

δ -Risk: Towards context-aware multi-objective privacy management in connected environments

Application to smart IoT healthcare domain

KARAM BOU-CHAAYA, Univ Pau & Pays Adour, E2S/UPPA, LIUPPA, EA3000, France

RICHARD CHBEIR, Univ Pau & Pays Adour, E2S/UPPA, LIUPPA, EA3000, France

MANSOUR NASER ALRAJA, Dhofar University, Oman

PHILIPPE ARNOULD, Univ Pau & Pays Adour, E2S/UPPA, LIUPPA, EA3000, France

CHARITH PERERA, Cardiff University, United Kingdom

MAHMOUD BARHAMGI, Université Claude Bernard Lyon 1, LIRIS lab, France

DJAMAL BENSLIMANE, Université Claude Bernard Lyon 1, LIRIS lab, France

In today's highly connected cyber-physical environments, users are becoming more and more concerned about their privacy and ask for more involvement in the control of their data. However, achieving effective involvement of users requires improving their privacy decision-making. This can be achieved by: (i) raising their awareness regarding the direct and indirect privacy risks they accept to take when sharing data with consumers; (ii) helping them in optimizing their privacy protection decisions to meet their privacy requirements while maximizing data utility. In this paper, we address the second goal by proposing a user-centered multi-objective approach for context-aware privacy management in connected environments, denoted δ -Risk. Our approach features a new privacy risk quantification model to dynamically calculate and select the best protection strategies for users based on their preferences and contexts. Computed strategies are optimal in that they seek to closely satisfy user's requirements and preferences while maximizing data utility and minimizing the protection cost. We implemented our proposed approach and evaluated its performance and effectiveness based on several use cases. Results show that δ -Risk: (1) handles privacy reasoning in real-time, which makes it able to support the user in various contexts, including ephemeral ones; and (2) always provides the user with at least one best strategy per context.

Additional Key Words and Phrases: Privacy management, Privacy risk quantification, Privacy by Design, Context-aware computing, Semantic reasoning, Internet of Things

ACM Reference Format:

Karam Bou-Chaaya, Richard Chbeir, Mansour Naser Alraja, Philippe Arnould, Charith Perera, Mahmoud Barhamgi, and Djamel Benslimane. 2020. δ -Risk: Towards context-aware multi-objective privacy management in connected environments: Application to smart IoT healthcare domain.

Authors' addresses: Karam Bou-Chaaya, Univ Pau & Pays Adour, E2S/UPPA, LIUPPA, EA3000, Anglet, France, karam.bou-chaaya@univ-pau.fr; Richard Chbeir, Univ Pau & Pays Adour, E2S/UPPA, LIUPPA, EA3000, Anglet, France, richard.chbeir@univ-pau.fr; Mansour Naser Alraja, Dhofar University, Salalah 2509, Oman, malraja@du.edu.om; Philippe Arnould, Univ Pau & Pays Adour, E2S/UPPA, LIUPPA, EA3000, Mont-de-Marsan, France, philippe.arnould@univ-pau.fr; Charith Perera, Cardiff University, United Kingdom, charith.perera@ieee.org; Mahmoud Barhamgi, Université Claude Bernard Lyon 1, LIRIS lab, Lyon, France, mahmoud.barhamgi@univ-lyon1.fr; Djamel Benslimane, Université Claude Bernard Lyon 1, LIRIS lab, Lyon, France, djamal.benslimane@univ-lyon1.fr.

1 INTRODUCTION

Advances in the fields of ubiquitous computing (e.g., Internet of Things), sensing technologies, and Big Data have allowed the fast evolution of smart connected environments. These environments are defined as physical infrastructures that host Cyber-Physical Systems (CPS), such as sensor networks, interconnected using various communication technologies. These systems are capable of collecting data that could be later processed to provide advanced services. Current CPS-based applications are impacting numerous application domains including healthcare (e.g. patient and elderly monitoring), building/housing (e.g., optimizing energy consumption, occupants' comfort), environmental (e.g., monitoring air and water pollution levels), etc.

