RADIO DATA MINING SYSTEM

The insights gained by the large-scale analysis of health-related data can have an enormous impact in public health and medical research, but access to such personal and sensitive data poses serious privacy implications for the data provider and a heavy data security and administrative burden on the data consumer. The discussion on what exactly it means to not disclose private data [4] and the discussion on policies for balancing between scientific advancement and privacy [6] are very relevant, but should be complemented by the equally relevant discussion of whether there is tension at all between data privacy and data-driven research. In other words, it is not straightforward if private data can be insulated from medical research workflows without compromising either.

As anonymization has been repeatedly proven to be inadequate [13], attention has turned to research in cryptography and distributed computation. These fields can provide methods for computing aggregates and statistics without revealing the specific data values involved in the computation, offering a much stronger guarantee of privacy than anonymization. However, from the perspective of the data mining practitioners and the medical researchers there is still a residue of functionality missing between their workflows over anonymized data and what is technically possible to achieve without accessing specific datapoints.

The scope of our discussion here is restricted to the data and processing required to empirically validate an already formulated hypothesis over a larger dataset than what can reasonably be made available to research. Naturally, part of the researchers' workflow involves browsing data in order to formulate a hypothesis. This initial hypothesis formulation remains in the scope of smaller experimental data specifically collected and licensed to be shared.

2.1 Entities

To make this more concrete, we will assume use cases from *ambient assisted living (AAL)* environments. AAL covers a wide range of concepts, hardware and software products, and services that facilitate better, healthier and safer life outside formal health-care institutions. These environments emphasise the automatic collection of health data in one's own environment and the secure sharing of such data with medical care providers. In such a system, health data is shared between the following entities:

- The *AAL agent* that is the data management component of the AAL environment. The AAL agent has unrestricted access to its user's sensitive data. The management and security of the data held by the AAL agent is primarily within the scope of network security.
- The *health-care provider* that needs access to sensitive data of a small set of individuals on a need-to-know basis, depending on the medical condition that necessitates the monitoring of each individual. The management and security of the data held by the health-care provider is primarily within the scope of network security and access control.
- The *medical researcher* that needs access to aggregate values computed over the sensitive data of a large set of individuals, but does not need to know any specific individual's data. It is the data transfer protocols between this agent and the AAL agents that are within the scope of the work described here.

2.2 Related Work

We see in the literature three major approaches to privacy-preserving computation: *differential privacy*, *homomorphic encryption*, and *secure multiparty computation*. *Differential privacy* is based on the property that a result of a statistical value can be approximated even if random noise has been added to the data. *Homomorphic encryption* supports computations over cipher-texts, so that the result can be obtained without decrypting individual datapoints. Finally, *secure multiparty computation* is based on

communication protocols between the agents to collaboratively compute a function over their private values without revealing the actual values.

Differential privacy preserves privacy by perturbing the datasets with randomized noise, such as symmetric exponential (Laplace) noise or with a use of a Geometric Distribution [16]. When the perturbed datasets are used in statistical analysis, knowledge of the distribution parameters of the noise applied allows approximating the analysis outcomes over the unperturbed data, but does not allow recovering any of the individual datapoints. To name an example, the PINQ data analysis platform [11] creates a differential privacy layer between the raw data and data analysis software. PINQ supplies the analyst with a set of transformations in operations like Where, Select, GroupBy and Join, in order to apply them to the data-set before applying operations for differential-privacy aggregations.

What should be noted about differential privacy is that it provides approximations and is only applicable where this is tolerated and where the datasets are large enough to allow for this approximation to be accurate enough for its purpose. In the the analysis for medical data, it is often the case that datasets are not large enough to give tolerable error margins or that outliers can lead to important insights and should be highlighted rather than smoothed out.

The second major strain of privacy-aware computation protocols is based on *homomorphic cryptosystems*, cryptographic mechanisms with the property that certain operators (such as addition) can be computed directly within the encrypted space without requiring that the individual operands can be decrypted. One of the most prominent homomorphic cryptosystems is Paillier's cryptosystem [14], which allows computing the cipher of the sum of two numbers given the ciphers of these numbers. Paillier's cryptosystem requires that all numbers are encrypted using derivatives of the public part of a master key; these derivatives are such that they cannot decrypt the cipher of other derivative keys, but the master key can decrypt the cipher of the sum. This algorithmic basis can be extended to provide further numerical and categorical operators beyond summation; for example Kissner and Song [10] proposed an extension that supports union, intersection and element reduction.

Although data providers are perfectly protected from their peers, the main weakness of homomorphic systems is the trust that must be placed on the entity that issues the master key [9]. The typical summation protocol based on Paillier's cryptosystem has a master agent issue a master key and a number of data agents that exchange their encrypted values between them in order to send a total encrypted summation back to the master agent. Privacy from the master agent is only guaranteed by the fact the master agent only receives the cipher of the end-result. If the master agent colludes with one malicious data agent, they can use the private part of the master key to reveal the private value of the victim agent, the data agent that passes its encrypted data to the malicious agent.

To lift the requirement that the master agent must be trusted, Shi et al. [16] proposed a framework that can compute statistics on medical data with the use of an *untrusted* data aggregator, by encrypting values that can be decrypted with the sum of multiple cipher-texts under different user keys. Shi et al. propose a method where each agent encrypts periodically its data with its respective private key. The data of every agent includes its private value combined with white noise. The untrusted aggregator receives all the encrypted values from the agents and decrypts them with its private key and with the use of a correlation between the private keys of all agents and a specific hash function, that is based on the time series. The algorithm needs an initial trusted setup phase, which does not allow agents to join or leave the system dynamically. The proposed protocol is based on differential privacy and as an implication the resulted statistic is an approximation of the real one, which may cause problems in medical data. Moreover, as authors report, in order for their approach to work efficiently, the plain text space should be small.

There are many studies that combine their secure mechanism with the use of a trusted third party that works as the aggregator. In trusted third party protocols, there is an external trusted party which receives the private data of the agents and computes a function by using them. Hanmanthu et al. [5] propose an enhanced protocol that combines a technique which perturbs distributed data with the use of a third party. Specifically, they define a protocol for constructing a Naive Bayes classifier. In this protocol, each

agent encrypts its perturbed data with its private key and sends it to a trusted third party. The trusted third party decrypts this data with the public key of the respective agent and constructs a perturbed Naive Bayes Classifier. Moreover, there are some studies that combine *secure multiparty computation (SMC)* systems with a trusted third party. Generally speaking, an SMC system deals with the computation of any function with any input in a distributed network, where the involved agents can learn only the total result and their own input. Thus, a common strategy to ensure trustworthiness is the use of a trusted third party. Ajmani et al. [1] present TEP, a trusted third party computation service that maintains generality. TEP offers flexibility because it fits in many SMC applications to guarantee privacy. However, this type of mechanism requires the existence of a trusted third party, so is inherently weaker than purely peer-to-peer networks.

Nevertheless, Sheikh et al. [15] proposed a SMC system that applies a secure summation protocol without the use of a trusted third party. The proposed protocol focuses on the increased computation complexity to avoid hacking. Each agent splits its data to a fixed number of segments and promotes a single segment to the next agent at each iteration. As an extension Sheikh et al. [15] define a master agent, which sets a random number during the initialization. Despite the fact that this protocol does not utilize a third trusted party, it is weak because if two neighbour agents collude, they can reveal the data value of the middle agent. Moreover, this technique imposes a considerable overhead in the communication between the agents.

Many recent research studies focus on privacy preserving on vertical and on horizontal partition of data. Our approach is oriented to horizontally distributed data, as each AAL agent keeps a private database with its values and each database contains the same set of attributes. Specifically, Karr et al. [7] propose a secure computation of linear regression for horizontally partitioned data without the use of a trusted third party. This is achieved by converting the linear regression equation to a summation form, where the quantities of each summation involve attribute values of the same agency. To protect data from the scope of the source and the values, they propose a SMC secure sum computation protocol. During the initialization of the protocol, a master agent adds his private value with a random number, that he previously produced, and forwards the summed value to the next agent. Each agent receives the aggregated value from the previous agent and forwards it to the neighbor agent, after the addition of his private value. The total summation result is returned back to the master agent, which removes his random number. This protocol is weak mostly because a private value of an agent can be revealed by the collusion of his neighbors. Also due to the circular mode of the algorithm, it can not be parallelised.

The study of Molina et al. [12] is closer to our approach. Specifically, they propose an application of homomorphic encryption to compute basic statistics on aggregated medical data which also guarantee the privacy of the medical data. Their SMC protocol preserves the privacy between the caregivers, where each one computes statistics for their corresponding patients. This is achieved with a double encryption, each one depending on a different public key — the public key of researcher and the public key of a caregiver chosen randomly to work as the aggregator. This approach can be mapped well in distributed systems because each caregiver can work in parallel to compute aggregates of their patients' data. However, privacy is relatively weak as the researcher and the aggregator can collude to reveal the plaintexts of each caregiver. Moreover, doubly homomorphic encryption schemes are not fully explored to define which statistics can be determined.

