Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Privacy by Evidence: A Software Development Methodology to Provide Privacy Assurance

Pedro Yóssis Silva Barbosa

Tese submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Metodologia e Técnicas da Computação

Andrey Elísio Monteiro Brito e Hyggo Oliveira de Almeida
(Orientadores)

Campina Grande, Paraíba, Brasil

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

B238p Barbosa, Pedro Yóssis Silva.

Privacy by evidence : a software development methodology to provide privacy assurance / Pedro Yóssis Silva Barbosa. – Campina Grande, 2018.

135 f. : il. color.

Tese (Doutorado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2018.

"Orientação: Prof. Dr. Andrey Elísio Monteiro Brito, Prof. Dr. Hyggo Oliveira de Almeida".

Referências.

1. Ciência da Computação - Privacidade. 2. Ciência da Computação - Segurança. 3. Engenharia de Software. I. Brito, Andrey Elísio Monteiro. II. Almeida, Hyggo Oliveira de. III. Título.

CDU 004.8(043)

Resumo

Em um mundo cada vez mais conectado, uma diversidade de softwares e sensores coletam dados dos ambientes e seus habitantes. Devido à riqueza das informações coletadas, privacidade se torna um requisito importante. Aplicações estão sendo desenvolvidas, e, apesar de existirem princípios e regras para lidar com a privacidade dos indivíduos, faltam metodologias para guiar a integração das diretrizes de privacidade em um processo de desenvolvimento. Metodologias existentes como o *Privacidade desde a Concepção* (do inglês *Privacy by Design – PbD*) ainda são vagas e deixam muitos questionamentos em aberto sobre como aplicá-las na prática. Neste trabalho, nós propomos o conceito de *Privacidade por Evidência* (do inglês *Privacy by Evidence – PbE*), uma metodologia de desenvolvimento de software para prover privacidade. Dada a dificuldade em prover privacidade total, propomos que as documentações das mitigações sejam em formas de evidências de privacidade, objetivando aumentar a confiança no projeto. Para validar a eficácia, *PbE* tem sido utilizada durante o desenvolvimento de quatro aplicações que servem como estudos de caso. O primeiro estudo de caso considerado é uma aplicação de medição inteligente de energia; o segundo considera uma aplicação de contagem e monitoramento de pessoas; o terceiro considera um sistema de monitoramento de eficiência energética; e o quarto considera um sistema de autenticação de dois fatores. Para estas aplicações, os times proveram sete, cinco, cinco e quatro evidências de privacidade, respectivamente, e concluimos que a *PbE* pode ser efetiva em ajudar a entender e a tratar as necessidades de proteção à privacidade quando se está desenvolvendo software.

Abstract

In an increasingly connected world, a diversity of software and sensors collect data from the environment and its inhabitants. Because of the richness of the information collected, privacy becomes an important requirement. Applications are being developed, and, although there are principles and rules regarding the privacy of individuals, there is still a lack of methodologies to guide the integration of privacy guidelines into the development process. Existing methodologies like the *Privacy by Design* (*PbD*) are still vague and leave many open questions on how to apply them in practice. In this work we propose the concept of *Privacy by Evidence* (*PbE*), a software development methodology to provide privacy assurance. Given the difficulty in providing total privacy in many applications, we propose to document the mitigations in form of evidences of privacy, aiming to increase the confidence of the project. To validate its effectiveness, *PbE* has been used during the development of four applications that serve as case studies. The first considered case study is a smart metering application; the second considers a people counting and monitoring application; the third considers an energy efficiency monitoring system; and the fourth considers a two factor authentication system. For these applications, the teams were able to provide seven, five, five, and four evidences of privacy, respectively, and we conclude that *PbE* can be effective in helping to understand and to address the privacy protection needs when developing software.

Agradecimentos

Gostaria de agradecer primeiramente a Deus. Sem ele, eu jamais teria chegado até aqui.

Aos meus pais, Antônio Eduardo e Doralice, grandes incentivadores por todo o apoio, pela educação e estrutura que me proporcionaram e pelos ensinamentos que me guiaram nas escolhas da vida.

Aos meus irmãos André, Clara e Paulo por me influenciarem, encorajarem e torcerem por mim. Por suas boas companhias nos momentos de descontração.

Agradeço à minha amada Larissa, por todo o companheirismo, pela compreensão das minhas ausências e apoio em todos os momentos. Também agradeço à toda a família dela, que também é minha família, por todo o acolhimento e conselhos que recebi.

Meus sinceros agradecimentos aos meus amigos e orientadores Andrey Brito e Hyggo Almeida, pela dedicação, empenho e disponibilidade. Sem suas ideias e conhecimentos, certamente eu não teria chegado até aqui.

Aos amigos e colegas do projeto PrivIoT: Diego Pereira, José Lucas, Fábio Silva, Helder Ronyer, Nazareno Andrade e Lesandro Ponciano; e a Dalton Cezane, do projeto Secure-Cloud. Com muito trabalho e competência, esta equipe foi responsável por grande parte dos resultados apresentados neste trabalho.

Aos meus amigos do ELT, time de competições de cyber-segurança, pelos momentos de descontração e mesmo assim de muita aprendizagem.

Aos professores e funcionários da COPIN e do DSC e finalmente, ao Governo Brasileiro, por meio da CAPES, pelo apoio financeiro fornecido para execução das atividades deste doutorado.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Contextualization

Privacy concerns exist wherever personally identifiable information or other sensitive information is improperly collected, stored, used or disclosed. It is known that conventional authentication and encryption mechanisms are not always enough to solve privacy concerns. In some cases, the entity able to authenticate and to decrypt the data may also play the role of the adversary and, therefore, may infer sensitive information from the received data. Moreover, such entities may:

- Negotiate and export data to third parties;
- Suffer external attacks;
- Suffer internal attacks (*e.g.*, malicious employees).

Previous research shows that internal attacks are the major concerns. In the health industry, the estimated loss of an external attack in 2013 was about $50.000, whereas that of an internal attack was about $2.7 million [40]. Previous research also shows that approximately 40% of people who have had their privacy violated have discontinued their relationships with the companies, and this may be one of the reasons for the existence of evidences that companies do not disclose all violations that occur (reporting 1/3, on average) [40].

Manipulating sensitive data in a irresponsible way can have significant negative legal, financial, and regulatory consequences on the data custodian. Some examples:

- With the intention of providing a real data set to be used by researchers working in the area of Internet search, AOL posted search queries from its clients on the web. *New York Times* reporters were able to re-identify one individual from these queries [90]. The bad publicity from this resulted in the CTO of the company resigning and the researcher who posted the data to lose his job.

- A re-identification attack on a movie ratings data set for a competition organized by Netflix [10] resulted in Netflix canceling a second competition and settling a class action lawsuit. This had financial and reputational impacts on the organization.

Various applications may collect data and provide benefits and useful utilities, however, from examples like many Internet of Things (IoT) applications, it is possible to observe that the concerns about privacy only grow, and being able to state that a company or application is "privacy-friendly" may be a competitive advantage [22; 39]. To avoid the development of harmful applications (not only for users, but also for companies), it is necessary to follow an appropriate methodology to apply techniques that are intrinsic to privacy. Almost every day, the media publishes news with evidences that show that not applying such techniques can have devastating consequences.

When developing a privacy-friendly application (*i.e.*, an application which has privacy as a non-functional requirement), it is important to deal with many concepts. For example, to provide their features, applications may collect and store large amounts of data and in different formats; table records, time series, graphs, text, images, among many others. The format of the data is related to each application. More importantly, these data possess different sensitivity levels for different users. There are also rules to follow and techniques to apply, but, unfortunately, preventive measures have the potential to limit data utility or to make some features unavailable or infeasible. Thus, we face a Privacy vs. Utility trade-off.

1.2 Problem

In this context, we address the following problem: In a software development, how to guide developers to take conscious decisions that will improve the privacy of users, and still allow the desired features to keep working? How to organize a sequence of steps to implement

privacy mechanisms considering a risk analysis process? Almost every day, the media publishes news regarding privacy incidents showing that not applying such mechanisms can have devastating consequences.

Unfortunately, there is a lack of methodologies to guide the development of privacy-friendly applications and to apply in practice the existing privacy guidelines and rules [46]. Companies, developers and users would benefit from the existence of a methodology.

1.3 Our Proposal

In this work, we propose *Privacy by Evidence* (*PbE*): a novel methodology that guides the implementation of privacy concepts in applications. This methodology includes risk assessment, mitigations and tests as crucial activities of the development cycle. Given the general impossibility in providing total privacy (*i.e.*, free of vulnerabilities), we propose to document the mitigations in form of evidences, aiming to increase the confidence. To document the argumentation and evidences, we use the *Goal Structuring Notation* (*GSN*) [45]. We validate the effectiveness of *PbE* through the development of four case studies.

1.4 Summary of Contributions

The first part of our research sought to identify methodologies that help developers and companies to develop privacy-friendly applications. We found several guides for specific activities, such as norms, user perception studies, privacy techniques, and attacks. However, we did not find a generic methodology that considers all these concepts together. Recommendations such as the *Privacy by Design* (*PbD*) [31] are abstract and rarely used in practice [46].

The second part of our research included the creation of a methodology that considers the various relevant privacy concepts found in the literature.

The third part of the research sought to improve and validate the proposed methodology through the development of three case studies with participation of the author. Through a smart energy metering, a people counting, and an energy efficiency application, we found that *PbE* might be an effective way to implement privacy protections in applications. We

identified that privacy protection activities must be in a constant cycle of evolution, since new risks can always be detected. The privacy techniques used in the case studies are in accordance with the definition by Stallings *et al.* [92]: "*Privacy assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed*".

Finally, the fourth part of the research sought to validate the methodology through the development of a case study (a two factor authentication system) without the participation of the author. For this, the Think Aloud method [27; 91] was used to gather information regarding the usage of *PbE* by the development team without influencing what participants say and do, and hence, use such useful information as feedback in order to improve the methodology.

1.5 Organization

This document is organized as follows: in Chapter 2 we provide the background and the privacy concepts. Following, in Chapter 3, we present the *PbE*, our proposed methodology. In Chapter 4 we present the first case study, a smart metering application which uses information regarding energy consumption. In Chapter 5 we present the second case study, a people counting and tracking application. In Chapter 6 we present the third case study, an energy efficiency application which correlates energy and people counting data aiming to detect consumption anomalies (inefficient periods). In Chapter 7 we present the fourth case study, a two factor authentication system. Finally, in Chapter 8, we return the general discussion regarding the contributions and the lessons learned.

Chapter 2

Background

There is currently a great interest in the data collected by modern applications. Data that reveal information regarding public environments, energy consumption, financial balance, health status, and geographical locations are just some examples of contexts. Large volumes of data can now be analyzed quite efficiently to gain new insights for users and service providers. "Big data" is one of the most discussed phrases today, specially in IoT [15; 36; 76; 52], however, such data analysis may also raise privacy concerns.

In addition to the application context, there are many other concepts that should be considered when dealing with privacy preserving. This chapter presents the necessary background presenting such concepts.

2.1 Related Work

This section presents a review of relevant works in the context of methodologies for developing privacy-friendly applications. Because privacy concerns and mitigations are relatively recent, there is not much research related to this methodological challenge. In the analysis of the related works, several limitations and open problems are identified.

To ensure privacy protection, we must take into account principles and guidelines about individuals' privacy, such as the *Health Insurance Portability and Accountability Act* (*HIPAA*) [96], and the *Privacy by Design* (*PbD*) [31]. *PbD* consists in 7 foundational principles to be followed when developing systems that should take into account users privacy:

- **Proactive not Reactive; Preventative not Remedial**: The *PbD* approach is charac-

terized by proactive rather than reactive measures. It anticipates and prevents privacy invasive events before they happen. *PbD* does not wait for privacy risks to materialize, nor does it offer remedies for resolving privacy infractions once they have occurred – it aims to prevent them from occurring. In short, *PbD* comes before-the-fact, not after.

- **Privacy as the Default Setting**: *PbD* seeks to deliver the maximum degree of privacy by ensuring that personal data are automatically protected in any given IT system or business practice. If an individual does nothing, their privacy still remains intact. No action is required on the part of the individual to protect their privacy – it is built into the system, *by default*.

- **Privacy Embedded into Design**: *PbD* is embedded into the design and architecture of IT systems and business practices. It is not bolted on as an add-on, after the fact. The result is that privacy becomes an essential component of the core functionality being delivered. Privacy is integral to the system, without diminishing functionality.

- **Full Functionality – Positive-Sum, not Zero-Sum**: *PbD* seeks to accommodate all legitimate interests and objectives in a positive-sum "win-win" manner, not through a dated, zero-sum approach, where unnecessary trade-offs are made. *PbD* avoids the pretense of false dichotomies, such as privacy vs. security, demonstrating that it is possible to have both.

- **End-to-End Security – Full Lifecycle Protection**: *PbD*, having been embedded into the system prior to the first element of information being collected, extends securely throughout the entire lifecycle of the data involved – strong security measures are essential to privacy, from start to finish. This ensures that all data are securely retained, and then securely destroyed at the end of the process, in a timely fashion. Thus, *PbD* ensures cradle to grave, secure lifecycle management of information, end-to-end.

- **Visibility and Transparency – Keep it Open**: *PbD* seeks to assure all stakeholders that whatever the business practice or technology involved, it is in fact, operating according to the stated promises and objectives, subject to independent verification. Its component parts and operations remain visible and transparent, to users and providers alike.

- **Respect for User Privacy – Keep it User-Centric**: Above all, *PbD* requires architects and operators to keep the interests of the individual uppermost by offering such measures as strong privacy defaults, appropriate notice, and empowering user-friendly options.

The definition of these principles alone are not enough. Gürses *et al.* [46] argue that while these principles are useful to guide development, they are still vague and leave many open questions on how to apply them in practice. For example, the fourth principle (Full Functionality – Positive-Sum, not Zero-Sum) classifies trade-offs as unnecessary because *PbD* seeks to accommodate the objectives in a "win-win" manner. However, there is no explanation and demonstration on how to achieve that. Through case studies, Gürses *et al*. demonstrate that there is not only one way to tackle privacy protection problems, and conclude that the development of a generalizable methodology is necessary. Our work acknowledges this and describes a methodology to ensure privacy preservation in various contexts.

The Nokia institute published a white paper [74] on how to use privacy protection policies. The paper states that there is a lack of privacy experts, so, a software engineering discipline called *Privacy Engineering & Assurance* is proposed. This discipline aims to be a systematic approach in the implementation of the *PbD* principles, and yet, foster the birth of a new professional, the *Privacy Engineer*. Despite the relevance, the introduction to new topics and concepts on its own is not enough. We still need a detailed methodology to know how to organize the development workflow and to guide the implementation of privacy-friendly applications.

Ionescu *et al.* [51] propose an argumentative structure to privacy cases, matching the privacy-protection goals to human and organizational privacy concerns. They seek to analyse privacy risks from the perspectives of different stakeholders, and provide convincing argumentation that support technical solutions in mitigating privacy risks. Our work recognizes the need of a solid and easy argumentative structure to provide evidences which support the privacy assurance.

The *Privacy Management Reference Model and Methodology (PMRM)* [2] is a proposal of the *Advancing Open Standards for the Information Society (OASIS)* which has high level guidelines to help business process engineers, IT analysts, architects, and developers in implementing privacy and security policies. By having a focus on policies, *PMRM* lacks guide-

lines on how to deal with many other important privacy concepts, such as user perceptions, utility tradeoffs, privacy techniques, metrics, and attacks. For not being dense, high level and focused on policies such as *PMRM*, *PbE* is a simple and development-driven proposal.

In another related work, Samani *et al.* [85] propose an architectural framework to ensure privacy based on the interactions that happen between IoT objects. The framework intends to restrict non-authorized operations, neutralizing the execution of such operations in the data of users. This neutralization happens with the integration of penalty (report privacy violations to authorities) and preventing operations (like anonymization techniques) to the communication protocol used by the objects. Although very useful, the work of Samani *et al.* does not address the Privacy vs. Utility trade-off and also does not suggest a sequence of acts of risk analysis and threat mitigation (for instance, investigating possible attacks).

Considering the related work, we can notice a lack of methodologies to guide the development and that considers important concepts like norms, perception studies, privacy techniques, trade-offs and attacks. Our novel proposed methodology considers such concepts and increases the confidence of the project through the provision of privacy evidences.

2.2 Conceptual Framework

In this section we describe the key concepts when dealing with privacy. The goal is to generate a manageable knowledge that is used in our proposed methodology. The conceptual framework with the key concepts identified by us is presented in Figure 2.1. We describe these concepts in the next sections.

2.2.1 Participants

Figure 2.2 presents the possible participants/stakeholders in a typical scenario when developing a privacy-friendly application. They are:

- *User*: The user of the application. Software and sensors may collect sensitive data from the user and transmit to remote servers.
- *Product Owner*: Entity that represents the executives. Initiates the project, finances it, contracts it out, and benefits from its output(s). Part of an owner responsibility is to

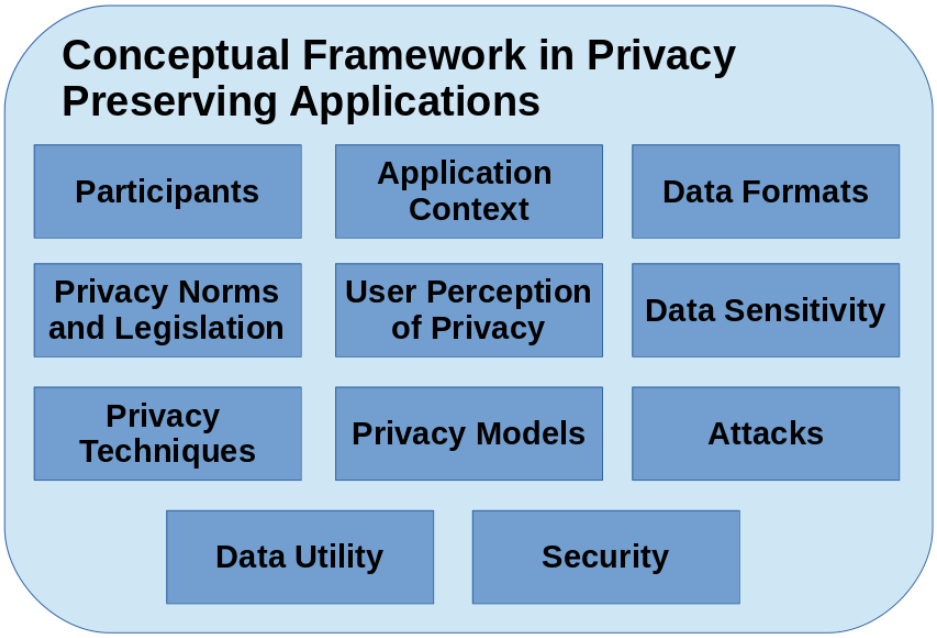


Figure 2.1: Key concepts when dealing with privacy.

have a vision of the requirements, and convey that vision to the development team.

- *Development Team*: Responsible for developing and delivering the project in accordance to the requirements (including data privacy and utility requirements).
- *Regulatory Agency*: Responsible for exercising autonomous authority and establishing privacy guidelines and rules. These preventive rules combined with penalty mechanisms can help preventing potentially dangerous activities.
- *Privacy Engineer*: Responsible for risk analysis, development and evaluation of privacy techniques, provision of evidences, simulation of attacks and assurance that the privacy norms established by the regulatory agencies are being followed.
- *Adversary*: Entity which seeks to violate the users privacy obtaining sensitive information. May perform many attacks depending on the application context, data format and privacy techniques.
- *Data Recipient*: Third party who is interested in the published data and may conduct data analysis in sensitive data.

The application environment contains the building blocks of the system, such as sensors,

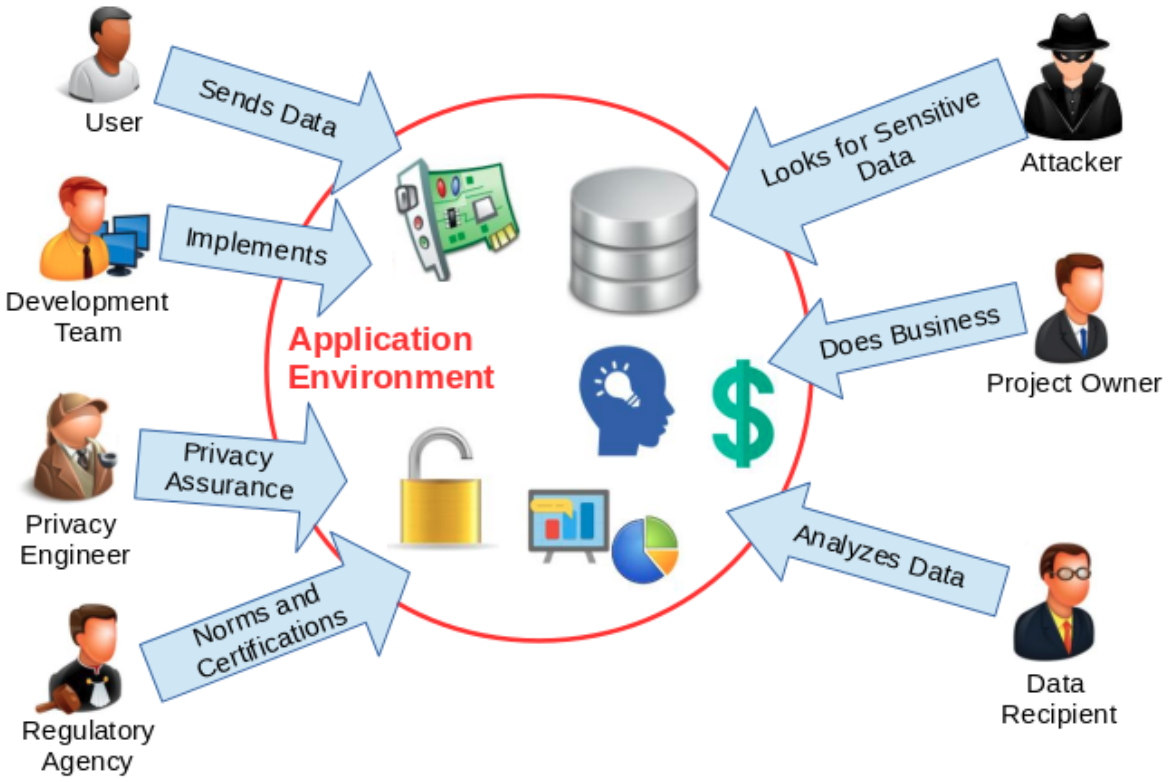


Figure 2.2: Possible participants when developing a privacy-friendly application.

servers, service APIs and databases. It may collect the data from users and publish to third parties or the public. A typical scenario of data collection and publishing is, for example, in health care. Through smart health devices, a hospital collects data from patients and publishes these records to an external medical center. In this example, the devices and the hospital are in the application environment, patients are users, and the medical center is the data recipient. The data analysis conducted at the medical center could be any task from a simple count of the number of men with diabetes to a sophisticated cluster analysis to make health insurances more profitable.

An adversary model may consider that the data recipient is untrusted. If the environment is trusted and users are willing to provide their personal information, there may be still a problem if the trust is transitive, *i.e.*, the data recipient be untrusted. In practice, different contexts may have different adversary assumptions and requirements.

2.2.2 Application Context

In different application contexts, the privacy and utility requirements may differ. Smart energy meters, social networks, geolocation systems, finance, and medical applications are some examples of application contexts. The identification of the context is essential, since different utilities can be provided when processing the data. More importantly, different application contexts can be target of different attacks and privacy violations, and therefore, the usage of different privacy protection techniques is required.

2.2.3 Data Format

In order to provide their features more precisely and specific for each individual, applications may collect large amounts of data. This data can exist in different formats; table records, time series, network traffic, graphs, text, images, among many others. The format in which the data is presented usually is related to the application context and different privacy protection techniques may be applied to different data formats.

2.2.4 Data Sensitivity

In the most basic form, a data unit may be or contain one of the following:

- **Explicit Identifier**: Set of attributes, such as name, email, phone number and IP address, containing information that explicitly identifies users.
- **Quasi Identifier**: Set of attributes that could potentially identify users such as ZIP code, sex and date of birth. We call QID the set of attributes and qid the values of this set.
- **Sensitive**: Consist of sensitive person-specific information such as disease, energy consumption, salary, and disability status.
- **Non-Sensitive**: Consist of all information that do not fall into the previous three categories.

More importantly, the sensitive data also possess different sensitivity levels (*e.g.*, low, medium and high). There are some standards of classification of sensitive data [93], however,

in practice, the classification depends on the context and population. Improper collection and usage of sensitive data may be a privacy violation.

2.2.5 Privacy Norms and Legislation

Establishing norms to restrict the usage of sensitive data is one of the preventive methods for privacy protection. Preventive norms combined with punishing mechanisms (such as reporting violations to authorities) can help preventing potentially dangerous activities. Many regulatory agencies have proposed privacy norms that must be followed.

As an example, the *HIPAA* (*Health Insurance Portability and Accountability Act*) [96] established the *Safe Harbor* standard, which is a precise method for anonymization of health information. It stipulates the removal or generalization of 18 variables from a data set:

1. Names;
2. All geographic subdivisions smaller than a state, including street address, city, county, precinct, ZIP code, and their equivalent geocodes, except for the initial three digits of a ZIP code if, according to the current publicly available data from the Bureau of the Census:
    a) The geographic unit formed by combining all ZIP codes with the same three initial digits contains more than 20,000 people.
    b) The initial three digits of a ZIP code for all such geographic units containing 20,000 or fewer people is changed to 000.
3. All elements of dates (except year) for dates directly related to an individual, including birth date, admission date, discharge date, and date of death; and all ages over 89 and all elements of dates (including year) indicative of such age, except that such ages and elements may be aggregated into a single category of age 90 or older;
4. Telephone numbers;
5. Fax numbers;
6. Electronic mail addresses;

7. Social security numbers;
8. Medical record numbers;
9. Health plan beneficiary numbers;
10. Account numbers;
11. Certificate/ license numbers;
12. Vehicle identifiers and serial numbers, including license plate numbers;
13. Device identifiers and serial numbers;
14. Web universal resource locators (URLs);
15. Internet protocol (IP) address numbers;
16. Biometric identifiers, including finger and voice prints;
17. Full face photographic images and any comparable images;
18. Any other unique identifying number, characteristic, or code.

The certainty and simplicity of *Safe Harbor* makes it quite attractive for health information custodians when disclosing data without patient consent, and it is used quite often in practice [40].

Another regulation is the *GDPR* (*General Data Protection Regulation*) [3], which is to be enforceable in the European Union from 25 May 2018. The aim of the *GDPR* is to "protect all European citizens from privacy and data breaches", and has the following key points:

- *Territorial Scope*: The *GDPR* applies to the processing of personal data of Europen ciyizens by controllers and processors, regardless their locations;
- *Penalties*: Organizations in breach of *GDPR* can be imposed for the infringements;
- *Consent*: Users consent must be clear and distinguishable from other matters and provided in an intelligible and easily accessible form, using clear and plain language.

Beyond the key points, *GDPR* stipulates *Privacy by Design* as a legal requirement during the development, and the European citizens have the following rights: the right to be notified when a data breach occurs, the right to access their data (and in a portable format), the right to be forgotten, and the right to restriction of processing.

Unfortunately, norms or regulations such as *GDPR* and *HIPAA* usually are not enough to guide developers in developing privacy-friendly applications. This is why the provision of concrete evidences of privacy and the implementation of other counter-measures such as conducting privacy perception studies, implementing privacy techniques, and evaluating potential attacks, becomes necessary.

2.2.6 Users Perception of Privacy

Different people usually exhibit different perceptions of privacy, *i.e.*, they associate the word privacy with a diversity of meanings. For example, some people believe that privacy is the right to control what information about them may be made public [87; 16; 66; 104]. Other people believe that if someone cares about privacy is because he/she is involved in wrongdoing [20].

Different perceptions of privacy originate different types of concerns about privacy. For example, some people tend to provide the information requested by the system only if it presents a privacy policy. Privacy policy is a legal document and software artifact that fulfills a legal requirement to protect the user privacy. It answers important questions about the software's operation, including what personal identifiable information is collected, for what purpose is it used, and with whom is it shared [23].

The application of surveys and/or the conduction of interviews may cover and help to understand many privacy aspects, such as: (i) general perceptions, beliefs and attitudes; (ii) perceptions about data collection and control; (iii) perceptions about information inference; (iv) perception about information usage; and (v) perceptions about possibilities of data exchange (not only leakage) [80].

2.2.7 Privacy Techniques

The raw data usually does not satisfy specified privacy requirements and it must be modified before the disclosure. The modification is done by applying privacy techniques which may come in several flavors, like anonymization [35], generalization [95], noise addition [13], zero-knowledge proofs [44] and the usage of homomorphic encryption [58]. As an example, anonymization refers to the approach that seeks to hide the identity of users, assuming that sensitive data must be retained for data analysis. Clearly, explicit identifiers of users must be removed.

Several techniques may work for the same context and data format. The objective of such techniques is to protect sensitive user data from possible privacy violations. It may also include the provision of the control of what information may be disclosed to the service, data recipients or other users.

2.2.8 Attacks

A privacy attack is an attempt to expose, steal or gain unauthorized access to sensitive data. Unfortunately, applying norms and privacy techniques may not be enough and real attackers may have success in violating the user privacy when exploiting flaws in the norms and privacy techniques. Thus, before the deployment of the application, privacy engineers may simulate attacks, seeking to explore and fix additional privacy breaches. Attack simulation reports may also assess potential impacts to the organization and suggest countermeasures to reduce risks.

As an example of attack, even with all explicit identifiers removed, an individual's name in a public voter list may be linked with his record in a published medical database through the combination of ZIP code, date of birth, and sex, as shown in Figure 2.3. Each of these attributes does not uniquely identify a user, but their combination, called the quasi-identifier (qid value), often singles out a unique or a small number of users. Research showed that 87% of the U.S. population had reported characteristics that made them unique based on only such quasi-identifiers [94].

In the example of Figure 2.3, the user is re-identified by linking his qid. To perform such linking attacks, the attacker needs two pieces of prior knowledge: the victim's record in the

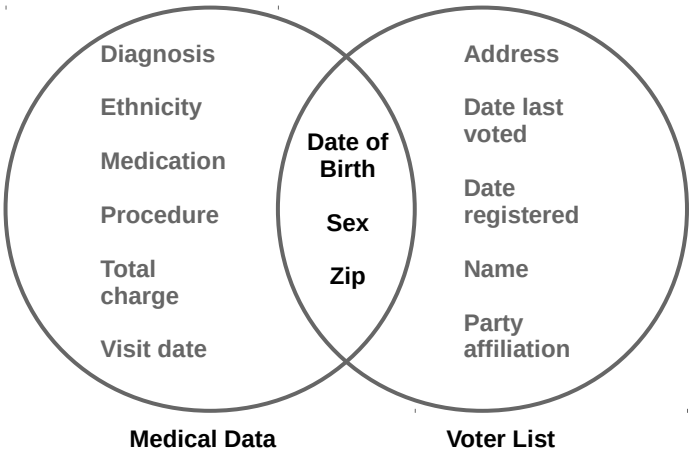


Figure 2.3: Linking attack to re-identify users [95].

released data and the qid of the victim. Such knowledge can be obtained by observations. For example, the attacker noticed that his boss was hospitalized, and, therefore, knew that his boss's medical record would appear in the released patient database. In another example, it is not difficult for an adversary to obtain his boss's ZIP code, date of birth, and sex, which could serve as the quasi-identifier in linking attacks.

Besides linking attacks, other examples of attacks may include filtering (when applying noise addition techniques), integer factorization and discrete logarithm computation (when applying homomorphic encryption, for example), and side-channel attacks.

2.2.9 Privacy Models

To prevent attacks such as linkage through quasi-identifiers and to quantify the privacy levels, it is necessary to validate the privacy technique using formal privacy models. Sweeney *et al.* [94] propose the notion of k-anonymity: If one user has some value qid, at least $k-1$ other records also have the value qid. In other words, the minimum equivalence group size on QID is at least k. A data set satisfying this requirement is called k-anonymous. In a k-anonymous data set, each user is indistinguishable from at least $k-1$ other records with respect to QID. Consequently, the probability of linking a victim to a specific user through QID is at most $1/k$.

Another insightful privacy model is the ϵ-differential privacy: the risk to the user's privacy should not substantially increase as a result of participating in a statistical database.

Dwork [37] proposes to compare the risk with and without the user's data in the published data. Consequently, the privacy model called ϵ-differential privacy ensures that the removal or addition of a single element does not significantly affect the outcome of any analysis.

Considering a data set as a set of participants, we say data sets $D1$ and $D2$ differ in at most one element if one is a proper subset of the other and the larger data set contains just one additional participant. Therefore, a randomized function F gives ϵ-differential privacy if for all data sets $D1$ and $D2$ differing on at most one element, and all $S \subseteq Range(F)$, $Pr[F(D1) \in S] \leq exp(\epsilon) \times Pr[F(D2) \in S]$, where the probability is taken over the randomness of function F [37].

The ϵ value is the privacy metric and for better privacy, a small value is desirable. A mechanism F satisfying this definition addresses concerns that any participant might have about the leakage of his personal information: even if the participant removed his data from the data set, no outputs (and thus consequences of outputs) would become significantly more or less likely.

Differential privacy is achieved by the addition of noise whose magnitude is a function of the largest change a single user could have on the output to the query function; this quantity is referred as the sensitivity of the function.

Besides k-anonymity and ϵ-differential privacy, there are many other privacy models, as presented in Table 2.1. The privacy model to use depends on the attack model, application context, data format, privacy technique, and other factors.

2.2.10 Data Utility

We consider data utility as the usefulness of the data for the achievement of the primary features of the application. A *utility metric* may measure the quality and the business value attributed to data within specific usage contexts.