Sharing data in exchange for goods and services presents an opportunity for users to improve their quality of life, however, it also exposes them to many privacy risks. In fact, processing and analyzing generated sensor data (e.g., location of individuals, patient's vital signs), which are spatio-temporal in nature [1], can lead to disclose many privacy-sensitive information about users [2, 3], such as health conditions, performed/daily activities, habits, preferences, etc. This disclosure may be intentional if users are aware of it and have entered into agreements with relevant providers. However, it can be harmful if the data/information of users is misused by providers, sold to interested third parties without user consent, or stolen by cybercriminals as providers are often victims of cyber-attacks that lead to data breaches.

Hence, involving users in the control of their privacy protection is currently receiving extensive attention on both legal and technical aspects [4–9]. Nonetheless, existing legal frameworks for data protection (e.g., GDPR [4]) might not necessarily deter data consumers from abusing, intentionally or unintentionally, the data of users. The Facebook-Cambridge Analytica [10] and Exactis [11] scandals are only few examples of a long series of data breach scandals that happened despite the existence of appropriate data protection laws. Moreover, these laws vary among countries, with some providing more protection than others (e.g., GDPR [4] for the European Union, CCPA [5] for the state of California). This makes it more difficult to manage and preserve the privacy of users, especially when users, providers, and third parties are located in different countries governed by different data protection laws. Therefore, all these constraints emphasize the need for user-centric technical solutions that guarantee the same level of privacy protection in all countries.

Current approaches of user-centric privacy preserving [6, 7, 9] have mostly relied on preference specification and policy enforcement, where users specify their privacy preferences and accept policies that enforce these preferences. However, they all share the following limitations:

- (1) *lack of user awareness*. The user may not be completely aware of the direct and indirect privacy risks associated with sharing his data with providers to correctly specify his preferences in the first place. He may simply not know what sensitive information can be revealed from his data when data pieces are analyzed in isolation or combined with each other or/and with side information about the user or his surroundings (e.g., information acquired from external sources such as social networks).
- (2) *lack of context-based privacy decision making*. The data sharing/protection decisions are often made/accepted by the user in a static way. This means that they remain unchanged regardless of the user-context changes. However, the sensitivity of data may vary from a context to another [2, 12], i.e., new privacy risks may emerge as others may lose their significance. This makes static decisions over-protective in some contexts, causing an unnecessary loss of data utility, which can downgrade the accuracy of associated services; or under-protective, which might lead to privacy breaches. Consequently, the user must be able to make dynamic adjustments to his privacy decisions according to the evolution of his context.

The objectives of our research work is to design an appropriate solution that addresses these limitations, and provides a complete Privacy by Design framework capable of: (i) sensitizing the user about the privacy risks involved in sharing his data with consumers; (ii) assisting the user in optimizing his privacy decisions according to his context and preferences; and (iii) securing a full life-cycle protection of the user’s privacy. To overcome the first limitation, we proposed in a previous work [2] a context-aware privacy risk inference approach that provides users with a dynamic overview of the privacy risks they take as their contexts evolve. The computed risk overview is intuitive enough to allow users to understand the implicit, direct and indirect implications of sharing their data with consumers. This paves the way for users to make informed adaptations of their privacy decisions. However, users might not always know the appropriate data protection measures to apply in their context. That is, over-protective measures limit the utility of shared data to eliminate the risks, but could also downgrade the accuracy of services. On the other hand, under-protective measures may improve the accuracy of services, but might also lead to privacy breaches. Hence, determining the optimal protection measure that answers the requirements of the user while maximizing the utility of shared data remains challenging. In addition, what makes it even more challenging is that user-decisions must be fast (i.e. in real-time). Therefore, the optimization solution must be simple (i.e. not complex for the user), fast to support the user in real-time, and scalable according to user preferences and context.