2.3 System Architecture

The system architecture can be perceived as a stack of three layers and each layer depends on the functionality provided from the layer at the lower stage. The upper layer, called the *Medical Researcher's interface*, accepts from the medical researcher the method with the initial parameters to be executed by the system. The purpose of this interface is to provide a familiar environment to the researcher and therefore in our current implementation this layer is developed in the R language.[1] The initial parameters are transformed appropriately in order to be passed to the next layer, which is the *Compilation*

[1] Cf. `https://www.r-project.org`

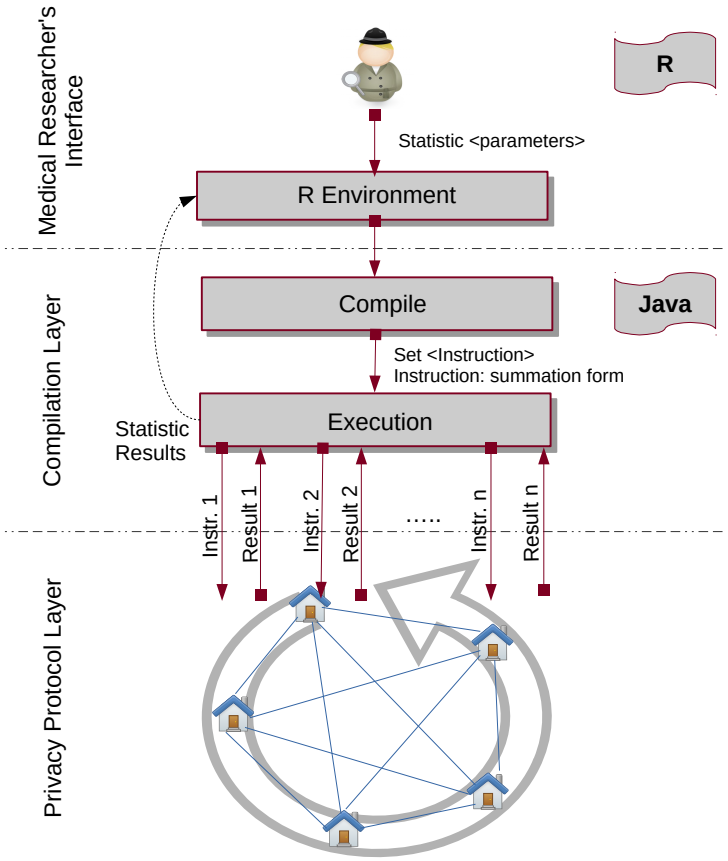

Figure 2: The system's architecture

Layer. At that stage, the high-level parameters and commands of the statistical method are transformed into low-level instruction for accessing the private databases of the agents. An instruction represents an aggregation over a selection of data. Currently, the aggregation operation is summation. However, the aggregations that are both feasible by the system and safe for preserving privacy depend on the secure protocol used. These instructions will be eventually evaluated by the lowest layer of the architecture, the *Privacy Protocol Layer*. Figure 2 depicts the system architecture and the information exchanged between the layers.

2.3.1 The Compilation Layer

This layer is responsible for the communication between the two other layers. Specifically, it translates the arguments of the *secure statistic* to a suitable format, thus it defines the appropriate data that are going to be used for the statistic computation. Moreover, it converts the simple statistic equations to a set of summations; a compatible format to achieve the secure summation protocol. Therefore, a set of instructions is composed where each instruction represents a summation equation of the statistic with the appropriate parameters set for its computation. During the execution, the compilation layer gives to the privacy protocol layer a single instruction at a time and it receives its result. After the execution of the whole set, it computes the statistic and the analysis parameters. The statistic result is sent back to the Medical Researcher's interface layer.

2.3.2 The Aggregation Protocol

This layer executes the privacy protocol between the AAL agents, To deal with the concurrent computation of each instruction, we model our agents as actors. Each actor makes the appropriate computations with respect to the given instruction and its private data. These computations can easily be done since

every AAL agent controls its corresponding health records. After the computation of the value, which represents the initial secret, the privacy protocol is executed. The protocol may involve all the actors to work collaboratively in order to compute the aggregation of their secrets without revealing the actual secrets to each other or the agent requesting the aggregation. The aggregated result is collected a designated actor. The selection of such actor is irrelevant and can be done randomly. Our proposed implementation for this layer is presented in more detail in Section 3.

2.3.3 Example

We will use a simple example to better demonstrate the proposed system. Suppose that a medical researcher needs to run a t-test to assess whether the means of two groups are statistically different from each other, that is to compute t in Eq. 1:

$$t = \frac{\bar{X} - \bar{Y}}{\sqrt{|X|^{-1}\,\sigma_X^2 + |Y|^{-1}\,\sigma_Y^2}} \tag{1}$$

where X and Y are the datapoints of the two groups, $\bar{X}$ and $\bar{Y}$ are their means, $|X|$ and $|Y|$ are their cardinalities, and σ_X^2 and σ_Y^2 their variances.

Assume, for instance, that a researcher wants to test the effect of medicine M_1 (Group 1) and medicine M_2 (Group 2) on blood pressure, with the further restriction that participants in Group 2 should be above 65 years old. A workflow using the R language would be:

- Select from a database the instances that match Group 1 criteria and store them in variable X
- Select from a database the instances that match Group 2 criteria and store them in variable Y
- Decide on the conditions of the T-test, such as the confidence level and alternation, and store them in variable C
- Pass X, Y, C as arguments to an implementation of t-test

Our architecture allows this workflow to remain essentially unaffected, except for the contents of X and Y. Instead of holding actual data arrays these now contain a representation of the Group 1 and Group 2 criteria, so that the selection can be executed in distributed manner. Using this representation, a privacy-aware implementation of t-test can produce the exact same result as the conventional implementation, except without ever accessing any individual data.

This representation declares a list of dependent variables and a list of eligibility criteria of the sample groups, as a set of $(\text{variable}, \text{operator}, \text{value})$ tuples. In our example, we assign to X and Y the criteria that we would have used to assign to them a value array if we had full access to the data:

- X = [("medicine", =, "M_1")]
- Y = [("medicine", =, "M_2"), ("age", >, "65")]

The compilation layer converts the t-test implementation into a set of instructions. Recall that each instruction is an aggregation over the private data of each agent, under the given selection restrictions. Table 1 defines the instructions needed to implement the t-test (Eq. 1), which is then implemented using the following pseudo-code:

1. $X =$ [("medicine", =, "M_1")];
   X is a representation of the secret values of all AAL agents where medicine M_1 is used.
2. $Y =$ [("medicine", =, "M_1"), ("age", >, "65")];
   Y is a representation of the secret values of all AAL agents where medicine M_2 is used and age is above 65.
3. $N_1 = \mathrm{add}(X); N_2 = \mathrm{add}(Y)$;
   N_1 is the sum of the secret values X and N_2 is the sum of the secret values Y.

Function	Definition
$\mathrm{add}(C)$	$\sum_i s_i(C)$, where $s_i(C)$ is the secret value of the i-th AAL agent if condition C is satisfied, 0 otherwise
$\mathrm{add}^2(C, k)$	$\sum_i (s_i(C) + k)^2$, where k is a constant and $s_i(C)$ is same as above
$\mathrm{cnt}(C)$	$\sum_i c_i(C)$, where $c_i(C)$ is 1 if the i-th AAL agent satisfies condition C, 0 otherwise.

Table 1: Characteristic instructions provided by the RASSP Protocol.

4. $C_1 = \mathrm{cnt}(X); C_2 = \mathrm{cnt}(Y)$;
   C_1 is the number of AAL agents with non-zero values in X and C_2 is the number of AAL agents with non-zero values in Y.
5. $\bar{X} = N_1/C_1; \bar{Y} = N_2/C_2$;
   This uses the values above to calculate means.
6. $\sigma_X^2 = \mathrm{add}^2(X, -\bar{X}); \sigma_Y^2 = \mathrm{add}^2(Y, -\bar{Y})$;
   This uses the values above to calculate variances.
7. $T = \left(\bar{X} - \bar{Y}\right) / \mathrm{sqroot}\left(\sigma_X^2/C_1 + \sigma_Y^2/C_2\right)$;

Each instruction is executed with the use of the secure summation protocol, obtaining the aggregate values specified in the instruction without obtaining the values themselves. From the perspective of the R interface user, the t-test functions operate as if they had been passed the actual value matrices as parameters.

2.4 Discussion

The proposed system architecture assumes that:

- The statistical analysis that is to be carried out can be implemented using the set of aggregation instructions provided by the aggregation protocol. In other words the algorithm should not depend on individual data points.
- A summation protocol exists that guarantees privacy.

The first assumption holds, since the most commonly used class of data mining algorithms can be expressed as an iteration of summation expressions [8]. If needed, categorical operators can be implemented based on summation [10].

Regarding the second assumption, we will now proceed to discuss the summation protocols that can be used in our architecture and, in Section 3, present the protocol we use in our reference implementation of the architecture.

Most of the related studies guarantee their privacy by utilising encryption or differential privacy techniques. These approaches do not fit in our problem, because we deal with medical history data that are distributed across AAL agents. In homomorphic techniques, a *master agent* shares a public key with the rest of the agents, in order to encrypt their data, and keeps a private key for the final decryption. Such a mechanism is privately weak in the case of collaborative computations, because if the medical researcher (master agent) and one AAL agent collude, they can learn another AAL agent's private value. This makes the technical protocol weak, as it places a heavy burden on non-technical policies and protocols to guarantee the integrity of the medical researcher. Since our main aim is to alleviate the need for non-technical policies and protocols and to make it easier for medical researchers to run

statistics over datapoint they are not meant to access directly, homomorphism encryption does not cover our requirements.