In the previous sections, we presented only concepts regarding privacy. However, regarding utility, in order to provide their features more precisely and specific for each individual, applications may collect a large of data which may be useful for many purposes. Privacy preventive measures may have the potential to limit the data utility, or even prevent the operation of some feature. Thus, we are facing a Utility vs. Privacy trade-off. In the development of an application, it is necessary to ensure that the privacy of the users will not be compromised

Table 2.1: Privacy Models [42]. A tick represents the coverage of an attack model.

Privacy Model	**Attack Model**			
	Record linkage	Attribute linkage	Table linkage	Probabilistic attack
k-Anonymity [95]	✓			
MultiR k-Anonymity [70]	✓			
l-Diversity [62]	✓	✓		
Confidence Bounding [98]		✓		
(a, k)-Anonymity [102]	✓	✓		
(X, Y)-Privacy [97]	✓	✓		
(k, e)-Anonymity [57]	✓	✓		
(ϵ, m)-Anonymity [61]		✓		
Personalized Privacy [103]		✓		
t-Closeness [72]		✓		✓
δ-Presence [70]			✓	
(c, t)-Isolation [34]	✓			✓
ϵ-Differential Privacy [37]			✓	✓
(d, γ)-Privacy [83]			✓	✓
Distributional Privacy [24]			✓	✓

and still allow the largest possible number of the services keep functioning.

2.2.11 Security

Despite the focus in privacy, it is necessary to notice the fact that the security of the system is also vitally important. It is known that privacy is a sub-area of security, more specifically, in confidentiality. In fact, it is very common to see even professionals of Information Technology mixing the concepts between security and privacy. However, more recently, research on privacy are taking different directions than research on security. For example, data analysis for infering human behavior is considered a hot topic in privacy, whereas software vulnerabilities is a hot topic in security.

A system with security vulnerabilities transitively compromises the data privacy and if an attacker seizes control of the application, the privacy will be at risk. To address these concerns, it is necessary the inclusion of practices used in security assurance methodologies, such as the execution of penetration testing to detect vulnerabilities in the involved elements. To ensure that privacy is present in the whole communication process, it is necessary to take preventive security measures, such as encrypting the data in transit and using signed certificates to avoid man-in-the-middle attacks (MiTM) [92].

2.3 Discussion

We presented a general conceptual framework to be used when dealing with privacy preserving in software development. However, one question remains: how to organize these key concepts when developing applications? We concluded that it would be interesting to have a methodology that guides the development, applying in practice all these concepts. The conceptual framework is to be used in subsequent parts of our research.

Chapter 3

Privacy by Evidence: Our Proposed Methodology

According to the first and third principles of *PbD* [31], the team must act in a preventive manner (not waiting for the violations to happen), and the analysis and risk mitigations must be embedded into the design of the project. For this reason, our methodology consists in the inclusion of a few activities conducted by a privacy team in parallel with the normal development cycle. The parallelism of the privacy preserving activities and the common development is important here, since the product owner does not want a long time to market. If the development takes long, it can be harmful for the company (*e.g.*, competitors may have the same ideia).

Privacy by Evidence (*PbE*) begins with the identification of the application context and the data formats, then, legislations are studied, providing evidences of privacy protection. After that, a list of data utilities and their potential privacy risks is made. The next activities are to apply privacy protection techniques and evaluate potential attacks. Since we are proposing an iterative methodology, the team can go back and re-execute the previous activities, being in accordance with the normal software evolution cycle. These activities are structured using a *UML Activity Diagram*, as shown in Figure 3.1. This chapter focuses on explaining the workflow to be executed by the privacy team (right part in Figure 3.1).

Although we use the term "privacy team", it is important to note that this team does not need to have privacy experts. In fact, our goal is to guide common developers in doing privacy preserving activities. We separate the teams just to avoid biasing. In the next sections,

we detail each of these activities.

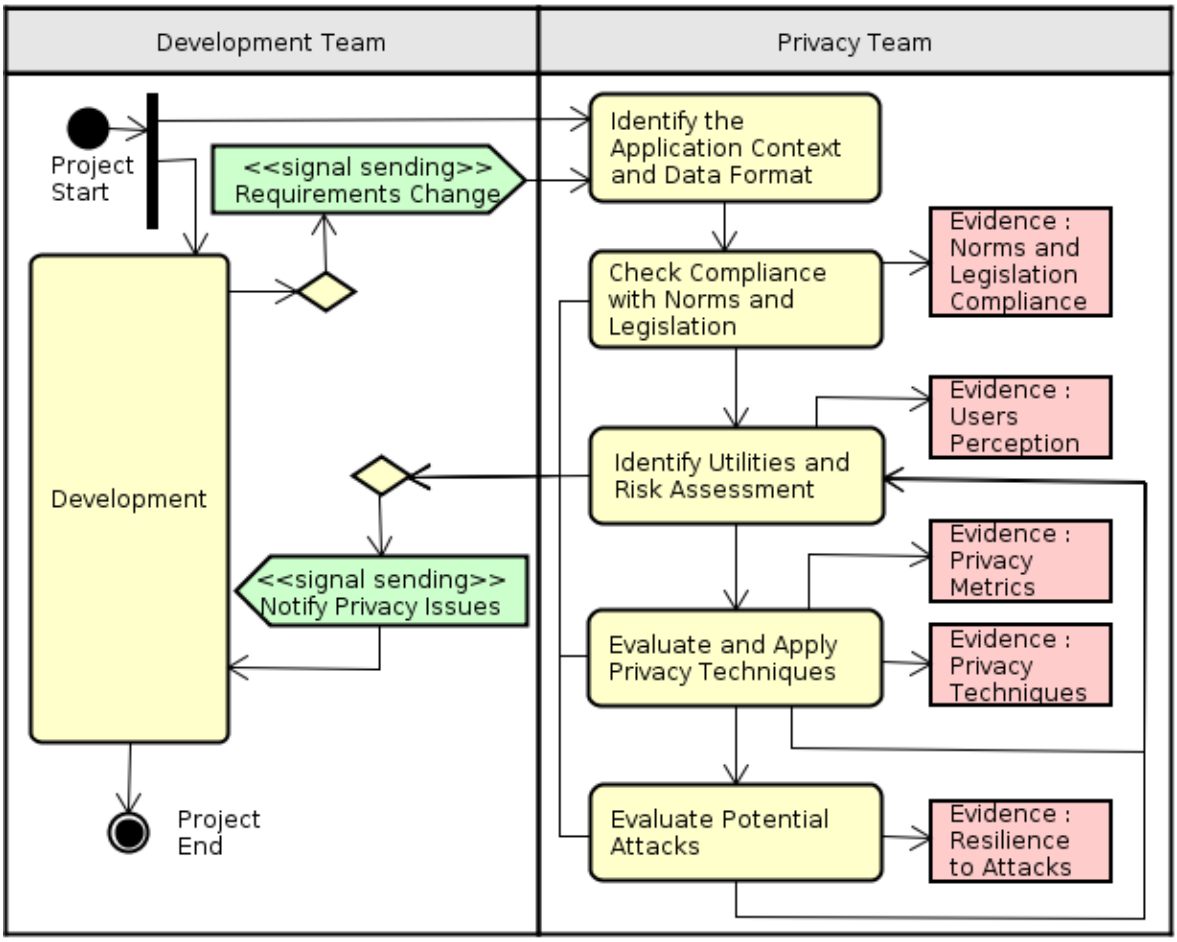


Figure 3.1: The activities of *Privacy by Evidence*, a software development methodology to provide privacy assurance.

3.1 Identify the Application Context and the Data Formats

In different application contexts, the collected data can exist in different formats. Smart meters, people counting, social networks, geolocation systems and medical applications are some examples of contexts. Time series, aggregated values, network traffic, graphs, geographic localizations and table records are some examples of data formats. The identification of the context and the data format is essential, since different utilities can be provided when processing the data. More importantly, different contexts and data formats can be target of different attacks and privacy violations, and therefore, the usage of different privacy

protection techniques is required.

Possible artifacts of this activity may include:

- **Engagement Report**: Information regarding the application context, goals and scope;
- **Datasets**: Samples of the data formats. Most of the time, they do not need to be real. They help the privacy team to understand how the user data would appear to other participants and consequently, possible data manipulations.

3.2 Check Compliance with Norms and Legislations

Creating norms to restrict usage of sensitive data is one of the preventive methods for privacy protection. Combined with punishing mechanisms (such as reporting violations to authorities) they can help preventing potentially dangerous activities. Many agencies have proposed privacy rules that must be followed. Being in accordance with these rules generates evidences that privacy is being taken into account and implemented in the development. During this activity, a study about existing norms must be made.

It is also valuable to study legislations, principles and guidelines proposed by organizations and governments. This activity precedes others because the team can not proceed to implement something that is not allowed by the norms and legislations.

Norms and legislations depend on the regions and the companies. For example, in Brazil, the rules for collecting and storing specific sensitive data could be very different from the rules in United States of America (actually, between different states of USA, these could be different too). Also, a company that develops software such as the Hewlett Packard (HP) may have internal privacy norms different from Sony.

Possible artifacts of this activity may include:

- **Summary of Norms**: For different regions and companies, it may be the case of having different norms and legislations;
- **Implementation of Norms**: Code parts of norm implementations and unit tests. For example, using information flow annotations in Java [69], it is possible to perform static checking on the fields specified by the *Safe Harbor* standard, presented in Section 2.2.5;

- **Compliance Proofs**: They are evidences of privacy. For example, certifications are compliance proofs ensuring that the privacy practices meet the constraints defined by the regulatory agencies.

3.3 Identify Utilities and Risk Assessment

After the identification of the context and the data format, the team should try to list all the potential utilities available after data processing (we consider as utilities everything that the data can be used for, including possible privacy violation activities). Once the team has a list of data utilities, it analyzes and classifies them in a risk level scale, which varies according to the sensitivity of the available data and the adversary model. For instance, low, medium or high level of privacy risk is a possible scale.

For a better classification of the privacy risks, we suggest the conduction of privacy perception studies, where questionnaires and interviews are executed and the answers of the target users are evaluated. These studies may help to understand the sensitivity of the data, to define the adversary model and to elaborate the privacy policy document. Conducting such perception studies and being in accordance with the concerns of the users are in accordance with the seventh principle of *PbD* and also generate evidences of privacy (perhaps, the most important ones).

Maybe for better perception studies, the privacy team wants to question users regarding the real intended system. However, in a first iteration of our methodology, we do not recommend to collect and use real user data. If serious privacy concerns are detected during the first iteration, this can already cause serious damages. Thus, we only suggest the use of real data in a second iteration, where a privacy policy document has already been presented to the user, and privacy techniques and possible attacks have already been evaluated.

Possible artifacts of this activity may include:

- **Utilities List**: For what the data can be used for, including legitimate and bad purposes. A utility has a sensitivity level, according to a defined scale;

- **Perception Questionnaires**: Set of questions designed to gain knowledge about users' perceptions of the system;

- **Perception Report**: To consider the users' perception in the design of the system is also an evidence of privacy. This report may include statistics, and general privacy recommendations for the system design;
- **Privacy Concerns**: Identified threats, based on the perception report, the developers experience, or evaluations;
- **Adversary Model**: Defines the possible adversaries, such as other users, remote services and third parties. It helps to understand the scope of the privacy mitigations.
- **Privacy Policy**: A document that fulfills a legal requirement to protect the user privacy.

3.4 Evaluate and Apply Privacy Techniques

Despite the existence and implementation of privacy norms, in fact, privacy protection is not achieved yet and for this reason, sometimes it is necessary to apply privacy techniques. Anonymization [35], generalization [95], noise addition [13], and use of homomorphic encryption [58] are examples of possible privacy techniques. In Appendix A we present a catalog of privacy techniques that may be useful in helping developers in choosing suited ones. The catalog was created based on the experience of the author. We do not consider technologies for the security of a system, for example firewalls, intrusion detection systems, etc., but our focus to the techniques which directly act upon the information content.

Several techniques may work for the same context and data format. Therefore, through evaluation of performance metrics such as computational time, cost, application complexity, accuracy, scalability and fault tolerance, and through evaluation of metrics of formal privacy models such as k-anonymity [95], l-diversity [62] and ϵ-differential privacy [37], it is decided the privacy techniques to be applied. The application of privacy techniques and evaluation through privacy metrics also generate evidences.

The privacy techniques are chosen based on the recommendations generated by the privacy perception study and in order to optimize the utilities list, created in the previous activity. The team chooses the techniques that provide the greatest possible amount of utilities that poses no threats and which inhibit the greatest possible amount of utilities that represent threats to users privacy.

The extraction of privacy goals from the privacy policy document may also help in choosing the privacy techniques. Bhatia *et al.* [23] introduce a semiautomated framework that uses a hybrid combination of crowdsourcing and natural language processing to improve the extraction of privacy goals.

The application of a privacy technique may be a software feature and therefore, code parts with the implementation may be generated. Unit tests can also be generated and attached into a continous integration platform, thus, being verified automatically in different stages of the software lifecycle.

Possible artifacts of this activity may include:

- **Summary of Techniques**: A catalog, with privacy techniques that can be applied, considering the risk assessment executed in the previous activity;
- **Techniques Report**: Includes a comparison between the possible privacy techniques. Besides the privacy models, the team may also consider other metrics, such as cost, complexity and accuracy. The choice of good privacy techniques and the achievement of high privacy levels are also evidences of privacy;
- **Implementation of Techniques**: Code parts with implementation of the privacy techniques and the corresponding tests.
- **New Utilities List**: With the chosen privacy techniques, it is desirable to provide utilities that poses no privacy threats and to inhibit the utilities that represent threats;

3.5 Evaluate Potential Attacks

After the application of privacy techniques, the team must assure that the unwanted utilities were in fact inhibited. For this reason, it simulates attacks, seeking to explore additional privacy breaches and still search for unwanted utilities. Each unsuccessful attack serves as an evidence that the privacy techniques were implemented in an effective manner and that the data is protected against that kind of attack. In the event of a successful attack, the team must execute the risk assessment again and then improve the techniques or seek for others that better address the detected vulnerabilities.

The simulation of attacks is considered as a software testing activity and can generate scripts that could be integrated with the tests of the software and a continous integration platform, thus, being verified automatically in different stages of the software lifecycle.

Possible artifacts of this activity may include:

- **Summary of Attacks**: A catalog, with the potential attacks. Includes the attack plannings and their hypotheses;
- **Attack Scripts**: The scripts of the attacks to be performed;
- **Attacks Report**: May include experimental and analytical results of the performed attacks as well as recommendations on how to fix identified flaws. Each unsuccessful attack is an evidence of privacy.

3.6 Documentation

Table 3.1 shows a framework for the checklist with the possible artifacts to be generated in each activity of the methodology. A goal of the privacy team may be to supply these artifacts and fill the *Supplied?* column with check marks (✓).

During the development process, the satisfability of privacy requirements and the generation of evidences and artifacts may be coupled into *User Stories*, as well as the definition of *Acceptance Criterias*.

Even taking privacy aspects into consideration and with many collected evidences, one can never state that a system is completely safe. New vulnerabilities can always be found and yet, real attackers can explore breaches that were not detected by the privacy team. For this reason, there is a need to a constant and iterative process, to analyze new risks and consequently inhibit them. To represent this need, the artifacts should include the review history and dates.

3.6.1 Privacy Case

The concept of the “safety case” has been adopted across many industries [54]. Based on the results found in the safety literature, we define the term “privacy case” and apply some of

Table 3.1: Framework for the checklist of the artifacts to be produced by *PbE*.

Activity	Artifact	Supplied?
Identify the Application Context and Data Formats	Engagement Report	
	Datasets	
Check Compliance with Norms and Legislations	Summary of Norms	
	Implementation of Norms	
	Compliance Proofs	
Identify Utilities and Risk Assessment	Utilities List	
	Perception Questionnaires	
	Perception Report	
	Privacy Concerns	
	Adversary Model	
	Privacy Policy	
Evaluate and Apply Privacy Techniques	Summary of Techniques	
	Techniques Report	
	Implementation of Techniques	
	New Utilities List	
Evaluate Potential Attacks	Summary of Attacks	
	Attack Scripts	
	Attacks Report	

the recommendations in our development process. We define the purpose of a privacy case in the following terms:

Definition 1 *A privacy case communicates a clear, comprehensive and defensible argument that a system preserves the users privacy when operating in a particular context.*

Both argument and evidence are crucial elements of the privacy case that must go hand-in-hand. An argument without supporting evidence is unfounded, and therefore unconvincing. Evidence without an argument is unexplained - it can be unclear that (or how) privacy goals have been satisfied. To provide an evidence of privacy, the team should make the most appropriate decision by using the most accurate information together with its knowledge, experience and evaluations. Therefore, we define “evidence of privacy” in the following

terms:

Definition 2 *An envidence of privacy is a best-effort documentation, indicating an effective privacy decision-making and an argument confirmation.*

Despite the focus in privacy cases, we must pay attention to the fact that the security of the system is also vitally important, as described in the fifth principle of *PbD*. A system with security vulnerabilities transitively compromises the data privacy and if an attacker seizes control of the system, the privacy will be at risk. To address these concerns, in parallel to our privacy assurance methodology, we suggest the inclusion of practices used in security assurance methodologies, such as the evaluation of penetration testing to detect vulnerabilities in the involved elements.

3.6.2 GSN

For the privacy case, we consider the evidences of privacy as essential and we propose that when collected, such evidences should be structured using the *Structured Assurance Case Metamodel* (*SACM*) [75]. *Goal Structuring Notation* (*GSN*) [45] is an extension of *SACM* and provides graphical argumentation notation that can be used to document explicitly the individual elements of any argument (claims, evidence and contextual information) and the relationships that exist between these elements (*i.e.*, how claims are supported by other claims, and ultimately by evidence, and the context that is defined for the argument). Arguments documented using *GSN* can help provide assurance of critical properties of systems, services and organizations. Within Europe, *GSN* has been adopted by a growing number of companies within safety-critical industries (such as aerospace, railways and defence) for the presentation of safety arguments within safety cases [54]. In this work, we reuse these concepts to what we defined as privacy cases.

When the elements of the *GSN* are linked together in a network, they are described as a "goal structure". The principal purpose of any goal structure is to show how goals (claims about the system) are successively broken down into sub-goals until a point is reached where claims can be supported by direct reference to available evidence. As part of this decomposition, using the *GSN* is also possible to make clear the argument strategies adopted, the rationale for the approach and the context in which goals are stated (*e.g.*, the system scope

or the assumed operational role).

The principal symbols of the notation are shown in Figure 3.2 (with example instances of each concept). In this structure, as in most, there exist "top level" goals – statements that the goal structure is designed to support. In this case, "*The user privacy is being preserved*", is the (singular) top level goal. Beneath the top level goal or goals, the structure is broken down into sub-goals, either directly or, as in this case, indirectly through a strategy. The argument strategy that addresses the top level goal is "*Argument about protection when the service is an adversary*". This strategy is then substantiated by a sub-goal. At some stage in a goal structure, a goal statement is put forward that need not be broken down and can be clearly supported by reference to some evidence. In this case, the goal "*An adversary can not infer behavioral information through energy data*", is supported by direct reference to the evidence "*Noise Addition has been implemented*".

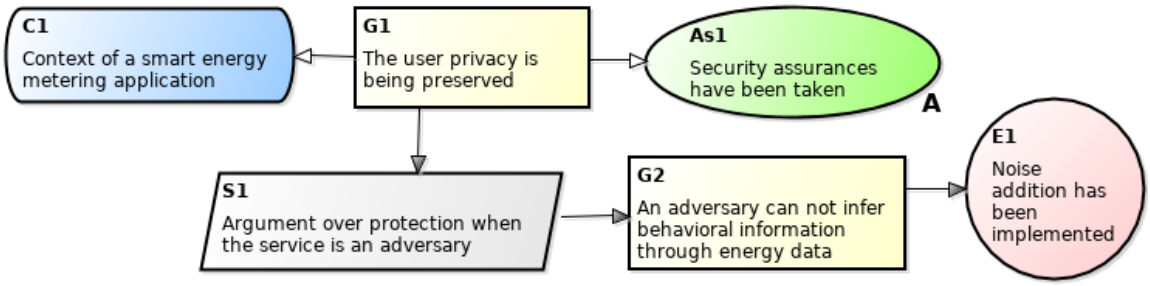


Figure 3.2: An example goal structure.

Table 3.2 describes a sheet for the evidence $E2$, making it easier for the participants of the project to relate it to the corresponding privacy goals and concerns. Every evidence has an identification and a reference to the *PbE* activity that generated such evidence, enabling the traceability into the corresponding artifacts and the possibility to provide visibility and transparency, according to the sixth principle of *PbD*. The description of an evidence provides information for how the evidence fulfills its driving goals. It may also contain links with more details regarding the provided evidence, status, review date, and an evidence weight. Some evidences may be more valuable than others and therefore, we encourage teams to choose the weight using a voting system similar to Scrum poker [68].

Table 3.2: Sheet for the evidence $E2$. Implementation of the privacy technique of noise addition.

E2	**Noise addition has been implemented**	**Status**: Done	**Review Date**: February 2015	**Weight**: 8
PbE Activity: *Evaluate and Apply Privacy Techniques*				
Driven by: G1 $\rightarrow$ S1 $\rightarrow$ G2; *In context of*: C1; *Assumptions*: As1				
Description: A noise addition privacy-preserving scheme for smart metering has been implemented. We claim that the solution meets the needs of consumers (privacy) and power providers (utility). The modification in the communication procedure between a smart meter and the power provider is just the generation of a random number and the addition of this number to the measurement to be sent to the power provider.				
References: Section 4.4.				

3.7 Validation of *PbE*

To validate the effectiveness of *PbE*, it has been conducted four case studies, executing the activities described here in smart energy metering, people counting, energy efficiency, and two factor authentication applications. These are examples of applications that are growing in numbers, and some of them may bring privacy risks, causing many people and even the media to show distrust about them.

The main objective of these case studies is to assess the ability of *PbE* in producing evidences of privacy. In order to measure the achievement of this objective, we use the *Goal, Question, Metric* (*GQM*) paradigm [17], a mechanism for defining and evaluating goals using measurement.

3.7.1 *GQM*

GQM defines a measurement model on three levels: conceptual level (Goal), operational level (Question) and quantitative level (Metric). *GQM* templates are a structured way of specifying goals. A *GQM* template contains the following fields: *purpose, object of study, focus, stakeholder and context*.

In our validation, here is the *GQM* template to express the goal of our study: The purpose

of this study is to *evaluate the ability* of the *Privacy by Evidence methodology* in *helping to produce privacy-friendly applications* from the point of view of *developers* in the context of *the development process*.

In the next sections, we conduct four case studies, therefore, four research questions were defined to characterize the measurement objects:

- **RQ1**: Is *PbE* helpful to develop a privacy-friendly smart energy metering application?
- **RQ2**: Is *PbE* helpful to develop a privacy-friendly people counting application?
- **RQ3**: Is *PbE* helpful to develop a privacy-friendly energy efficiency application?
- **RQ4**: Is *PbE* helpful to develop a privacy-friendly two factor authentication application?

To answer these three research questions, we use the metrics of number of provided evidences and their sum of weights. If the development of an application is being able to produce evidences of privacy protection, then this indicates that *PbE* is being helpful.

3.8 Discussion

In this chapter, we proposed the *Privacy by Evidence* (*PbE*), a novel methodology to guide the implementation of privacy concepts when developing applications and to provide evidences of privacy. *PbE* is in accordance with all the 7 principles defined by *PbD*, and therefore, *PbE* may be considered as an extension of *PbD*. We specified a framework for the checklist of the possible artifacts. Some of these artifacts are evidences of privacy, and we propose to structure them using the *Goal Structuring Notation* (*GSN*). In the next chapters, we seek to validate *PbE* through case studies.

Chapter 4

Case Study I: Smart Metering

In this chapter, we present the first case study, a smart metering application which uses information regarding energy consumption. Although the main objective is to validate the *PbE*, we believe that the results presented in this chapter are of importance for the smart metering area. These results were to be applied in the LiteMe[1] application, a project developed by the Smartiks[2] company in partnership with the *Federal University of Campina Grande* (*UFCG*).

4.1 Context and Data Formats

Smart Grids are systems that combine the traditional power grid with modern information technologies to enable a more efficient and robust grid. With these systems, power providers can monitor, analyze and control the network and communicate with the consumers to improve the energy quality, reduce energy consumption and cost, and maximize the transparency and reliability of the energy supply chain. In this scenario, a key component on the consumers' side is the use of smart meters.

Smart meters are devices that measure electricity consumption in real time and transmit this data to remote servers. These devices may represent a turning point in the energy industry and foster the development of new services and improvement of existing ones. In a typical smart metering architecture, the analysis of the collected data can help power providers to learn how to better manage the areas within their networks. Thus, helping to understand the

[1] `http://liteme.com.br`

[2] `http://www.smartiks.com`

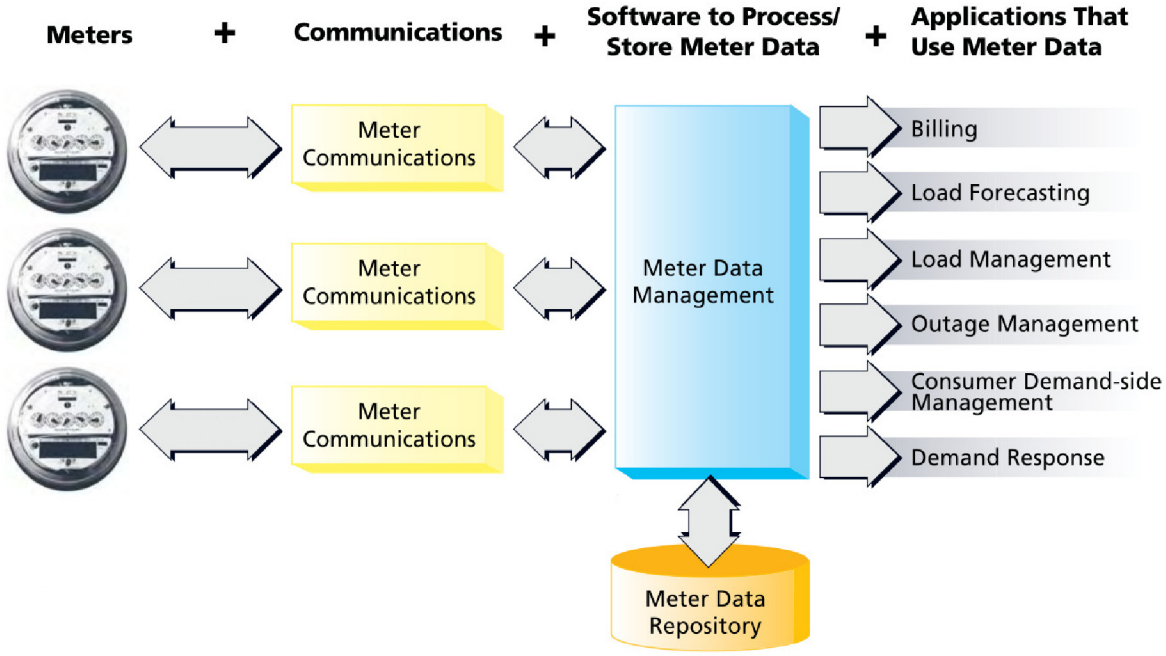


Figure 4.1: Smart metering system architecture and data usage applications [100].

business benefits of investing in Smart Grids. Figure 4.1 presents a smart metering system architecture.

Despite the benefits, smart meters raise concerns about the privacy of consumers. Electricity data may contain private sensitive information, such as which appliances are being used, if the house is empty, when people take a shower or shut down the television. The privacy issues are some of the main reasons why smart meters were still not deployed in many countries [55]. Clearly, there is a trade-off between utility and privacy.

As an example of the data format, Figure 4.2 shows a daily profile of a residential consumer.[3] Using advanced power signature analysis tools, such as the *Non-Intrusive Appliance Load Monitoring* (*NIALM*), it is possible to find out private information about the consumer's lifestyle. Batra *et al.* [19] designed some methodologies to identify the use of appliances from load profiles. In Figure 4.2, the appliance with highest wattage and easier to identify is the laundry dryer. If the load monitoring algorithm is running remotely, the consumers may not know that their behaviors are being monitored.

In a traditional industrial setting, such behavior information is not in principle useful

[3]Combining appliance signatures it is possible to generate arbitrary large populations and measurement frequency. Several databases of appliance signatures are available online (*e.g.*, Tracebase [84]).

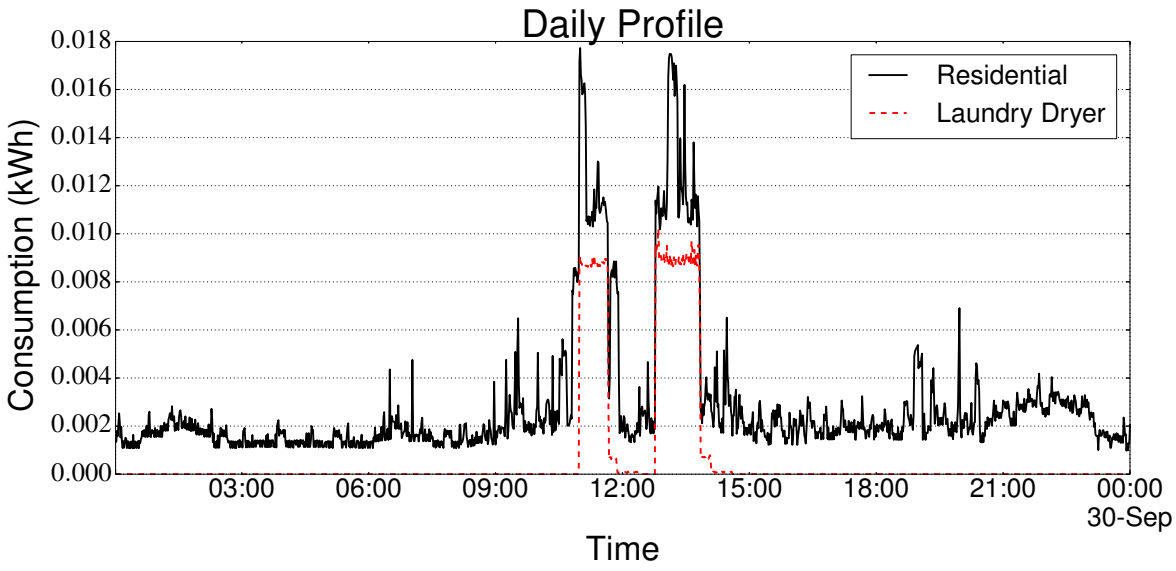


Figure 4.2: Residential (black solid) and Laundry Dryer (red dashed) daily profiles with measurements at each 1 minute.

for a power provider. However, currently this type of information is of interest to many businesses that want to identify the profile of a potential consumer of their products and services. Therefore, disclosing such profiles and habits of consumers evokes issues about privacy. Clear rules are needed to protect consumers from misuse of their behavioral data and to avoid that Smart Grids become a new type of Big Brother [25]. Unfortunately, protection laws may take decades to be applied whereas smart meters are already operational.

4.2 Norms and Legislation

Regarding the activity of checking compliance with norms and legislations for the energy data, the *Brazilian National Agency of Electrical Energy* (*ANEEL*) establishes a normative resolution[4] with the following rule of privacy for smart metering: "*In the hypothesis of a metering system with a remote communication, the power provider has to adopt procedures and technologies to ensure the security of the data traffic and, specially, of the collected personal information*". In other words, this rule determines that security mechanisms such as encryption and certification must be used.

[4]ANEEL, Normative Resolution Nº 502, Chapter III, Art. 7. `www.aneel.gov.br/cedoc/ren2012502.pdf`

In the same article, *ANEEL* establishes another rule: "*It is forbidden for the power provider to disclose to third parties the data collected from consumer units without authorization of the owners*". These rules could help to provide some evidences of privacy, but unfortunately they were suspended by the agency since February 11, 2014. Moreover, even if they come back in the future, it is known that there are possibilities of external hackers to penetrate systems (including the Meter Data Management in the power provider), or internal malicious employees to export the data after a decryption.

Another rule[5] established by *ANEEL* is: "*The meters must have mass storage capable of storing active and reactive energy data, demand and tension, considering the direct and reverse flow of energy according to the usage, at programmable intervals of 5 (five) to 60 (sixty) minutes*". The granularity of 5 minutes is the important information here. This could help to provide evidences of privacy, however it is known that *NIALM* algorithms can still identify appliance usages with this granularity. Moreover, this rule is only applied to official and regulated meters. The energy meter used by the LiteMe application is just an additional device that does not have the goal to replace existing traditional meters. Therefore, for many purposes, this device has the ability to collect and send data with time intervals of 1 second.

Despite the lack of norms and legislation to be applied in this application, the study and the summary of possibilities suggest a concern for privacy in this project, generating this, an evidence of privacy. Table 4.1 describes a sheet for this evidence $E1$ and Figure 4.3 presents the first part of the *GSN* for the privacy case of a smart metering application. This representation is still to grow, according to the normal software evolution and our proposed methodology. In this application context, assuming that security assurances have been taken, there is an argument that the privacy is being preserved according to the provided evidences. For now, there is only one evidence provided ($E1$).

4.3 Utilities and Risk Assessment

For the risk assessment, first we identified what is possible to do with the collected energy data. Since individual values (such as the consumption of a house in an instant of time)

[5]ANEEL, PRODIST, Module 5, Section 4.1.3.1. `www.aneel.gov.br/arquivos/PDF/Modulo5_Revisao\_2.pdf`

Table 4.1: Sheet for the evidence $E1$. Norms and legislations for smart metering.

E1	**Norms and legislations have been studied**	**Status**: Done	**Review Date**: August 2015	**Weight**: 1
PbE Activity: *Check Compliance with Norms and Legislation*				
Driven by: G1 $\rightarrow$ G2; *In context of*: C1; *Assumptions*: As1				
Description: Possible norms and legislations that could be applied for the smart metering application have been studied and summarized. In Brazil, it was not found any norm or law dealing with the collection, storage and processing of energy metering data that could affect the design of the LiteMe application. There are norms proposed by the *ANEEL*, but only applied to official and regulated meters.				
References: Section 4.2.				