To cope with these challenges, and to answer the second limitation, this paper proposes a new user-centered, context-aware and multi-objective privacy management approach, denoted δ -Risk. The proposed approach assists the user in optimizing his privacy decisions, by providing him with dynamic and best protection strategies that could be adopted in his context. Each of these strategies minimizes the risks inferred in the present context to meet the privacy requirements of the user while maximizing the utility of data and minimizing the protection cost. A delivered strategy is composed of the best combination of protection levels to be assigned to shared attributes according to the user’s preferences and context. To validate our proposal, we developed a Java-based prototype that performs real-time reasoning and generates dynamic/contextual protection strategies. We evaluated the performance of the δ -Risk process by considering several use cases, and we formally studied its effectiveness. Results show that δ -Risk: (1) handles reasoning in real-time, which makes it able to support the user in various contexts, including ephemeral ones (i.e. contexts with small timescale); (2) always identifies all possible appropriate strategies that answer the data utility/privacy protection trade-off; (3) delivers the best strategies to the user; and (4) provides the user with at least one best strategy per context.

The remainder of this paper is organized as follows. Section 2 introduces a scenario that motivates our proposal and identifies the challenges to tackle. Section 3 presents our Privacy Oracle framework, and provides formal definitions of the key concepts used in the paper. Section 4 details the δ -Risk approach. Section 5 outlines the experiments and tests performed. Section 6 highlights the Privacy by Design standard and discusses existing context-aware privacy preserving approaches in connected environments. Finally, Section 7 concludes the paper and discusses future research directions.

2 MOTIVATING SCENARIO

To motivate our proposal, we investigate a real-life scenario to showcase some of the privacy risks that might be resulted from sharing data with consumers, and to emphasize the need for dynamic/contextual adaptations of the user-privacy decisions. Fig.1 illustrates the proposed scenario. Assume that Alice is a COPD (Chronic Obstructive Pulmonary Disease) patient. She pursues her medical treatment remotely using an NIV (Non-Invasive Ventilation) device deployed at home. Consider that Alice shares fine-grained data with the following service providers:

- **Electricity provider:** Alice shares the energy consumption of her home through a deployed smart energy meter. In return, the provider offers Alice personalized recommendations to reduce her energy consumption and bills.
- **Healthcare provider:** Alice shares her real-time location through a mobile application to benefit from an emergency care system. This system provides smart healthcare services, such as a smart ambulance service, that she would use in case of respiratory distress.

The trust relationship between Alice and the providers is not static. It varies depending on her context, the parties with which the provider is communicating Alice's data, etc. For example, assume that both providers have signed contracts with third parties interested in exploiting the data of Alice for different purposes, including marketing companies and government agencies. Marketing companies could be interested in exploiting consumption data to analyze the lifestyle of Alice in order to send her targeted advertisements (e.g., advertisement about appliances that she owns or does not own). Government agencies could be interested in identifying users involved in wrongdoing (e.g., fraud, crimes, etc.).

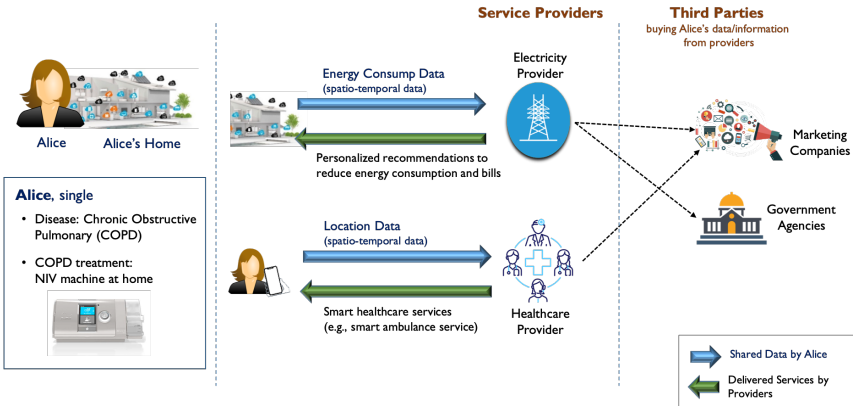


Fig. 1. Motivating Scenario

Even though Alice is notified, through agreed policies, about consumers who have access to her data, she might not be necessarily aware of the privacy risks involved with this sharing. These risks may vary between: (i) mono-source risks, that might be generated from analyzing data pieces in isolation; and (ii) multi-source risks, more complex risks that might be generated from analyzing combined data pieces together or/and with other background knowledge information acquired from external data sources.