In addition, differential privacy is also not applicable, from both the perspective of the medical researcher as well as from that of the AAL agent. From the perspective of the medical researcher, differential privacy computes *approximations*, which can be a problem as discussed in Section 2.2 above. From the perspective of the AAL agent, the secret value can be approximated by its repeated querying, since a different perturbation of the real secret needs to be computed for each query. The AAL agent cannot produce a single perturbed value and use this for all queries, since it needs to be re-computed to follow the distribution parameters requested by the medical researcher. This might be less of a problem in time-series data (such as power grid data or traffic data), but can result in substantial information leaking in static historical data, such as health records.

2.5 Implementation

The project's source code is organized in three modules, each one implementing one of the layers in our architecture:

- `proto` implements the *aggregation protocol*
- `stats` is the implementation of statistical analysis primitives over an aggregation protocol, and implements the *compilation layer*
- `RStats` implements the R interface for the medical researcher over the compilation layer.

3 THE PRIVACY PRESERVING PROTOCOL

The RADIO data mining system presented in Section 2 is unaware of the underlying privacy-preserving protocol that it is using. In this chapter we will present the RASSP (RADIO Secure Summation Protocol) that satisfies the requirements needed by the system to ensure privacy preservation.

The rest of the chapter is organized as follows. Section 3.1 provides a brief but necessary introduction of the theoretical foundations of the privacy preserving protocol and the secret sharing schemes in general. Section 3.2 builts on the theoretical foundations to construct a practical privacy-preserving protocol that support summation as the core operation on the cluster of RADIO Homes.

3.1 Background

Secret sharing schemes divide a secret into many *shares* which can be distributed to n mutually suspicious agents. The initial secret can be revealed if any k of these n agents combine their shares. We will call such schemes, (k, n)-threshold schemes. If such a scheme also possesses the *homomorphism* property, then multiple secrets can be combined by direct computation only on the shares. Such schemes are usually called *composite secret sharing schemes* [2].

More specifically, assume n mutually suspicious agents and each agent holds a secret s_i. The desired computation is combination into a super-secret s under an operation $\oplus$, namely $s = s_1 \oplus \cdots \oplus s_n$. Using a secret sharing scheme each s_i can be split into k shares $d_{i_1}, \ldots, d_{i_k}$ such that given a known function F_I it is the case that:

$$s_i = F_I(d_{i_1}, \ldots d_{i_k})$$

We will say the (k, n) threshold scheme has the $(\oplus, \otimes)$-homomorphism property if whenever $s = F_I(d_1, \ldots, d_k)$ and $s' = F_I(d'_1, \ldots d'_k)$ then

$$s \oplus s' = F_I(d_1 \otimes d'_1, \ldots, d_k \otimes d'_k)$$

The composition of the shares d_1, d'_1 yield a *super-share* $d_1 \otimes d'_1$. In other words, the $(\oplus, \otimes)$-homomorphism property implies that the composition of the shares under the operator $\otimes$ are shares of the composition under the operator $\oplus$.

Overall, the advantage of having a composite secret sharing scheme is that secret cannot be obtained, only if k or more agents collude and combine their sub-shares. In addition, this protocol is suitable to our approach from the AAL agent's point of view, because it does not use a trusted third party or depends on cryptographic assumptions, while at the same time it is k-secure. This approach represents a secure summation protocol that can easily be applied to collaborative agent systems.

Based on this mathematical foundation, we will now proceed to present the RASSP protocol, a $(+, +)$-homophorphic composite secret sharing scheme.

3.2 The RASSP Protocol

Assume that we have n AAL agents, where each one has its private value $v_i, i \in [1..n]$. Each AAL produces random breakdown of v_i into n terms $r_{ij}, j = 1..n$ such that $v_i = \sum_{j=1}^{n} r_{ij}$. These terms are computed by first producing $n - 1$ random terms $r_{ij}, j = 1..i - 1, i + 1..n$ and then setting

$$r_{ii} = v_i - \sum_{j \in [1..n] - \{i\}} r_{ij}$$

The r_{ij} terms are called *sub-shares* and are (except for r_{ii}) shared with the rest of the AAL agents, one per agent. In this manner, each AAL agent shares $n - 1$ values and receives $n - 1$ values from the rest

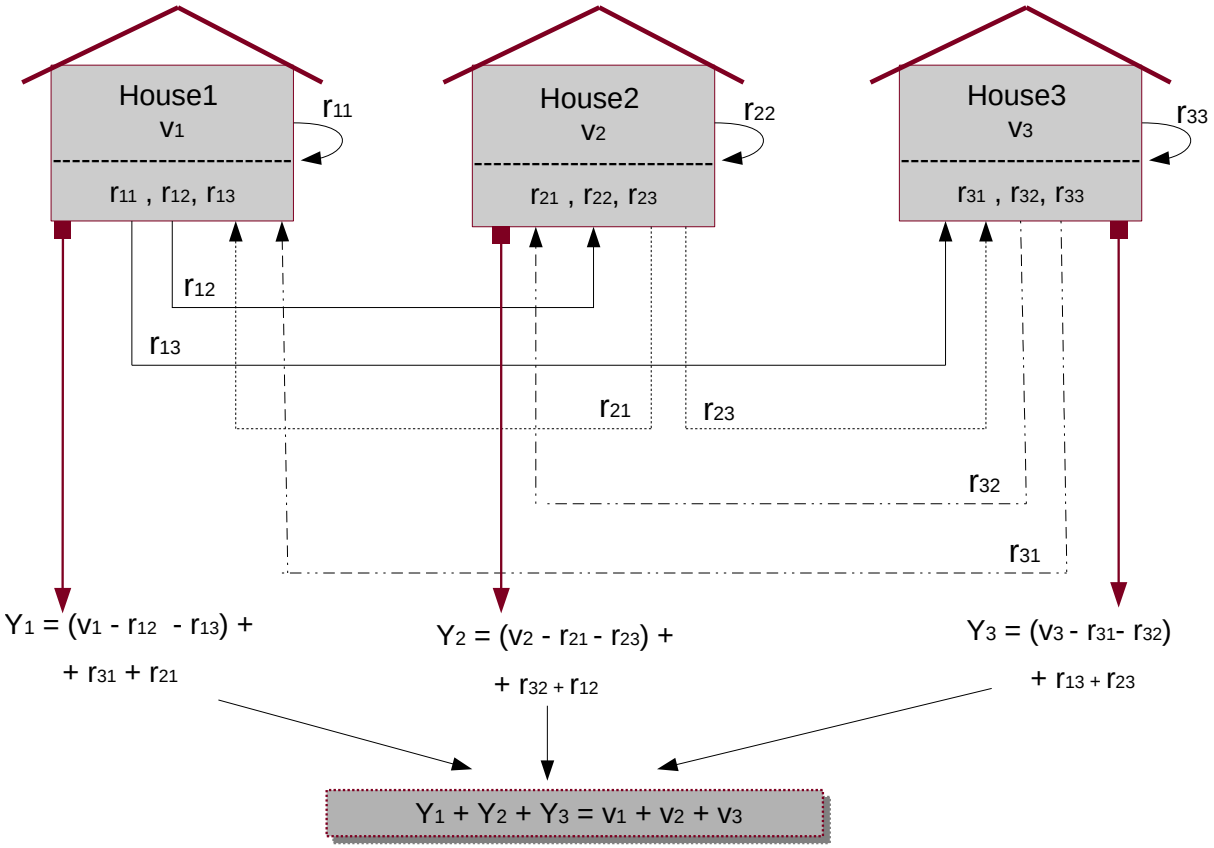

Figure 3: The RASSP secure summation protocol.

of the AAL agents. The *super-share* Y_i for each agent is defined as:

$$Y_i = r_{ii} + \sum_{k \in [1..n]-\{i\}} r_{ki} \tag{2}$$

Notice how the super-share of AAL agent i is the sum of the sub-share that it has not shared and of all the sub-shares that it has received from the other AAL agents. Finally, we define a function F_I as the sum of the super-shares:

$$F_I(Y_1, \ldots, Y_n) = \sum_{i=1}^{n} Y_i \tag{3}$$

It is straightforward to verify that $F_I(Y_1, \ldots, Y_n)$ is equal to the sum of all secrets. It is also straightforward to verify that only random numbers and obscured data values are shared between AAL agents and between AAL agents the researcher. Notice also that only if $(n-1)$ AAL agents collude to merge their sub-shares can the private value of the n-th agent be revealed. Therefore, our system guarantees $(n-1)$-security.

Figure 3 gives an example of the above description for a system of three AAL agents. In this example House1 has the private value v_1 and produces three numbers: r_{11}, r_{12}, r_{13}. Then, it shares r_{12} and r_{13} with House2 and House3, keeping r_{11} hidden. House1 receives two numbers (r_{21}, r_{31}) from the other AAL agents. In then shares the computed Y_1, so that F_I can be computed by summing all Y_i. $F_I(Y_1, Y_2, Y_3)$ computes the sum of all three AAL agents' secret values.

The described secure summation protocol is suitable for computing medical statistics and preserve privacy at the same time. The only constraint is that the resulted outcome is a sum of the private values, thus the statistic equations should be converted in a summation form. The summation form results in accurate values and not approximations, while simultaneously it can easily be parallelised [3]. Besides, medical researchers typically use descriptive statistics which utilise numerical descriptors such as mean

and standard deviation. These descriptors can easily be converted into a summation form, thus they can be computed by our system.