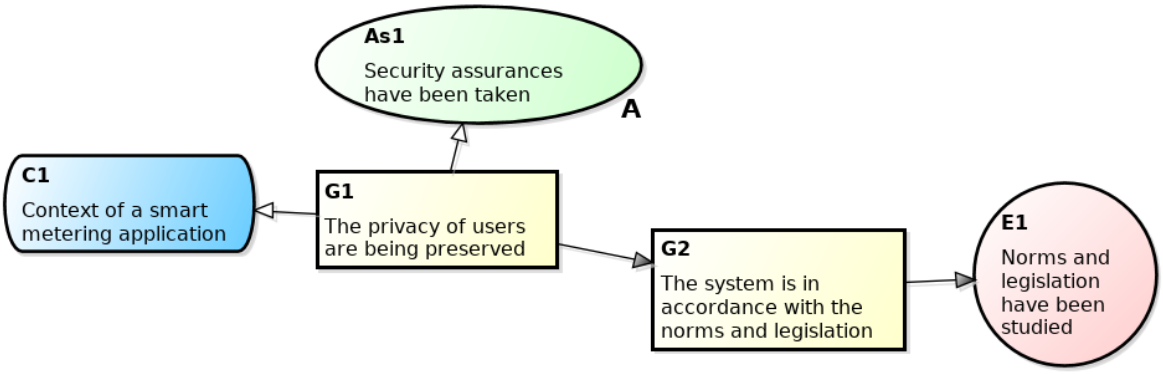


Figure 4.3: First iteration of the construction of the *GSN* representation for the privacy case of a smart metering application.

are more sensitive than aggregate values (such as the total consumption in a region with N consumers, or the total consumption of a house in the end of a billing period), the privacy techniques need to focus on hiding individual values.

Many utilities that use metering data were listed in Table 4.2. From this list, using a risk binary scale, “load forecasting for individual consumers”, “individual data analytics” and “demand-based rates” were considered as privacy threats that needed to be solved. This classification is based on the usage of individual values. The check if a utility uses individual values (and thus if it is a privacy risk) can be found in the references.

Unfortunately, “demand-based rates” may be a legitimate utility but regarded as a privacy risk. Power providers would charge based on the instantaneous power, but to do that, it is

Table 4.2: List of metering data utilities.

Feature / Benefit	**Privacy Risk?**
Billing optimization [13]	No
Load monitoring and management for specific groups or regions [13]	No
Energy theft/losses detection [5]	No
Load forecasting for specific groups or regions [49]	No
Load forecasting for individual consumers [49]	Yes
Time-based rates (*e.g.*, different prices based on time of day and season) [13]	No
Demand-based rates (*e.g.* different prices based on demand levels) [81]	Yes
Individual data analytics (*e.g. NIALM* and marketers) [11; 73]	Yes
In-home feedback tools: estimated bills, device profiles etc [105]	No

necessary to know the individual measurements.

4.3.1 Adversary Model

In the smart metering privacy literature, it is common to consider the power provider as an adversary (honest, but curious) and, transitively, the data exposure to third parties becomes a threat to privacy too. Therefore, it is resonable to consider the power provider as an adversary and to apply the privacy techniques on the consumers' side [30; 41; 43; 8; 53; 65; 107; 8; 53; 65; 107]. Figure 4.4 presents the second iteration of the construction of the *GSN* representation for this case study. The diagram now considers the adversary model, *i.e.*, the assumption that the sensor is not an adversary and the strategy to provide mitigations considering that the service is an adversary.

4.4 Privacy Techniques

With the considerable amount of privacy-preserving techniques that can be adopted to ensure privacy in smart metering, the need to better understand and compare these solutions rise. In general, there are techniques based on homomorphic encryption [30; 41; 43], the techniques that use rechargeable batteries [8; 53; 65; 107], and the ones that make use of noise addition

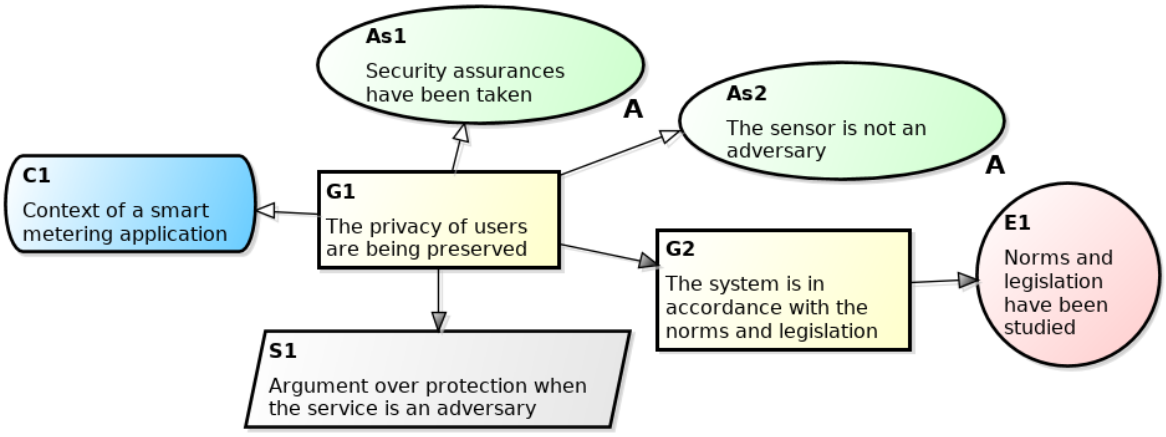


Figure 4.4: Second iteration of the construction of the *GSN* representation for the privacy case of a smart metering application.

[99; 26; 47; 12]. Since the main privacy issues are derived from individual data, these privacy preserving approaches tend to reveal aggregate data and hide individual data.

To enable power providers to process smart meter measurements, while preventing them to access private data, cryptographic tools like homomorphic encryption may be used. With these approaches, before sending its measurement, each smart meter runs a cryptographic routine. The power provider receives encrypted measurements, but can still perform useful computations and output the correct aggregate values, like the total consumption of a consumer during a billing period or the total consumption in the region in an instant of time. Thus, some benefits of using smart metering are still provided, while the consumer privacy is maintained.

Techniques based on the usage of rechargeable batteries consist in using a battery between the smart meter and home appliances. Therefore, the disclosed information is the battery load profile, but not the daily activities and appliance usages of consumers.

Through noise addition techniques, individual measurements are masked by adding random numbers. This masking happens in a way that does not affect the outcome of the aggregating operations but hides the individual measurements.

The privacy techniques were evaluated [14], and it was concluded that noise addition stands out from the others. Despite the disadvantage in accuracy, it has a low complexity, scalability, meters' independence, low cost and low environmental impact (this one, when compared with rechargeable batteries approach). Therefore, the next paragraphs are focused

on the noise addition approach. Rechargeable batteries and homomorphic encryption techniques are presented in Appendix B.

4.4.1 Noise Addition

Noise addition is a privacy preserving technique to mask the data. Wang *et al.* [99] propose an approach to mask the data adding random numbers from a *Gaussian Mixture Models* (*GMM*), whereas Bohli *et al.* [26] and He *et al.* [47] propose approaches to mask the data using Gaussian noise. Noise addition is a promising and efficient technique, however, these mentioned approaches do not have formal models to calculate the amount of noise that should be added to guarantee desired privacy and utility levels. In fact, He *et al.* [47] argue that for real world system design, a proper trade-off between privacy protection and accuracy should be considered.

We propose that for every measurement, the smart meter reads the consumption and adds a random number [13]. Thus, after an aggregating operation (such as the calculation of the total consumption in a region or the total consumption of a consumer in the end of a billing period), the result will be:

$$\sum_{i=1}^{N} c_i \approx \sum_{i=1}^{N} (c_i + x_i)$$

where N is the total number of measurements, x_i is a random number generated from a probabilistic distribution and c_i is an individual consumption measurement.

The previous formalization can also be rewritten as follows:

$$\sum_{i=1}^{N} c_i = \sum_{i=1}^{N} (c_i + x_i) - e_o$$

where e_o is the obtained error by the addition of random numbers. Therefore, e_o is the sum of all added random values:

$$e_o = \sum_{i=1}^{N} x_i \, .$$

We developed many analytical models using probability theory for different distributions [11]. Here we consider the Laplace distribution. Let x_i be a random variable generated from this distribution. Its variance is $\sigma_x^2 = 2b^2$, where b is a scale parameter. Now, for a large N, the central limit theorem ensures that the obtained error for billing purpose follows a normal

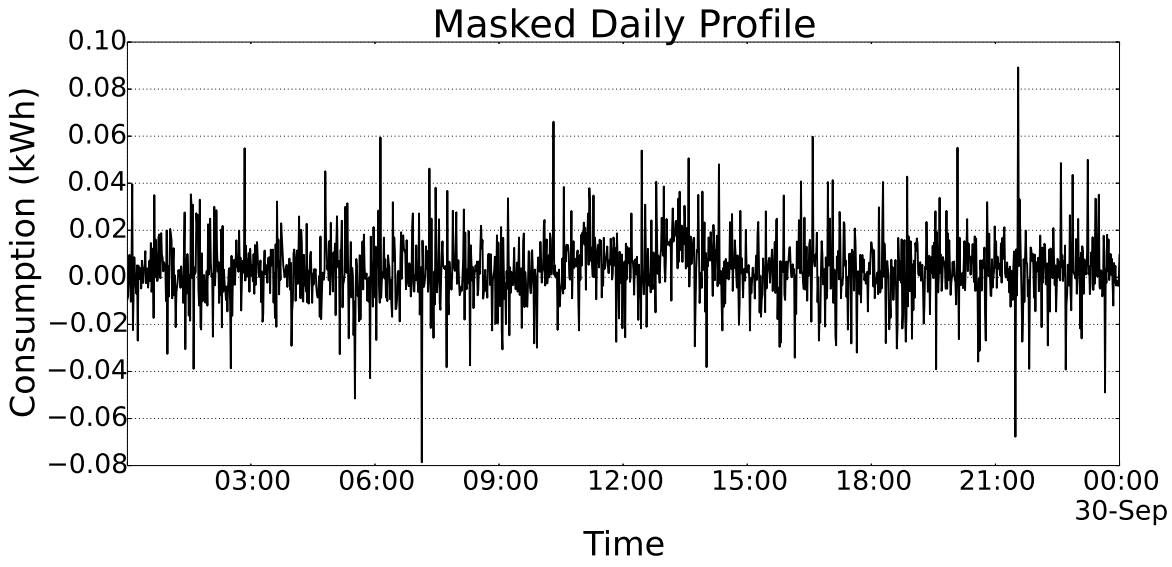


Figure 4.5: Residential masked daily profile with measurements at each 1 minute.

distribution with mean $\mu_{e_o} = 0$ and variance:

$$\sigma^2_{e_o} \;=\; N^2(\sigma^2_x \,/\, N) \;=\; N\sigma^2_x \;= 2Nb^2\,. \tag{4.1}$$

In other words, to have an obtained error between two accepted values (with high probability), we can use the following normal distribution:

$$e_o \;\sim\; N\left(0,\; 2Nb^2\right)\;.$$

As an example, Figure 4.2 shows a daily profile of a residential consumer. There are 16 appliances in this consumption profile. However, the appliance with highest wattage and easier to identify is the laundry dryer. Assuming the billing period as one month, the total consumption of this consumer during the billing period (one month of 31 days) is 131.978 kWh. Considering a maximal allowed error of 5% for billing purpose, we have 6.5989 kWh. Thus, the variance for a high probability (*e.g.*, 0.98) of not exceeding this value is $\sigma^2_{e_o} = 8.04626$. Isolating the scale parameter b from Equation 4.1, we have (for measurements at each 1 minute, $N = 44,640$):

$$b \;=\; \sqrt{\sigma^2_{e_o}/\,(2N)} \;=\; 0.0094934\,.$$

Figure 4.5 presents the daily profile of Figure 4.2 masked using this Laplacian noise. Considering that at the end of the month the power provider sums the informed masked

values by the consumer, it obtains a value of 133.7977 kWh. The real value is 131.978 kWh. The difference between these values is an error of 1.3788%, less than the maximum allowed error (5%). This error may be less if the consumer does not mask the measurements all the time. Also, if the meter firmware accumulates the sum of the added random numbers and sends that with the last measurement of the month, the error is zero.

The existence of errors in applications of electrical networks are often found nowadays (since these errors should be less than established limits). For example, in Brazil, the INMETRO (National Institute of Metrology, Standardization and Industrial Quality) establishes percentual error limits to measurements for billing purposes. For residential customers (defined as class B), the percentual relative error for active energy must stay between +/- 2%, while for industrial customers (defined as class A), this error must stay between +/-3%. For more information see the ordinance number 375 of September 27, 2011 [50].

A key feature of the approach is the possibility to allow the consumer and the power provider to negotiate the privacy and utility levels by just changing the allowed error/noise magnitude. Moreover, since this masking approach considers only the data that is disclosed to the power provider, it does not affect the customer's ability of analyzing his own real consumption profile inside his home and identifying appliance usage [105].

We claim that the proposed approach is lightweight [13] because in order to preserve privacy the approach just generates a random number. Also, it is possible to provide differential privacy [37] guarantees for appliance usages, making them indistinguishable in a consumption profile [12].

Differential Privacy for Appliances

Dwork [37] proposed the notion of differential privacy for general datasets. A mechanism of obfuscation is differentially private if its outcome is not significantly affected by the removal or addition of a single dataset participant. Here, we instantiate this notion for datasets of energy consumption profiles.

Since the privacy is related with appliance usage (*i.e.*, the consumer behavior), we aim to achieve differential privacy for appliances in metering data without affecting the aggregate data. Thus, appliances are participants in consumption profiles and an adversary learns approximately the same information about any individual appliance, regardless of its presence

or absence in the original profile.

Considering that a consumption profile is a set of appliances, we say profiles $P1$ and $P2$ differ in at most one appliance if one is a proper subset of the other and the larger dataset profile contains just one additional appliance.

Definition 3 *An obfuscation mechanism K gives ϵ-differential privacy if for all profiles $P1$ and $P2$ differing in at most one appliance, and all $S \subseteq Range(K)$,*

$$Pr[K(P1) \in S] \leq exp(\epsilon) \times Pr[K(P2) \in S]$$

where the probability is taken over the randomness of K.

The ϵ value is the privacy metric and for better privacy, a small value is desirable. A mechanism K satisfying this definition addresses concerns that any appliance might have about the leakage of its information: even if the appliance has been removed from the dataset, no outputs (and thus consequences of outputs) would become significantly more or less likely.

Differential privacy is achieved by the addition of noise whose magnitude is a function of the largest change a single appliance could have on the output profile; this quantity is referred as the sensitivity of the function.

Definition 4 *For $f : P \rightarrow R^k$, the sensitivity of f is*

$$\Delta f = \max_{P1,P2} \parallel f(P1) - f(P2) \parallel_1$$

for all $P1$, $P2$ differing in at most one appliance.

In particular, when $k = 1$ the sensitivity of f is the maximum difference in the values that the function f may take on a pair of profiles that differ in only one appliance.

The privacy mechanism, denoted K for a query function f, computes $f(X)$ and adds noise with a Laplace distribution with mean $\mu = 0$ and scale parameter b:

$$b = \Delta f/\epsilon \ \therefore \ \epsilon = \Delta f/b \, . \tag{4.2}$$

The variance of the noise distribution is $2b^2$. On query function f the privacy mechanism K responds with

$$f(X) + (Lap(\Delta f/\epsilon))^k$$

adding noise with distribution $Lap(\Delta f/\epsilon)$ independently to each of the k components of $f(X)$. Note that decreasing ϵ, a publicly known parameter, flattens out the $Lap(\Delta f/\epsilon)$ curve, yielding larger expected noise magnitude.

Theorem 1 *For $f : P \rightarrow R^k$, the mechanism K that adds independently noise with distribution $Lap(\Delta f/\epsilon)$ gives ϵ-differential privacy.*

The proof of Theorem 1 is straightforward and Dwork gives this proof for general datasets in [38]. This theorem describes a relationship between Δf, b, and the differential privacy. To achieve ϵ-differential privacy, one must choose $b \geq \Delta f/\epsilon$. Given a sufficiently small ϵ, differential privacy limits the ability of an adversary to identify an appliance in the consumption profile.

Note that it is not hard for a user (or an automated mechanism) to identify the wattage of an appliance purchased. That said, the global sensitivity of the profile from Figure 4.2 is the maximum variation of the appliance with highest wattage (laundry dryer):

$$\Delta f = 0.01022 \, .$$

We can conclude that, from Equation 4.2, using an utility requirement of 5%, the achieved privacy level in this example is:

$$\epsilon = \frac{\Delta f}{b} = 1.077 \, .$$

In the literature of differential privacy, some researchers argue that the value of ϵ may be relative because for the same value of ϵ, the probability of identifying a participant is dependent on the context. Lee et al. [59] propose a technique to, in accordance with a context, find a suitable value of ϵ:

$$\epsilon = \frac{\Delta f}{\Delta v} \ln \frac{(n-1)p}{1-p} \tag{4.3}$$

where Δv is the largest distance of possible values, p is the probability of identifying the presence of an individual and n is the number of individuals in the data set. In the previous example for the consumption profile, we have $\Delta v = 0.0178$ (highest value presented in the profile) and $n = 16$ (number of appliances). Thus, considering that the probability of identifying the usage of an appliance is $1/3$, it is suggested to have:

$$\epsilon \leq \frac{0.01022}{0.0178} \ln \frac{15 \cdot 3}{3 \cdot 2} = 1.1568 \, .$$

Since in our example the obtained value of ϵ is 1.077, we can conclude that an attacker has a probability of identifying the laundry dryer less than $1/3$. In other words, on average, he/she has to try more than three moments to be able to guess one moment of usage of the laundry dryer (and maybe he/she does not know that).

Rearranging Equation 4.3, we can determine the probability:

$$p = \frac{1}{1 + (n-1) \cdot exp(-\frac{\epsilon \Delta v}{\Delta f})} \, . \tag{4.4}$$

In Table 4.3 we present the wattages and achieved privacy levels for some appliances of the profile from the previous example. The wattages were obtained from the Tracebase dataset [84], the ϵ values were obtained from Equation 4.2, and the probability values of identifying appliances were obtained from Equation 4.4.

As discussed before, lower values of ϵ or p imply better privacy. Naturally, the consumer behavior and privacy are correlated with the appliance usage and some of these information may be more sensitive than others.

As we can see in Table 4.3, the appliance wattage is very correlated with the privacy level. However, some appliances with higher wattage have better privacy than some appliances with lower wattage. This is because the accumulated consumption (kWh) is not only dependent on the wattage, but also on the time of use of the appliance.

It is known that less measurements implies higher privacy. Taking an extreme example, if a consumer sends to the power provider only one measurement with the total consumption at the end of the billing period, the achieved privacy is much better than sending measurements at each 1 minute. However, when the number of measurements in a time period is large, our approach generates noise to hide each individual measurement, and the achieved privacy could be as higher as sending only one measurement with the total consumption at the end of the billing period.

Using the proposed model, it is also possible to calculate the achieved utility level for a given privacy requirement. From Equation (4.1) and (4.2) we have:

$$\sigma^2_{e_o} = \frac{2 \cdot N \cdot \Delta f^2}{\epsilon^2} \, . \tag{4.5}$$

Therefore, with this variation, the maximum obtained error (utility level) is easily calculated using the inverse cumulative density function (quantiles) of the normal distribution.

Table 4.3: Privacy levels achieved for each appliance from the profile of Fig. 4.2. Lower values of ϵ and p imply better privacy.

Appliance	**Max. Wattage**	**Privacy (ϵ)**	**Probability (p)**
Charger Smartphone	∼6	0.001	0.0626
Router	∼9	0.003	0.0628
Playstation 3	∼130	0.037	0.0663
TV LCD	∼140	0.075	0.0706
PC Desktop	∼300	0.077	0.0708
Refrigerator	∼1000	0.099	0.0733
Printer	∼600	0.127	0.0767
Water Fountain	∼260	0.143	0.0787
Toaster	∼700	0.257	0.0944
Cooking Stove	∼900	0.276	0.0973
Vacuum Cleaner	∼1100	0.336	0.1068
Iron	∼1500	0.347	0.1087
Coffee Maker	∼1300	0.353	0.1097
Microwave Oven	∼1400	0.457	0.1287
Washing Machine	∼3000	0.903	0.2431
Laundry Dryer	∼3500	1.077	0.3031

Example of Load Monitoring in a Region

If each consumer masks his data based on the billing period, the power provider may obtain accurate values for billing. However, to obtain accurate values for load monitoring in a region, the number of consumers should be as many as possible because it is desired that the added noise should be unnoticed in the aggregated data used for load monitoring.

The data used in the next example are measurements collected at each 30 minutes from real residential consumers (anonymised) from Ireland (*Commission for Energy Regulation – CER*) [32]. Suppose that the power provider wants to compute the total consumption in a region with many consumers through time for load monitoring (*e.g.*, find peak times, leak detection, load forecasting and many other applications). Using a billing period of 1 month

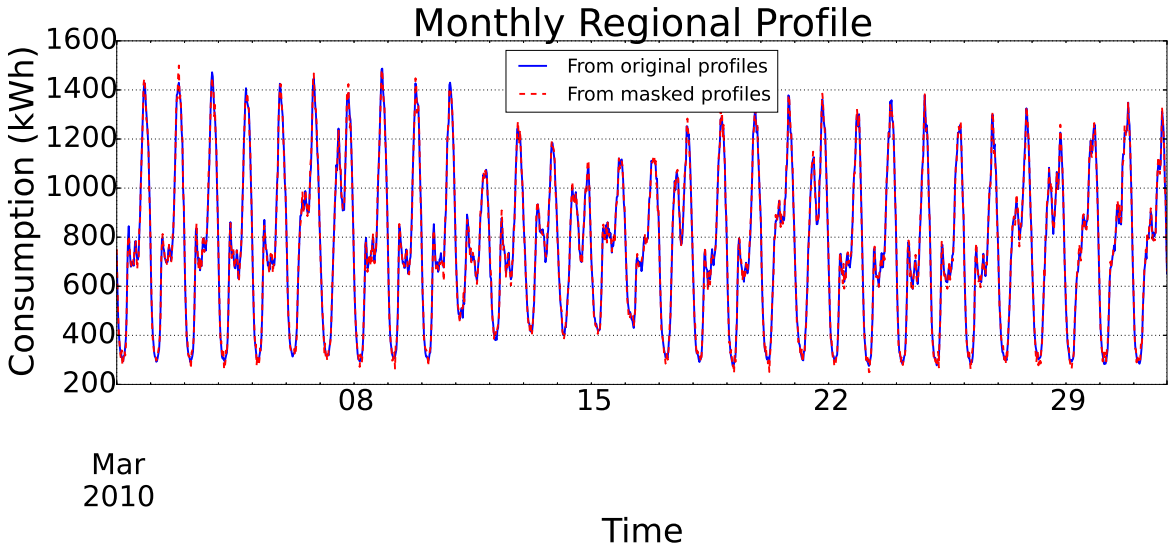


Figure 4.6: Regional profile using original data (blue solid) versus using masked data (red dashed) with measurements of 30 min. The profiles are very similar.

and measurements at each 30 minutes, it has $N = 1488$ measurements for each consumer during a month with 31 days (March). So, in this experiment, we consider a region with 1488 consumers (thus, the data forms a square matrix). Figure 4.6 shows the regional profile during March obtained from original consumer profiles versus the regional profile obtained from masked profiles.

As depicted in Figure 4.6, visually there is no difference between the two profiles. However, the obtained errors depend on the population behavior. For example, in a high consumption period the obtained error has a different proportion from the obtained error in a low consumption period. The large errors in Figure 4.6 were obtained in periods of low consumption (*e.g.*, during the night), because while consuming less, consumers are still masking their data using a noise level based on the billing period. The blue dashed line in Figure 4.7 presents the obtained errors through time for this scenario (Figure 4.6). These errors may be lower if not all consumers decide to mask their data all the time.

The possibility of having consumers deciding when to send masked (vs. original) measurements is in accordance with the privacy definition mentioned by Stallings *et al.* [92] and can be implemented through a privacy switch (*i.e.*, enable or disable the masking). As not all consumers are masking the measurements all the time (making the obtained error to be lower), this implies in a utility improvement.

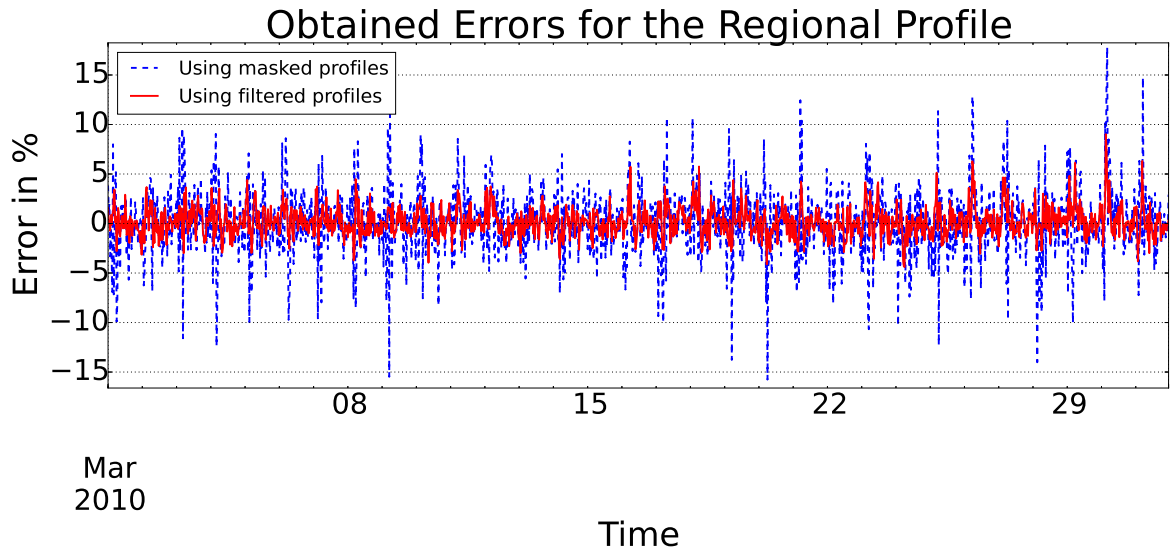


Figure 4.7: Obtained errors for the regional profile using masked profiles versus using filtered profiles. Using filtered profiles implies in better accuracy for load monitoring.

4.4.2 Performance Evaluation of Privacy Techniques

In order to analyze and compare solutions, we describe the performance aspects that are considered in our studies. This analysis is important because, for example, solutions might excel at low computational complexity but their installation and usage has high costs and harms the environment. Thus, the need to consider different aspects of performance. In this section, we present an experiment that aims to compare the response time (which is related to computational complexity) of the approaches. We also discuss other aspects, such as: scalability, meters' independence, cost, environmental impact and accuracy. Table 4.4 presents a summary and comparison of the discussed approaches.

Computational Complexity

Low complexity is desired mainly because most of the deployed smart meters are low-cost microcontrollers and with limited computational resources. Through the analysis of the solutions' running time growth order, we have the complexities presented in Table 4.5. We consider that, regarding computational complexity, the noise addition and rechargeable batteries approaches stand in relation to the homomorphic encryption approaches.

Although estimation through asymptotic complexity is a good way to estimate computational complexity, we claim that experimental analysis is essential to present concrete results

Table 4.4: Comparison of the privacy perserving approaches in smart metering.

	NA	**RB**	**HE**
Low complexity	✓	✓	
Scalability	✓	✓	
Meters' independence	✓	✓	
Low cost	✓		✓
Low environmental impact	✓		✓
Accuracy		✓	✓

Legend:

- **NA**: Noise Addition [11];
- **RB**: Rechargeable Batteries [65];
- **HE**: Homomorphic Encryption [30; 43; 41];

when comparing different proposals. Therefore, we implemented simulators[6] in the *C* programming language. These simulators make use of a few functions from the *libgmp*[7], *libpaillier*[8] and *libcrypto*[9] libraries to implement algorithms that mimic the protocols described in this chapter and in Appendix B. The simulators were executed in a machine with 1.6 GHz Intel Core i5 processor, 6 GB of RAM memory and the Ubuntu 14.04 operating system.

In our simulations, using different configuration scenarios (number of meters, ranging from 1 to 200) to calculate the total consumption in the region, we measured the processing time of each meter (Figure 4.8) and the aggregator (Figure 4.9).

Each scenario was executed 10 times and the average values were considered. This amount of repetitions were enough to get precise average values. Due to the very low obtained variations, the confidence intervals are being omitted here, except for the aggregation in the approach proposed by Busom *et al.* [30], which needs a trial and error mechanism to solve the discrete logarithm problem. The confidence intervals in these cases are of 95%.

[6]The source codes can be found at our GitHub repository (`https://git.lsd.ufcg.edu.br/pedroysb/privacy-performance-smart-metering/tree/master`).

[7]*libgmp*: `https://gmplib.org`

[8]*libpaillier*: `http://acsc.cs.utexas.edu/libpaillier`

[9]*libcrypto*: `https://www.openssl.org/docs/manmaster/crypto/crypto.html`

Table 4.5: Complexity analysis for different privacy preserving approaches in smart metering.

	NA		**RB**		**MEE**		**PESS**		**MPE**	
Operation	SM	AG	SM	AG	SM	AG	SM	AG	SM	AG
Encryption	-	-	-	-	$O(1)$	-	$O(N)$	-	$O(1)$	-
Decryption	-	-	-	-	-	$O(M)$	$O(1)$	-	$O(1)$	-
Transmission	$O(1)$	$O(N)$	$O(1)$	$O(N)$	$O(1)$	$O(N)$	$O(N)$	$O(N^2)$	$O(N)$	-
Sum	$O(1)$	$O(N)$	$O(1)$	$O(N)$	-	-	$O(1)$	$O(N)$	$O(N)$	-
Product	-	-	-	-	-	$O(N)$	-	$O(N^2)$	$O(N)$	-

Legend:

- **NA**: Noise Addition [11];
- **RB**: Rechargeable Batteries [65];
- **MEE**: Modified ElGamal Encryption [30];
- **PESS**: Paillier Encryption and Secret Sharing [43];
- **MPE**: Modified Paillier Encryption [41];
- **SM**: Smart Meter;
- **AG**: Aggregator;
- **N**: Number of consumption measurements;
- **M**: Total (aggregate) consumption value.

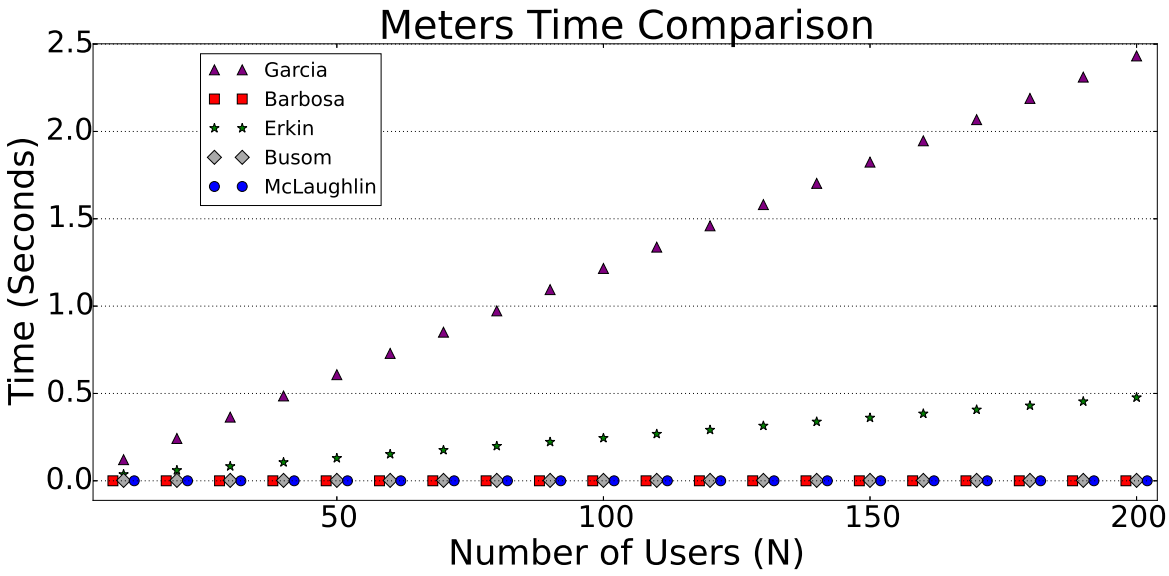


Figure 4.8: Processing time of smart meters in 5 different privacy preserving approaches.