For instance, analyzing the energy consumption data (see the signature in Fig. 2) can entail various mono-source risks for Alice, such as the risks of disclosing her presence/absence hours at home, waking/sleeping cycles, some of her habits and activities (e.g., cooking, TV watching, sports activity using a treadmill) [13]. Moreover, existing works (e.g., [3]) show that consumption signatures can be mined to identify the use of specific appliances (e.g., medical devices). This would reveal the health condition of Alice if the use of her NIV machine was identified. The analysis of location data can also involve significant mono-source risks for Alice such as the risks of disclosing her habits, behaviors and health conditions by analyzing her trajectory patterns (cf. Fig. 3). For example, if Alice is located twice per week in a pulmonary rehabilitation center for COPD patients, then she is very likely to be a COPD patient.

On the other hand, consumers can also exchange the data of Alice between them (cf. Fig.1), which may imply further multi-source risks for her. For example, assume that Alice has unlawfully certified that she is living alone to be eligible for a welfare program when she submitted her application. A marketing company having access to both location and consumption data can infer this fraud (it suffices to identify the usage of some specific devices such as microwaves and TVs while Alice is located outside her home).

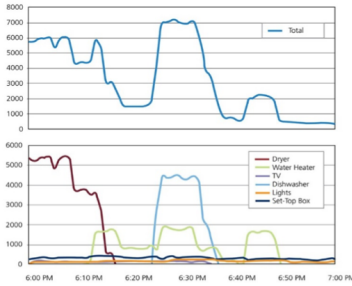


Fig. 2. Energy consumption signature

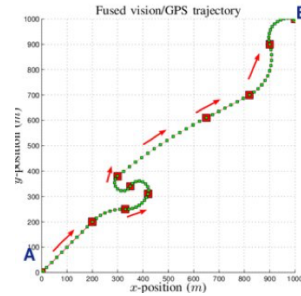


Fig. 3. Location data pattern

After alerting Alice of the risks involved in her context, adapting her privacy protection measures becomes essential. Nonetheless, such an adaptation can be difficult for her, especially as it may impact the utility of shared data, and thus the accuracy of associated services, including important services for Alice. Assume that the services offered by the healthcare provider are important for Alice. She may want to minimize her risks when being located in the pulmonary rehabilitation center, but without completely losing the healthcare services. In this case, Alice may not know the appropriate amount of protection to assign to her shared attributes, as she may not know the impact of this protection on associated risk values. This raises the need for a system that can support Alice in real-time to optimize her privacy decisions while keeping the process simple to her. However, building this dynamic context-dependent system requires to address the following scientific challenges:

- **Challenge 1. Privacy Risk Quantification:** The privacy risks are associated to the data shared by the user in his context. This means that protecting these data will lead to minimize the risk values. Hence, quantifying privacy risks becomes an important challenge to address in order to study the impact of data protection levels on the risk values.
- **Challenge 2. User-centric Privacy:** Users may have different levels of expertise to express their requirements/preferences. Therefore, the proposed solution must be simple and user-friendly, in that it assists the user based on his level of expertise, while masking the complexity of correlations that exist among user preferences, contexts, and protection measures.
- **Challenge 3. Optimal Protection Strategies:** The strategies to be delivered to the user depend on his context and preferences. Therefore, the proposed solution should be able to monitor the evolution of the user context, and dynamically identify and select the best protection strategies that satisfy the preferences of the user while maximizing data utility and minimizing the protection cost.