4 MEDICAL RESEARCHER'S INTERFACE

4.1 Implemented Statistics in RASSP

The statistics currently implemented in RASSP are shown in Table 2. What is important to note is that our implementation of the statistical tests presents an interface identical to the standard R implementation of the t-test.[2] The underlying difference is that the arguments in each function do not point to actual data matrices but to instances of our structure, which hold the information needed by the compilation layer in order to distribute the computation to the participating nodes. In the following section we go through each one of the statistics implemented and provide examples of the syntax in RASSP.

Table 2: Current list of statistic implementations offered to medical researcher.

R Function	Description
`mean`	Mean (average)
`var`	Variance
`stdev`	Standard Deviation
`normality`	Checks whether a sample follows a normal distribution
`plotnorm`	Plots Gaussian distribution
`ttest`	T test
`anova`	Analysis of variance
`cor`	Correlation Coefficient statistic
`lr`	Simple Linear Regression statistic
`chisq.test`	Chi-Square (Pearson's method) Test statistic

4.2 Examples

4.2.1 Definition of dataset and parameters

We used as an example a dataset taken from `http://www.cookbook-r.com/Statistical_analysis/ANOVA/`. This dataset contains measurements of the dependent variable (DV) 'value' for 30 participants, as well as information about the independent variables (IV) of *Sex*(Male or Female) and *Age*(Young or Old). The within participants DV 'value' has two levels of *time*(Before and After) (IV).

In R this dataset would be defined as a table containing all the aforementioned variables. For use in RASSP, the data for each participant are retained in the form of a JSON message.

```
{
  "age": [ "old" ],
  "sex": [ "F" ],
  "value_time_before": [9.5],
  "value_time_after": [7.1],
  "value_time_avg": [8.3],
   ...
}
```

[2] Cf. `http://www.statmethods.net/stats/ttest.html`

The arguments in RASSP methods are defined as lists of dependent variables or independent variables as follows:

```
# GroupStat structures define groups based on IVs, e.g.:
group1 <- GroupStat(list(c("sex","=","F")))
group2 <- GroupStat(list(c("sex","=","M"), c("age","=","old")))
groupX <- GroupStat(list(c("time", "=", "before")))

#Parameters structures define lists on DVs and specify IV conditions/levels
methodParameters <- Parameters(list("value"), list(group1, group2))
```

4.2.2 mean(parameters)

Summary The mean value of the list described by parameters.

Arguments `Parameters` structure

Results A `Double` number that represents the mean result.

Example 1. Calculate the mean of the DV `value_time_before`, for all participants.

```
# Define the DV through the arguments of the Parameters
mParam <- Parameters(list("value_time_before"), NULL)
#Execute the secure statistic
mean(mParam)
```

Example 2. Calculate the mean of the DV `value_time_before` only for the male population (`sex=M`).

```
#Define IV condition
Group <- GroupStat(list(c("sex","=","M")))
#Define DV list for Group condition
meanParam <- Parameters(list("value_time_before"), list(Group))
#Execute the secure statistic
mean(meanParam)
```

4.2.3 var(parameters)

Summary The variance of the list described by parameters.

Arguments `Parameters` structure

Results A `Double` number that represents the variance result.

Example 3. Calculate the variance of the variable `value_time_before` only for the male population (`sex=M`).

```
#Define IV condition
Group <- GroupStat(list( c("sex","=","M")))
#Define DV list for Group condition
varParam <- Parameters(list("value_time_before"), list(Group))
#Execute the secure statistic
var(varParam)
```

4.2.4 stdev(parameters)

Summary The standard deviation value of the list described by parameters.

Arguments `Parameters` structure

Results A `Double` number that represents the standard deviation result.

Example 4. Calculate the standard deviation of the variable `value_time_before` only for the female population (`sex=F`).

```
#Define IV condition
Group <- GroupStat(list(c("sex","=","F")))
#Define DV list for Group condition
stdParam <- Parameters(list("value_time_before"), list(Group))
#Execute the secure statistic
stdev(stdParam)
```

4.2.5 normality(parameters, conf.level)

Summary Checks whether a sample, that is described through params, follows a normal distribution (D Agostino-Pearson omnibus method).

Arguments

- `Parameters` structure;
- `conf.level` (default 0.95), which specifies the confidence level for the statistic.

Results Returns test statistic for skewness `Zg1`, for kurtosis `Zg2`, and the D Agostino-Pearson omnibus `pvalue` `https://brownmath.com/stat/shape.htm`.

Example 5. Check whether the DV `value_time_after` follows a normal distribution for the female population(`sex=F`).

```
#Define IV condition
Group <- GroupStat(list(c("sex", "=", "F")))
#Define DV list for Group condition
normParam <- Parameters(list("value_time_after"),list(Group))
#Execute the secure statistic
normality(normParam)
```

4.2.6 plotnorm(parameters)

Summary Plots the Gaussian distribution of a sample that is described by parameters

Arguments `Parameters` structure

Results

- The plot of the normal distribution
- The mean and standard deviation of the distribution

Example 6. Plot the Gaussian distribution of the variable `value_time_after` for the female population(`sex=F`).

```
#Define IV condition
Group <- GroupStat( list( c("sex", "=", "F") ))
#Define the list for Group condition
pltParam <- Parameters( list("value_time_after") , list(Group))
#Execute the secure statistic
plotnorm(pltParam)
```

4.2.7 ttest(parameters, alternative, mu, varEq, conf.level)

Summary The T-test statistic and the statistic's attributes

Arguments
- `Parameters` structure for two sample groups
- `alternative`: A character string specifying the alternative hypothesis, must be one of `two.sided` (default), `greater` or `less`. You can specify just the initial letter.
- `mu`: A double number indicating the difference in means and declares the Ho hypothesis.
- `varEq`: A logical variable indicating whether to treat the two variances as being equal. If `TRUE` then the Student T-test is used otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.
- `conf.level`: The confidence level of the interval.

Results The output of the T-test execution gives the `t-test value`, the `p-value` and the degrees of freedom of the t-test`df`.

Example 7. Test the hypothesis that that the DV `value_time_before` is significantly different between males and females.

```
#Define IV condition
Gr1 <- GroupStat(list( c("sex", "=", "M")))
Gr2 <- GroupStat(list( c("sex", "=", "F")))
#Define the DV list for Gr1 and Gr2 groups.
tParam <- Parameters(list("value_time_before"), list(Gr1, Gr2))
#Execute the secure statistic
ttest(tParam, varEq = FALSE)
```

To demonstrate the differences and similarities between RASSP interface and traditional R methods the following listing accomplishes the same effect by loading the data from a single file and calling the `mean` function from the standard R library.

```
Dataset <- read.table(header=TRUE, 'data.csv')
t.test(Dataset["before"][sex=="F"],Dataset["before"][sex=="M"])
```

4.2.8 anova(parameters, method)

Summary The ANOVA statistic described by params. Regarding the method the following types can be computed:
- One-Way Between participants ANOVA
- One-Way Within / Repeated ANOVA
- Mixed ANOVA

Arguments
- A `Parameters` structure. The One-Way Between ANOVA as well as One-Way Repeated ANOVA needs a single DV and at least two sample groups/conditions. On the other hand, Mixed ANOVA needs a single DV and exactly two groups (the one describing the sample groups and the other the conditions).
- `method`: The method of anova that will be applied; `'b'` is for one-way between subjects, `'r'` is for repeated measures (within participants and `'m'` for mixed anova

Results A set of resulted components (summary table) after the ANOVA execution, such as `df`, `ssq`, `ms`, `Fvalue`, `pvalue` etc.

Example 8 (Between participants)**.** Check the effect of `sex=M/F` on the DV `value_time_before`. Firstly we should define the `Parameters`:

```
#Define the two groups
Group1 <- GroupStat(list(c("sex", "=", "M")))
Group2 <- GroupStat(list(c("sex", "=", "F")))
#Define the DV list for Group1 and Group2 groups.
btwParam <- Parameters(list("value_time_before"), list(Group1, Group2))
#Execute the secure statistic
anova(btwParam, "b")
```

The following listing achieves the same using standard non privacy preserving ANOVA in R.

```
Dataset <- read.table(header=TRUE, 'data.csv')
aov(before ~ sex, data=Dataset)
```

Example 9 (Within Participants)**.** Check the effect of `time=before/after` on the DV `value`.

```
#Define the two levels of time
Level1 <- GroupStat(list(c("time", "=", "before")))
Level2 <- GroupStat(list(c("time", "=", "after")))
#Define the DV list for Level1  and Level2 conditions.
repParam <- Parameters(list("value") , list(Level1, Level2))
#Execute the secure statistic
anova(repParam, "r")
```

The following listing achieves the same using standard non privacy preserving ANOVA in R.

```
Dataset <- read.table(header=TRUE, 'data.csv')
aov(value ~ time + Error(subject/time), data=DataSet)
```

Example 10 (Mixed)**.** Run a 2x2 ANOVA, checking the effect of the within participants factor of `time=before/after` and the between participants factor of `sex=M/F` on the DV `value`.

```
#Define Between and Within participant factors.
BPfactor <- GroupStat(list( c("sex", "=", "M"), c("sex", "=", "F")))
WPfactor <- GroupStat(list( c("time", "=", "before"), c("time", "=", "after
    ")))
#Define the DV list for between and within participant factors.
mixParam <- Parameters(list("value"), list(BPfactor, WPfactor))
#Execute the secure statistic
anova(mixParam, "m")
```

The following listing achieves the same using standard non privacy preserving ANOVA in R.

```
Dataset <- read.table(header=TRUE, 'data.csv')
aov(value ~ age*time + Error(subject/time), data=Dataset)
```

4.2.9 cor(parameters)

Summary Correlation Coefficient statistic.