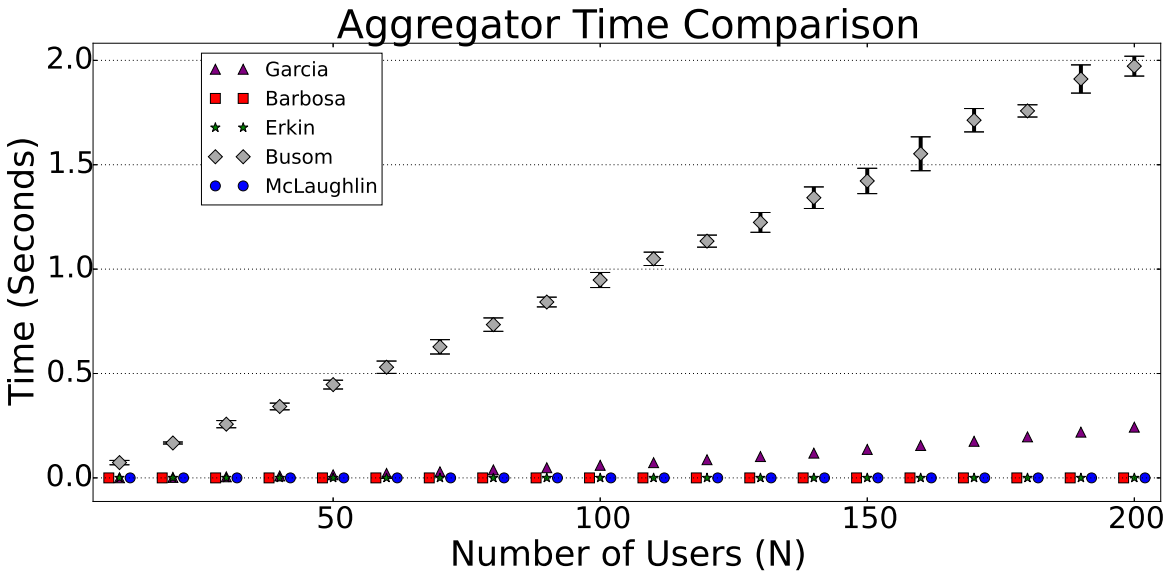


Figure 4.9: Processing time of the aggregator in 5 different privacy preserving approaches.

From these measurements, we conclude that the noise addition approach and the one that uses rechargeable batteries presented very low response times, whereas the homomorphic encryption approaches presented considerable delays. It is also important to note that there are many message exchanges in the homomorphic encryption approaches, but we are not considering possible network delays in our experiments.

Cost and Environmental Impact

As mentioned, usage of rechargeable batteries can help to diminish many privacy issues. Nevertheless, it is hard to ignore the environmental effects and the costs of using batteries. Mclaughlin *et al*. [65] stipulate that a lead-acid battery of 50 Ah which operates at 12 V may cost approximately $100, and to achieve a typical residential nominal voltage of 120 V it is required 10 of such batteries (aprox. $1,000). The lifetime of each battery is approximately two years. Therefore, these solutions can not be considered low cost and cause a high environmental impact. Noise addition and homomorphic encryption approaches do not have these limitations.

Meters' Independence and Scalability

In order to exchange randomly generated numbers, keys and secret shares, the homomorphic encryption protocols require communication between meters and/or the power provider. Some solutions even require a large distribution of keys and certification. For this reason, in these approaches, the meters are not independent. This overhead can be the bottleneck of the smart metering system. Hence, actually optimizing communication at the meters is a hard task that has not been fully addressed in homomorphic encryption approaches. Additionally, if one meter fails in any message exchange, the aggregation may become impossible because it requires computations using all distributed keys or secret shares. Therefore, these solutions may raise scalability issues when used in an area with a large number of meters. Noise addition and rechargeable batteries approaches do not have these limitations.

Accuracy

As mentioned, noise addition masks the data and introduces some error in an aggregation operation. This error is controlled and usually smaller than an acceptable value. Also, allow-

ing the enabling or disabling of masking would make errors smaller in both dimensions: for billing, since a consumer would not mask all the time, and for load monitoring in a region, since not all consumers would mask in an instant of time. However, the noise addition privacy preserving approach is still not considered one hundred percent accurate. In the case of Section 4.4.1, the use of noise will introduce errors in the billing reports. These errors tend to be cancelled over time, but there is a probability that a higher value can occur. Rechargeable batteries and homomorphic encryption approaches do not have this limitation.

4.4.3 Discussion

We reviewed five solutions, one based on the use of noise addition, other based on the use of rechargeable batteries and three others based on homomorphic encryption schemes. We evaluated these approaches considering the main needed performance aspects and conclude that each solution has its advantages and disadvantages, but the noise addition stands out with relation to the others. Therefore, we choose to implement the noise addition in the LiteMe system.

To validate the utility, many utilities or benefits that use metering data were listed and evaluated to check whether they are still supported when using masked data. Since the noise addition approach preserves the aggregated values, the procedure to check if a feature is supported can be mapped as a checking if the feature uses only aggregated values or also uses individual values.

Table 4.6 presents the list from Table 4.2 but with an additional column (supported or not). The check if a utility uses only aggregated values (and thus if it is supported) can be found in the references. From the features that are not supported, “load forecasting for individual consumers” and “individual data analytics” can be considered as privacy threats that were solved. Since the approach masks the profile using noise addition, the individual demand or consumption levels are not original and the feature “demand-based rates” is not supported. Therefore, if the power provider wants to use this feature, it will see this as a limitation.

We implemented the noise addition approach as a privacy switch in the smart meter, as presented in Figure 4.10. There are many components in this sensor, however the most important ones are:

Table 4.6: Metering data utilities and the masking impact.

Feature / Benefit	**Privacy Risk?**	**Support**
Billing optimization [13]	No	✓
Load monitoring and management for specific groups or regions [13]	No	✓
Energy theft/losses detection [5]	No	✓
Load forecasting for specific groups or regions [49]	No	✓
Load forecasting for individual consumers [49]	Yes	
Time-based rates (*e.g.*, different prices based on time of day and season) [13]	No	✓
Demand-based rates (*e.g.* different prices based on demand levels) [81]	Yes	
Individual data analytics (*e.g. NIALM* and marketers) [11; 73]	Yes	
In-home feedback tools: estimated bills, device profiles etc [105]	No	✓

1. **ACS712**: Hall-Effect-Based Linear Current Sensor. Able to measure AC/DC current up to $30A$.

2. **Switching power supply**: Set as $220V$ input and $3.3V$ output - $1A$.

3. **CS5490**: IC of power management. Two channels used for current and voltage. $4kHz$ sampling rate and 24 bit resolution. Calculates powers (active, reactive and apparent), power factor, peak and RMS values.

4. **Voltage transformer**: Set as $220V$ primary and $12V$ secondary.

5. **ESP8266**: WiFi module with memory ($512KB$ flash storage) and processing capacity ($80MHz$ CPU), GPIOs and ADC converter (not being used).

6. **Privacy switch**: Allows enabling/disabling the data masking mechanism.

Our focus here is the privacy switch (number 6). This privacy implementation is in accordance with the privacy definition mentioned by Stallings *et al.* [92]. When the switch is in the disabled position, the meter sends to the remote server the original power measurements. If the switch is in the enabled position, the meter sends to the remote server masked measurements. The pseudorandom number generator used in this implementation uses $/dev/urandom$, which is considered cryptographically secure [1].

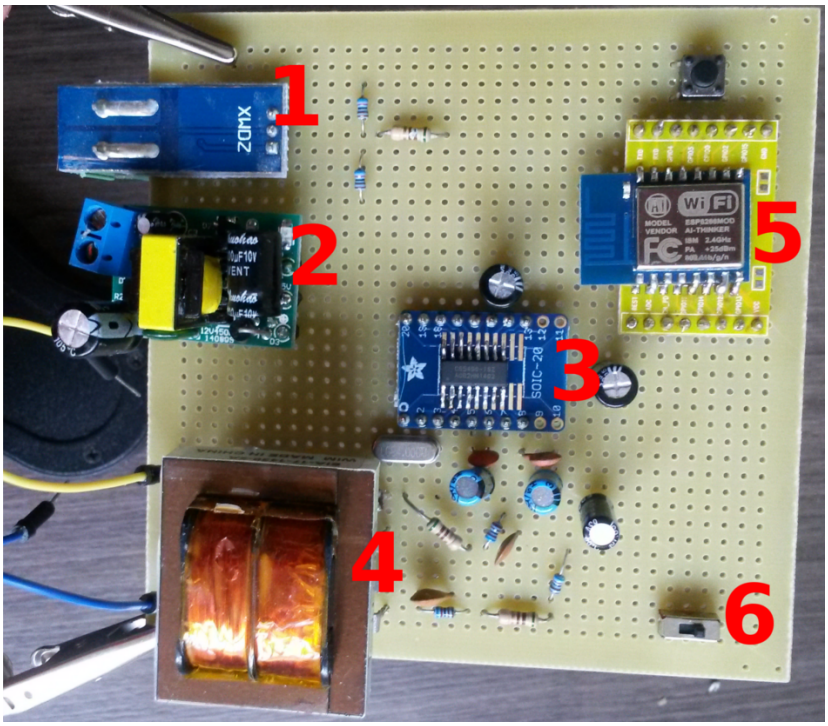


Figure 4.10: LiteMe sensor device.

The implementation of the noise addition technique consists in an evidence of privacy. Table 4.7 describes a sheet for this evidence $E2$. Since the evaluation and choice of the best evaluated techniques increase the confidence, we also have the evidence of privacy $E3$, regarding the comparison of the privacy techniques, as presented in Table 4.8. The implemented noise addition approach provides differential privacy for appliances, as described in Section 4.4.1. Therefore, there is another evidence of privacy, $E4$, as described in the sheet of Table 4.9.

Figure 4.11 presents the third iteration of the construction of the *GSN* representation for this case study. The diagram now contains the representation of the evidences $E2$, $E3$ and $E4$.

4.5 Potential Attacks

We evaluated three types of privacy attacks to the noise addition approach. The first is a filtering attack, based on the calculation of moving arithmetic means throughout the consumption profile. The second evaluated attack is a *Non-Intrusive Appliance Load Monitoring*

Table 4.7: Sheet for the evidence $E2$. Implementation of the privacy technique of noise addition.

E2	**Noise addition has been implemented**	**Status**: Done	**Review Date**: February 2015	**Weight**: 8
PbE Activity: *Evaluate and Apply Privacy Techniques*				
Driven by: G1 $\rightarrow$ S1 $\rightarrow$ G3; *In context of*: C1; *Assumptions*: As1, As2				
Description: A noise addition privacy-preserving scheme for smart metering has been implemented. We claim that the solution meets the needs of consumers (privacy) and power providers (utility). The modification in the communication procedure between a smart meter and the power provider is just the generation of a random number and the addition of this number to the measurement to be sent to the power provider.				
References: Section 4.4.				

Table 4.8: Sheet for the evidence $E3$. Performance evaluation of different privacy techniques.

E3	**Homomorphic encryption, rechargeable batteries and noise addition techniques have been evaluated**	**Status**: Done	**Review Date**: August 2015	**Weight**: 2
PbE Activity: *Evaluate and Apply Privacy Techniques*				
Driven by: G1 $\rightarrow$ S1 $\rightarrow$ G3; *In context of*: C1; *Assumptions*: As1, As2				
Description: Five solutions have been reviewed. One based on the use of noise addition, other based on the use of rechargeable batteries and three others based on homomorphic encryption schemes. These approaches have been evaluated considering the main needed performance aspects. Each solution has its advantages and disadvantages, but the noise addition stands in relation to the others. Due the simplicity and low complexity, the noise addition approach can be deployed in low-cost microcontrollers.				
References: Section 4.4.2, [14].				

Table 4.9: Sheet for the evidence $E4$. Differential privacy for appliance usages.

E4	**The approach provides differential privacy for appliances**	**Status**: Done	**Review Date**: August 2015	**Weight**: 8
PbE Activity: *Evaluate and Apply Privacy Techniques*				
Driven by: G1 → S1 → G3 → G4; *In context of*: C1; *Assumptions*: As1, As2				
Description: Since the inference of the consumer behavior may be a privacy concern and it is very correlated with appliance usage, the privacy levels achieved by appliances are measured through the state of the art in privacy model (*i.e.*, differential privacy). Differential privacy is a recent area of research that brings mathematical rigor to the problem of privacy-preserving analysis of data. Informally, the definition stipulates that any appliance should be indistinguishable in the disclosed masked consumption profile. Thus, an attacker cannot learn anything regarding appliances on the disclosed profile, even in the presence of any auxiliary information the attacker may have.				
References: Section 4.4, [12].				

(*NIALM*). Finally, the third attack is based on the hypothesis that a consumer tends to have a similar weekly behavior and, therefore, the attacker may collect the consumer's data and calculate an expected week for this consumer.

4.5.1 *Filtering Attack*

In this section we present a filtering attack. It is based on the calculation of moving arithmetic means throughout the profile. The algorithm for the filtering attack works as follows:

- Let T be a time series, which represents the masked profile and T_f a new series (the resulting filtered profile).
- The first P values from T_f will be equal to the first P values of T and the last P values from T_f will be equal to the last P values of T.
- The value with index $P + 1$ from T_f will be the mean of the values with indexes from 1 to $2P + 1$ from T. The value with index $P + 2$ will be the mean of the values with indexes from 2 to $2P + 2$ from T, and so on for all remaining values. This procedure creates moveable means to eliminate the high-frequency noise.

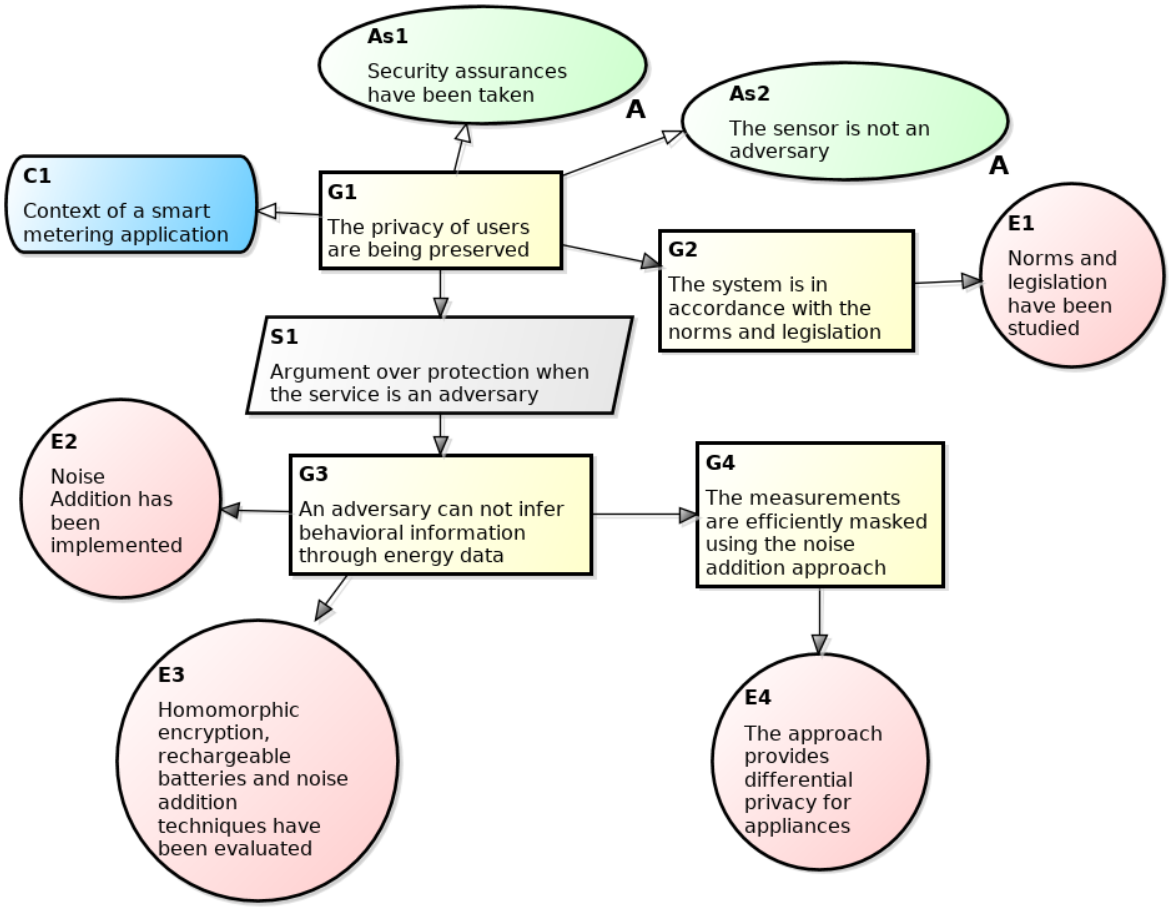


Figure 4.11: Third iteration of the construction of the *GSN* representation for the privacy case of a smart metering application.

To analyze the attack effectiveness, we made experiments to verify if the Pearson correlation coefficient[10] between the original profile and the masked profile is increased after the filtering. For example, using the masked profile from Figure 4.5, we verified if the added noise is removed and if the correlation with the original profile is increased. As presented in Table 4.10, we made this procedure using different P values. In this example, the configuration which presented best result was $P = 30$.

As observed in Table 4.10, the effectiveness of filtering is increased when we increment the P value because the high frequency noise is increasingly eliminated. In fact, the correlation between the masked profile and the original profile is 0.2135, whereas filtering with

[10]The Pearson correlation coefficient is a measure of the linear dependence between two variables. It has a value between +1 and -1 inclusive, where 1 is total positive linear correlation, 0 is no linear correlation, and -1 is total negative linear correlation.

Table 4.10: Filtering effect using different values of P. In this example, $P = 30$ is the best value.

P	Correlation	P	Correlation
0	0.2135	29	0.7363
2	0.4154	30	0.7368
4	0.5101	31	0.7362
8	0.6149	32	0.7351
16	0.6859	256	0.4164

$P = 30$ we obtain a correlation of 0.7368. In other words, we can consider that this attack has some effect on privacy. However, the attacker may not know which P value to choose and if the chosen one is greater than 30, the obtained correlation is dwindled due the filtering saturation. In this example, choosing P values greater than 30, the filter will eliminate not only the added noise, but also the original characteristics of the profile.

Experiments also showed that coarse-grained measurements (with longer time intervals) implies less filtering efficiency. This is because each measurement becomes less correlated with the adjacent measurements. For the same consumer of the previous example, the Figure 4.12 presents the mean of the best values of P (vertical axis) for different granularity of measurements (horizontal axis). As observed for this consumer, using measurement intervals of 36 minutes (or greater) will make filtering attacks ineffective because the best setting is using a P equals to 0 (*i.e*, no filtering). This means that with a granularity greater than or equals to 36 minutes, the original profile is more correlated with the masked profile than with the filtered profile. Therefore, after evaluating the experimental results of the filtering attack, we considered this attack as unsuccessful, and we have the evidence of privacy $E5$, as presented in the sheet of Table 4.11.

4.5.2 *NIALM*

We also performed some *NIALM* attacks and the results were evaluated. Figure 4.13 presents the workflow of this testing procedure. The noise addition approach is applied in the green task, whereas in the orange boxes, we applied the Improved *NIALM* using *load DIvision and*

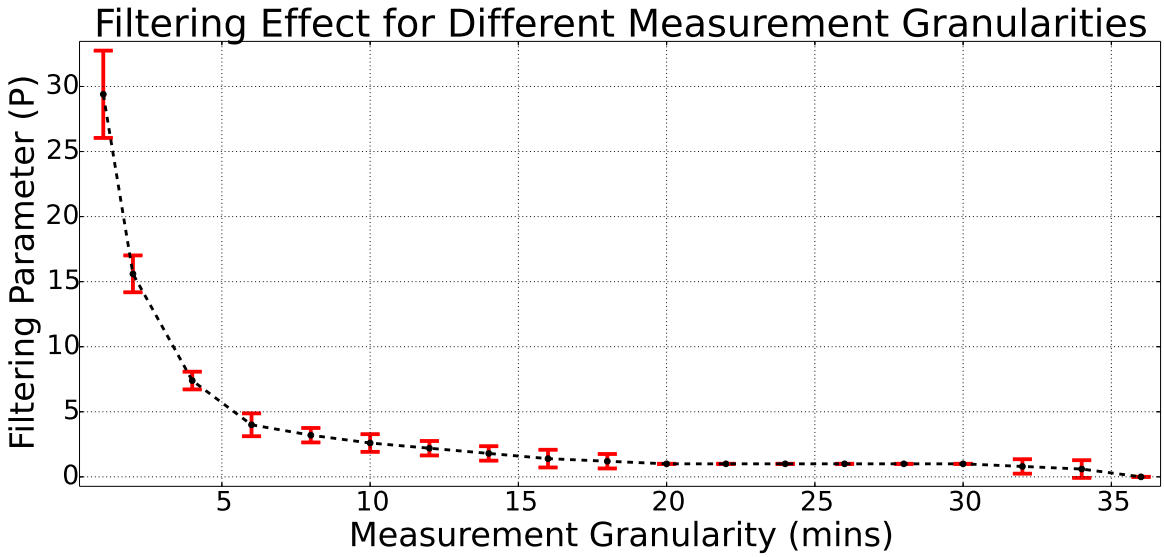


Figure 4.12: Filtering parameter (P) for different measurement granularities. The confidence intervals are of 95%.

Table 4.11: Sheet for the evidence $E5$. Resilience to the filtering attack.

E5	**The approach is resilient to the filtering attack**	**Status**: Done	**Review Date**: June 2015	**Weight**: 3
PbE Activity: *Evaluate Potential attacks*				
Driven by: G1 → S1 → G3 → G4; *In context of*: C1; *Assumptions*: As1, As2				
Description: Experiments were conducted and the filtering attack was considered as unsuccessful. The attacker may not know which parameters to choose and the filtering may be saturated. Choosing a wrong parameter value, the filter will eliminate not only the added noise, but also the original characteristics of the profile.				
References: Section 4.5.1, [12].				

Calibration (*INDIC*) algorithm [18].

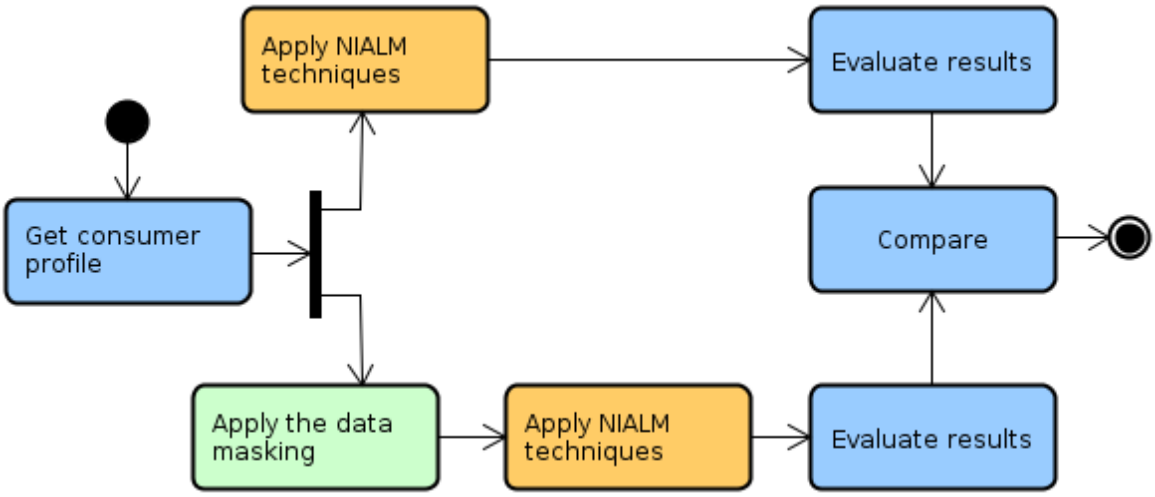


Figure 4.13: Workflow for privacy validation through *NIALM* attacks.

Figure 4.14 presents a weekly profile obtained from the popular *REDD* dataset [56]. In this profile, there are many appliances being used, such as a microwave and a refrigerator. Figure 4.15 presents the real disaggregated microwave profile. A microwave was chosen as an example because in this profile it is a very characteristic appliance and one of the most difficult to be hidden (since it has many peak variations).

After applying the *INDIC* algorithm to perform *NIALM* on the profile from Figure 4.14, the appliances were disaggregated. Figure 4.16 presents the predicted microwave profile. As observed, this predicted profile is very similar with the real microwave profile (at least the peak moments and levels are almost the same).

We applied our noise addition approach and using an allowed error of (e_a) of 5% to mask the aggregated profile from Figure 4.14, the masked profile from Figure 4.17 was obtained. Because negative demand values are impossible and *NIALM* tools can not work with that, we vertically shifted the profile to perform the *NIALM* attack. This modification was simply the addition of the absolute minimum value to every measurement of the profile, i.e., $\forall c \in P$, we made $c = c + |min(P)|$, where P is the profile and c is an individual measurement. For the *NIALM* algorithm, this procedure should be the same as considering that an appliance which demands $|min(P)|$ was always on usage.

After applying the same *INDIC* algorithm to perform the *NIALM*, we observed that the technique lost significantly its ability to detect appliance usages. Figure 4.18 presents the

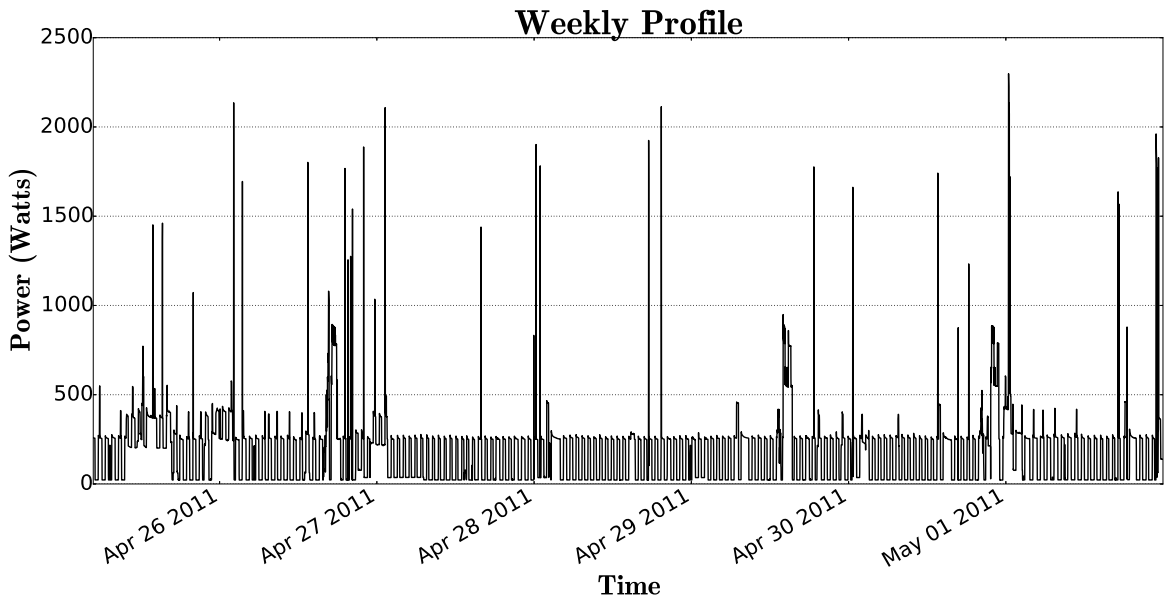


Figure 4.14: Example week obtained from the *REDD* dataset (measurements are at each 1 minute).

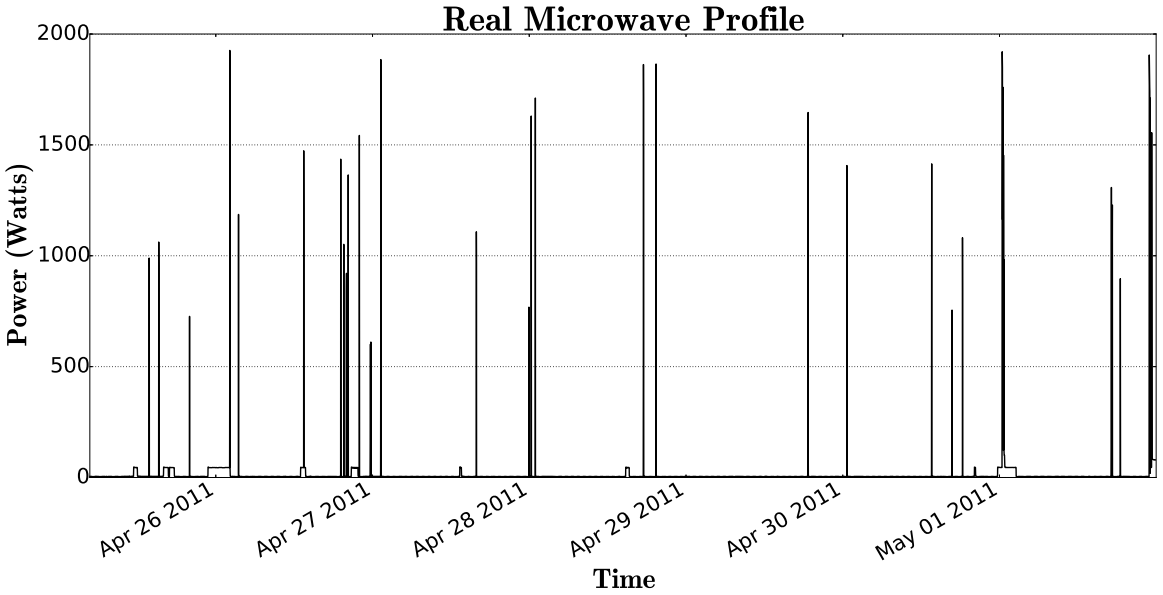


Figure 4.15: Real disaggregated microwave obtained from profile of Figure 4.14.

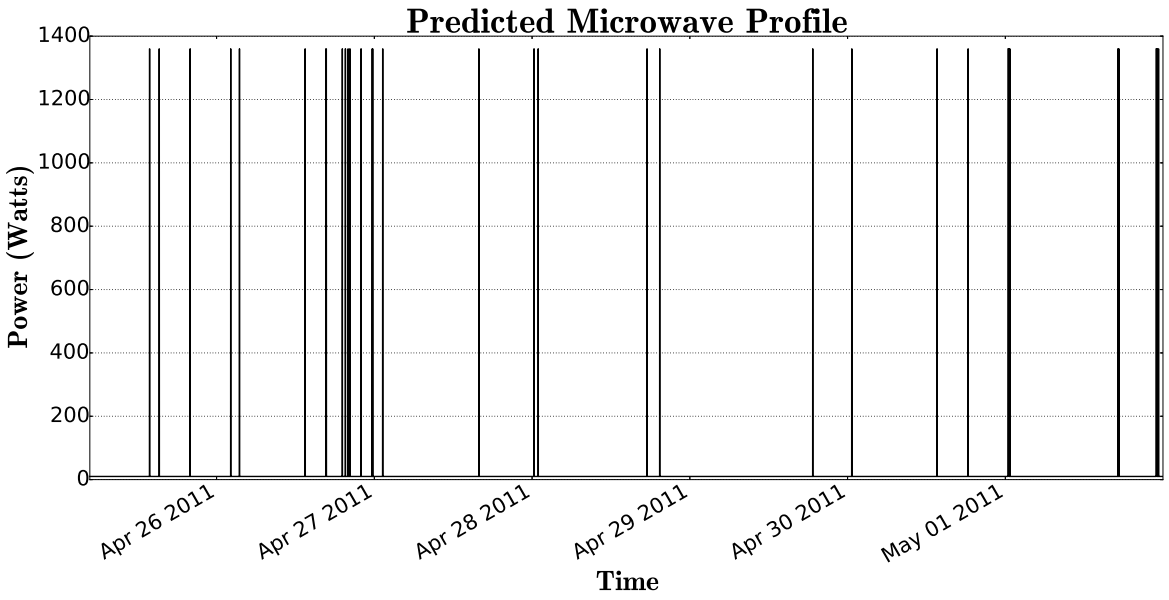


Figure 4.16: Disaggregated microwave predicted by *INDIC* over profile of Figure 4.14.

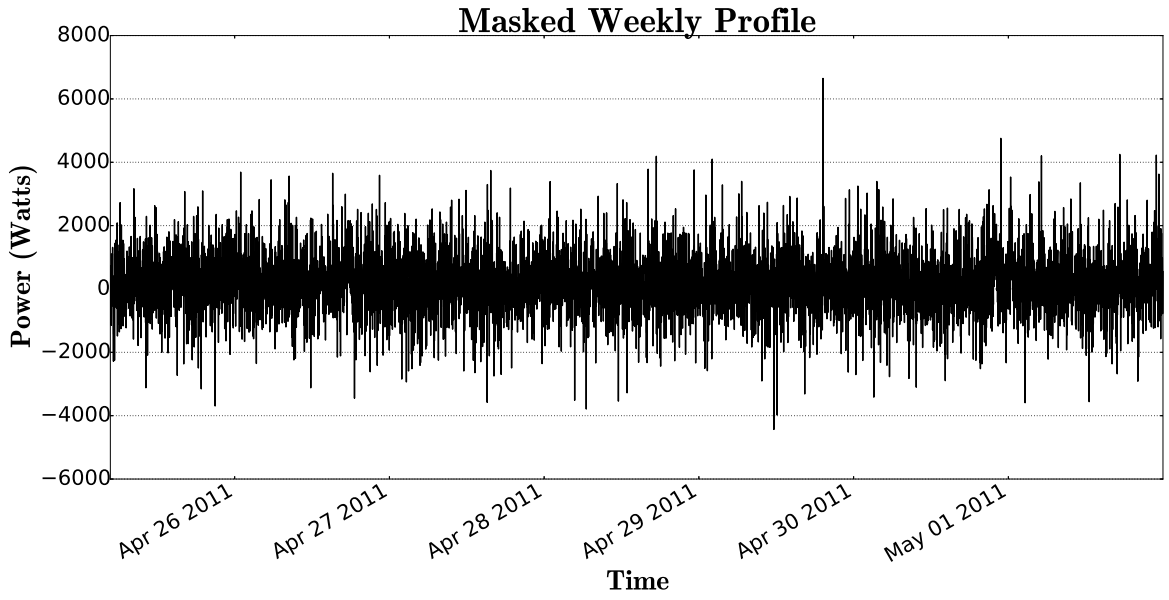


Figure 4.17: Aggregated profile of Figure 4.14 masked using e_a of 5%.

predicted microwave profile by the *INDIC* algorithm applied on the masked profile from Figure 4.17.