3 PRIVACY ORACLE: PRIVACY BY DESIGN FRAMEWORK FOR CONTEXT-AWARE PRIVACY MANAGEMENT IN CONNECTED ENVIRONMENTS

In order to stress the usage of the δ -Risk approach, we present in this section an overview of our Privacy by Design framework for context-aware privacy management in connected environments. We start by explaining the functioning of the solution, and formally defining the key concepts used in the paper. Then, we briefly describe the framework modules.

The aim of our framework is to build-up a user-centered and context-aware reasoning system, denoted **Privacy Oracle**, that can assist the user in controlling and managing his privacy protection (cf. Fig. 4). This system can be embedded on a user device as middleware between the user and the connected providers, such that it manages the user's data before being released to providers. To do so, the user starts by specifying: (1) the list of attributes that are currently shared with data consumers; (2) his preferences, which vary between mandatory and optional preferences (user preferences are detailed in Section 4). The system captures these information pieces from its side and begins to collect the data flow of attributes. Besides, the system continuously gathers background knowledge information that describe the user or/and his surrounding physical environment from the user's Web environments (e.g., social networks). This information gathering process is ensured by the use of Bots that continuously monitor and analyze the Web environments of the user.

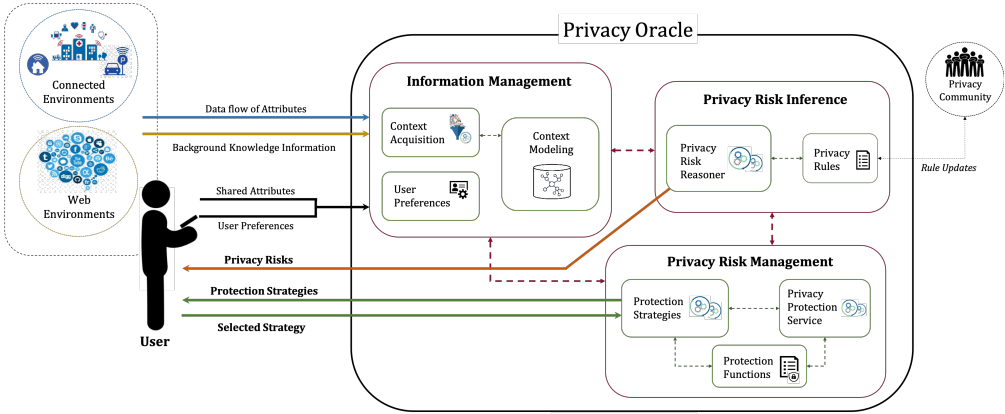


Fig. 4. Privacy Oracle Framework

Let u denotes the user of interest.

Definition 1 (Data-Exchange Node). Let DEN be the *set of data-exchange nodes* $\{dn_1, \dots, dn_n\}$ responsible for exchanging data of u (i.e. generate or receive the data of u). dn can be the source from which the data is collected (e.g., sensor, social network), or the data consumer with which the data is shared (i.e. service provider or third party). $dn \in DEN$ is formalized as follows:

$dn : \langle desc ; id \rangle$, where:

- $desc$ is the textual description of dn (e.g., gps-sensor, heart-rate-sensor, facebook)
- id denotes the identity of dn , expressed as an IP address or a uniform resource identifier (URI) \square

Definition 2 (Physical Environment). Let E_u be the *set of physical environments* $\{env_1, \dots, env_n\}$ controlled by u , or where u is/was located. $env \in E_u$ is formalized as follows:

$env : \langle desc ; sz ; System \rangle$, where:

- *desc* denotes the textual description of env (e.g., home, office, mall)
- *sz* expresses the spatial zone of env (cf. Definition 4)
- *System* is the set of systems (e.g., sensor, device) deployed in env , $System \sqsubseteq DEN$ \square

Definition 3 (Spatial Zone). A *spatial zone*, sz , is defined as a geographical surface bounded by a set of locations, such that:

$sz : \langle loc_1 ; loc_2 ; \dots ; loc_n \rangle$, where:

- *loc* is a *location* instance defined as 3-tuple $loc : \langle long ; lat ; alt \rangle$, such that:
 - *long* denotes the *longitude* of loc
 - *lat* denotes the *latitude* of loc
 - *alt* denotes the *altitude* of loc \square

Example 1. The home of Alice can be defined as follows:

home :

- *desc* : home of Alice
- *sz* : *homeZone* : $\langle loc_1 ; loc_2 ; loc_3 ; loc_4 \rangle$
- loc*₁ : *loc*₂ : *loc*₃ : *loc*₄ :
- *long* : -1.52308 - *long* : -1.52222 - *long* : -1.51503 - *long* : -1.51534
- *lat* : 33.0585 - *lat* : 33.0884 - *lat* : 34.1381 - *lat* : 34.1431
- *alt* : 200.03 - *alt* : 205.14 - *alt* : 216.57 - *alt* : 218.13
- *System* = {*sensor1* : $\langle desc : energy-consump-sensor ; id : 46.193.0.164 \rangle$ }

Physical environments may be interdependent (i.e. spatial dependency), making the associated information interdependent. For instance, the system may receive information describing the home and city of the user, where home is located inside the city, making the information collected on the two environments interdependent. At this stage, we do not consider the interdependency of environments, and we reason on each environment in isolation.

Definition 4 (Attribute). Let SA be the *set of attributes* $\{a_1, a_2, \dots, a_n\}$ that u might share with data consumers. $a \in SA$ is an attribute whose data flow can be controlled and managed by the system. a is formalized as follows:

$a : \langle desc ; source ; D_{consumer} ; ent ; Log \rangle$, where:

- *desc* denotes the textual description of a (e.g., location, energy-consump, heart-rate)
- *source* expresses the system (e.g., sensor, device) from which the data values of a are captured, such that $source \in DEN$
- $D_{consumer}$ represents a set of data consumers with whom a is shared, such that:

$D_{consumer} = \{ dc_1 ; dc_2 ; \dots ; dc_n \} \cup \perp$, where:

- dc_i is a data consumer, such that $dc_i \in DEN$
- $D_{consumer} = \emptyset$ indicates that data consumers are unknown
- $D_{consumer} = \perp$ denotes that a is a public attribute (i.e. shared with everyone)

- **ent** represents the entity described by **a**, which can be **u** or a physical environment controlled by **u** (i.e. environment containing **source**). **ent** is defined as follows:

$$\mathbf{ent} \in \{u, \mathbf{env}\} \mid \mathbf{env} \in E_u \text{ and } \mathbf{source} \in \mathbf{env}.\mathbf{System}$$

- **Log** = $\{\langle rv ; M \rangle\}$ is the set of data values of **a** (e.g., spatio-temporal data stream). **Log** can be seen as the log file of **a**, such that:
 - **rv** is a raw data value
 - **M** denotes the set of metadata characterizing **rv** (e.g., time of capture, location of capture, data-type, data-unit) \square

Example 2. Alice's shared attributes can thus be represented as follows:

$a_1 :$ - desc : energy consumption - source : sensor1 - $D_{consumer} : prov-1$ prov-1: $\langle desc : elect-prov ; id : 58.17.37.23 \rangle$ - ent : home - log : $\langle 89 ; \{t_{capture}:21:05:00 ; d_{unit} : kWh\} \rangle$ $\langle 115 ; \{t_{capture}:21:15:00 ; d_{unit} : kWh\} \rangle$	$a_2 :$ - desc : location - source : sensor2 : $\langle desc : GPS ; id : 46.89.1.47 \rangle$ - $D_{consumer} : prov-2$ prov-2: $\langle desc : health-prov ; id : 64.31.3.12 \rangle$ - ent : u - log : $\langle (-33.0534, 16.3103) ; \{t_{capture}:11:00:00\} \rangle$ $\langle (-36.0534, 17.4401) ; \{t_{capture} : 11:01:00\} \rangle$
---	--