Arguments A `Parameters` structure defining a dependent and an independent variable.

Results A `Double` number between the range $[-1.0, 1.0]$, which denotes the linearly relation of the two variables.

Example 11. Calculate the correlation coefficient between `value_time_before` and `value_time_after`

```
#Define variables to be correlated.
corParam <- Parameters(list("value_time_before", "value_time_after"), NULL)
#Execute the secure statistic
cor(corParam)
```

The traditional R code for calculating the correlation coefficient will look like the following.

```
Dataset <- read.table(header=TRUE, 'data.csv')
cor(Dataset["before"],Dataset["after"])
```

4.2.10 lr(parameters)

Summary Simple Linear Regression statistic, described by parameters. The statistic fits a straight line: $y = \beta_1 + \beta_0 * x$.

Arguments A `Parameters` structure defining one independent variable (x) (1st argument) and one dependent variable(y) (2nd argument).

Results

- The coefficients of the linear regression.
- The plot of the fitted model.
- Parameters of the statistics computation, such as `xbar`, `ybar`.
- Other output of related to model fitting, such as `Sx`, `Sy`, `R2`.

Example 12. linear regression model of `value_time_after` depending on `value_time_before`.

```
#Define Parameters structure.
lParam <- Parameters(list("value_time_before", "value_time_after"), NULL)
#Execute the secure statistic
lr(lParam)
```

The traditional R code for calculating the correlation coefficient will look like the following.

```
Dataset <- read.table(header=TRUE, 'data.csv')
lm(formula = before ~ $after, data=Dataset)
```

4.2.11 chisq.test(params)

Summary The Chi-Square (Pearson's method) Test statistic, described by parameters.

Arguments A `Parameters` variable, which needs two sample groups, each one defining the categories of each attribute.

Results Analytics of the Pearson's chi-square, such as `df`, `X2`, `pvalue`.

Example 13. We want to apply a chi-square test between the following two variables: `sex` (with categories `M` and `F`) and `age` (with categories `old`, `young`).

```
#Define Parameters structure. Notice that there is no DV in this occasion.

chiGr1 <- GroupStat(list(c("sex", "=", "M"), c("sex", "=", "F")))
chiGr2 <- GroupStat(list(c("age", "=", "old"), c("age", "=", "young")))
chiParam <- Parameters(list(), list(chiGr1,chiGr2))
#Execute the secure statistic
```

```
chisq.test(chiParam)
```

The traditional R code for calculating the correlation coefficient will look like the following.

```
Dataset <- read.table(header=TRUE, 'data.csv')
chisq.test(Dataset["sex"],Dataset["age"])
```

All the secure statistics are tested with the use of the given dataset, resulting in accurate values.

5 System and Network Security

This chapter presents the general requirements and guidelines that should be taken into account for the project.

5.1 Data protection and Privacy

5.1.1 Data Protection Directive (Directive 95/46/EC)

The Data Protection Directive [D_95/45/EC] (officially Directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data) is a European Union directive which regulates the processing of personal data within the European Union. It is an important component of EU privacy and human rights law.

Personal data are defined as:

> Any information relating to an identified or identifiable natural person ('data subject'); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity. (Art. 2a)

This definition is meant to be very broad. Data are 'personal data' when someone is able to link the information to a person, even if the person holding the data cannot make this link. Some examples of 'personal data' are: address, credit card number, bank statements, criminal record, etc.

The notion processing means:

> Any operation or set of operations which is performed upon personal data, whether or not by automatic means, such as collection, recording, organization, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, blocking, erasure or destruction; (Art. 2b)

The responsibility for compliance rests on the shoulders of the 'controller', meaning the natural or artificial person, public authority, agency or any other body which alone or jointly with others determines the purposes and means of the processing of personal data (Art. 2d).

The data protection rules are applicable not only when the controller is established within the EU, but whenever the controller uses equipment situated within the EU in order to process data (Art. 4). Controllers from outside the EU, processing data in the EU, will have to follow data protection regulation. In principle, any online business trading with EU citizens would process some personal data and would be using equipment in the EU to process the data (i.e. the customer's computer). As a consequence, the website operator would have to comply with the European data protection rules. The directive was written before the breakthrough of the Internet, and to date there is little jurisprudence on this subject.

5.1.2 Directive on Privacy and Electronic Communications (2002/58/EC)

Directive 2002/58/EC on Privacy and Electronic Communications, otherwise known as E-Privacy Directive, is an EU directive on data protection and privacy in the digital age. It presents a continuation of earlier efforts, most directly the Data Protection Directive. It deals with the regulation of a number of important issues such as confidentiality of information, treatment of traffic data, spam and cookies. This Directive has been amended by Directive 2009/136, which introduces several changes, especially in what concerns cookies, that are now subject to prior consent.

The first general obligation in the Directive is to provide security of services. The addressees are providers of electronic communications services. This obligation also includes the duty to inform the subscribers whenever there is a particular risk, such as a virus or other malware attack.

The second general obligation is for the confidentiality of information to be maintained. The addressees are Member States, who should prohibit listening, tapping, storage or other kinds of interception or surveillance of communication and 'related traffic', unless the users have given their consent or conditions of Article 15(1) have been fulfilled.

Data retention and other issues The directive obliges the providers of services to erase or anonymize the traffic data processed when no longer needed, unless the conditions from Article 15 have been fulfilled. Retention is allowed for billing purposes but only as long as the statute of limitations allows the payment to be lawfully pursued. Data may be retained upon a user's consent for marketing and value-added services. For both previous uses, the data subject must be informed why and for how long the data is being processed.

Subscribers have the right to non-itemised billing. Likewise, the users must be able to opt out of calling-line identification.

Where data relating to location of users or other traffic can be processed, Article 9 provides that this will only be permitted if such data is anonymized, where users have given consent, or for provision of value-added services. Like in the previous case, users must be informed beforehand of the character of information collected and have the option to opt out.

Spam Article 13 prohibits the use of email addresses for marketing purposes. The Directive establishes the opt-in regime, where unsolicited emails may be sent only with prior agreement of the recipient. A natural or legal person who initially collects address data in the context of the sale of a product or service, has the right to use it for commercial purposes provided the customers have a prior opportunity to reject such communication, either where it was initially collected or subsequently. Member States have the obligation to ensure that unsolicited communication will be prohibited, except in circumstances given in Article 13.

Two categories of emails (or communication in general) will also be excluded from the scope of the prohibition. The first is the exception for existing customer relationships and the second for marketing of similar products and services. The sending of unsolicited text messages, either in the form of SMS messages, push mail messages or any similar format designed for consumer portable devices (mobile phones, PDAs) also falls under the prohibition of Article 13.

Cookies The Directive provision applicable to cookies is Article 5(3). Recital 25 of the Preamble recognizes the importance and usefulness of cookies for the functioning of modern Internet and directly relates Article 5(3) to them but Recital 24 also warns of the danger that such instruments may present to privacy. The change in the law does not affect all types of cookies. For cookies that are deemed to be 'strictly necessary for the delivery of a service requested by the user' the consent of the user is not needed. An example of a 'strictly necessary' cookie is when you press 'add to basket' or 'continue to checkout' when shopping online. It is important that the browser remembers information from a previous web page in order to complete a successful transaction.

The article is technology neutral, not naming any specific technological means which may be used to store data, but applies to any information that a website causes stored in a user's browser. This reflects the EU legislator's desire to leave the regime of the directive open to future technological developments.

The addressees of the obligation are Member States, who must ensure that the use of electronic communications networks to store information in a visitor's browser is only allowed if the user is provided with 'clear and comprehensive information', in accordance with Data Protection Directive, about the purposes of the storage of, or access to, that information; and has given his or her consent.

The regime so set-up can be described as opt-in, effectively meaning that the consumer must give his or her consent before cookies or any other form of data is stored in their browser. The UK Regulations allow for consent to be signified by future browser settings, which have yet to be introduced but which must be capable of presenting enough information so that a user can give their informed consent and

indicating to a target website that consent has been obtained. Initial consent can be carried over into repeated content requests to a website. The Directive does not give any guidelines as to what may constitute an opt-out, but requires that cookies, other than those 'strictly necessary for the delivery of a service requested by the user' are not to be placed without user consent.

5.1.3 Patients' rights in cross-border healthcare

Directive 2011/24/EU of the European Parliament and of the Council of 9 March 2011 on the application of patients' rights in cross-border healthcare is in force since the 24th April 2011. The Directive has been formally adopted by the Council.

The Directive helps patients who need specialised treatment, for example those who are seeking a diagnosis or treatment for a rare disease. It brings about closer and improved health co-operation between Member States, including the recognition of prescriptions. Health experts across Europe are better able to share best practices on healthcare and establish and maintain standards of excellence.

5.1.4 Guidelines of practice

Information is one of RADIO Project's most important assets. Protection of information assets is necessary to establish and maintain trust between the healthcare institution and its patients, maintain compliance with the law, and protect the reputation of the project. Timely and reliable information is required regarding collected information in order to support healthcare institutions needs and accurate patient diagnosis. A healthcare institution's reputation and integrity can be adversely affected if information becomes known to unauthorized parties, is altered, or is not available when it is needed. Information security is the process with which an organization protects and secures its systems, media, and facilities that process and maintain information vital to its operations.