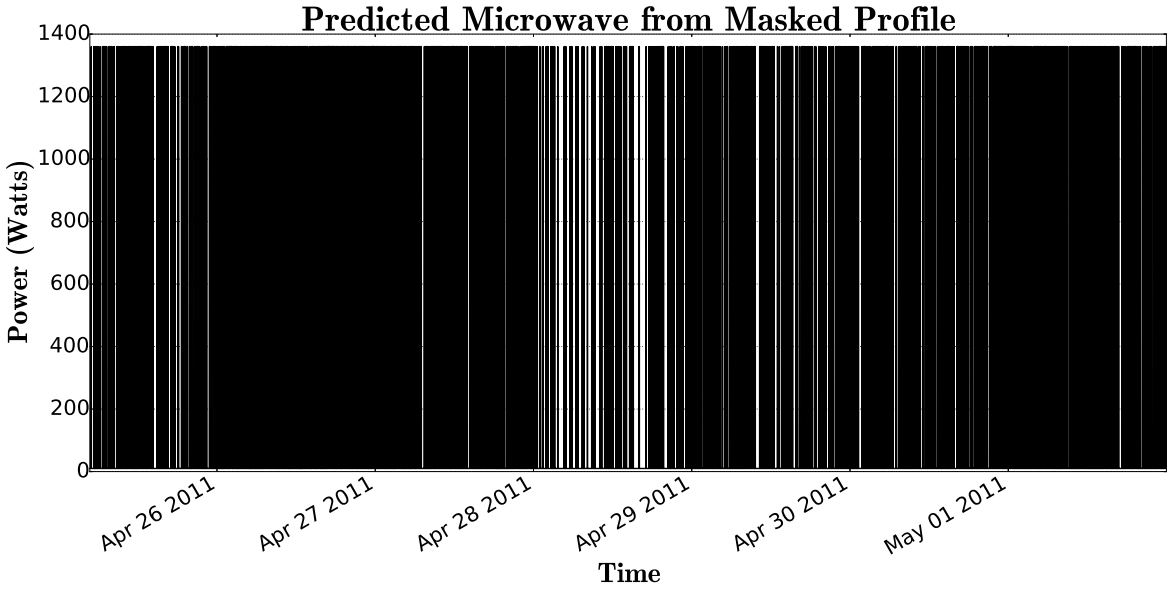


Figure 4.18: Disaggregated microwave predicted by *INDIC* over the masked profile of Figure 4.17.

To evaluate the potential of *NIALM* approaches in detecting appliance usages, Batra *et al.* [18] use two metrics: *Mean Normalized Error* (*MNE*) and *Root Mean Square Error* (*RMS*). Lower values of *MNE* and *RMS* imply in better accuracy of the *NIALM*. Using different configurations, we evaluated the masking approach for the two most significant appliances of the profile from Figure 4.14, as presented in Table 4.12 and Table 4.13. As observed, even using a high utility level and a low obfuscation (choosing a small e_a), the noise addition approach still affects significantly the *NIALM* potential to detect appliance usages (the *MNE* and *RMS* values are increased significantly). Therefore, we consider this attack as unsuccessful, and we have the evidence of privacy $E6$, as presented in the sheet of Table 4.14.

4.5.3 *Weekly Behavior Attack*

This attack is based on the hypothesis that the consumer tends to have a similar weekly behavior (expected week). An expected week is composed by the seven expected days (from Sunday to Saturday), and an expected day is composed by the averages of each instant of time from this specific day (*e.g.*, the attacker calculates the expected Sunday from all available

Table 4.12: *MNE* values by *INDIC* using different masking configurations (no masking, masking with allowed error of 1%, 2% and 5%).

Appliance	**No Masking**	$e_a = 1\%$	$e_a = 2\%$	$e_a = 5\%$
Microwave	70.89	225.69	2976.94	6768.89
Refrigerator	28.09	248.31	247.59	278.4

Table 4.13: *RMS* values by *INDIC* using different masking configurations (no masking, masking with allowed error of 1%, 2% and 5%).

Appliance	**No Masking**	$e_a = 1\%$	$e_a = 2\%$	$e_a = 5\%$
Microwave	54.16	193.47	649.45	495.72
Refrigerator	70.17	143.34	216.79	189.67

Table 4.14: Sheet for the evidence $E6$. Resilience to the *NIALM* attack.

E6	**The approach is resilient to the *NIALM* attack**	**Status**: Done	**Review Date**: June 2015	**Weight**: 3
PbE Activity: *Evaluate Potential attacks*				
Driven by: G1 $\rightarrow$ S1 $\rightarrow$ G3 $\rightarrow$ G4; *In context of*: C1; *Assumptions*: As1, As2				
Description: Experiments were conducted and it was concluded that even using a high utility level and a low obfuscation (choosing a small e_a), the noise addition approach still affects significantly the potential of *INDIC* (a *NIALM* algorithm) in detecting appliance usages.				
References: Section 4.5.2, [11].				

Number of available weeks to the attacker	**Average correlation between masked and real weeks**	**Average correlation between the expected and real weeks**
2	(0,499; 0,510)	(-0,046; -0,033)
4	(0,335; 0,344)	(-0,005; 0,004)
8	(0,369; 0,374)	(0,154; 0,166)
16	(0,326; 0,331)	(0,171; 0,180)
32	(0,436; 0,439)	(0,182; 0,189)
52	(0,432; 0,434)	(0,316; 0,323)

Table 4.15: Effect of the attack of the similar weekly behavior for a residential consumer.

Sundays). Using the expected week, the attacker can try to predict the consumer behavior in future weeks. The attack effect is dependent on the amount of data available to the attacker.

From the CER dataset (real residential consumers of Ireland), we selected a consumer who always repeats his/her behavior and considered in our experiments. The results are presented in Table 4.15. The average Pearson correlation between masked weeks and real weeks was compared versus the average Pearson correlation between the expected week and real weeks to analyze the attack effect. We used confidence intervals with significance levels of 95%.

As it can be seen in Table 4.15, when the number of weeks available to the attacker increases, the attack effect increases also, because the correlation between the expected week and real weeks is higher. But for our experiments, even choosing a consumer who repeats a behavior almost always and using a long period of observation (52 weeks or a full year), these correlations are less than the correlations between masked weeks and real weeks. It means that it is better to guess the consumer behavior from the own masked week than from the expected week. Therefore, we considered this attack as unsuccessful, and we have the evidence of privacy $E7$, as presented in the sheet of Table 4.16.

After performing the attacks and providing the evidences $E5$, $E6$ and $E7$, the *GSN* representation was expanded to the one presented in Figure 4.19.

Table 4.16: Sheet for the evidence $E7$. Resilience to the similar weekly behavior attack.

E7	**The approach is resilient to the similar weekly behavior attack**	**Status**: Done	**Review Date**: June 2015	**Weight**: 3
PbE Activity: *Evaluate Potential attacks*				
Driven by: G1 → S1 → G3 → G4; *In context of*: C1; *Assumptions*: As1, As2				
Description: Experiments were conducted and it was concluded that even for a consumer who repeats a behavior almost always and using a long period of observation (52 weeks or a full year), the correlations between the expected week and the real weeks are less than the correlations between masked weeks and real weeks. It means that it is better to guess the consumer behavior from the own masked week than from the expected week and therefore, the approach is considered as resilient to this attack.				
References: Section 4.5.3, [13].				

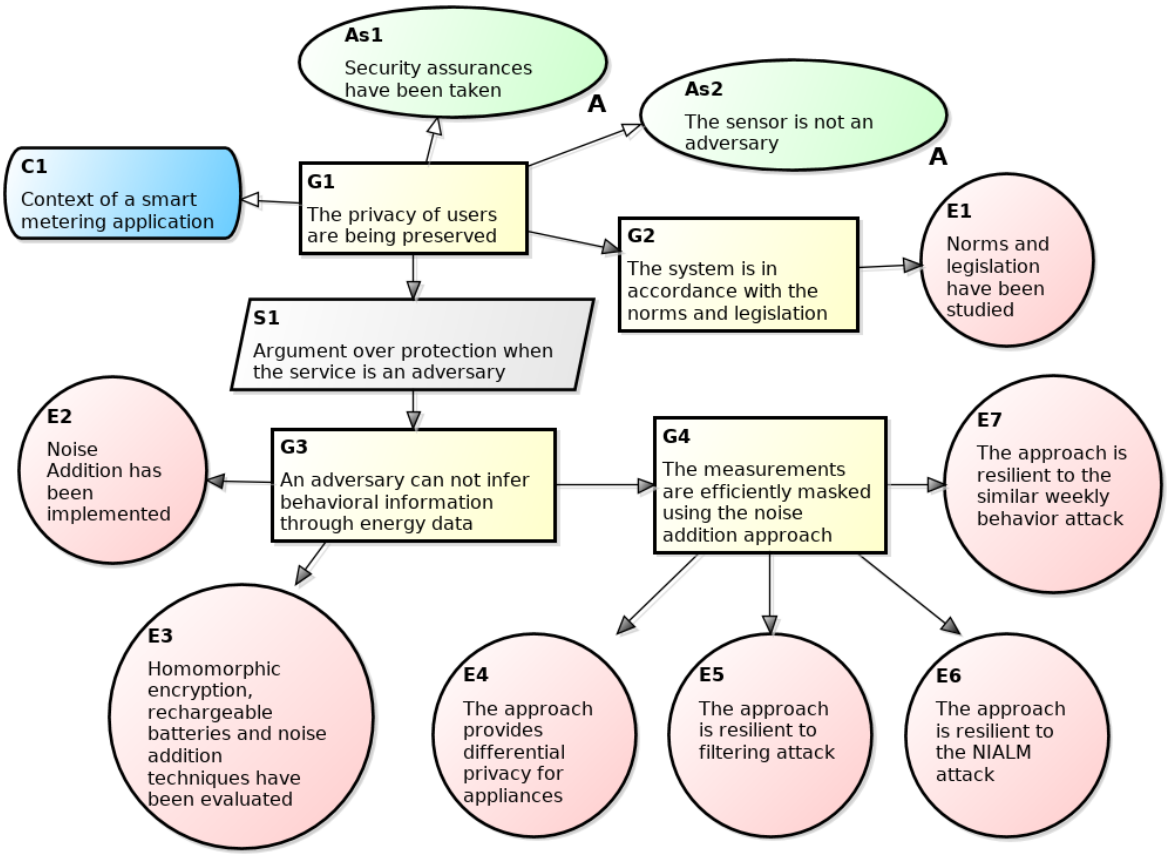


Figure 4.19: Fourth iteration of the construction of the *GSN* representation for the privacy case of a smart metering application.

Table 4.17: Checklist of the artifacts produced in the smart metering case study.

Activity	Artifact	Supplied?
Identify the Application Context and Data Formats	Engagement Report	✓
	Datasets	✓
Check Compliance with Norms and Legislations	Summary of Norms	✓
	Implementation of Norms	✓
	Compliance Proofs	✓
Identify Utilities and Risk Assessment	Utilities List	✓
	Perception Questionnaires	
	Perception Report	
	Privacy Concerns	✓
	Adversary Model	✓
	Privacy Policy	
Evaluate and Apply Privacy Techniques	Summary of Techniques	✓
	Techniques Report	✓
	Implementation of Techniques	✓
	New Utilities List	✓
Evaluate Potential Attacks	Summary of Attacks	✓
	Attack Scripts	✓
	Attacks Report	✓

4.6 Concluding Remarks

In this chapter, we validated the *Privacy by Evidence* (*PbE*) methodology through a case study of a smart metering application. Table 4.17 shows the checklist with the collected artifacts in this case study. We did not conduct any perception study in the risk assessment, however, in the next chapters, we present examples of such studies.

Figure 4.19 presented the *GSN* for the privacy case of a smart metering application considering the mentioned adversary model. This representation is still to grow, according to the normal software evolution and our proposed methodology. In this application context, assuming that security assurances have been taken and thay the sensor is not an adversary, there is an argument that the privacy is being preserved according to the provided evidences. Ev-

ery evidence has an identification to enable the traceability into the corresponding artifacts. There are other smart energy meter solutions such as OPower[11] and Eyedro[12], however, we claim that LiteMe is the first application to consider techniques to preserve users' privacy.

After validating the *Privacy by Evidence* (*PbE*) through a case study of a smart metering application, we conclude that this methodology can be regarded as an effective way to implement privacy protections mechanism. Seven privacy evidences were provided and the sum of their weights results in 28. Therefore, we positively support the research question $RQ1$. It is important to note that such mitigations must be an iterative work, and thus the stages in the methodology must happen in a constant cycle, once new risks can always be detected.

[11] Opower, opower.com

[12] Eyedro, eyedro.com

Chapter 5

Case Study II: Pulso Application

In this chapter, we present the second case study, the Pulso application. This application uses information regarding people tracking in public places, and therefore, may bring many privacy concerns. The Pulso application is a project developed by the *Federal University of Campina Grande* (*UFCG*) in partnership with the *Hewlett Packard*[1] (*HP*) company.

5.1 Context and Data Formats

Pulso is a mobile application that helps people to obtain information about public places. When it comes to the features, Pulso shows the estimated number of people in the place, the forecast of people that will be at this place in the next hours and also the "familiar faces" feature, that indicates the people a user may know, from the amount of times that two individuals were in the same place in the past. Figure 5.1 presents two screens of the Pulso application.

For the data collection, there are sensors (common access points, with modified firmwares) which retrieve probe requests from WiFi signals of mobile phones. The data flow is as follows:

1. The mobile phone sends broadcast packets, scanning for nearby WiFi networks;
2. A sensor catches these packets, which contain the media access control (MAC) addresses of the mobile phone, and sends them to a remote service/database;

[1] `http://www.hp.com`

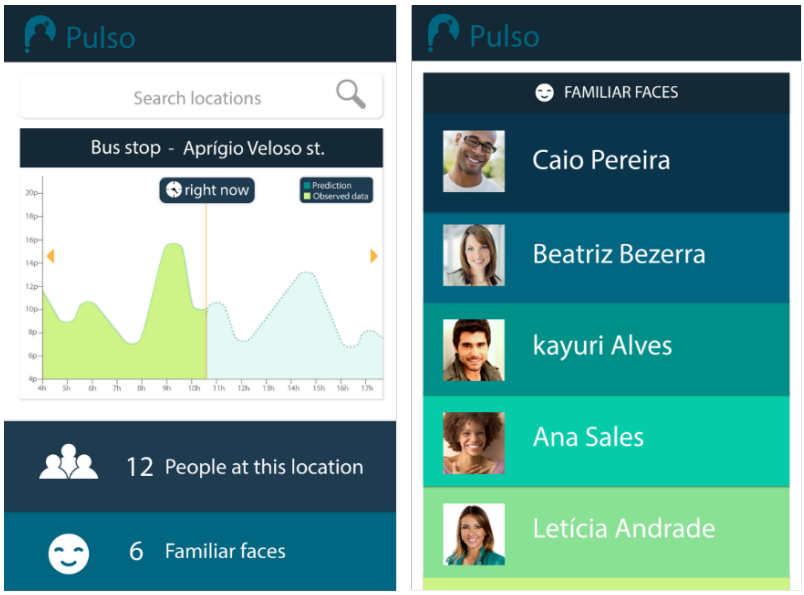


Figure 5.1: Two screens of the Pulso application.

3. The service processes the data and provides an Application Programming Interface (API). The Pulso application can retrieve the data by querying from this API and presenting it on the screen.

As it can be noticed, the Pulso application may generate many useful utilities but also some privacy concerns. If the wrong person has access to the data, sensitive information, like who was where, when, and with whom, is compromised. We analysed many samples of the collected data by the Pulso application. Basically, the collected data are tuples in the form $< sensor\_id, mac\_address, timestamp >$, where the $sensor\_id$ attribute also refers to the monitored place.

5.2 Norms and Legislation

In Brazil, we did not find any law dealing with the collection, storage and processing of people tracking data[2]. From these laws, the most important ones are the *Marco Civil da*

[2]Full report of laws: `https://drive.google.com/open?id=0By4xbln0HFnRNll5ckVJcWpPYjQ`

Internet[3] (law 12.965/2014) and the *Projeto de Lei de Proteção de Dados Pessoais*[4] (law project 5276/2016). These laws are regarding data collection through the usage of Internet connections. From our interpretation of the *Marco Civil da Internet*, its main purpose is to regulate the relationship between Internet providers and consumers. Pulso collects data of presence/absence of people in public places in an local manner, not from the Internet.

The *HP* company defines its own norms, according to their experience in the market and their ethical principles, when manipulating *Personally Identifiable Information - PII*. However, the stakeholders of the project defined the Pulso application as a product of the *UFCG*, and therefore, it does not need to follow the norms established by *HP*.

Despite the lack of norms and legislation to be applied in this application of IoT, the study and the summary of possibilities suggest a concern for privacy in this project, consisting in an evidence of privacy. Table 5.1 describes a sheet for this evidence $E1$.

Table 5.1: Sheet for the evidence $E1$. Norms and legislations for Pulso application.

E1	**Norms and legislations have been studied**	**Status**: Done	**Review Date**: November 2016	**Weight**: 1
PbE Activity: *Check Compliance with Norms and Legislation*				
Driven by: G1 $\rightarrow$ G2; *In context of*: C1; *Assumptions*: As1, As2, As3				
Description: We studied and summarized possible norms and legislations that could be applied for the Pulso application. In Brazil, we did not find any norm or law dealing with the collection, storage and processing of people tracking data that could affect the design of the application. There are norms regarding data collection through the usage of internet applications, but it is not the case of Pulso.				
References: Section 5.2.				

5.3 Utilities and Risk Assessment

For the risk assessment in the smart metering case study (Section 4.3), we did not conduct any privacy perception study with possible users of the application. However, our method-

[3] `http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2014/lei/l12965.htm`

[4] `http://www.camara.gov.br/proposicoesWeb/prop_mostrarintegra?codteor=1457459`

ology suggests this activity for better understanding the risks and to build an application in accordance with the users perception of privacy. Therefore, for the case study of the Pulso application, we focus on this activity. The following subsection were from an analysis carried out in collaboration with Lesandro Ponciano, a human computation researcher, about the perception study.

5.3.1 Privacy Perception Study

We organized the privacy perception study into five dimensions that represent perspectives in which it is possible to approach users privacy, which are: 1) people's general perceptions, concerns, and beliefs; 2) data collection; 3) information inference; 4) information use trade-offs; and 5) information exchange [80]. In order to assess people's privacy perceptions and concerns towards the Pulso system, we built a survey with a set of questions derived from these five dimensions. To consider the users perception in the design of the system consists in an evidence of privacy, as presented in Table 5.2.

Table 5.2: Sheet for the evidence $E2$. A privacy perception study of the Pulso application.

E2	**A privacy perception study has been conducted**	**Status**: Done	**Review Date**: November 2016	**Weight**: 5
PbE Activity: *Identify Utilities and Risk Assessment*				
Driven by: G1 $\rightarrow$ G2; *In context of*: C1; *Assumptions*: As1, As2, As3				
Description: For the Pulso application, we conducted a privacy perception study with a survey considering five dimensions that represent perspectives in which it is possible to approach users privacy in IoT systems, which are: 1) people's general perceptions, concerns, and beliefs; 2) data collection; 3) information inference; 4) information use trade-offs; and 5) information exchange. 58 people answered the survey and the results have been considered into the design of the Pulso application.				
References: Section 5.3.1, [80].				

The context of the system was explained to the respondents by presenting screens of the application and a video prototype showing how the system works – the used video prototype is available at `https://youtu.be/Sx_XORwpAbk`. The survey was designed by using Google Forms tools; the form is currently retired and permanently available at `https:`

`//goo.gl/forms/2HpdQmvg9uHCL86f1`. Before the application of the survey, a pilot test was conducted with 21 respondents in order to identify and to fix problems.

After validating the survey in the pilot test, answers from participants were collected in a transportation hub in Campina Grande, Brazil. For each respondent, it was presented a tablet with access to the survey. They answered the survey individually, without any external interference. To sum up, 58 people answered the survey, being 38 (66%) female and 20 (34%) male respondents, aging mostly between 25 and 34 years old (36, 65.5%). Most of them completed high school (27, 46.6%) or under-graduation (27, 46.6%). The detailed results obtained from the survey, considering the five defined dimensions, are presented below.

General perceptions, beliefs, and attitudes

Summary of the results: Most participants (35, 60.3%) believe that privacy is a right guaranteed by law. Also, most participants (36, 62.1%) tend to provide the information requested by the system if it explains its privacy policy.

Instructions to the designers: The general guideline drawn from these results is that, in order to encourage data provisioning, the Pulso system must have a privacy policy document which can be easily found by the user. To support users in the decision about the provision of data, the document must explain how the privacy issues are addressed. Also, because the respondents believe that privacy is a right guaranteed by law, legal issues must also be stated in the privacy policy document.

Data collection

Summary of the results: Most people (37, 63.8%) understand that when using the Pulso system, it will collect data about the locations where their mobiles were throughout the day. In addition, most people (46, 79.3%) feel from no concern to moderate concern about the collection of such data. Most people (43, 74.1%) would not know what to do to stop the Pulso system from collecting their personal data.

Instructions to the designers: Despite the understanding of which data is collected and the few concerns about such collection, people do not know how to control/interrupt the collection in case they want to do so. System designers must pay attention to the fact that, while using the system and perceiving privacy concerns, it is essential that users have clear ways to stop the collection and use of their personal data. According to the respondents, it was not clear how to do so.

Information inference

Summary of the results: Only a fraction of 39.7% (23) of people believe that, by collecting data on time and location where users' smart phone have been over a day, the Pulso system is able to find out which users were together in the same place. However, this inference is possible and it is the main assumption behind the feature about "familiar faces" provided by the Pulso system.

Instructions to the designers: People do not fully understand the additional information that can be inferred by the system from the collected data. Designers must consider that if users are not able to understand the information that can be inferred from the data collected by the system, then they do not know the real risks of providing data to the system. Thus, any information inference decision based on a risk-benefit analysis may be biased. How and why users data are processed and used should be clearly defined in the system privacy policy.

Information use trade-offs

Summary of the results: Two designing alternatives of the Pulso system were analyzed: 1) informing how many people are "familiar faces", and 2) informing also the names and photos of those people who are familiar faces.

- 60.3% (35) of people think that the feature that informs users how many people are familiar faces is very useful or extremely useful. In order to make this feature possible, about 94.8% (55) of people are willing to allow the system to count how many people have been in the same places that they have been. This information that they are willing to allow the system to collect, causes from moderate concern to extreme concern in about 34.6% (20) of people.

- 50% (29) of people think that the feature that informs the photos and names of people who are familiar faces is very useful or extremely useful. In order to make this feature possible, about 75.9% (44) of people would be willing to provide such data. This information that they are willing to provide causes from moderate concern to extreme concern in about 58.6% (34) of people.

Instructions to the designers: In choosing the feature that will be implemented, the designers must consider the fact that the feature related to the name and photos of people may compromise the adoption and use of the system. Users tend to perceive less utility, have

more concern and be less inclined to provide data related to such feature. One alternative is to allow users to configure the features that they want to enable.

Information exchange

Summary of the results: About the Pulso system providing users' data to third parties, there is no general consensus that can be presented. It seems that the aspect that causes more concern to users is their personal data being provided to government agencies. This causes concern above the moderate level in about 55.2% (32) of people. On the other hand, the aspect that causes less concern to users is their personal data being provided to members of their families. It causes concern below the moderate level in 53.4% (31) of people.

Instructions to the designers: Because people's perception about who can access their data is very diverse, it could be good if the system designers implement an interface that allows each user to inform explicitly who can access his/her data.

5.3.2 Adversary Model

The possible participants in the information flow of the Pulso application are:

- **Users**: Disclose their information and may see the information regarding other users;
- **Service**: Collect the user's information and may disclose to other users or third parties;
- **Third parties**: May obtain user's information through the service.

The stakeholders of the project decided that only other users should be considered adversaries. They assume that the service is not an adversary and the privacy techniques may be applied into the service. This model is similar to the ones used by popular social networks, such as Facebook. Also, they assume that the service will not exchange the user's data with third parties (they do not intend to do that in the near future). Therefore, dealing with such privacy cases is out of the scope of the project.

5.4 Privacy Techniques

Given the instructions generated from the privacy perception study, we mapped them into privacy techniques, as presented in Figure 5.2. Next, we detail each of these techniques.

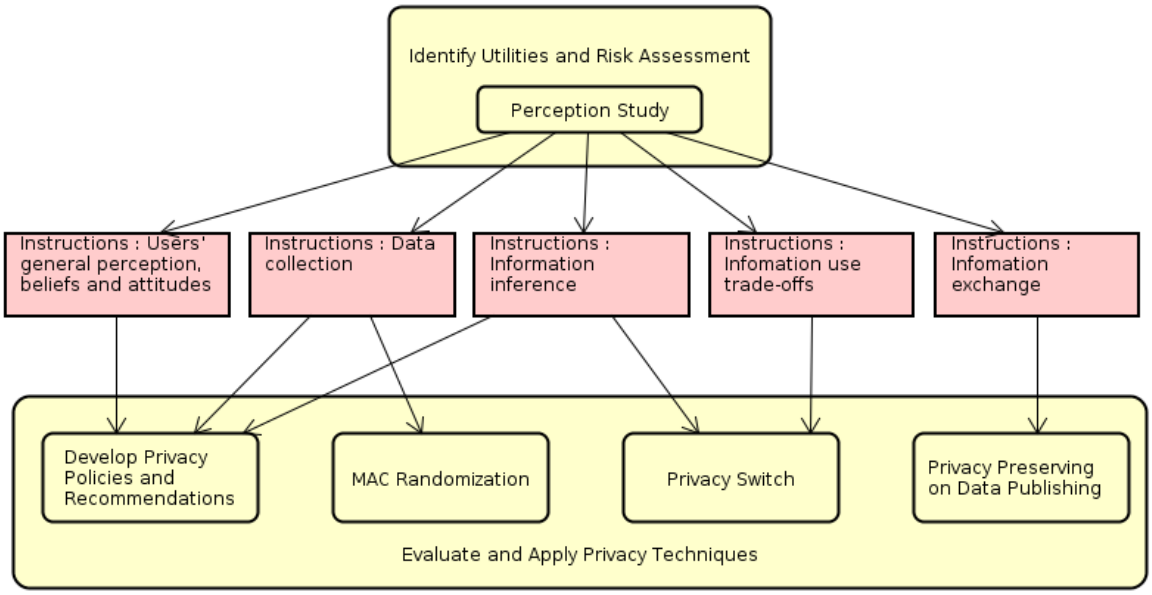


Figure 5.2: Mapping of the instructions generated from the privacy perception study for the Pulso application into the privacy techniques.

5.4.1 Privacy Policies and Recommendations

A privacy policy is a statement that discloses the ways a party collects, uses, discloses, and manages a user's data. It fulfills a legal requirement to protect a user's privacy. It informs the user what specific information is collected, and whether it is kept confidential or shared. The exact contents of a privacy policy will depend upon the applicable law and may need to address requirements across geographical boundaries and legal jurisdictions. Most countries have their own legislation and guidelines of who is covered, what information can be collected, and what it can be used for.

For the Pulso application, the designers must create a document, that explains how the privacy concerns are being addressed and that there are no rules or legislation to be applied for the users in the specific region (Brazil). Privacy recommendations may also be spread over different views of the application, such as in famous applications, like Facebook. The providing of privacy policies and recommendations for the users consist in an evidence of privacy, as presented in Table 5.3.

Table 5.3: Sheet for the evidence $E3$. Privacy policies and recommendations for the Pulso application.

E3	**Privacy policies and recommendations**	**Status**: Done	**Review Date**: November 2016	**Weight**: 5
PbE Activity: *Evaluate and Apply Privacy Techniques*				
Driven by: G1; *In context of*: C1; *Assumptions*: As1, As2, As3				
Description: For the Pulso application, the designers must create a document, that explains how the privacy concerns are being addressed and that are no rules or legislation to be applied for the users in the specific region (Brazil). Privacy recommendations may also be spread over different views of the application.				
References: Section 5.4.1, [80].				

5.4.2 MAC Randomization

Even the users who do not use the Pulso application, have WiFi probe requests collected, disclosed to the remote server, and accounted in the people count information. This may imply privacy risks and with the worst magnitude: users not aware of their information being collected and stored. When this is the case, users can still have privacy without the need to turn off their WiFi, by enabling the MAC address randomization which is a feature in many modern smartphones [106]. With new random MAC addresses periodically being generated by the user devices, it should no longer be possible to track them. This possibility also consists in an evidence of privacy, as presented in Table 5.4.

Table 5.4: Sheet for the evidence $E4$. MAC Randomization.

E4	**Users can activate MAC randomization and be unidentifiable**	**Status**: Done	**Review Date**: November 2016	**Weight**: 3
PbE Activity: *Evaluate and Apply Privacy Techniques*				
Driven by: G1 $\rightarrow$ G4; *In context of*: C1; *Assumptions*: As1, As2, As3				
Description: Users can have privacy without the need to turn off their WiFi, by enabling the option of MAC address randomization. With new random MAC addresses periodically being generated by the user devices, it should no longer be possible to track them.				
References: Section 5.4.2.				

5.4.3 Privacy Switch

Considering other users as possible adversaries, it may be the case that users have their privacy violated when the application discloses their presence/absence to others. For this reason, a privacy switch was implemented. This switch has three possible states: public (everybody can see the user presence/absence status), private (only declared friends can see the user status) and invisible (nobody can see the user status). To have a balance in the privacy-utility trade-off, only users with the switch in the public position are allowed to see public people. Users with the switch in the private or invisible position can only see their friends that are not invisible. The implementation of a privacy switch which gives control to the users is in accordance with the privacy definition mentioned by Stallings *et al.* [92] and consists in another evidence of privacy, as presented in Table 5.5.

Table 5.5: Sheet for the evidence $E5$. Privacy switch.

E5	**A privacy switch, with options of public, private and invisible was implemented**	**Status**: Done	**Review Date**: December 2016	**Weight**: 5
PbE Activity: *Evaluate and Apply Privacy Techniques*				
Driven by: G1 $\rightarrow$ S1 $\rightarrow$ G5; *In context of*: C1; *Assumptions*: As1, As2, As3				
Description: Considering that other users may be adversaries, we implemented a privacy switch. This switch has three possible states: public (everybody can see the user presence/absence status), private (only declared friends can see the user status) and invisible (nobody can see the user status).				
References: Section 5.4.3.				

5.4.4 Privacy Preserving on Data Publishing

The results of the perception study show that people's perception about information exchange is very diverse, and a good scenario is that the system designers implement an interface that allows each user to inform explicitly which third parties can access his/her data. However, trustiness on the service is an assumption in this application, thus, dealing with data disclosure is out of scope. If this becomes a requirement in the future, concepts and techniques of *Privacy Preserving on Data Publishing*, such as those presented by Fung *et al.* [42] and by

Emam [40] may be applied.

5.5 Potential Attacks

We also implemented a connection timeout mechanism. The system computes the gap between the last day someone used the application, and the current date. If this gap reaches an established threshold, the system stops associating that person's MAC address with its personal data (*e.g.* name and picture), which is equivalent to set the user as "Invisible" in the privacy switch mechanism. The need for this mechanism was detected during the attack-/testing activity.

5.6 Concluding Remarks

In this chapter, we validated the *Privacy by Evidence* (*PbE*) methodology through the case study of the Pulso application. Table 5.6 shows the checklist with the collected artifacts in this case study.

Figure 5.3 presents the *GSN* for the privacy case of the Pulso application considering the mentioned adversary model. This representation may grow, according to the normal software evolution and our proposed methodology. In this application context, assuming that security assurances have been taken, the service and the sensors are not adversaries, and the data will not be disclosed to third parties, there is an argument that the privacy is being preserved according to the provided evidences. Every evidence has an identification to enable the traceability into the corresponding artifacts.

Even when invisible, a user still appears in the people counting of the monitored place. This may lead to some privacy breaches and it is possible that, in some cases, the user can still be identifiable [95]. However, the stakeholders of the project decided to do not consider this possibility as a privacy risk. If it starts to be a privacy risk in the future, it is possible to implement a privacy technique which adds some noise to the people counting value. This creates uncertainty regarding presence/absence of users and enables a high privacy level according to the differential privacy model [37], making users indistinguishable in the people counting value.

Table 5.6: Checklist of the artifacts produced in the Pulso case study.