Definition 5 (Background Knowledge Information). Let **BI** be a **set of background knowledge information** $\{b_1, b_2, \dots, b_n\}$ describing **u** or/and his surrounding physical environment. **b** \in **BI** is a piece of information that cannot be controlled or managed by the system. It can be publicly available on the Web through user profiles on social networks, public databases (e.g., voting/medical records), etc. **b** is formalized as follows:

$$b : \langle desc ; source ; ent ; value \rangle, \text{ where:}$$

- **desc** denotes the textual description of **b** (e.g., age, marital-status)
- **source** expresses the Web environment source from which **b** is captured (e.g., social network, public database), such that **source** \in **DEN**
- **ent** represents the entity described by **b**, which can be **u**, or a physical environment where **u** is/was located (e.g., home, mall, street). **ent** \in $\{u \cup \mathbf{env}\} \mid \mathbf{env} \in E_u$
- **value** : $\langle rv ; M \rangle$, such that:
 - **rv** expresses the raw data value
 - **M** denotes the set of metadata characterizing **rv** (e.g., time of capture, location of capture, data-type, data-unit) \square

Example 3. Assume that the system has captured the following background knowledge information about Alice from her Facebook account:

$b_1 :$ - desc : marital-status - source : socialAccount1: $\langle desc : facebook ; id : https://www.facebook.com/Alice \rangle$ - ent : u - value : $\langle single ; \{t_{capture}:12:00:00 ; d_{type}:String\} \rangle$
--

The Privacy Oracle system models acquired context information, reasons over it while relying on imported *privacy rules* (cf. Definition 8) in order to infer the *privacy risks* (cf. Definition 10) involved in the actual context. If no risk is inferred, the system keeps releasing the data values of attributes to consumers without adding any extra protection. Otherwise, it alerts the user by providing him with a clear overview of his risks, and a list of best protection strategies that could be adopted in his context. Meanwhile the system stops sharing data with consumers and waits for the user's response. When the user selects the strategy to be implemented in his context, the system protects the attribute data accordingly before communicating it to consumers. The system continues to apply the same protection strategy to data flows until a new context emerges (cf. Definition 6). At this stage, the system examines the similarity between consecutive contexts (cf. Definition 7). If contexts are similar, the system keeps applying the same protection strategy, otherwise the entire reasoning process is relaunched. Therefore, the risk inference and strategy identification process is executed by default once per consecutive similar contexts.

Definition 6 (User Context). A *user context*, c , is a spatio-temporal context in which u shares a fixed set of *attributes* with data consumers, and the system has a fixed set of *background knowledge information* describing u and his environments. c is formalized as follows:

$$c : \langle t ; s ; CI \rangle, \text{ where:}$$

- t denotes the time period of c , which can be a time instant or a time interval. A *time interval*, ti , is defined as 2-tuple $ti : \langle t_{start} ; t_{end} \rangle$, where t_{start} and t_{end} are two time instants expressing respectively the lower and upper boundaries of ti
- s expresses the *spatial zone* of c (cf. Definition 3)
- CI represents the set of *context information* characterizing c . CI is composed of the fixed set of shared *attributes* in c , SA_c , combined with the fixed set *background knowledge information* known about the user and his environments in c , BI_c , such that: $CI = SA_c \cup BI_c$

A context-change takes place if at least one of the context parameters varies. \square

Definition 7 (Context Similarity). Let c_1, c_2 be two contexts. The *context similarity* of c_1 and c_2 is determined by computing the similarity between the two sets of context information characterizing these contexts. The similarity score is computed as follows:

$$sim(c_1, c_2) = sim(CI_{c_1}, CI_{c_2}) \rightarrow [0; 1], \text{ where:}$$

- sim is a unit similarity function comparing the two sets CI_{c_1}, CI_{c_2} based on their instances and their parameters' values. sim returns a score between $[0; 1]$, where 0 expresses a total divergence and 1 a complete similarity.

Therefore, c_1 and c_2 are said to be similar contexts only if $sim(c_1, c_2) = 1$ \square

Performing rule-based reasoning to infer the risks involved in the user context requires relying on a reference schema that introduces a list of pre-defined *privacy rules*.