On a broad scale, the RADIO Consortium members have a primary role in protecting the patients personal information, upon collected and processed from information systems within RADIO project scope. The Guidelines of Practice provide guidance to consortium members during architecture, implementation and service delivery to assess the level of security risks to the RADIO Project.

5.2 RADIO Security Objectives

The RADIO system is appointed to develop a platform to manage and analyse acquired multimodal and advanced technology data from brain and body activities of epileptic patients emphasising privacy and security issues. It is therefore important to analyse the security risks of the system and implications of compromise of the system or data. A secure system will protect data confidentiality and integrity as well as protect its availability.

Information security enables RADIO to meet its objectives by implementing multimodal data collection with due consideration of information technology (IT) capacity and constraints. RADIO members meet this goal by striving to accomplish the following security related objectives.

5.2.1 Availability

The ongoing availability of systems addresses the processes, policies, and controls used to ensure authorized users have prompt access to information. This objective protects against intentional or accidental attempts to deny legitimate users access to information or systems.

Certain RADIO scenarios have several time critical functions. For example in emergency life threatening seizure detection, a situation where every second is important, doctors should have feasible access to the data they need. Also, appropriately appointed caregivers should be alarmed in a timely way.

Integrity of Data or Systems—System and data integrity relate to the processes, policies, and controls used to ensure information has not been altered in an unauthorized manner and that systems are free from unauthorized manipulation that will compromise accuracy, completeness, and reliability.

RADIO features such patient risk assessment and decision support for the professionals, rely on accurate

information. Corrupted data may cause unexpected behavior on the system. With fabricated data, a malicious party may try to affect the behavior of the RADIO system. Data can be corrupted during transmission or while stored. Attempts may be made to enter fabricated data to the system through normal RADIO input devices e.g. touch screen or via open communication channel.

5.2.2 Confidentiality of Data or Systems

Confidentiality covers the processes, policies, and controls employed to protect information of external parties and RADIO against unauthorized access or use. The RADIO applications will access and use information about the patient that is sensitive e.g. health status. Actual or perceived risk of such information being available to unauthorized personnel will affect negatively the acceptability of the RADIO solution.

Protection of user privacy is thus important. Confidentiality of the RADIO data can be compromised at data storage, during data transmission or gaining access to one of the devices through which RADIO provides data output e.g. health professional's computer.

5.2.3 Accountability

Clear accountability involves the processes, policies, and controls necessary to trace actions to their source. Accountability directly supports non-repudiation, deterrence, intrusion prevention, security monitoring, recovery, and legal admissibility of records.

5.2.4 Assurance

Assurance addresses the processes, policies, and controls used to develop confidence that technical and operational security measures work as intended. Assurance levels are part of the system design include availability, integrity, confidentiality, and accountability. Assurance highlights the notion that secure systems provide the intended functionality while preventing undesired actions.

5.3 Security Control Implementation

In this section critical concepts regarding the implementation of security control will be presented from a system wide perspective.

5.3.1 Access Control

The goal of access control is to allow access by authorized individuals and devices and to disallow access to all others. Authorized individuals as part of RADIO are considered the consortium members being part of the project. Access should be authorized and provided only to individuals whose identity is established, and their activities should be limited to the minimum required for project purposes. An effective control mechanism includes numerous controls to safeguard and limits access to key information system assets at all layers in the network stack. This section addresses logical and administrative controls, including access rights administration for individuals and network access issues.

5.3.2 Access Rights Administration

System devices, programs, and data are system resources. Each system resource may need to be accessed by individuals (users) in order for work to be performed. Access beyond the minimum required for work to be performed exposes the project's systems and information to a loss of confidentiality, integrity, and availability. Accordingly, the goal of access rights administration is to identify and restrict access to any particular system resource to the minimum required for work to be performed.

Formal access rights administration for users consists of four processes:

1. An enrollment process to add new users to the system;
2. An authorization process to add, delete, or modify authorized user access to operating systems, applications, directories, files, and specific types of information;
3. An authentication process to identify the user during subsequent activities; and

4. A monitoring process to oversee and manage the access rights granted to each user on the system.

The enrollment process establishes the user's identity and anticipated business needs for information and systems. During enrollment and thereafter, an authorization process determines user access rights. In certain circumstances the assignment of access rights may be performed only after the manager responsible for each accessed resource approves the assignment and documents the approval. In other circumstances, the assignment of rights may be established by the user's role or group membership, and managed by pre-established authorizations for that group. External parties, on the other hand, may be granted access based on their relationship with the project.

Authorization for privileged access should be tightly controlled. Privileged access refers to the ability to override system or application controls. Good practices for controlling privileged access include

- Identifying each privilege associated with each system component,
- Implementing a process to allocate privileges and allocating those privileges either on a need-to-use or an event-by-event basis,
- Documenting the granting and administrative limits on privileges,
- Finding alternate ways of achieving the business objectives,
- Assigning privileges to a unique user ID apart from the one used for normal business use,
- Logging and auditing the use of privileged access,
- Reviewing privileged access rights at appropriate intervals and regularly reviewing privilege access allocations and
- Prohibiting shared privileged access by multiple users.

The access rights process programs the system to allow the users only the access rights they were granted. Since access rights do not automatically expire or update, periodic updating and review of access rights on the system is necessary. Updating should occur when an individual's business needs for system use changes. Many job changes can result in an expansion or reduction of access rights. Job events that would trigger a removal of access rights include transfers, resignations, and terminations. When these job events occur, project stakeholders should take particular care to promptly remove the access rights for users who have remote access privileges, access to patient information, and perform administration functions for the project's systems.

Because updating may not always be accurate, periodic review of user accounts is a good control to test whether the access right removal processes are functioning and whether users exist who should have their rights rescinded or reduced.

Access rights to new software and hardware present a unique problem. Typically, hardware and software are shipped with default users, with at least one default user having full access rights. Easily obtainable lists of popular software exist that identify the default users and passwords, enabling anyone with access to the system to obtain the defaultuser's access. Default user accounts should either be disabled, or the authentication to the account should be changed. Additionally, access to these default accounts should be monitored more closely than other accounts.

Sometimes software installs with a default account that allows anonymous access. Anonymous access is appropriate, for instance, where the general public accesses an informational Web server. Systems that allow access to or store sensitive information, including customer information, should be protected against anonymous access.

The access rights process also constrains user activities through an acceptable-use policy (AUP). Users who can access internal systems typically are required to agree to an AUP before using a system. An AUP details the permitted system uses and user activities and the consequences of noncompliance. AUPs can be created for all categories of system users, from internal programmers to external parties. An AUP

is a key control for user awareness and administrative policing of system activities. Examples of AUP elements for internal network and stand-alone users include

- The specific access devices that can be used to access the network;
- Hardware and software changes the user can make to their access device;
- The purpose and scope of network activity;
- Network services that can be used and those that cannot be used;
- Information that is allowable and not allowable for transmission using each allowable service;
- Bans on attempting to break into accounts, crack passwords, or disrupt service;
- Responsibilities for secure operation; and
- Consequences of noncompliance.

External parties may be provided with a Web site disclosure as their AUP. Based on the nature of the Web site, RADIO may require external parties to demonstrate knowledge of and agreement to abide by the terms of the AUP. That evidence can be paper based or electronic.

Authorized users may seek to extend their activities beyond what is allowed in the AUP and unauthorized users may seek to gain access to the system and move within the system. Network security controls provide many of the protections necessary to guard against those threats.

5.3.3 Authentication

Authentication is the verification of identity by a system based on the presentation of unique credentials to that system. The unique credentials are in the form of something the user knows, something the user has, or something the user is. Those forms exist as shared secrets, tokens, or biometrics. More than one form can be used in any authentication process. Authentication that relies on more than one form is called multi-factor authentication and is generally stronger than any single-factor authentication method. Authentication contributes to the confidentiality of data and the accountability of actions performed on the system by verifying the unique identity of the system user.

Authentication over the RADIO delivery channel presents unique challenges. That channel does not benefit from physical security and controlled computing and communications devices like internal local area networks (LANs), and is used by people whose actions cannot be controlled. It should be considered the use of single-factor authentication in that environment, as the only control mechanism, to be inadequate for high-risk transactions involving access to patient information or the movement of healthcare information to other parties. Authentication does not provide assurance that the initial identification of a system user is correct.

5.3.4 Shared Secret Systems

Shared secret systems uniquely identify the user by matching knowledge on the system to knowledge that only the system and user are expected to share. Examples are passwords, pass phrases, or current transaction knowledge. A password is one string of characters (e.g., ‘t0Ol@Tyme’). A pass phrase is typically a string of words or characters (e.g., ‘My car is a shepherd’) that the system may shorten to a smaller password by means of an algorithm. Current transaction knowledge for a financial institution for example could be the account balance on the last statement mailed to the user/customer. The strength of shared secret systems is related to the lack of disclosure of and about the secret, the difficulty in guessing or discovering the secret, and the length of time that the secret exists before it is changed.

A strong shared secret system only involves the user and the system in the generation of the shared secret. In the case of passwords and pass phrases, the user should select them without any assistance from any other user, such as the help desk. One exception is in the creation of new accounts, where a temporary shared secret could be given to the user for the first log-in, after which the system requires the user to create a different password. Controls should prevent any user from re-using shared secrets

that may have been compromised or were recently used by them.