Activity	Artifact	Supplied?
Identify the Application Context and Data Formats	Engagement Report	✓
	Datasets	✓
Check Compliance with Norms and Legislations	Summary of Norms	✓
	Implementation of Norms	✓
	Compliance Proofs	✓
Identify Utilities and Risk Assessment	Utilities List	
	Perception Questionnaires	✓
	Perception Report	✓
	Privacy Concerns	✓
	Adversary Model	✓
	Privacy Policy	✓
Evaluate and Apply Privacy Techniques	Summary of Techniques	✓
	Techniques Report	✓
	Implementation of Techniques	✓
	New Utilities List	
Evaluate Potential Attacks	Summary of Attacks	✓
	Attack Scripts	
	Attacks Report	

When considering that the service is not an adversary, we considered the sensors that collect data as part of the service and therefore, trusted too. This is the case for the current purposes of deployment of the Pulso system. However, if in the future the sensors start to be managed by other parts, such as regulatory agencies, it is possible to implement privacy techniques into the sensors (*i.e.*, before disclosing to remote servers), and therefore, the assumption that "the service is not an adversary" may be changed to an assumption that "the sensor is not an adversary". A possible privacy technique to be implemented into the sensor may be to, instead of disclosing real MAC addresses, disclose salted hashes or encrypted blocks (symmetric or asymetric). This privacy technique may still provide some utility for the application, such as the people counting in a public place. However, due to the anonymization, for the users who wish to be identified, a comunication mechanism be-

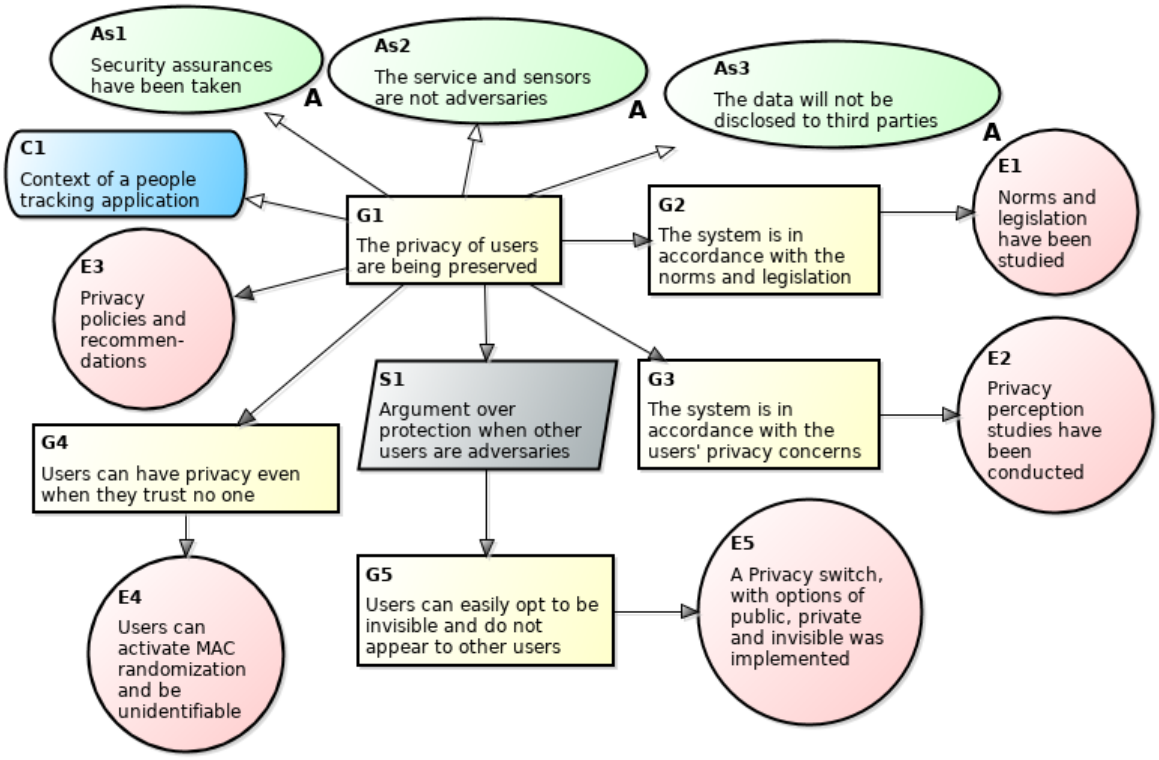


Figure 5.3: The *GSN* representation for the privacy case of the Pulso application.

tween the Pulso mobile application and the sensors would be necessary (*i.e.* to deactivate the anonymization).

After validating the *Privacy by Evidence* (*PbE*) through the case study of the Pulso application, we reinforce the conclusion that this methodology can be regarded as an effective way to implement privacy protections. Five privacy evidences were provided and the sum of their weights results in 19. Therefore, we positively support the research question $RQ2$. Once again, it is important to note that such mitigations must be an iterative work, and thus the stages in the methodology must happen in a constant cycle, since new risks can always be detected.

Chapter 6

Case Study III: Lumen Application

In this chapter, we present the third case study, the Lumen[1] application. This application uses information regarding energy consumption and people counting, therefore, may bring many privacy concerns, as already presented before. The Lumen application is a project developed by the *Federal University of Campina Grande* (*UFCG*) in partnership with the *Hewlett Packard*[2] (*HP*) company.

6.1 Context and Data Formats

The Lumen application uses energy consumption and people counting data for energy efficiency purposes. Therefore, it uses two sensor types: people counters (such as the ones of the Pulso application) and smart meters (such as the ones of the LiteMe application).

We analysed many samples of the collected data by the Lumen application. When deployed in an environment (*e.g.*, residential or an organizational building), the Lumen application is able to identify when there is an efficient or inefficient energy consumption. This is based on the correlation between people counting and energy consumption: when there is a high energy consumption and a few people in the environment, there is an inefficient consumption. On the other hand, when there is a low energy consumption and enough people in the environment, there is an efficient consumption. The formalization is presented below:

[1] `http://web.cloud.lsd.ufcg.edu.br:45194/eff_app`

[2] `http://www.hp.com`

$$Ineff = \frac{W_t - W_e}{people + 1}, \tag{6.1}$$

where $Ineff$ is the metric for inefficiency, W_t is the current total power usage, W_e is the total power usage when the building is empty and $people$ is the current people counting. The subtraction of W_e is to obtain the power consumption only resulted by the presence of people, *i.e.*, excluding the devices which are powered on most of the time.

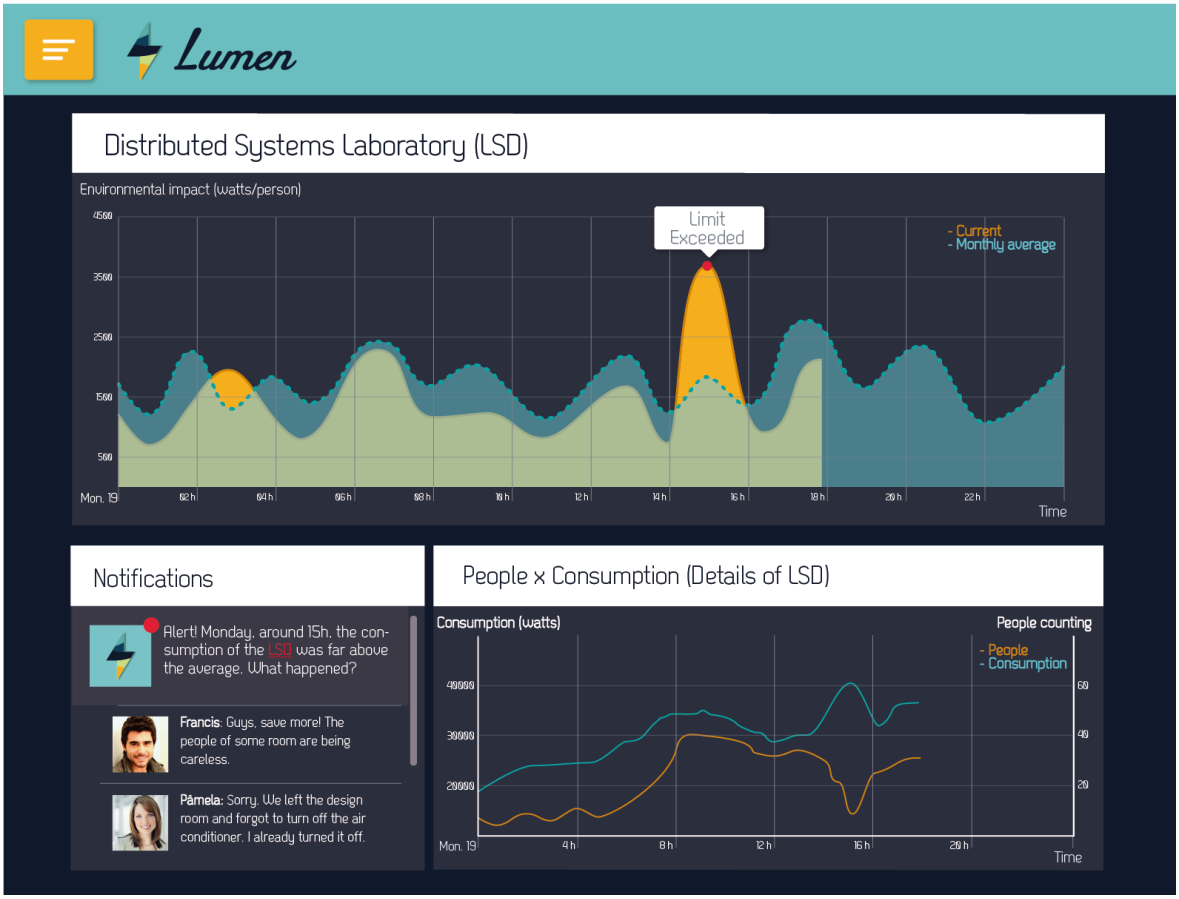


Figure 6.1: A screen of the Lumen application.

When the $Ineff$ metric reaches a threshold, an alarm is triggered in the application and the collaborators are notified. Since the application collects the MAC addresses of the users, the notification is able to blame the responsible(s) for the inefficient consumption. For example, if an employee goes home and forgets to turn off the air conditioner of his workstation, the boss may receive a notification that such employee is being responsible for an inefficient consumption. The application also allows the employee to add a comment in the generated notification, for example, to apologize for his lapse.

Figure 6.1 presents a screen of the Lumen application. In the top of the screen is a chart with the current inefficiency profile versus the average inefficiency in previous days. The bottom-left panel presents the comments of the users regarding the generated notifications and the bottom-right chart presents detailed energy consumption and people counting profiles.

6.2 Norms and Legislation

Just like it was with Pulso and LiteMe applications, in Brazil, we did not find any law dealing with the collection, storage and processing of people tracking and energy consumption data for the Lumen application. The energy meter used by the Lumen application is just an additional device that does not have the goal to replace existing traditional meters, and therefore, the stipulated norms by the *ANEEL* are not applied. Also, Lumen collects data from a specific monitored place in an local manner, not from the internet, and therefore, the *Marco Civil da Internet* law is not applied too.

The *HP* company defines its own norms, according to their experience in the market and their ethical principles, when manipulating *Personally Identifiable Information - PII*. However, the stakeholders of the project defined the Lumen application as a product of the *UFCG*, and therefore, does not need to follow the norms established by the *HP*.

Despite the lack of norms and legislation to be applied in this application, the study and the summary of possibilities suggest a concern for privacy in this project, consisting in an evidence of privacy. Table 6.1 describes a sheet for this evidence $E1$.

6.3 Utilities and Risk Assessment

Our methodology suggests the conduction of a privacy perception study for better understanding the risks and to build an application in accordance with the users perception of privacy. Therefore, for the case study of the Lumen application, we conducted such this study. The following subsection were from an analysis carried out by Lesandro Ponciano, a human computation researcher, about the perception study.

Table 6.1: Sheet for the evidence $E1$. Norms and legislations for Lumen application.

E1	**Norms and legislations have been studied**	**Status**: Done	**Review Date**: December 2016	**Weight**: 1
PbE Activity: *Check Compliance with Norms and Legislation*				
Driven by: G1 $\rightarrow$ G2; *In context of*: C1; *Assumptions*: As1, As2, As3				
Description: We studied and summarized possible norms and legislations that could be applied for the Lumen application. In Brazil, we did not find any norm or law dealing with the collection, storage and processing of people tracking and energy consumption data that could affect the design of the application. There are norms regarding data collection through the usage of internet applications, but it is not the case of Lumen. There are also norms proposed by the *ANEEL*, but only applied to official and regulated energy meters.				
References: Section 6.2.				

6.3.1 Privacy Perception Study

In order to assess people's privacy perceptions and concerns towards the Lumen system, we built a survey with a set of questions derived from five dimensions: 1) people's general perceptions, concerns, and beliefs; 2) data collection; 3) information inference; 4) information use trade-offs; and 5) information exchange [80]. To consider the users perception in the design of the system consists in an evidence of privacy, as presented in Table 6.2.

Two alternative features of the system were evaluated. In the first feature, the notifications of inefficient energy consumption informs to users the room in the organization that is related to a disproportionate consumption. In the second feature, in turn, the system informs to users the name of the user(s) who is(are) related to an inefficient energy consumption. Figure 6.2 presents these two situations. One of the key points investigated in these features was how the users perceive the possibility of their rooms or their names being publicly associated to the cause of an inefficient consumption in the organization.

The context of the system is explained to the respondent by presenting screens of the application and a video prototype that explain how the whole system works – the video prototype is available at `https://youtu.be/XyB6MmHmNYA`. The survey was designed by using Google Forms tools; it is now retired and permanently available at `https://goo.gl/forms/y9W1TVrxQ81mPI483`. A pilot survey was conducted with 8 re-

Table 6.2: Sheet for the evidence $E2$. A privacy perception study of the Lumen application.

E2	**A privacy perception study has been conducted**	**Status**: Done	**Review Date**: December 2016	**Weight**: 5
PbE Activity: *Identify Utilities and Risk Assessment*				
Driven by: G1 $\rightarrow$ G2; *In context of*: C1; *Assumptions*: As1, As2, As3				
Description: For the Lumen application, we conducted a privacy perception study with a survey considering five dimensions that represent perspectives in which it is possible to approach users privacy, which are: 1) people's general perceptions, concerns, and beliefs; 2) data collection; 3) information inference; 4) information use trade-offs; and 5) information exchange. 55 people answered the survey and the results have been considered into the design of the Lumen application.				
References: Section 6.3.1, [80].				

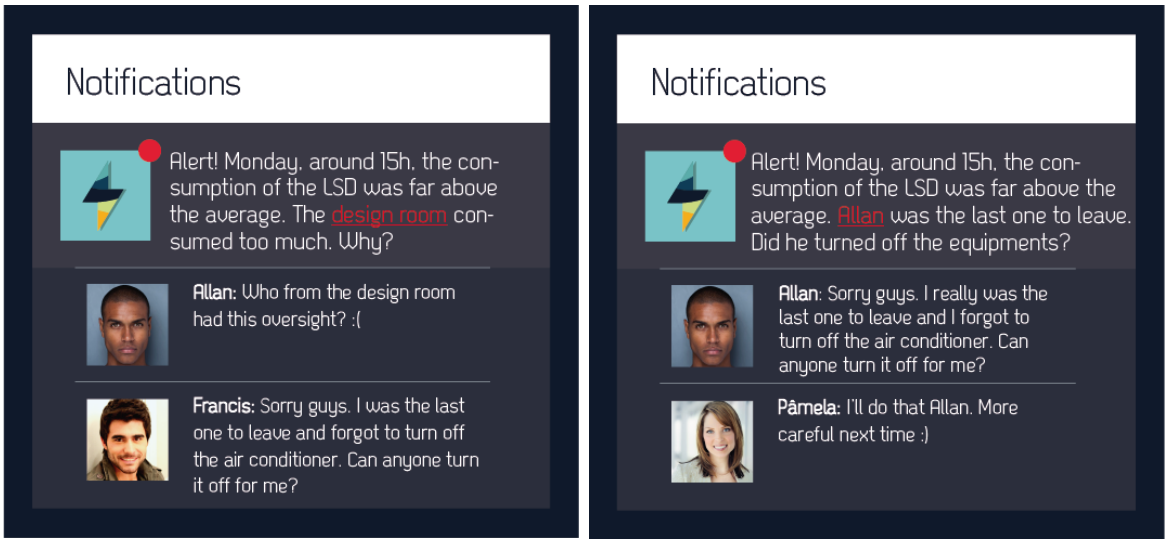
spondents in order to identify and to fix problems in the survey.

After validating the survey, answers from participants were collected in three organizations from Campina Grande, Brazil. The organizations include computer science research laboratories and a software development company. Each respondent had access to the link of the survey. They answered individually, without any external interference. In all, 55 people answered the survey, being 50 (90.9%) male and 5 (9.1%) female respondents, aging mostly between 25 and 34 years old (36, 65.5%). Most of them completed undergraduate (45, 81.8%) or master degree (6, 10.9%). This population of respondents is clearly skewed. However, it is representative of the target audience of the Lumen system. Next, we turn to detail the results obtained from the survey according to the defined five dimensions.

General perception, beliefs and attitudes

Summary of the results: Most people (34, 61.8%) believe that privacy is a right guaranteed by law. The main criteria considered by most people (32, 58.2%) to decide about providing information requested by the system is if it explains its privacy policy. Only few people (7, 12.7%) base their data provision decision on the fact of the system offers to them something in exchange to their private data.

Instructions to the designers: For such users' profile, the following guidelines should be considered by the designers: 1) to convince the user to provide data, the system should rely



(a) Informing the room in the organization that is related to a disproportionate consumption.

(b) Informing the name of the user who is related to a disproportionate consumption.

Figure 6.2: Two possibilities of notifications generated by the Lumen application.

more on providing to them data control guarantees than on proposing to them a bargain in which it offers something in exchange for the data; 2) the system should provide a privacy policy document about data collection and control that can be easily found by the users; 3) relevant aspects foreseen in the legislation must be stated in the privacy policy document.

Data collection

Summary of the results: Users are able to understand which kind of data is collected by the system: most people (46, 83.6%) correctly identify that the system collects data on the locations and times that their cell phones were in the organization, and only 1.8% (1) of users incorrectly identify the system is able to collect data about their health conditions. However, many users (26, 47.3%) do not understand how they would stop the collection of their personal data if they want to do so. In general, the information collected by the system does not cause great privacy concern to users; for example: few users (7, 12.7%) express concern above the moderate level in the collection of times that they came and left work, few users (8, 14.5%) express concern above the moderate level on the collection of which other mobiles were close to their mobiles; and also few users (4, 7.3%) express concern above the moderate level on the collection of data about electrical devices that they use the organization. The minority who feels concern about such collections should have ways in

the system to control the collection of their data or have privacy guarantees that make them feel less concerned.

Instructions to the designers: Two guidelines for system design can be drawn from these results: 1) the system should make it clearer to users how they can stop the collection of their personal data or make it clearer if the collection is done in a way that guarantees their privacy; 2) Control over the collection should be available to each kind of data collected by the system.

Information inference

Summary of the results: In general, users are able to understand which kind of information is inferred by the system: for example, 85.5% (47) of people correctly identify that the system is able to infer the places where users were in the organization throughout the day, and 83.5% (46) of people correctly identify that the system is able to infer which users were together in the same place in the organization. The information inferred by the system does not cause great concern to users; information about places in the organization where users have been over a day (4 users or 7.2% of users express concern above the moderate level); information about people with whom users have been in the organization (5 users or 9.1% of users express concern above the moderate level); and information about electrical devices that users use in the organization (2 users or 3.6% of users express concern above the moderate level). Thus, in general, the current picture of data inference by the system does not cause privacy concern in the majority of the users.

Instructions to the designers: Three guidelines for system design can be drawn from these results: 1) The system should provide ways that the minority of users who feel concerned about information inference can exert control over it; 2) Because the users' level of concerns vary with the type of inferred information, the system may allow users to control the inference of each information differently; 3) the privacy policy document should provide details about which kind of information is inferred by the system and whether the system do the inference in a way to provide privacy for the users.

Information use trade-offs

Summary of the results: Two alternative features were evaluated in the survey. In the first feature, the system informs to users the room in the organization that is related to a disproportionate energy consumption. Most users perceive high utility in this feature (34

people or 61% of people perceive the feature as very useful or extremely useful), and they consider that this feature is low or moderate harmful in terms of privacy (52 people or 95% of people feel moderated concerned or lower in terms of privacy). In the second feature, in turn, the system informs to users the name of the user who is related to a disproportionate energy consumption in the organization. Few users perceive high utility in this feature (21 people or 38.1% of people perceive the feature as very useful or extremely useful) and many of them consider this feature low or moderate harmful in terms of privacy (39 people or 70.9% of people feel moderated concerned or lower in terms of privacy).

Instructions to the designers: From these results, the following advice can be draw: 1) providing notifications that inform the room related to a disproportionate energy consumption in the organization would be well perceived (in terms of utility and privacy) by the users, so it can be used in the system; 2) providing notifications of a disproportionate energy consumption that associate the problem with a name of a user is something perceived as less useful and more worrying in terms of privacy.

Information exchange

Summary of the results: In general, users are concerned about their data being leaked or made available to third parties. Many of them feel very or extremely concerned of their data being provided to their colleagues (26, 47.3%) or being provided to their superiors (24, 43.6%). Also, the majority of them feel very or extremely concerned of their data being provided to other systems (34, 61.81%) or being provided to government agencies (32, 58.18%). Thus, users care about which third parties can have access to the data that the system collects about them.

Instructions to the designers: 1) The system should made explicit to which third parties user's data is provided and which kind of data is provided; 2) Users disagree among them in terms of whether their data may be accessible or not to their colleagues and superiors; thus, concerns in this regard must be addressable to each individual user; 3) User's data cannot be provided to government agencies or to other systems; procedures that are different from this must be stated to the users. 4) The system must protect user's data against unauthorized access from third parties.

6.3.2 Adversary Model

We considered the possible participants in the information flow of the Lumen application:

- **Users**: Disclose their information and may see the information regarding other users;
- **Service**: Collect the user's information and may disclose to other users or third parties;
- **Third parties**: May obtain user's information through the service.

The stakeholders of the project decided that only other users should be considered adversaries. They assume that the service is not an adversary and the privacy techniques may be applied into the service. Also, they assume that the service will not exchange the user's data with third parties (they do not intend to do that in the near future). Therefore, dealing with such privacy cases is out of the scope of the project.

6.4 Privacy Techniques

Given the instructions generated from the privacy perception study, we mapped them into privacy techniques, as presented in Figure 6.3. Next, we detail each of these techniques.

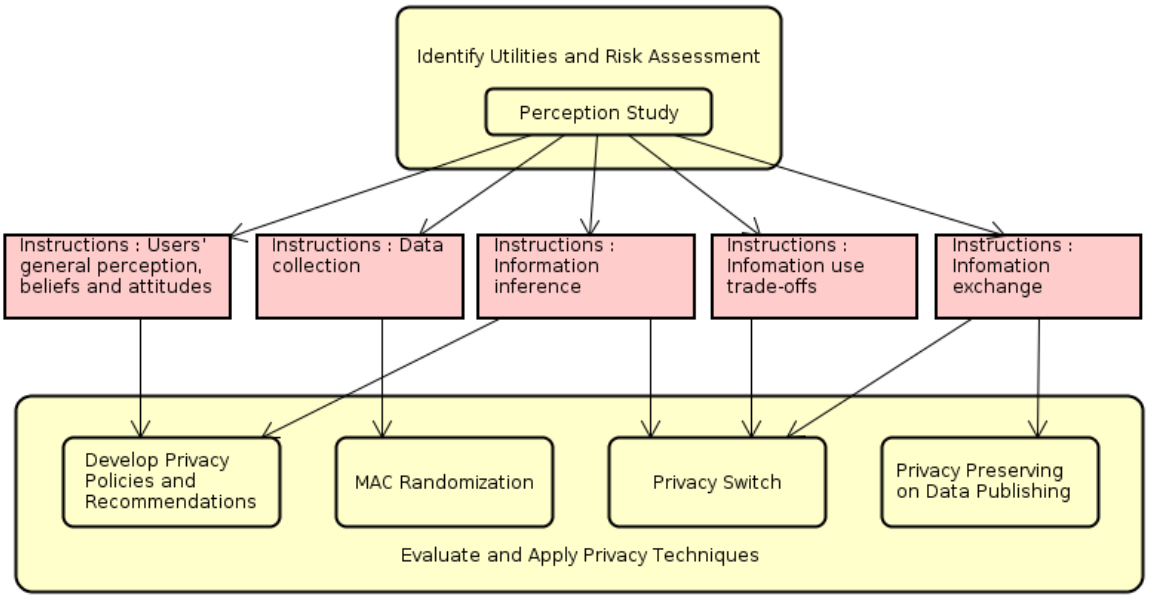


Figure 6.3: Mapping of the instructions generated from the privacy perception study for the Lumen application into the privacy techniques.

6.4.1 Privacy Policies and Recommendations

For the Lumen application, the designers must create a document, that explains how the privacy concerns are being addressed and that there are no rules or legislation to be applied for the users in the specific region (Brazil). Privacy recommendations may also be spread over different views of the application, such as occur in famous applications, like Facebook's. The providing of privacy policies and recommendations for the users consist in an evidence of privacy, as presented in Table 6.3.

Table 6.3: Sheet for the evidence $E3$. Privacy policies and recommendations for the Lumen application.

E3	**Privacy policies and recommendations**	**Status**: Done	**Review Date**: December 2016	**Weight**: 5
PbE Activity: *Evaluate and Apply Privacy Techniques*				
Driven by: G1; *In context of*: C1; *Assumptions*: As1, As2, As3				
Description: For the Lumen application, the designers must create a document, that explains how the privacy concerns are being addressed and that are no rules or legislation to be applied for the users in the specific region (Brazil). Privacy recommendations may also be spread over different views of the application.				
References: Section 6.4.1, [80].				

6.4.2 MAC Randomization

Even the users who do not use the Lumen application, have WiFi probe requests collected, disclosed to the remote server, and accounted in the people count and energy efficiency information. When this is the case, users can still have privacy without the need to turn off their WiFi, by enabling the option of MAC address randomization [106]. With new random MAC addresses periodically being generated by the user devices, it should no longer be possible to track them. This possibility also consists in an evidence of privacy, as presented in Table 6.4.

Table 6.4: Sheet for the evidence $E4$. MAC Randomization.

E4	**Users can activate MAC randomization and be unidentifiable**	**Status**: Done	**Review Date**: December 2016	**Weight**: 3
PbE Activity: *Evaluate and Apply Privacy Techniques*				
Driven by: G1 $\rightarrow$ G4; *In context of*: C1; *Assumptions*: As1, As2, As3				
Description: Users can have privacy without the need to turn off their WiFi, by enabling the option of MAC address randomization. With new random MAC addresses periodically being generated by the user devices, it should no longer be possible to track them.				
References: Section 6.4.2.				

6.4.3 Privacy Switch

Although the privacy perception study points that providing notifications of a disproportionate energy consumption that associate the problem with a name of a user (such as presented in Figure 6.2b) is something perceived as less useful and more worrying in terms of privacy, the stakeholders of the project believe that users will be more conscious in saving energy if this feature is available. Therefore, it became a requirement of the system to associate the problem with the name of the user (blaming).

For the users who want privacy and do not want to be associated, we implemented a privacy switch. This switch has two possible states: visible (in a notification, the name of the user will be associated to the disproportionate energy consumption) and invisible (the name of the user will not be associated in the notification). To find a balance in the privacy and utility tradeoff, users are encouraged to keep their privacy switch in the visible position. There is a gamification scheme, where visible users gain points in a ranking. In the end, users with more points (*i.e.*, who more save energy) earn some awards.

The implementation of a privacy switch which gives control to the users is in accordance with the privacy definition mentioned by Stallings *et al.* [92]. With this privacy technique, we have another evidence of privacy, as presented in Table 6.5.

Table 6.5: Sheet for the evidence $E5$. Privacy switch.

E5	**A privacy switch, with options of public and invisible was implemented**	**Status**: Done	**Review Date**: December 2016	**Weight**: 5
PbE Activity: *Evaluate and Apply Privacy Techniques*				
Driven by: G1 → S1 → G5; *In context of*: C1; *Assumptions*: As1, As2, As3				
Description: Considering that other users may be adversaries, we implemented a privacy switch. This switch has two possible states: visible (in a notification, the name of the user will be associated to the disproportionate energy consumption) and invisible (the name of the user will not be associated in the notification).				
References: Section 6.4.3.				

6.4.4 Privacy Preserving on Data Publishing

The results of the perception study show that users disagree among them in terms of whether their data may be accessible or not to third parties or to their colleagues and superiors. Trustiness on the service is an assumption in this application, thus, dealing with data disclosure to third parties is out of scope. If this becomes a requirement in the future, concepts and techniques such as those presented by Fung *et al*. [42] and by Emam [40] may be applied. For the colleagues and superiors, the privacy switch technique, presented in Section 6.4.3, should mitigate the problem.

6.5 Potential Attacks

We also implemented a connection timeout mechanism. The system computes the gap between the last day someone used the application, and the current date. If this gap reaches an established threshold, the system stops associating that person's MAC address with a notification of inefficient consumption, which is equivalent to set the user as "Invisible" in the privacy switch mechanism. The need for this mechanism was detected during the attack/testing activity.

6.6 Concluding Remarks

In this chapter, we validated the *Privacy by Evidence* (*PbE*) methodology through the case study of the Lumen application. Table 6.6 shows the checklist with the collected artifacts in this case study.

Table 6.6: Checklist of the artifacts produced in the Lumen case study.

Activity	Artifact	Supplied?
Identify the Application Context and Data Formats	Engagement Report	✓
	Datasets	✓
Check Compliance with Norms and Legislations	Summary of Norms	✓
	Implementation of Norms	✓
	Compliance Proofs	✓
Identify Utilities and Risk Assessment	Utilities List	
	Perception Questionnaires	✓
	Perception Report	✓
	Privacy Concerns	✓
	Adversary Model	✓
	Privacy Policy	✓
Evaluate and Apply Privacy Techniques	Summary of Techniques	✓
	Techniques Report	✓
	Implementation of Techniques	✓
	New Utilities List	
Evaluate Potential Attacks	Summary of Attacks	✓
	Attack Scripts	
	Attacks Report	

Figure 6.4 presents the *GSN* for the privacy case of the Lumen application considering the mentioned adversary model. This representation may grow, according to the normal software evolution and our proposed methodology. In this application context, assuming that security assurances have been taken, the service and the sensors are not adversaries, and the data will not be disclosed to third parties, there is an argument that the privacy is being preserved according to the provided evidences. Every evidence has an identification

to enable the traceability into the corresponding artifacts.

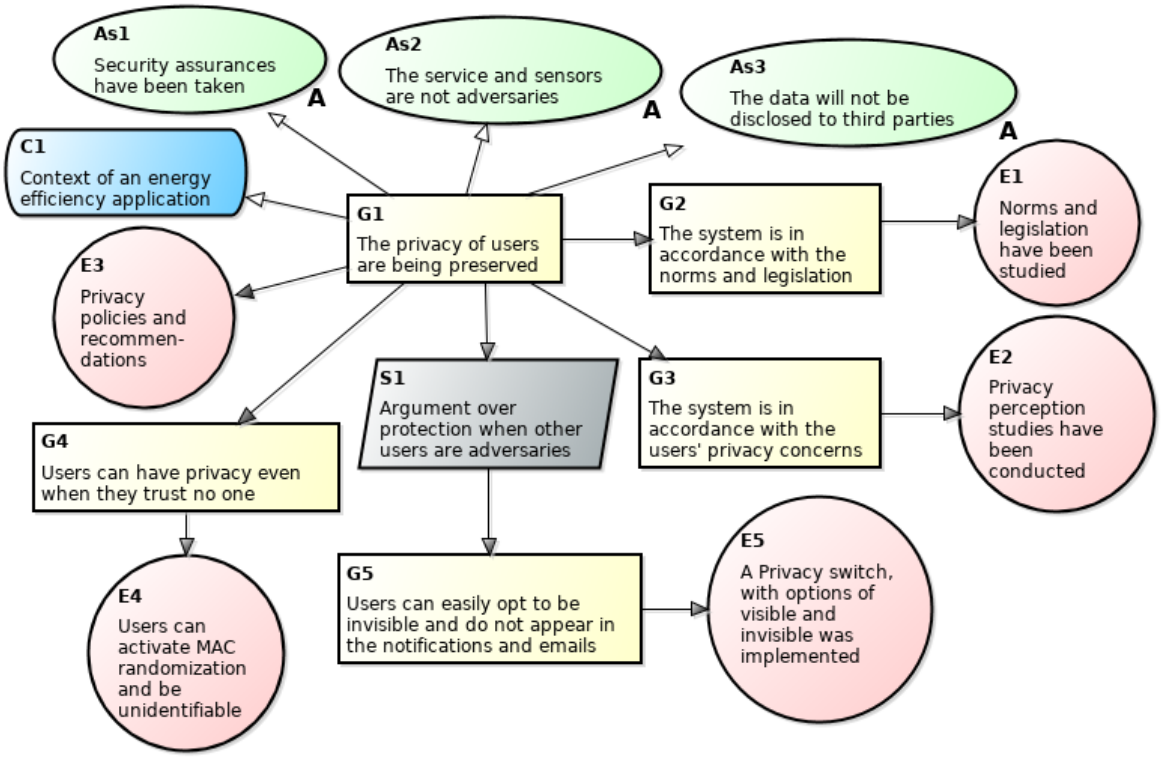


Figure 6.4: The *GSN* representation for the privacy case of the Lumen application.

Even when invisible, a user still appears in the people counting from the monitored place. This may lead to some privacy breaches and it is possible that, in some cases, the user can still be identifiable [95]. However, the stakeholders of the project decided to do not consider this possibility as a privacy risk. If it starts to be in the future, it is possible to implement a privacy technique which adds some noise to the people counting value. This creates uncertainty regarding presence/absence of users and enables a high privacy level according to the differential privacy model [37], making users indistinguishable in the people counting value.

Another possible privacy risk is that even when invisible, the consumption habits of a user still appears in the consumption profile and from this, it is possible to infer much information, as presented in Chapter 4. However, the stakeholders of the project decided to do not consider this possibility as a privacy risk. If it starts to be in the future, it is possible to implement a privacy technique which adds some noise to the energy consumption value, as presented in Section 4.4.1.

When considering that the service is not an adversary, we considered the sensors that collect data as part of the service and therefore, trusted too. If in the future the sensors start to be managed by other parts, such as regulatory agencies, it is possible to implement privacy techniques into the sensors (*i.e.*, before disclosing to remote servers), and therefore, the assumption that "the service is not an adversary" may be changed to an assumption that "the sensor is not an adversary".

After validating the *Privacy by Evidence* (*PbE*) through the case study of the Lumen application, we reinforce the conclusion that this methodology can be regarded as an effective way to implement privacy protections. Five privacy evidences were provided and the sum of their weights results in 19. Therefore, we positively support the research question $RQ3$. Once again, it is important to note that such mitigations must be an iterative work.

Chapter 7

Case Study IV: Two Factor Authentication System

In this chapter, we present the fourth case study, a two factor authentication system. This application uses information regarding people presence, therefore, may bring many privacy concerns, as already presented before.

The development team consisted in a manager and a developer. To avoid biasing, different than the previous case studies, this application was developed without the participation of the author. For this, the Think Aloud method [27; 91] was used to gather information regarding the usage of *PbE* by the development team without influencing what participants say and do, and hence, use such useful information as feedback in order to improve the methodology.