5.3.5 Token Systems

Token systems typically authenticate the token and assume that the user who was issued the token is the one requesting access. One example is a token that generates dynamic passwords after a set number of seconds. When prompted for a password, the user enters the password generated by the token. The token's password-generating system is identical and synchronized to that in the system, allowing the system to recognize the password as valid. The strength of this system of authentication rests in the frequent changing of the password and the inability of an attacker to guess the seed and password at any point in time.

Another example of a token system uses a challenge/response mechanism. In this case, the user identifies him/herself to the system, and the system returns a code to enter into the password-generating token. The token and the system use identical logic and initial starting points to separately calculate a new password. The user enters that password into the system. If the system's calculated password matches that entered by the user, the user is authenticated. The strengths of this system are the frequency of password change and the difficulty in guessing the challenge, seed, and password.

5.3.6 Public Key Infrastructure

Public key infrastructure (PKI), if properly implemented and maintained, can provide a strong means of authentication. By combining a variety of hardware components, system software, policies, practices, and standards, PKI can provide for authentication, data integrity, defenses against customer repudiation, and confidentiality. The system is based on public key cryptography in which each user has a key pair—a unique electronic value called a public key and a mathematically related private key. The public key is made available to those who need to verify the user's identity.

The private key is stored on the user's computer or a separate device such as a smart card. When the key pair is created with strong encryption algorithms and input variables, the probability of deriving the private key from the public key is extremely remote. The private key must be stored in encrypted text and protected with a password or PIN to avoid compromise or disclosure. The private key is used to create an electronic identifier called a digital signature that uniquely identifies the holder of the private key and can only be authenticated with the corresponding public key.

5.3.7 Device Authentication

Device authentication typically takes place either as a supplement to the authentication of individuals or when assurance is needed that the device is authorized to be on the network. Devices are authenticated through either shared secrets, such as pre-shared keys, or the use of PKI. Authentication can take place at the network level and above. At the network level, IPv6 has the built-in ability to authenticate each device. Device authentication is subject to the same shared-secret and PKI weaknesses as user authentication, and is subject to similar offsetting controls. Additionally, similar to user authentication, if the device is under the attacker's control or if the authentication mechanism has been compromised, communications from the device should not be trusted.

5.3.8 Examples of Common Authentication Weaknesses, Attacks, and Offsetting Controls

All authentication methodologies display weaknesses. Those weaknesses are of both a technical and a nontechnical nature. Many of the weaknesses are common to all mechanisms. Examples of common weaknesses include social engineering, client attacks, replay attacks, man-in-the-middle attacks, and hijacking. Frequently, the authentication data is encrypted; however, dictionary attacks make decryption of even a few passwords in a large group a trivial task. A dictionary attack uses a list of likely authenticators, such as passwords, runs the likely authenticators through the encryption algorithm, and compares the result to the stolen, encrypted authenticators. Any matches are easily traceable to the pre-encrypted authenticator.

Dictionary and brute force attacks are viable due to the speeds with which comparisons are made. As

microprocessors increase in speed, and technology advances to ease the linking of processors across networks, those attacks will be even more effective. Because those attacks are effective, great care should be taken in securing authentication databases. Upon use of one-way hashes the insertion of secret bits should be considered (also known as 'salt') to increase the difficulty of decrypting the hash. Salt has the effect of increasing the number of potential authenticators that attackers must check for validity, thereby making the attacks more time consuming and creating more opportunity for system administrators to identify and react to the attack.

Social engineering involves an attacker obtaining authenticators by simply asking for them. For instance, the attacker may masquerade as a legitimate user who needs a password reset or as a contractor who must have immediate access to correct a system. An attack may also try all possible combinations of the allowed character set. By using persuasion, being aggressive, or using other interpersonal skills, the attackers encourage a legitimate user or other authorized person to give them authentication credentials. Controls against these attacks involve strong identification policies and user training.

Client attacks are an area of vulnerability common to all authentication mechanisms. Passwords, for instance, can be captured by hardware- or software-based keystroke capture mechanisms. PKI private keys could be captured or reverse-engineered from their tokens. Protection against these attacks primarily consists of physically securing the external party systems, and, if a shared secret is used, changing the secret on a frequency commensurate with risk.

Replay attacks occur when an attacker eavesdrops and records the authentication as it is communicated between an external party and the ARMOM system and then later uses that recording to establish a new session with the system and masquerade as the true user. Protections against replay attacks include changing cryptographic keys for each session, using dynamic passwords, expiring sessions through the use of time stamps, expiring PKI certificates based on dates or number of uses, and implementing liveness tests for biometric systems.

Man-in-the-middle attacks place the attacker's computer in the communication line between the server and the client. The attacker's machine can monitor and change communications. Controls against man-in-the-middle attacks include prevention through host and client hardening, appropriate hardening and monitoring of domain name service (DNS) servers and other network infrastructure, authentication of the device communicating with the server, and the use of PKI.

Hijacking is an attacker's use of an authenticated user's session to communicate with system components. Controls against hijacking include encryption of the user's session and the use of encrypted cookies or other devices to authenticate each communication between the client and the server.

5.3.9 Encryption

Encryption is used to secure communications and data storage, particularly authentication credentials and the transmission of sensitive information. It can be used throughout technological environment, including the operating systems, middleware, applications, file systems, and communications protocols. Encryption can be used as a preventive control, a detective control, or both. As a prevention control, encryption acts to protect data from disclosure to unauthorized parties. As a detective control, encryption is used to allow discovery of unauthorized changes to data and to assign responsibility for data among authorized parties. When prevention and detection are joined, encryption is a key control in ensuring confidentiality, data integrity, and accountability.

Properly used, encryption can strengthen the security of a project's systems. Encryption also has the potential, however, to weaken other security aspects. For instance, encrypted data drastically lessens the effectiveness of any security mechanism that relies on inspections of the data, such as anti-virus scanning and intrusion detection systems. When encrypted communications are used, networks may have to be reconfigured to allow for adequate detection of malicious code and system intrusions.

Although necessary, encryption carries the risk of making data unavailable should anything go wrong with data handling, key management, or the actual encryption. For example, a loss of encryption keys or

other failures in the encryption process can deny the member's access to the encrypted data. The products used and administrative controls should contain robust and effective controls to ensure reliability.

An encryption strength should be employed sufficient to protect information from disclosure until such time as the information's disclosure poses no material threat. For instance, authenticators should be encrypted at strength sufficient to detect and react to an authenticator theft before the attacker can decrypt the stolen authenticators.

Decisions regarding what data to encrypt and at what points to encrypt the data are typically based on the risk of disclosure and the costs and risks of encryption. The costs include potentially significant overhead costs on hosts and networks. Generally speaking, authenticators are encrypted whether on public networks or on project's network. Sensitive information is also encrypted when passing over a public network and also may be encrypted within RADIO. Encryption cannot guarantee data security. Even if encryption is properly implemented, for example, a security breach at one of the endpoints of the communication can be used to steal the data or allow an intruder to masquerade as a legitimate system user.

5.3.10 Encryption Types

Three types of encryption exist: the cryptographic hash, symmetric encryption and asymmetric encryption. A cryptographic hash reduces a variable-length input to a fixed-length output. The fixedlength output is a unique cryptographic representation of the input. Hashes are used to verify file and message integrity. For instance, if hashes are obtained from key operating system binaries when the system is first installed, the hashes can be compared to subsequently obtained hashes to determine if any binaries were changed. Hashes are also used to protect passwords from disclosure. A hash, by definition, is a one-way encryption. An attacker who obtains the password cannot run the hash through an algorithm to decrypt the password. However, the attacker can perform a dictionary attack, feeding all possible password combinations through the algorithm and look for matching hashes, thereby assuming the password. To protect against that attack, 'salt', or additional bits, are added to the password before encryption. The addition of the bits means the attacker must increase the dictionary to include all possible additional bits, thereby increasing the difficulty of the attack.

Symmetric encryption is the use of the same key and algorithm by the creator and reader of a file or message. The creator uses the key and algorithm to encrypt, and the reader uses both to decrypt. Symmetric encryption relies on the secrecy of the key. If the key is captured by an attacker, either when it is exchanged between the communicating parties, or while one of the parties uses or stores the key, the attacker can use the key and the algorithm to decrypt messages or to masquerade as a message creator.

Asymmetric encryption lessens the risk of key exposure by using two mathematically related keys, the private key and the public key. When one key is used to encrypt, only the other key can decrypt. Therefore, only one key (the private key) must be kept secret. The key that is exchanged (the public key) poses no risk if it becomes known. For instance, if individual A has a private key and publishes the public key, individual B can obtain the public key, encrypt a message to individual A, and send it. As long as individual A keeps his private key secure from discovery, only individual A will be able to decrypt the message.

5.3.11 Examples of Encryption Uses

Asymmetric encryption is the basis of public key infrastructure. In theory, PKI allows two parties who do not know each other to authenticate each other and maintain the confidentiality, integrity, and accountability for their messages. PKI rests on both communicating parties having a public and a private key, and keeping their public keys registered with a third party they both trust, called the certificate authority, or CA. The use of and trust in the third party is a key element in the authentication that takes place. For example, assume individual A wants to communicate with individual B. A first hashes the message, and encrypts the hash with A's private key. Then A obtains B's public key from the CA and encrypts the message and the hash with B's public key. Obtaining B's public key from the trusted CA

provides A assurance that the public key really belongs to B and not someone else. Using B's public key ensures that the message will only be able to be read by B. When B receives the message, the process is reversed. B decrypts themessage and hash with B's private key, obtains A's public key from the trusted CA, and decrypts the hash again using A's public key. At that point, B has the plain text of the message and the hash performed by A. To determine whether the message was changed in transit, B must re-perform the hashing of the message and compare the newly computed hash to the one sent by A. If the new hash is the same as the one sent by A, B knows that the message was not changed since the original hash was created (integrity). Since B obtained A's public key from the trusted CA and that key produced a matching hash, B is assured that the message came from A and not someone else (authentication). Various communication protocols use both symmetric and asymmetric encryption.

Transaction layer security (TLS), the successor to Secure Socket Layer (SSL) uses asymmetric encryption for authentication, and symmetric encryption to protect the remainder of the communications session. TLS can be used to secure healthcare applications and other transmissions between the RADIO and external parties. TLS may also be used to secure e-mail, telnet, and FTP sessions.

IPSec is a complex aggregation of protocols that together provide authentication and confidentiality services to individual IP packets. It can be used to create a VPN over the Internet or other untrusted network, or between any two computers on a trusted network. Since IPSec has many configuration options, and can provide authentication and encryption using different protocols, implementations between vendors and products may differ.

SSL and TLS are frequently used to establish encrypted tunnels between a service provider and its users. They are also used to provide a different type of VPN than that provided by IPSec.

Secure Shell (SSH) is frequently used for remote server administration. SSH establishes an encrypted tunnel between a SSH client and a server, as well as authentication services.

Encryption may also be used to protect data in storage. The implementation may encrypt a file, a directory, a volume, or a disk.

5.4 Security Monitoring

Security monitoring focuses on the activities and condition of network traffic and network hosts. Activity monitoring is primarily performed to assess policy compliance, identify non-compliance with the project security goals, and identify intrusions and support an effective intrusion response. Because activity monitoring is typically an operational procedure performed over time, it is capable of providing continual assurance.

Monitoring of condition is typically performed in periodic testing. The assurance provided by condition monitoring can relate to the absence of an intrusion, the compliance with authorized configurations, and the overall resistance to intrusions. Condition monitoring does not provide continual assurance, but relates to the point in time of the test.

Risk drives the degree of monitoring. In general, risk increases with system accessibility and the sensitivity of data and processes. For example, a high-risk system is one that is remotely accessible and allows direct access to diagnosis, personal or sensitive healthcare data. Information-only Web sites that are not connected to any internal institution system or transaction-capable service are lower-risk systems. Information systems that exhibit high risks should be subject to more rigorous monitoring than low-risk systems.

Project's security monitoring should, commensurate with the risk, be able to identify control failures before a security incident occurs, detect an intrusion or other security incident in sufficient time to enable an effective and timely response, and support post-event forensics activities.

5.5 Activity Monitoring

Activity monitoring consists of host and network data gathering, and analysis. Host data is gathered and recorded in logs and includes performance and system events of security significance. Host performance is important to identify anomalous behavior that may indicate an intrusion. Security events are important both for the identification of anomalous behavior and for enforcing accountability. Examples of security events include operating system access, privileged access, creation of privileged accounts, configuration changes, and application access. Privileged access may be subject to keystroke recording. Sensitive applications should have their own logging of significant events. Host activity recording is typically limited by the abilities of the operating system and application. Network data gathering is enabled by sensors that typically are placed at control points within the network. For example, a sensor could record traffic that is allowed through a firewall into the perimeter networking (often reffered to as demilitarized zone or DMZ), and another sensor could record traffic between the DMZ and the internal network. As another example, a sensor could be placed on a switch that controls a subnet on the internal network and record all activity into and out of the subnet.

Network data gathering is governed by the nature of network traffic. The activity recorded can range from parts of headers to full packet content. Packet header information supports traffic analysis and provides such details as the endpoints, length, and nature of network communication. Packet header recording is useful even when packet contents are encrypted. Full packet content provides the exact communications traversing the network in addition to supporting traffic analysis. Full packet content recording allows for a more complete analysis, but entails additional collection, storage, and retrieval costs.

Many types of network sensors exist. Sensors built into some popular routers record activity from packet headers. Host-based sniffer software can be used on a device that does not have an IP address. Some sensors are honeypots, or hosts configured to respond to network communications similar to other hosts, but exist only for the purpose of capturing communications. Other sensors contain logic that performs part of the analysis task, alerting on the similarity between observed traffic and preconfigured rules or patterns.

5.5.1 Log Transmission, Normalization, Storage and Protection

Network and host activities typically are recorded on the host and sent across the network to a central logging facility. The data that arrives at the logging facility is in the format of the software that recorded the activity. The logging facility may process the logging data into a common format. That process is called normalization. Normalized data frequently enables timely and effective log analysis.

Log files are critical to the successful investigation and prosecution of security incidents and can potentially contain sensitive information. Intruders will often attempt to conceal any unauthorized access by editing or deleting log files. Therefore, RADIO should strictly control and monitor access to log files whether on the host or in a centralized logging facility. Some considerations for securing the integrity of log files include

- Encrypting log files that contain sensitive data or that are transmitting over the network;
- Ensuring adequate storage capacity to avoid gaps in data gathering;
- Securing back-up and disposal of log files;
- Logging the data to a separate, isolated computer;
- Logging the data to write-only media like a write-once/read-many (WORM) disk or drive; and
- Setting logging parameters to disallow any modification to previously written data.

6 Conclusion

In this report we described the overall architecture of the RADIO ecosystem and established the interconnections of the various components, entities and sites.

In the proposed architecture, special care has been taken regarding data protection requirements. Sensitive data is only accessible by authenticated and authorized personel. Transmission of raw data points is avoided, while trasmission of derived abstract data is encrypted when leaving the boundaries of a RADIO Home. Moreover, privacy preservation in data analysis is enabled from the architecture by enforcing the data to remain local and allow only certain aggregation to be performed.

REFERENCES

[1] Sameer Ajmani, Robert Morris, and Barbara Liskov. A trusted third-party computation service. Technical report, MIT-LCS-TR-847, MIT, 2001.

[2] Josh Cohen Benaloh. Secret sharing homomorphisms: Keeping shares of a secret secret. In *Advances in Cryptology: Proceedings of CRYPTO '86*, volume 263 of *LNCS*, pages 251–260. Springer, 1986.

[3] Cheng-tao Chu, Sang Kyun Kim, Yi-An Lin, YuanYuan Yu, Gary Bradski, Andrew Y Ng, and Kunle Olukotun. Map-reduce for machine learning on multicore. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19: Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS 2007), Vancouver, BC, Canada, 3–5 December 2007*, pages 281–288. MIT Press, 2007.

[4] Chris Clifton, Murat Kantarcioglu, and Jaideep Vaidya. Defining privacy for data mining. In *Proceedings of the National Science Foundation Workshop on Next Generation Data Mining, Baltimore, USA, 1-3 November 2002*, 2002.

[5] B Hanmanthu, B Raghu Ram, and P Niranjan. Third party privacy preserving protocol for perturbation based classification of vertically fragmented data bases. *arXiv preprint arXiv:1304.6575*, 2013.

[6] Eric Horvitz and Deirdre Mulligan. Data, privacy, and the greater good. *Science Magazine*, 2015.

[7] Alan F Karr, Xiaodong Lin, Ashish P Sanil, and Jerome P Reiter. Secure regression on distributed databases. *Journal of Computational and Graphical Statistics*, 14(2):263–279, 2005.

[8] Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

[9] Florian Kerschbaum. Privacy-preserving computation. In *Privacy Technologies and Policy: Revised Selected Papers from the First Annual Privacy Forum (APF 2012), Limassol, Cyprus, 10-11 October 2012*, pages 41–54. Springer, 2012.

[10] Lea Kissner and Dawn Song. Privacy-preserving set operations. In *Advances in Cryptology: Proceedings of the 25th Annual International Cryptology Conference (CRYPTO 2005), Santa Barbara, California, USA, 14–18 August 2005*, volume 3621 of *LNCS*, pages 241–257. Springer, 2005.

[11] Frank D McSherry. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data (SIGMOD 2009)*, pages 19–30. ACM, 2009.

[12] Andres D Molina, Mastooreh Salajegheh, and Kevin Fu. Hiccups: health information collaborative collection using privacy and security. In *Proceedings of the first ACM workshop on Security and privacy in medical and home-care systems (SPIMACS 2009)*, pages 21–30. ACM, 2009.

[13] Paul Ohm. Broken promises of privacy: Responding to the surprising failure of anonymization. *UCLA Law Review*, 57, 2010.

[14] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT '99), Prague, Czech Republic, May 2-6, 1999*, volume 1592 of *LNCS*, pages 223–238. Springer, 1999.

[15] Rashid Sheikh, Beerendra Kumar, and Durgesh Kumar Mishra. Privacy preserving k secure sum protocol. *arXiv preprint arXiv:0912.0956*, 2009.

[16] Elaine Shi, TH Hubert Chan, Eleanor Rieffel, Richard Chow, and Dawn Song. Privacy-preserving aggregation of time-series data. In *Proceedings of the 18th Annual Network and Distributed System Security Symposium (NDSS 2011)*, volume 2, pages 1–17, 2011.