In usability testing [78], the Think Aloud method is a protocol to help the observer understand the emotional and practical user experience of a product or prototype. It requires the user to not only say his or her thoughts out loud, but also to explain and justify them. For example, an observer in a think aloud usability test for a website may expect a user clicking on a link and justifying this action by explaining his/her interest on it. This may produce valuable data on how to improve the user interface. The observer may be very surprised when a user does not click on a button that has been optimized and positioned in a prominent spot of the page layout. This could be due to the color of the button or the text on the button label. The product or prototype is then modified based on this feedback.

There are other proposed protocols too, but Think Aloud may be the single most valuable usability engineering method [71]. We instantiated this protocol to our work (*i.e.*, instead

of observing users testing an interface design, we observe a development team using our proposed methodology, *PbE*). The author found this instantiation reasonable, since it does not lose the essence of the Think Aloud protocol. During this experiment, the following activities were conducted:

- The development team develops the application using *PbE*. During the development, they write their thoughts and information regarding their produced artifacts in a electronic document (Google docs);
- The observer monitors the development in person and takes notes regarding the decisions taken by the development team;
- The observer also monitors the electronic document and verifies if the content is in accordance with his notes. In case of divergence, the observer asks questions to the development team, without biasing the team, however. All the artifacts, including the provided evidences of privacy and code repositories are monitored too;
- Confusions and misunderstandings made by the development team are used to review and improve the methodology. However, to avoid biasing, such occurrences are not revealed to the team and are considered only after the experiment;
- Developers may also implicitly suggest improvements on the methodology. The suggestions are considered only after the end of the experiment;

7.1 Methodology

First, the developer was asked to read about the *PbE* methodology and write a summary. In his own words, *PbE* is composed by:

- Privacy team acting together with development team
- Steps:
    a) Identify application context and data format
    b) Check compliance with norms and legislation

  - Evidence: norms and legislation compliance

c) Identify utilities and risk assessment

  - Evidence: users perception

d) Evaluate and apply privacy techniques

  - Evidences: privacy metrics and techniques

e) Evaluate potential attacks

  - Evidences: resilience to attacks

After making sure that the developer read about *PbE*, we started the experiment. The hypothesis considered in this experiment was based on $RQ4$. Recall that:

- **RQ4**: Is *PbE* helpful to develop a privacy-friendly two factor authentication application?

The considered metric is the number of provided evidences and their sum of weights. If the development of the two factor authentication application is being able to produce evidences of privacy protection, then this indicates that *PbE* is being helpful. Therefore, the considered hypothesis is:

- **H0**: If the team uses *PbE*, it will provide evidences of privacy.

7.1.1 Threats to Validity

The hypothesis H_0 may be considered a good hypothesis and, at first glance, appears easily testable. The problem is that, in a solid experimental design, the opposite (contrapositive) should also be true. The design of experiment dictates that, if a certain event does not occur, the tested outcome will not happen, a subtle but crucial factor [101]. The reason for this is that it ensures that there is a genuine causal relationship between the independent and dependent variables. Therefore, the following statement should also be true.

- **H1**: If the team does not use *PbE*, it will not provide evidences of privacy.

The problem is, in this experiment, for avoid biasing, it would be necessary to have another development team developing the same two factor authentication application, but

without using *PbE*. Actually, even that would not be sufficient. It would be more interesting to have a set of development teams (subjects) for H_1 and another set for H_0. That experiment would provide results with statistical confidence, but would require a lot of human and financial resources.

Because of this, the examining committee of this dissertation defense suggested the experiment with the usage of the Think Aloud protocol to observe one development team and to gather useful information as feedback, in order to improve the proposed methodology. We recognize this as a good plan, and went in this direction. However, we could not avoid recognizing the threats to the validity, and therefore, leaving the experiment of evaluating a set of development team using *PbE* and another set not using it as a possible future work.

Once the author was not part of the development, it is important to note that the information contained in the next sections may not be in accordance with his opinion and beliefs. In Section 7.7 is presented a critical point of view and the lessons learned from the feedback of this experiment.

7.2 Context and Data Formats

The application aims to authenticate users by using two factors (password and presence of the MAC address of the mobile phone). To make this possible, it should provide operations like registration, removal, and authentication of users. A user is registered with name, easy and hard passwords, and MAC address. For removal operation, just the username is needed. To authenticate, a user must specify the name, type the easy password and the system must detect the presence of his/her MAC address. If the easy password and MAC address are correct, the user is authenticated; if the MAC address is not validated/sensed, the user still can be authenticated, but using the hard password.

The information of authenticated users could be used for many purposes and by different external applications, such as email systems, code repositories, social networks, access to physical buildings, computer logins, and many others, similar to *OAuth* services [29]. In this case study, it was considered just an application of access control to a physical building (the Distributed Systems Laboratory at Federal University of Campina Grande).

The data regarding user registration is exchanged in *JSON* format, like the example be-

low:

```
{
  "username": "Dalton",
  "easy": "1234",
  "hard": "F!t0miRlr3t:Dj",
  "MAC": "00:28:f8:4f:38:13"
}
```

The easy password has 4-digit format and the hard password has between 8 and 15 characters.

7.3 Norms and Legislation

Just like it was with the LiteMe, Pulso, and Lumen applications, in Brazil, the development team did not find any laws dealing with the collection, storage and processing of people presence for the two factor authentication system. This reinforces our argument that in Brazil there is a lack of laws concerning protection of personal data.

Different than the previous case studies, the development team did not consider the effort of searching for appropiate laws and norms as an evidence of privacy, although the application can still be considered as in accordance with the set of Brazilian privacy laws (because none is applied).

7.4 Utilities and Risk Assessment

For the risk assessment, the team first identified what is possible to do with the collected MAC addresses and timestamps. Some utilities that use such data were listed in Table 7.1. From this list, using a risk binary scale, “Investigate people’s behaviour at the laboratory” and “Identify presence of people in the laboratory” were considered as privacy threats that needed to be solved.

The team did not carry out the perception study, so it was not possible to generate a perception report including threats and privacy recommendations. However, the developer, based on his experience, listed the following privacy concerns:

Table 7.1: List of data utilities for the two factor authentication system.

Feature / Benefit	**Privacy Risk?**
2 factor authentication system to allow/deny access to the laboratory	No
Identify the users/members of the laboratory	No
Investigate people's behaviour (time of entrance/exit, people counting, etc.)	Yes
Identify presence of people in the laboratory	Yes

- The easy password must not be discovered by an adversary;
- The hard password must not be discovered by an adversary;
- The MAC address must not be discovered by an adversary.

From the privacy concerns, we could observe that the team is concerned not only with the MAC addresses, but also with the attributes of easy and hard password. Based on this, the team created a privacy policy: *Our privacy policy guarantees that no one can see the data stored by the system. This includes all the passwords (easy and hard) and the MAC addresses. Only the usernames can be visualized by the administrators*.

7.4.1 Adversary Model

An adversary considered by the team was someone with the ability of accessing and *sniffing* the Intranet and getting the communication messages between sensors, server and clients (man-in-the-middle). Another considered adversary is an *evil admin*, a malicious administrator of the system with the ability of, for example, dumping the database containing the MAC addresses and passwords. With such information, both adversaries could be authenticated in place of a victim, or even observe the behaviour of people in the laboratory.

Similar to the previous case studies, the sensors were not considered as adversaries. Therefore, it could, for example, be regulated by a regulatory agency, and have implementation of privacy techniques.

7.5 Privacy Techniques

To achieve the requirements of the privacy policy, some techniques can be applied. The first considered technique was the usage of encrypted and authenticated communication between sensors, server and clients. This was achieved through the usage of *HTTPS* (*Hyper Text Transfer Protocol Secure*) and, therefore, the application can be regarded as resilient to the man-in-the-middle adversary. The usage of this technique consists in an evidence of privacy, as presented in Table 7.2.

Table 7.2: Sheet for evidence $E1$. HTTPS communication.

E1	**Data transmissions are over HTTPS**	**Status**: Done	**Review Date**: December 2017	**Weight**: 20
PbE Activity: *Evaluate and Apply Privacy Techniques*				
Driven by: G1 $\rightarrow$ G2; *In context of*: C1; *Assumptions*: As1, As2				
Description: Users can have privacy during data exchange because the sensitive data can not be discovered during the transmission. Communications between a client application and the server and between a sensor and the server are over HTTPS.				
References: Section 7.5.				

Another privacy technique was to store *salted* hashes of the easy and hard passwords. Therefore, an *evil admin* with the ability to dump the database would not be able to extract the original passwords. This leads us to the second evidence of privacy, $E2$, as presented in Table 7.3.

Another privacy technique was to store *salted* hashes of the MAC addresses. Therefore, an *evil admin* with the ability to dump the database would not be able to extract the original MAC addresses. This leads us to the third evidence of privacy, $E3$, as presented in Table 7.4. The thought is that, since the *evil admin* is not able to obtain the original MAC addresses, he or she will not be able to track the users.

So far, with the provision of the three evidences of privacy, the developer was satisfied with the privacy techniques and moved on to the next *PbE* activity (evaluate potential attacks). In this stage, the developer noticed possibilities that he have not noticed before, more specifically, possible privacy attacks. He noticed that if the *evil admin* tracks the process into

Table 7.3: Sheet for evidence $E2$. *Salted* hashes of the easy and hard passwords.

E2	**Salted hashes of passwords are stored.**	**Status**: Done	**Review Date**: December 2017	**Weight**: 13
PbE Activity: *Evaluate and Apply Privacy Techniques*				
Driven by: G1 → G3; *In context of*: C1; *Assumptions*: As1, As2				
Description: The easy and hard passwords of the users are not stored in plaintext. They are stored in the form of salted hashes. For this, the key derivation function *PBKDF2* (*Password-Based Key Derivation Function 2*) with SHA256 was used. The salt is the own username, which is stored alongside the password hashes.				
References: Section 7.5.				

the server (*e.g.*, using ptrace system call), or even develop his/her own malicious application, it is possible to dump the easy and hard passwords and the MAC addresses before the hashes computation. In other words, the attack vector was not only the database, but many other areas into the server. To mitigate this issue, the developer decided to use features provided by Intel SGX (Software Guard eXtensions) and generated another evidence of privacy, as presented in Table 7.5.

Intel SGX is available on recent off-the-shelf processors based on the Skylake microarchitecture or newer, and already has a wide variety of research publications related to its applicability on real world scenarios [48]. It is a hardware-based technology for ensuring privacy of sensitive data from disclosure or modification that enables user-level applications to allocate protected areas of memory called enclaves [64]. Such memory areas are cryptographically protected even from code running with higher privilege levels. This technology also provides means for users/applications to verify if a given application is running inside an Intel SGX enclave, and even check if the code of the application is indeed the one expected to be running there, a process called Remote Attestation [6].

In this case study, the server running the two factor authentication system uses an enclave to receive and process the passwords from the clients and the MAC addresses from the sensors. The communications with the enclave (bindings) are protected through Remote Attestation processes.

Table 7.4: Sheet for evidence $E3$. *Salted* hashes of the MAC addresses.

E3	**Salted hashes of MAC addresses are stored.**	**Status**: Done	**Review Date**: December 2017	**Weight**: 13
PbE Activity: *Evaluate and Apply Privacy Techniques*				
Driven by: G1 → G4; *In context of*: C1; *Assumptions*: As1, As2				
Description: The MAC addresses of the users are not stored in plaintext. They are stored in the form of salted hashes. For this, the key derivation function *PBKDF2* (*Password-Based Key Derivation Function 2*) with SHA256 was used. The salt is the easy password, which is sent by the user when requesting to log in. Since the *evil admin* is not able to obtain the original MAC addresses, he or she will not be able to track the users.				
References: Section 7.5.				

When the enclave receives a current detected MAC address from the sensor, it stores in a data structure for 30 minutes. When a user logs in, the client sends to the enclave the username and the easy password. Thus, the enclave computes the hash of the easy password (using the username as salt), and iterates through the data structure of MAC addresses and also computing the hashes (using the easy password as salt). If a combination of username, hash of easy password, and hash of MAC address matches with a record in the database stored in the file system, then the user is authenticated. Figure 7.1 presents the architecture of this implementation.

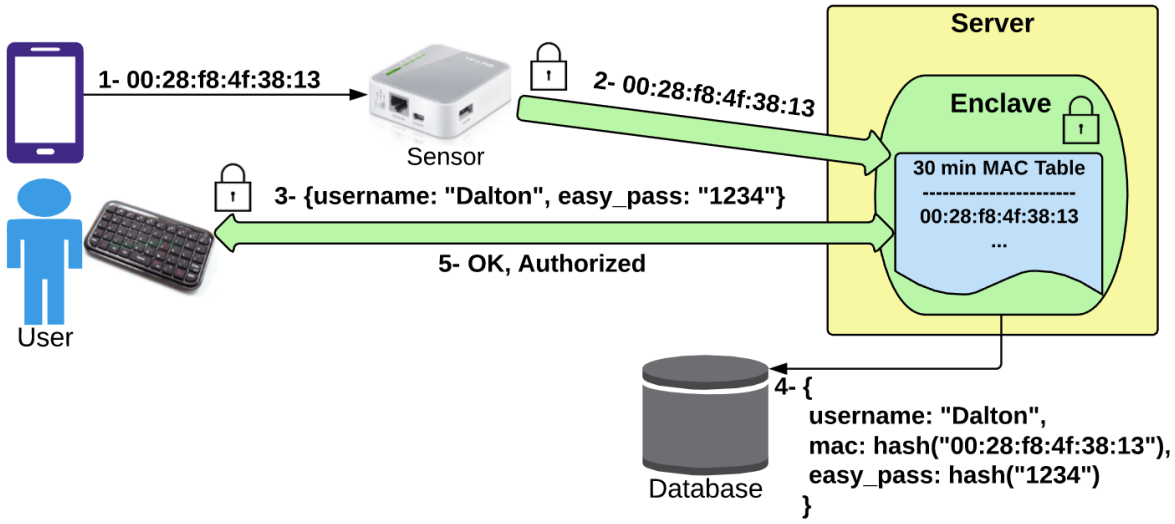


Figure 7.1: Architecture of the two factor authentication system.

Table 7.5: Sheet for evidence $E4$. Usage of Intel SGX features.

E4	**The server application runs in an Intel SGX enclave.**	**Status**: Done	**Review Date**: December 2017	**Weight**: 40
PbE Activity: *Evaluate and Apply Privacy Techniques*				
Driven by: G1 → G5; *In context of*: C1; *Assumptions*: As1, As2				
Description: Intel SGX is a hardware-based technology that enables user-level applications to allocate protected areas of memory, called enclaves, for ensuring privacy of sensitive data from disclosure or modification. Such memory areas are cryptographically protected even from code running with higher privilege levels. The server running the two factor authentication system uses an enclave to receive the passwords from the clients and the MAC addresses from the sensors. The communications which the enclave (bindings) are protected through Remote Attestation processes.				
References: Section 7.5.				

Table 7.6 presents the list from Table 7.1 but with an additional column (supported or not). From the features that are not supported, "Investigate people's behaviour (time of entrance/exit, people counting, etc.)" and "Identify presence of people in the laboratory" can be considered as privacy threats that were solved.

Table 7.6: List of data utilities and the impact after the usage of the privacy techniques.

Feature / Benefit	**Privacy Risk?**	**Support**
2 factor authentication system to allow/deny access to the laboratory	No	✓
Identify the users/members of the laboratory	No	✓
Investigate people's behaviour (time of entrance/exit, people counting, etc.)	Yes	
Identify presence of people in the laboratory	Yes	

7.6 Potential Attacks

In this stage, the developer thought and detailed the following attacks:

- **Man-in-the-middle**: some attacker with access to the network, sniffing it, trying to get

sensitive data. This attack is mitigated with the encrypted and authenticated channels.

- **Evil admin**: a worker/collaborator with access to the server trying to steal some sensitive data. Initially, the developer mitigated this possibility with the usage of salted hashes. However, in the attack phase, he noticed other attack vectors, such as *process tracing*. This motivated him to go back to the activity of *implementation of privacy techniques* and to use Intel SGX.

- **MAC Spoofing**: the developer was always aware that an attacker (man-in-the-middle, or *evil admin*) with access to the easy passwords and MAC addresses would be able to pretend to be a valid user, matching the password together with a spoofed MAC address.

7.7 Concluding Remarks

In this chapter, *Privacy by Evidence* (*PbE*) methodology was validated through the case study of a two factor authentication system. Table 7.7 shows the checklist with the collected artifacts in this case study.

Figure 7.2 presents the *GSN* for the privacy case of the two factor authentication system considering the mentioned adversary model. This representation may grow, according to the normal software evolution and our proposed methodology. In this application context, assuming that security assurances have been taken, and that the sensors are not adversaries, there is an argument that the privacy is being preserved according to the provided evidences. Every evidence has an identification to enable the traceability into the corresponding artifacts.

After validating the *Privacy by Evidence* (*PbE*) through the case study of the two factor authentication system, we reinforce the conclusion that this methodology can be regarded as an effective way to implement privacy protections. Four privacy evidences were provided and the sum of their weights results in 86. Therefore, we positively support the research question $RQ4$. Once again, it is important to note that such mitigations must be an iterative work.

During the experiment with the think aloud protocol, the developer had a few considera-

Table 7.7: Checklist of the artifacts produced in the two factor authentication system case study.

Activity	Artifact	Supplied?
Identify the Application Context and Data Formats	Engagement Report	✓
	Datasets	✓
Check Compliance with Norms and Legislations	Summary of Norms	✓
	Implementation of Norms	
	Compliance Proofs	
Identify Utilities and Risk Assessment	Utilities List	✓
	Perception Questionnaires	
	Perception Report	
	Privacy Concerns	✓
	Adversary Model	✓
	Privacy Policy	✓
Evaluate and Apply Privacy Techniques	Summary of Techniques	✓
	Techniques Report	✓
	Implementation of Techniques	✓
	New Utilities List	✓
Evaluate Potential Attacks	Summary of Attacks	✓
	Attack Scripts	
	Attacks Report	

tions to improve the methodology:

- Criticized the provision of evidences of privacy regarding norms/legislations when there is no norm/legislation to be complied with.
- Initially, the "New Utilities List" artifact was provided before the "Implementation of Techniques" but he suggested to provide after, once the utilities impact are defined just after the implementation of the privacy techniques. We accepted the suggestion and made the modification into the checklist.
- Initially, the "Privacy Concerns" artifact was part of the "Perception Report" artifact. He suggested to separate them, once one can define possible privacy concerns without

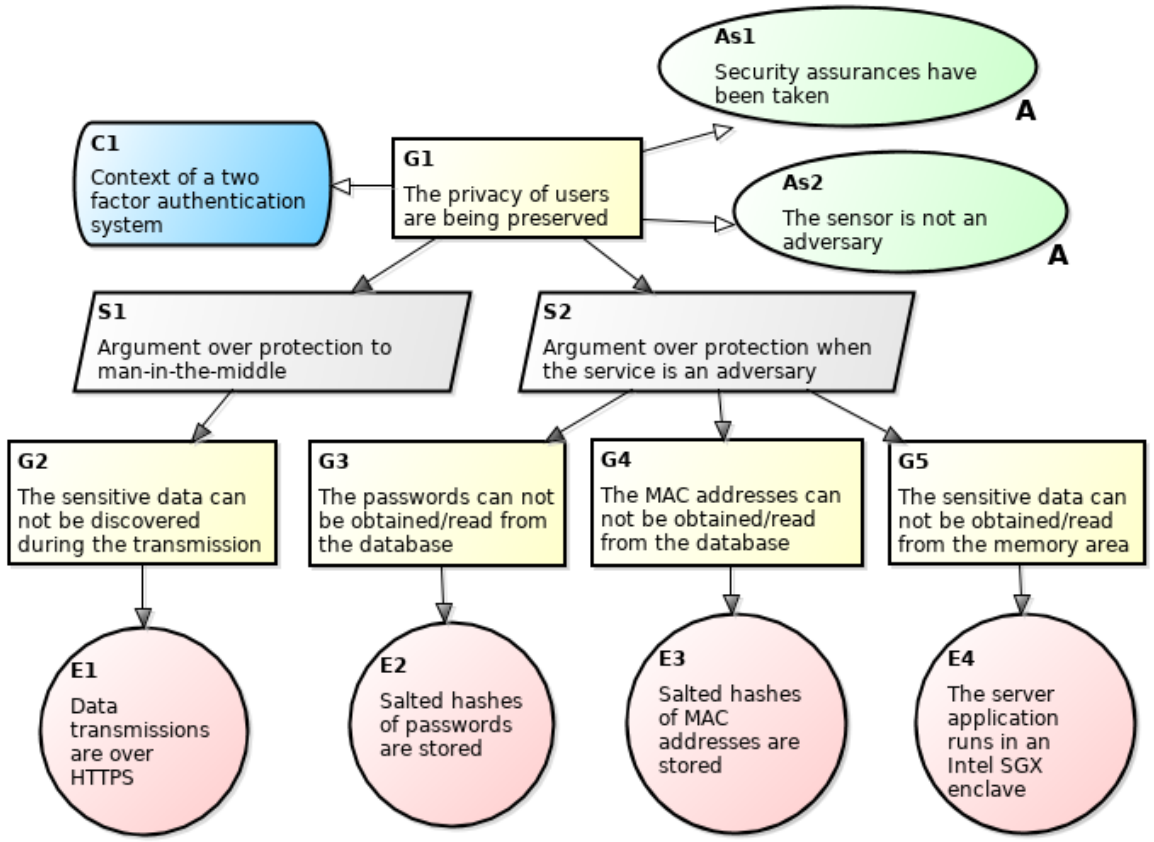


Figure 7.2: The *GSN* representation for the privacy case of the two factor authentication system.

conducting a privacy perception study (*e.g.*, based on the experience of the developer or on a conducted analysis). We accepted the suggestion and made the modification into the checklist.

Beyond the provision of the evidences of privacy and the confirmation of hypothesis H_0, we also noticed an important action that was triggered due the usage of *PbE*: during the "Evaluate Potential Attacks" activity, the developer detected the attack of extracting sensitive information from the memory area of the process and reexecuted the "Evaluate and Apply Privacy Techniques" activity, aiming to mitigate the detected issue. We suspect that, without the attacking phase, the developer would proceed without detecting the threat.

We also observed that, in some situations, the developer may have confused the concepts of security and privacy. For example, we consider the case of the attacker extracting the easy password and the MAC address of a user as a security issue. The attacker would have access in place of the victim, but in the first place it is not a privacy issue. Of course that privacy

violations may be derived from this issue, but indirectly. Thus, we believe that would be appropriate to consider these mitigations as "security assumptions", and keep the efforts on the privacy issues from the defined scope.

The confusion between the concepts of security and privacy suggests that, before starting the development, it would be useful for the developers to have a small period of training regarding security, privacy, and their differences. The duration and the content of the training would depend on the application context and the experience of the team. Recall that some authors divide the goals of information security as availability, integrity and confidentiality. Also, some authors classify privacy as a sub-goal of confidentiality [92; 86]. Lauren Henry [86] presents a discussion about the differences between information security and privacy, showing that they have separate objectives and sometimes can even be in opposition to each other (known as a Privacy vs. Security trade-off).

From the data manipulated by the two factor authentication system, we see examples of privacy as discovering who was where, when, and with whom. Such information may be extracted from the “30 minute MAC Table” stored into the process memory area. Therefore, we consider that the development team was right in defining the usage of Intel SGX as a privacy technique.

We do not consider the designed salted hashes scheme as secure. The *evil admin* is still able to perform the following attack:

- Get the username of the victim and the salted hash of the easy password and replace into the values of the *salt* and *hash_easy_pwd* variables from the Python script below. Running in a machine with i7-7600U CPU of 2.80GHz and four cores, this script takes on average 98 seconds to find the correct easy password. This time is lower if using multiple threads.

```
1 from passlib.hash import pbkdf2_sha256
2
3 salt = "Dalton"
4 hash_easy_pass = "$pbkdf2-sha256$29000$RGFsdG9u$yYUBIt.MIfJOmILOy
                                        J0OtXOp8CRbk6U7aBuubTraWAw"
5
6 for i in range(0, 9999 + 1):
7   easy_pass = "%04d" % i
```

```
8     if pbkdf2_sha256.hash(easy_pass, salt=salt) == hash_easy_pass:
9       print("Found easy password: " + easy_pass)
10      break
```

- Identify the vendor of the mobile phone of the victim or guess it (*e.g.*, Samsung, Apple or Sony). The IEEE Standards Association defines a MAC address as six bytes, where the first three bytes identifies the vendor[1](named as *OUI - Organizationally Unique Identifier*) and the last three bytes identifies the device. Knowing the *OUI* of the vendor of the victim's mobile phone, the easy password discovered in the previous step, and the salted hash of the MAC address, replace into the values of *oui*, *salt* and *hash_mac* variables from the Python script below. Running the script in a machine with i7-7600U CPU of 2.80GHz and four cores, this script takes on average 24 hours, 17 minutes and 4 seconds, to find the correct MAC address.

```
1  from passlib.hash import pbkdf2_sha256
2  from threading import Thread
3  import os
4
5  oui = "\x00\x28\xf8"
6  salt = "1234"
7  hash_mac = "$pbkdf2-sha256$29000$MTIzNA$9zqSyuzuJg6p2MV/.gcKI1zNfW
                                        joBJCqDP5o2expMu4"
8
9  def brute(begin, end):
10   for b1 in range(begin, end):
11     for b2 in range(0, 0xff + 1):
12       for b3 in range(0, 0xff + 1):
13         mac = oui + chr(b1) + chr(b2) + chr(b3)
14         if pbkdf2_sha256.hash(mac, salt=salt) == hash_mac:
15           print("Found MAC address: " + mac.encode("hex"))
16           os._exit(0)
17
18 delta = (0xff + 1)/4
19
20 Thread(target=brute, args=(0, delta)).start()
```

[1] `https://macvendors.com/` presents a tool to query for MAC address vendors

```
21 Thread(target=brute, args=(delta, 2*delta)).start()
22 Thread(target=brute, args=(2*delta, 3*delta)).start()
23 Thread(target=brute, args=(3*delta, 4*delta)).start()
```

- With the MAC address obtained in the previous step, change the MAC address of the wireless network interface. Most of the Linux distributions provide the *macchanger*[2] command in their repository. For example, the command below changes the current MAC address to *00:28:f8:4f:38:13*.

  ```
  sudo macchanger --mac 00:28:f8:4f:38:13 wlan0
  ```

  After following these steps, and sending the obtained easy password to the server, the attacker obtains access to the rest of the system.

Despite using a recommended hash algorithm (PBKDF2 has additional computational cost, aimed to make brute force attacks much slower), the search spaces of easy passwords and MAC addresses are small. Also the usage of a fixed salt (*username*) is not recommended. These facts reinforce the need of training developers in security and privacy aspects, before starting the development.

We do not consider the presence of security vulnerabilities as an invalidation of the *PbE* methodology. First, we acknowledge that there is no 100% secure system and actually this is our main motivation for providing the mitigations just as "evidences of privacy", not guarantees. Second, the unauthorized access, in principle, may not be considered as a privacy issue, as we mentioned before. Although security issues may impact privacy, the recommendation of *PbE* is to delegate security responsibilities to a separate team, considering them as assumptions. However, the developer took care on protecting the table containing the timestamps and the MAC addresses of the current people (*i.e.*, the data that may contain the sensitive information of who was where, when, and with whom) and because of this, we consider that *PbE* was validated in achieving its goal.

[2]GNU MAC Changer `http://www.gnu.org/software/macchanger`

Chapter 8

Final Considerations

The internet is connecting more devices every day and this growth carries several benefits. However, there are many concerns of privacy. For a company, being able to state that its product is "privacy-friendly" is a competitive advantage. When carrying forward the concern of privacy preserving, there are rules to follow and techniques to apply. Unfortunately, there is a lack of methodologies to guide the development, applying in practice the rules and privacy techniques.

Privacy by Design (*PbD*) consists in 7 foundational principles to be followed when developing systems that should take into account users privacy. However, these principles alone are not enough. They are still vague and leave many open questions on how to apply them in practice. In this work, we aimed to develop a generalizable methodology to fill this gap.

We identified the key concepts to be considered when dealing with privacy-friendly applications and described a conceptual framework, aiming to generate a manageable knowledge to be used in our proposed methodology. The concepts are: participants (users, project owner, development team, regulatory agency, privacy engineer, attacker and data recipient); application context; data formats; data sensitivity; privacy norms and legislation; users perception of privacy; privacy techniques; attacks; privacy models; data utility; and security.

We proposed the concept of *Privacy by Evidence* (*PbE*), a software development methodology to provide privacy assurance. *PbE* is in accordance with all the 7 principles defined by *PbD*, and therefore, *PbE* may be considered as an extension of *PbD*. *PbE* consists in the execution of five activities in parallel with the development of the application. The activities are: identify the application context and data format; check compliance with norms and

legislation; identify utilities and risk assessment; evaluate and apply privacy techniques; and evaluate potential attacks. Given the difficulty in providing total privacy (*i.e.*, free of vulnerabilities), we propose to document the mitigations in form of evidences, aiming to increase the confidence of the project. The evidences should be structured using the *Goal Structuring Notation* (*GSN*), a graphical argumentation notation that can be used to document explicitly the individual elements of any argument and, perhaps more significantly, the relationships that exist between these elements. Arguments documented using *GSN* can help to provide privacy assurance.

To validate the effectiveness of *PbE*, we conduct four case studies, executing the activities described here in smart energy metering, people counting, energy efficiency, and a two factor authentication applications. These are examples of applications that are growing in numbers, and their usage may bring privacy risks, causing many people and even the media to show distrust about them.

The main objective of these case studies is to assess the ability of *PbE* in producing evidences of privacy. In order to achieve this objective we defined three research questions using the *Goal, Question, Metric* (*GQM*) paradigm, a mechanism for defining and evaluating goals using measurement. The defined research questions were:

- **RQ1**: Is *PbE* helpful to develop a privacy-friendly smart energy metering application?
- **RQ2**: Is *PbE* helpful to develop a privacy-friendly people counting application?
- **RQ3**: Is *PbE* helpful to develop a privacy-friendly energy efficiency application?
- **RQ4**: Is *PbE* helpful to develop a privacy-friendly two factor authentication application?

To answer these three questions, we use the metrics of number of provided evidences and their sum of weights. For the smart energy metering application (LiteMe), using the *PbE*, we were able to provide seven evidences of privacy, and the sum of their weights resulted in 28. For the people counting application (Pulso), we were able to provide five evidences and the sum of their weights resulted in 19. For the energy efficiency application (Lumen), we were able to provide five evidences, and the sum of their weights resulted in 19 too. The two factor authentication was developed by an external team and we used the Think Aloud protocol to

observe the process. In this case study, the team provided four evidences of privacy, and the sum of their weights resulted 86. We positively support all the four research questions ($RQ1$, $RQ2$, $RQ3$ and $RQ4$) and claim that *PbE* is an effective way to develope privacy-friendly applications.

During the condution of the case studies, we had many lessons learned and received many feedbacks in order to improve the methodology. For example, the activity of "Identify Utilities and Risk Assessment" demonstrated to be hard and subjective. Therefore, the condution of privacy perception studies is a useful task to help in this activity and to identify and quantify privacy concerns. Another insight was that the activity of "Evaluate Potential Attacks" demonstrated to be important when developing privacy-friendly applications. The identification of working attacks in this activity and the need to return to previous activities in order to mitigate the risks indicate that without the attack evaluations, developers would just proceed to deployment/production phases. Another insight was that less experienced developers may confuse the concepts of security and privacy. This suggests that, before starting the development, it would be useful for the developers to have a small period of training regarding security, privacy, and their differences.

Based on the experience obtained from the analysis of the case studies, we noticed that all the artifacts presented in the checklist of Table 3.1 are important. However, a minimal set of artifacts that we recommend would be the following: *Engagement Report, Summary of Norms, Privacy Concerns, Adversary Model, Summary of Techniques, Implementation of Techniques,* and *Summary of Attacks*. This minimal set was defined based on the need to maintain the usual workflow of *PbE* and to allow the privacy understanding and the actual state by the participants of the project.

A limitation of this work is that we did not conduct a quantitative experiment with a set development teams using *PbE* and another set of teams not using *PbE*. That experiment would provide results with statistical confidence regarding the effectiveness of *PbE* in developing privacy-friendly applications. Therefore, we leave the conduction of this experiment as a possible future work.

Bibliography

[1] Myths about /dev/urandom. https://www.2uo.de/myths-about-urandom/, March 2018.

[2] OASIS Privacy Management Reference Model (PMRM) TC. https://www.oasis-open.org/committees/pmrm, March 2018.

[3] The EU General Data Protection Regulation (GDPR). https://www.eugdpr.org, February 2018.

[4] The Tor project. https://www.torproject.org, February 2018.

[5] M. Anas, N. Javaid, A. Mahmood, S. M. Raza, U. Qasim, and Z. A. Khan. Minimizing electricity theft using smart meters in AMI. In *Proc. of the IEEE Seventh International Conf. on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, pages 176–182, Victoria, Canada, August 2012.

[6] I. Anati, S. Gueron, S. Johnson, and V. Scarlata. Innovative technology for CPU based attestation and sealing. https://software.intel.com/en-us/articles/innovative-technology-for-cpu-based-attestation-and-sealing, August 2013.

[7] M. Andrew, M. Malte, L. Kevin, and N. Arvind. An Empirical Analysis of Linkability in the Monero Blockchain. https://arxiv.org/abs/1704.04299, April 2017.

[8] M. Backes and S. Meiser. Differentially Private Smart Metering with Battery Recharging. In *Data Privacy Management and Autonomous Spontaneous Security*, pages 194–212, New York, NY, USA, 2014. Springer-Verlag New York, Inc.

[9] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (Im)Possibility of Obfuscating Programs. 59:6:1–6:48, May 2012.

[10] M. Barbaro and J. T. Zeller. A face is exposed for AOL searcher no. 4417749. New York Times. http://www.nytimes.com/2006/08/09/technology/09aol.html, August 2006.

[11] P. Barbosa, A. Brito, and H. Almeida. Defending Against Load Monitoring in Smart Metering Data Through Noise Addition. In *Proc. of the 30th Annual ACM Symposium on Applied Computing (SAC)*, pages 2218–2224, Salamanca, Spain, April 2015.

[12] P. Barbosa, A. Brito, and H. Almeida. A Technique to provide differential privacy for appliance usage in smart metering. *Information Sciences*, 370-371:355–367, November 2016.

[13] P. Barbosa, A. Brito, H. Almeida, and S. Clauß. Lightweight privacy for smart metering data by adding noise. In *Proc. of the 29th Annual ACM Symposium on Applied Computing (SAC)*, pages 531–538, Gyeongju, South Korea, March 2014.

[14] P. Barbosa, J. L. Silva, A. Brito, and L. Silva. Privacy Preserving Techniques in Smart Metering: An Overview. In *XVI Brazilian Symposium on Information and Computational Systems Security (SBSeg)*, pages 30–43, Rio de Janeiro, Brazil, November 2016.

[15] G. Barnett. Harnessing Data in the Internet of Things, Strategies for managing data in a connected world. https://tdwi.org/whitepapers/2015/07/ibm_harnessing-data-in-the-internet-of-things.aspx, 2015.

[16] L. Baruh and Z. Cemalcilar. It is more than personal: Development and validation of a multidimensional privacy orientation scale. *Personality and Individual Differences*, 70:165 – 170, 2014.

[17] R. V. Basili. Software Modeling and Measurement: The Goal/Question/Metric Paradigm. Technical report, College Park, MD, USA, 1992.

[18] N. Batra, H. Dutta, and A. Singh. INDiC: Improved Non-intrusive Load Monitoring Using Load Division and Calibration. In *IEEE 12th International Conf. on Machine Learning and Applications (ICMLA)*, volume 1, pages 79–84, Dec 2013.

[19] N. Batra, J. Kelly, O. Parson, H. Dutta, W. Knottenbelt, A. Rogers, A. Singh, and M. Srivastava. NILMTK: An Open Source Toolkit for Non-intrusive Load Monitoring. In *Proceedings of the 5th International Conference on Future Energy Systems*, e-Energy '14, pages 265–276, New York, NY, USA, 2014. ACM.

[20] R. Beckwith. Designing for ubiquity: The perception of privacy. *IEEE Pervasive computing*, 2(2):40 – 46, 2003.

[21] M. Bellare and O. Goldreich. On Defining Proofs of Knowledge. In *12th Annual International Cryptology Conference*, pages 390–420, Santa Barbara, CA, USA, August 1993. Springer Berlin Heidelberg.

[22] A. R. Beresford, D. Kübler, and S. Preibusch. Price versus privacy: an experiment into the competitive advantage of collecting less personal information. In *forthcoming in Electronic Commerce Research*, August 2013.

[23] J. Bhatia, T. D. Breaux, and F. Schaub. Mining Privacy Goals from Privacy Policies Using Hybridized Task Recomposition. *ACM Trans. Softw. Eng. Methodol.*, 25(3):22:1–22:24, May 2016.

[24] A. Blum, K. Ligett, and A. Roth. A Learning Theory Approach to Non-interactive Database Privacy. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 609–618, New York, NY, USA, 2008. ACM.

[25] C. Boccuzzi. Smart Grid e o big brother Energético. *Metering International América Latina*, 3:82–83, 2010.

[26] J. Bohli, C. Sorge, and O. Ugus. A privacy model for smart metering. In *Proc. IEEE International Conf. Communications Workshops (ICC)*, pages 1–5, Cape Town, South Africa, May 2010.

[27] T. Boren and J. Ramey. Thinking aloud: Reconciling theory and practice. 43:261 – 278, October 2000.

[28] C. Bowman, A. Gesher, J. K. Grant, and D. Slate. *The Architecture of Privacy: On Engineering Technologies that can Deliver Trustworthy Safeguards*. Oreilly and Associate Series. O'Reilly Media, Incorporated, 2015.

[29] R. Boyd. *Getting Started with OAuth 2.0*. Oreilly and Associate Series. O'Reilly Media, Incorporated, 2012.

[30] N. Busom, R. Petrlic, F. Sebé, C. Sorge, and M. Valls. Efficient smart metering based on homomorphic encryption. *Computer Communications*, pages 95–101, September 2015.

[31] A. Cavoukian. *Privacy by Design - The 7 Foundational Principles*, August 2009.

[32] CER. Commission for Energy Regulation – CER smart metering project. Smart Meter Electricity Trial data, December 2012.

[33] D. Chaum. The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability. *Journal of Cryptology.*, 1(1):65–75, March 1988.

[34] S. Chawla, C. Dwork, F. McSherry, A. Smith, and H. Wee. Toward Privacy in Public Databases. In *Proceedings of the Second International Conference on Theory of Cryptography*, TCC'05, pages 363–385, Berlin, Heidelberg, 2005. Springer-Verlag.

[35] T. Dalenius. Finding a Needle In a Haystack or Identifying Anonymous Census Records. *Journal of Official Statistics*, 2(3):329–336, 1986.

[36] Datameer. Big Data Analytics and the Internet of Things. https://www.datameer.com/pdf/eBook-Internet-of-Things.pdf, 2015.

[37] C. Dwork. Differential Privacy. In *Proc. of the 33rd International Colloquium on Automata, Languages and Programming, part II (ICALP)*, pages 1–12, Venice, Italy, July 2006.

[38] C. Dwork. Differential Privacy: A Survey of Results. In *Proc. of the 5th International Conf. of Theory and Applications of Models of Computation (TAMC)*, pages 1–19, Xi'an, China, April 2008.

[39] EDPS. European Data Protection Supervisor – Opinion on coherent enforcement of fundamental rights in the age of big data, September 2016.

[40] K. E. Emam. *Guide to the de-identification of personal health information*. CRC Press, 2013.

[41] Z. Erkin and G. Tsudik. Private Computation of Spatial and Temporal Power Consumption with Smart Meters. In *Proc. of the 10th Int. Conf. on Applied Cryptography and Network Security (ACNS)*, pages 561–577, June 2012.

[42] B. C. M. Fung, K. Wang, A. W-C. Fu, and P. S. Yu. *Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*. Chapman and Hall/CRC, 2010.

[43] F. D. Garcia and B. Jacobs. Privacy-Friendly Energy-Metering via Homomorphic Encryption. In *Security and Trust Management*, volume 6710, pages 226–238, September 2010.

[44] J. Gehrke, E. Lui, and R. Pass. Towards Privacy for Social Networks: A Zero-knowledge Based Definition of Privacy. In *Proceedings of the 8th Conference on Theory of Cryptography*, TCC'11, pages 432–449, Berlin, Heidelberg, 2011. Springer-Verlag.

[45] The GSN Working Group Online. *Goal Structuring Notation (GSN)*, November 2011. www.goalstructuringnotation.info.

[46] S. Gurses, C. Troncoso, and C. Diaz. Engineering Privacy by Design. In *Computers, Privacy & Data Protection*, page 25, Brusels, Belgium, 2011.

[47] X. He, X. Zhang, and C. C. J. Kuo. A Distortion-Based Approach to Privacy-Preserving Metering in Smart Grids. *IEEE Access*, 1:67–78, May 2013.

[48] M. Hoekstra, R. Lal, P. Pappachan, V. Phegade, and J. Del Cuvillo. Using innovative instructions to create trustworthy software solutions. In *Proceedings of the 2Nd International Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '13, pages 11:1–11:1, New York, NY, USA, 2013. ACM.

[49] D. Ilić, P. G. Silva, S. Karnouskos, and M. Jacobi. Impact assessment of smart meter grouping on the accuracy of forecasting algorithms. In *Proc. of the 28th Annual ACM Symposium on Applied Computing (SAC)*, pages 673–679, Coimbra, Portugal, March 2013.

[50] INMETRO. Portaria número 375, de 27 de Setembro de 2011. http://www.inmetro.gov.br/legislacao/rtac/pdf/RTAC001738.pdf, 2011.

[51] T. Ionescu and G. Engelbrecht. The Privacy Case. In *Joint Workshop on Cyber-Physical Security and Resilience in Smart Grids*, Vienna, AUT, April 2016.

[52] J. S. John. Big data on the smart grid: 2013 in review and 2014 outlook. http://www.greentechmedia.com/articles/read/Big-Datas-5-Big-Steps-to-Smart-Grid-Growth-in-2014, 2013.

[53] G. Kalogridis, C. Efthymiou, S. Z. Denic, T. A. Lewis, and R. Cepeda. Privacy for smart meters: towards undetectable appliance load signatures. In *IEEE 1st International Conf. on Smart Grid Communications (SmartGridComm)*, pages 232–237, Gaithersburg, USA, October 2010.

[54] T. Kelly and R. Weaver. The goal structuring notation - a safety argument notation. In *Proc. of the dependable systems and networks workshop on assurance cases*, Florence, July 2004.

[55] O. Koehle. *Just say no to big brother's smart meters. The latest in bio-hazard technology*. ARC Reproductions, 2012.

[56] J. Z. Kolter and M. J. Johnson. REDD: A public data set for energy disaggregation research. In *Proc. of the ACM workshop on Data Mining Applications in Sustainability*, pages 1–6, USA, August 2011.

[57] N. Koudas, T. Yu, D. Srivastava, and Q. Zhang. Aggregate Query Answering on Anonymized Tables. *2007 IEEE 23rd International Conference on Data Engineering*, pages 116–125, 2007.

[58] K. Lauter, M. Naehrig, and V. Vaikuntanathan. Can Homomorphic Encryption be Practical? In *Proc. of 3rd ACM workshop on Cloud computing security (CCSW)*, pages 113–124, Illinois, USA, October 2011.

[59] J. Lee and C. Clifton. How Much is Enough? Choosing ϵ for Differential Privacy. In *Proc. of the 14th International Conf. on Information Security*, ISC'11, pages 325–340, Xi'an, China, October 2011.

[60] F. Li, B. Luo, and P. Liu. Secure Information Aggregation for Smart Grids Using Homomorphic Encryption. In *IEEE 1st International Conf. on Smart Grid Communications*, pages 327–332, Gaithersburg, USA, October 2010.

[61] J. Li, Y. Tao, and X. Xiao. Preservation of Proximity Privacy in Publishing Numerical Sensitive Data. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 473–486, New York, NY, USA, 2008. ACM.

[62] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. L-diversity: Privacy Beyond K-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1), March 2007.

[63] K. Makanda and J. C. Jeon. Key Distribution Protocol on Quantum Cryptography. In *2013 International Conference on IT Convergence and Security (ICITCS)*, pages 1–2, Dec 2013.

[64] F. McKeen, I. Alexandrovich, A. Berenzon, C. v. Rozas, H. Shafi, V. Shanbhogue, and U. R. Savagaonkar. Innovative instructions and software model for isolated execution. In *Proceedings of the 2Nd International Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '13, pages 10:1–10:1, New York, NY, USA, 2013. ACM.

[65] S. McLaughlin, P. McDaniel, and W. Aiello. Protecting consumer privacy from electric load monitoring. In *Proc. of the 18th ACM Conf. on Computer and Communications Security (CCS)*, pages 87–98, Illinois, USA, October 2011.

[66] R. Mekovec. Online privacy: overview and preliminary research. *Journal of Information and Organizational Sciences*, 34(2):195 – 209, 2010.

[67] I. Miers, C. Garman, M. Green, and A. D. Rubin. Zerocoin: Anonymous Distributed E-Cash from Bitcoin. In *2013 IEEE Symposium on Security and Privacy*, pages 397–411, May 2013.

[68] K. Molokken-Ostvold, N. C. Haugen, and H. C. Benestad. Using planning poker for combining expert estimates in software projects. *J. Syst. Softw.*, 81(12):2106–2117, December 2008.

[69] A. C. Myers and B. Liskov. Protecting Privacy Using the Decentralized Label Model. *ACM Trans. Softw. Eng. Methodol.*, 9(4):410–442, October 2000.

[70] M. E. Nergiz, C. Clifton, and A. E. Nergiz. Multirelational k-Anonymity. *IEEE Trans. on Knowl. and Data Eng.*, 21(8):1104–1117, August 2009.

[71] J. Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[72] L. Ninghui, L. Tiancheng, and S. Venkatasubramanian. *t-Closeness: Privacy beyond k-anonymity and l-diversity*, pages 106–115. 2007.

[73] NIST. National Institute of Standards and Technology – Guidelines for smart grid cybersecurity: Vol. 2 - Privacy and the smart grid, September 2014.

[74] Nokia. *Privacy Engineering & Assurance*, September 2014. www.w3.org/2014/privacyws/pp/Hirsch.pdf.

[75] Object Management Group - OMG. *Structured Assurance Case Metamodel (SACM)*, July 2015. www.omg.org/spec/SACM.

[76] Bankinter Foundation of Innovation. The Internet of Things In a Connected World of Smart Objects. https://www.fundacionbankinter.org/documents/20183/42758/PDF+Internet+of+things/397ad508-7dde-45b6-97cb-459dcb723023, 2011.

[77] I. Oliver. *Privacy Engineering: A Dataflow and Ontological Approach*. 2014.

[78] pidoco. POWERFUL PROTOTYPING. Think Aloud. https://pidoco.com/en/help/ux/think-aloud, 2017.

[79] B. Pinkas, T. Schneider, G. Segev, and M. Zohner. Phasing: Private Set Intersection Using Permutation-based Hashing. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 515–530, Washington, D.C., 2015. USENIX Association.

[80] L. Ponciano, P. Barbosa, F. Brasileiro, A. Brito, and N. Andrade. Designing for Pragmatists and Fundamentalists: Privacy Concerns and Attitudes on the Internet of Things. In *XVI Brazilian Symposium on Human Factors in Computing Systems (IHC)*, Joinville, Brazil, October 2017.

[81] Procel. Manual de tarifação da energia elétrica. Programa nacional de conservação de energia. Eletrobras, August 2011.

[82] B. Raghunathan. *The Complete Book of Data Anonymization*. CRC Press, 2013.

[83] V. Rastogi, D. Suciu, and S. Hong. The Boundary Between Privacy and Utility in Data Publishing. In *Proceedings of the 33rd International Conference on Very Large Data Bases*, VLDB '07, pages 531–542. VLDB Endowment, 2007.

[84] A. Reinhardt, P. Baumann, D. Burgstahler, M. Hollick, H. Chonov, M. Werner, and R. Steinmetz. On the Accuracy of Appliance Identification Based on Distributed Load Metering Data. In *Proc. of the 2nd IFIP Conf. on Sust. Internet and ICT for Sustainability*, pages 1–9, 2012.

[85] A. Samani, H. Ghenniwa, and A. Wahaishi. Privacy in Internet of Things: A Model and Protection Framework. *Procedia Computer Science*, 52:606–613, 2015.

[86] L. H. Scholz. Information Privacy and Data Security. In *Cardozo Law Review de Novo*, June 2015.

[87] K. S. Schwaig, A. H. Segars, V. Grover, and K. D. Fiedler. A Model of Consumers' Perceptions of the Invasion of Information Privacy. *Inf. Manage.*, 50(1):1–12, January 2013.

[88] A. Shamir. How to Share a Secret. *Communications of the ACM*, 22(11):612–613, November 1979.

[89] R. Silva, P. Barbosa, and A. Brito. DynSGX: A Privacy Preserving Toolset for Dinamically Loading Functions into Intel(R) SGX Enclaves. In *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 314–321, Dec. 2017.

[90] R. Singel. Netflix Spilled Your Brokeback Mountain Secret, Lawsuit Claims. https://www.wired.com/2009/12/netflix-privacy-lawsuit/, December 2009.

[91] M. Someren, Y. Barnard, and J. Sandberg. *The Think Aloud Method - A Practical Guide to Modelling CognitiveProcesses*. January 1994.

[92] W. Stallings and L. Brawn. *Computer Security: Principles and Practice*. Pearson, 3 edition, 2015.

[93] J. Stweart, M. Chapple, and D. Gibson. *Certified Information Systems*. SYBEX, 7 edition, 2015.

[94] L. Sweeney. Simple Demographics Often Identify People Uniquely. In *Carnegie Mellon University, Data Privacy*, USA, 2000.

[95] L. Sweeney. k-Anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, pages 557–570, 2002.

[96] U.S. Departament of Health & Human Services. *Guidance Regarding Methods for De-identification of Protected Health Information in Accordance with The Health Insurance Portability and Accountability Act (HIPAA), Privacy Rule*, November 2012.

[97] K. Wang and B. C. M. Fung. Anonymizing Sequential Releases. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 414–423, New York, NY, USA, 2006. ACM.

[98] K. Wang, B. C. M. Fung, and P. S. Yu. Handicapping Attacker's Confidence: An Alternative to K-anonymization. *Knowl. Inf. Syst.*, 11(3):345–368, April 2007.

[99] S. Wang, L. Cui, J. Que, D.-H. Choi, X. Jiang, and L. Xie. A randomized response model for privacy preserving smart metering. *IEEE Trans. on Smart Grid*, 3:1317–1324, September 2012.

[100] G. Waters. Conquering Advanced Metering Cost and Risk. *Electric Energy T&D Magazine*, 10:22–25, November 2006.

[101] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering*. Kluwer Academic Publishers, 2000.

[102] R. C. Wong, J. Li, A. W. Fu, and K. Wang. (α, K)-anonymity: An Enhanced K-anonymity Model for Privacy Preserving Data Publishing. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 754–759, New York, NY, USA, 2006. ACM.

[103] X. Xiao and Y. Tao. Personalized Privacy Preservation. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, SIGMOD '06, pages 229–240, New York, NY, USA, 2006. ACM.

[104] M. Z. Yao, R. E. Rice, and K. Wallis. Predicting user concerns about online privacy. *Journal of the American Society for Information Science and Technology*, 58(5):710 – 722, 2007.

[105] L. Ying-Xun, L. Chin-Feng, H. Yueh-Min, and C. Han-Chieh. Multi-appliance recognition system with hybrid SVM/GMM classifier in ubiquitous smart home. *Information Sciences*, 230:39–55, 2013.

[106] Zebra. Analysis of iOS 8 MAC Randomization on Locationing. http://mpact.zebra.com/ documents/iOS8-White-Paper.pdf, 2015.

[107] J. Zhao, T. Jung, Y. Wang, and X. Li. Achieving differential privacy of data disclosure in the smart grid. In *Proc. of IEEE INFOCOM*, pages 504–512, April 2014.

Appendix A

Catalog of Privacy Techniques

This appendix presents a catalog of privacy techniques, aiming to guide developers in choosing the most suited ones, according to their specific context. The catalog contains a number of techniques for manipulating and measuring information for the purpose of enhancing privacy which can be applied together as required to architect a solution. This list is not exaustive, and is based on the experience of the author.

Privacy Technique	**Comments**
Asymmetric Encryption	Uses two keys, one to encrypt and another one to decrypt [92], [82]. Examples of asymmetric encryption algorithms are RSA, Elgamal and ECC.
Symmetric Encryption	Same key used to encrypt and decrypt [92], [82]. Examples of symmetric encryption algorithms are AES, 3DES and RC4.
Hashing	Hash functions are typically one-way functions such that computing their inverse is computationally expensive [92], [82], [77]. Examples of hashing algorithms are SHA256 and scrypt.
Homomorphic Encryption	A form of encryption that allows computation on ciphertexts, generating an encrypted result which, when decrypted, matches the result of the operations as if they had been performed on the plaintext [58], [60], [43], [41], [30].
Secret Exchange	Enables two parties to produce a shared random secret known only to them, which can, for example, be used as a key to encrypt and decrypt messages. Examples of algorithms are Diff-Hellman and QKD [63].
Secret Sharing	Splits a secret amongst a group of participants, each of whom is allocated a share of the secret. The secret can be reconstructed only when a sufficient number of shares are combined together; individual shares are of no use on their own [88].

Zero-knowledge proof	A method by which one party (the prover) can prove to another party (the verifier) that a given statement is true, without conveying any information apart from the fact that the statement is indeed true [21].
Private Set Intersection	Allows two parties to compute the intersection of private sets while revealing nothing more than the intersection itself [79].
Trusted Execution Environment	Allows software developers control of the security of sensitive code and data by creating trusted domains within applications to protect critical information during execution and at rest [48], [64].
Dynamic Private Code Execution	Loads and unloads code to be executed in a trusted execution environment. Protects application code and libraries, defending against reverse engineering [89].
Code Obfuscation	Makes code difficult for humans to understand and programmers may deliberately obfuscate code to conceal its purpose (security through obscurity), primarily, in order to prevent reverse engineering [9].
Access Control	Means of ensuring proportionality in data access, controlling data usage, and enhancing security beyond the broad system-access level [28].
Generalization	Replaces a specific value with a general value, according to a given taxonomy. For a numerical attribute, exact values can be replaced with an interval that covers exact values [42]. The usage of this technique usually provides K-anonymity properties.
Pseudonymity	The act of hiding the identity through the usage of false ones [42].
Supression	Remove values or replace them with a special value, indicating that the replaced values are not disclosed [42].
Anatomization	De-associates the relationship between the sensitive data and the identities into different groups. The major advantage is that the data in both sets are unmodified [42].
Permutation	Same idea of anatomization, but after de-associating, shuffles the sensitive values within each group [42].
Noise Addition	The general idea is to replace the original sensitive value s with $s + r$ where r is a random value drawn from some distribution [42], [12]. The usage of this technique usually provides Differential-Privacy properties.
Data Swapping	The general idea is to exchange values of sensitive attributes among individual records while the swaps maintain the low-order frequency counts or marginals for statistical analysis [42], [82].
Synthetic Data	The general idea is to build a statistical model from the data and then to sample points from the model. These sampled points form the synthetic data, to be used instead of the original data [42].

DC-Nets	Provide anonymity of both sender and receiver by establishing some anonymous channels for message transmission [33].
Relay Network	Enables anonymity in the network for users (*e.g.*, Tor browser) and servers (*e.g.*, onion services) [4], defending against network surveillance and traffic analysis.
Private cryptocurrencies	Some cryptocurrencies, such as Bitcoin, lack privacy guarantees. However, other proposals that allow anonymity such as Zerocoin [67] and Monero [7] have been suggested.

Appendix B

Other Privacy Techniques in Smart Metering

B.1 Rechargeable Batteries

Rechargeable batteries between appliances and smart meters can help to reduce the privacy issues as the appliance signatures are no longer legible [8; 53; 65; 107].

Mclaughlin *et al.* [65] propose an approach called *Non-Intrusive Load Leveling* (*NILL*). The goal of a *NILL* system is to level the load profile to a constant *target load*, thus removing appliance signatures. When an appliance turns ON, it will exert a load beyond the target load. Thus, *NILL* will discharge the battery to partially supply the load created by the appliance, maintaining the target load. Similarly, if an appliance enters the OFF state, the load profile will decrease below the target load. These opportunities are used to charge the battery while restoring the target load.

The *NILL* system consists of two parts: a battery and a control system that regulates the battery's charge and discharge based on the present load and battery state. The controller attempts to maintain a steady state target load K_{SS}, but will go into one of two special states K_L or K_H if the battery needs to recover from a low or high state of charge.

The essence of *NILL* is described by the equation, $u(t) = d(t) + b(t)$, where $b(t)$ is the battery's rate of charge overtime, $d(t)$ is the actual load profile of the residence and $u(t)$ is the load under the influence of *NILL* as perceived by the smart meter and what is disclosed to the power provider. If $b(t) > 0$, the battery is charging, otherwise $b(t) < 0$ and the battery

is discharging. Finally, $c(t)$ is used to represent the battery's state of charge, thus:

$$c(t) = \int_{t_0}^{t} b(t)dt + c(t_0) \; .$$

Therefore, $c(t)$ is monitored. If $c(t) < L$, where L is the lower safe limit on the battery's state of charge, then the battery needs to be recharged and the system goes to the K_L state. Similarly, if $c(t) > H$, the system goes to the K_H state and the battery is discharged.

B.2 Using a Modified ElGamal Encryption

The first homomorphic encryption solution that we consider is based on a modification in the ElGamal encryption, a cryptographic system that relies on the discrete logarithm problem. The ElGamal cryptosystem proceeds as follows:

- *Set up*: a large prime q is chosen. Next, a generator g of the cyclic group $\mathbb{Z}_q^*$ is selected.
- *Key generation*: a secret key x is generated by setting its value as a random number $x \in_R \mathbb{Z}_q^*$. The corresponding public key is computed as $y = g^x$.
- *Encryption*: a message $m \in G$ is encrypted under public key y by taking a random number $r \in_R \mathbb{Z}_q^*$ and computing $c = g^r$ and $d = m \cdot y^r$. The ElGamal encryption of m under public key y, $E_y(m)$, is the tuple (c, d).
- *Decryption*: a ciphertext $E_y(m)$ is decrypted using the private key x by computing $m = d \cdot c^{-x}$.

Given messages m_1 and m_2, we can obtain an encryption of $m_1 \cdot m_2$ by computing:

$$\begin{aligned} E_y(m_1) \cdot E_y(m_2) &= (c_1 \cdot c_2, d_1 \cdot d_2) \\ &= (g^{r_1+r_2}, m_1 \cdot m_2 \cdot y^{r_1+r_2}) \\ &= E_y(m_1 \cdot m_2) \; . \end{aligned}$$

Hence, ElGamal is a multiplicative homomorphic cryptosystem.

To calculate the total consumption in a region, Busom *et al.* [30] propose a protocol which uses an additive ElGamal cryptosystem. Given $E_y(g^{m_1})$ and $E_y(g^{m_2})$, then, $E_y(g^{m_1}) \cdot E_y(g^{m_2}) = E_y(g^{m_1} \cdot g^{m_2}) = E_y(g^{m_1+m_2})$.

Initially, each smart meter possess the following values: a big prime number q and its generator g; a secret key x_i; a public key $y_i = g^{x_i}$. To encrypt the measurements, it is necessary a global public key $y = \prod_{i=1}^{N} y_i$.

Let m_i denote the measurement of a smart meter. To calculate the total consumption in the region, the following protocol is executed:

1. Each meter generates a random noise value $z_i \in \mathbb{Z}_q^*$ and computes a ciphertext as $C_i = E_y(g^{m_i+z_i}) = (c_i, d_i)$ which is sent to the aggregator (which can be the power provider).

2. The aggregator combines all the messages as $C = (\prod_{i=1}^{N} c_i, \prod_{i=1}^{N} d_i) = (c, d)$ and sends c to each meter.

3. Each meter computes $T_i = c^{x_i} \cdot g^{z_i}$ and sends the result to the aggregator. After that, each meter removes z_i from its memory.

4. Finally, the aggregator computes $D = d \cdot (\prod_{i=1}^{N} T_i)^{-1}$ and $log_g D = M = \sum_{i=1}^{N} m_i$, where M is the total consumption in the region.

Notice that, since M is a relatively small number, the discrete logarithm problem in step 4 can be solved in a short time. In step 2, the aggregator computes:

$$C = (\prod_{i=1}^{N} g^{r_i}, \prod_{i=1}^{N} g^{m_i+z_i} \cdot y^{r_i}) = (g^r, g^{M+z} \cdot y^r) = (c, d) \ ,$$

and in step 3, each meter computes:

$$T_i = c^{x_i} \cdot g^{z_i} = g^{r \cdot x_i} \cdot g^{z_i} = g^{x_i \cdot r} \cdot g^{z_i} = y_i^r \cdot g^{z_i} \ .$$

Therefore, the protocol works because in step 4 the aggregator computes:

$$D = d \cdot (\prod_{i=1}^{N} T_i)^{-1} = \frac{g^{M+z} \cdot y^r}{\prod_{i=1}^{N} (y_i^r \cdot g^{z_i})} = \frac{g^{M+z} \cdot y^r}{(\prod_{i=1}^{N} y_i^r) \cdot g^z} = \frac{g^{M+z} \cdot y^r}{g^z \cdot y^r} = g^M \ .$$

B.3 Using Paillier Encryption and Secret Sharing

A protocol based on Paillier encryption and *secret sharing* was proposed by Garcia *et al.* [43]. The Paillier cryptosystem proceeds as follows:

- *Set up*: two large primes p and q are chosen, $n = p \cdot q$, and $\lambda = lcm(p-1, q-1)$. A random number $g \in_R \mathbb{Z}^*_{n^2}$ is chosen in such a way that $gcd(b, n) = 1$, where $b = L(g^\lambda \ mod \ n^2)$ and $L(u) = \frac{(u-1)}{n}$.

- *Key generation*: let μ be the modular multiplicative inverse of b modulo n, *i.e.*, $\mu = b^{-1} \ mod \ n$. Thus, the public key is $P_k = (n, g)$ and the private key is $S_k = (n, \lambda, \mu)$.

- *Encryption*: a message m is encrypted under public key P_k by taking a random number $r \in_R \mathbb{Z}^*_{n-1}$ and computing $E_{P_k}(m) = g^m \cdot r^n \ mod \ n^2$.

- *Decryption*: a ciphertext $c = E_{P_k}(m)$ is decrypted using the private key S_k by computing $m = L(c^\lambda \ mod \ n^2) \cdot \mu \ mod \ n$.

Given messages m_1 and m_2, we can obtain an encryption of $m_1 + m_2$ by computing:

$$\begin{aligned} E_{P_k}(m_1) \cdot E_{P_k}(m_2) &= g^{m_1} \cdot r_1^n \cdot g^{m_2} \cdot r_2^n \ mod \ n^2 \\ &= g^{m_1+m_2} \cdot (r_1 \cdot r_2)^n \ mod \ n^2 \\ &= E_{P_k}(m_1 + m_2) \ . \end{aligned}$$

Hence, Paillier is an additive homomorphic cryptosystem.

Garcia *et al*. [43] propose that each smart meter possess a public key P_{ki} and a private key S_{ki}. Let m_i denote the measurement of the meter. To calculate the total consumption in the region, the following protocol is executed:

1. Each meter sends its public key to the aggregator.

2. The aggregator receives all public keys and shares them with all meters. Thus, each meter stays with its private key S_{ki} and all the public keys $\{P_{k1}, P_{k2}, \ldots, P_{kn}\}$.

3. Each meter calculates N secret shares for its measurement m_i, in such a way that $m_i = \sum_{j=1}^{N} s_{ij}$. Then, the meter keeps s_{ii} privately and sends to the aggregator all the other secret shares encrypted with the public keys of the other $N-1$ meters, *i.e.*, it sends $E_{P_{kj}}(s_{ij})$ for $j = 1, \ldots, i-1, i+1, \ldots, N$.

4. After receiving all the encrypted secret shares, the aggregator multiplies the ones encrypted with the same public key. Due to the Paillier homomorphic property, for each

meter i, it has $E_{P_{ki}}(m'_i) = \prod_{j \neq i}^{N} E_{P_{ki}}(s_{ji}) = E_{P_{ki}}(\sum_{j \neq i}^{N} s_{ji})$. Then, the aggregator sends $E_{P_{ki}}(m'_i)$ for each meter i.

5. Using its private key S_{ki}, each meter decrypts $E_{P_{ki}}(m'_i)$ and adds its s_{ii}, obtaining $\sum_{j=1}^{N} s_{ji}$. The meter then sends this value to the aggregator.

6. Finally, the aggregator can sum all the received values, obtaining the total consumption in the region $M = \sum_{i=1}^{N} \sum_{j=1}^{N} s_{ji}$.

In this approach, the total consumption in the region is computed and at the same time, neither the aggregator nor any other consumer has access to any real measurement from a consumer, for they can only access random shares. Since in step 6 the aggregator simply sum all the secret shares, the proof that the protocol works is straightforward.

B.4 Using a Modified Paillier Encryption

Erkin *et al.* [41] propose a protocol based on a modification in the Paillier encryption. Starting, there is a single pair of Paillier keys (P_k and S_k) shared with all N meters. Let m_i denote the measurement of the meter. To calculate the total consumption in the region, the following protocol is executed:

1. Each meter generates $N - 1$ random numbers, one for every one of the other meters, and sends them using secure communication (*e.g.*, RSA encryption between meters). Thus, there is a total of $N \cdot (N - 1)$ message exchanges in this step.

2. After receiving all the random numbers generated by the other meters, the meter computes $R_i = n + \sum_{j \neq i}^{N} r_{(i \to j)} - \sum_{j \neq i}^{N} r_{(j \to i)}$, where n is the Paillier modulo and $r_{(i \to j)}$ is the random number generated by the meter i for the meter j.

3. Following, the meter computes a hash h_t using the timestamp of the current measurement m_i. This hash must be coprime with the Paillier modulo n, *i.e.*, $gcd(h_t, n) = 1$. Since the timestamp is synchronized, the obtained hash is the same for all meters.

4. After computing R_i and h_t, the meter encrypts m_i using the following modified scheme of Paillier: $E_{P_k}(m_i) = g^{m_i} \cdot h_t^{R_i}$. Then, this encrypted measurement is disclosed to all other $N - 1$ meters.

5. Finally, after receiving all the encrypted measurements of the other meters, the meter calculates $E_{P_k}(M) = \prod_{i=1}^{N} E_{P_k}(m_i) = E_{P_k}(\sum_{i=1}^{N} m_i)$. This is true due the homomorphic property.

Possessing $E_{P_k}(M)$, the meter can decrypt this value and then send the total consumption in the region M to the aggregator. This way, the total consumption is computed and privacy is preserved, for the meter does not have access to the other measurements in plaintext.

The protocol works because in step 5, the meter computes:

$$E_{P_k}(M) = g^{m_1+m_2+\cdots+m_N} \cdot h_t^{(\sum_{i=1}^{N} n)+(\sum_{i=1}^{N}\sum_{j\neq i}^{N} r_{(i\to j)})-(\sum_{i=1}^{N}\sum_{j\neq i}^{N} r_{(j\to i)})},$$

and

$$E_{P_k}(M) = g^{M} \cdot h_t^{N\cdot n}\,.$$

Considering that $r = h_t^N$, this configuration represents the original paillier cryptosystem.