Definition 8 (Privacy Rule). Let PR be the *set of privacy rules* $\{pr_1, pr_2, \dots, pr_n\}$ that define the risks to be detected by the system (i.e. mono-source and multi-source risks). A *privacy rule*, $pr \in PR$, is a reasoning rule that indicates what attribute, or combination of attributes together or/and with other background knowledge information, if processed, leads to reveal what *privacy-sensitive information* $psi \in PSI$ (cf. Definition 9) about u . Therefore, pr should include at least one attribute. $pr \in PR$ is formalized as follows:

$$pr : A \wedge B \rightarrow psi, \text{ where:}$$

- $A = a_1 \wedge \dots \wedge a_n$ denotes the combined *attributes* such that: $A \sqsubseteq SA$ and $\min|A| = 1$
- $B = b_1 \wedge \dots \wedge b_n$ denotes the combined *background knowledge information*, $B \sqsubseteq BI$
- $psi \in PSI$ expresses the *privacy-sensitive information* to be disclosed by this combination \square

Definition 9 (Privacy-Sensitive Information). A *privacy-sensitive information*, $psi \in PSI$, is a personal information about the user, that if disclosed, could be harmfully used against him. psi is a sensitive information that might be revealed when processing and analyzing the knowledge acquired about u and his surroundings. psi has a primitive data type of String and belongs to a controlled set PSI , such that:

$$PSI = \{ psi_1 ; psi_2 ; \dots ; psi_n \}$$

The National Institute of Standards and Technology (NIST) guidelines for smart grid cybersecurity [13] has identified several psi instances, including: (i) User-profile information (e.g., disease, salary), (ii) user habits (e.g., daily activities), (iii) behaviors, (iv) preferences, (v) presence/absence, (vi) sleep/wake cycles, (vii) appliances and medical devices used, and (viii) fraud \square

Example 4. A privacy rule pr_1 states that processing the energy consumption of the user's home can lead to reveal the presence/absence of the user at home. pr_1 can be represented as follows:

- Let $a_3 : \langle desc : energy-consump ; ent : home \rangle$ and $psi_1 = \text{presence/absence at home}$
 $\Rightarrow pr_1 : a_3 \rightarrow psi_1$

Definition 10 (Privacy Risk). A *privacy risk*, r , is defined as the probability of disclosing a privacy-sensitive piece of information $psi \in PSI$ about u . r is linked to a pre-defined *privacy rule* $pr \in PR$. It is generated when the associated pr is satisfied in the user context, and remains valid for the entire time period of the context. r is probabilistic with a value between $[0, 1]$, where 0 indicates that r is eliminated and 1 the highest risk level. The default value of r when inferred is 1. r is formalized as follows:

$$r = P(psi)_{pr} \mid r \leftrightarrow pr \in PR, r \in [0, 1]$$

The number of risks to infer in a single iteration is fix, it depends on the number of imported privacy rules. Moreover, it is possible to have several risks r_i mapped to the same psi in cases where the psi could be disclosed through various possible combinations of information defined by different privacy rules. \square

Example 5. Assume that when launching the reasoning process, the rule pr_1 is satisfied in the context of Alice. Therefore, one risk, r_1 , is inferred for Alice:

$$r_1 = P(psi)_{pr_1} = P(\text{presence/absence at home}) = 1$$

3.1 Privacy Oracle Framework Modules

As illustrated in Fig.4, Privacy Oracle relies on a modular framework comprised of three modules: information management module, privacy risk inference module, and privacy risk management module. These modules are detailed in what follows.

3.1.1 Information Management. Inferring context-aware risks requires first to build up a global view of the user context. This is done by gathering information describing the user of interest and his surrounding cyber-physical environment. Hence, this module is responsible for managing (i.e., capturing and modeling) context information. It comprises the following components: (i) *context acquisition*, in charge of capturing context information (i.e. shared attributes and background

knowledge) from the user and his Connected/Web environments; (ii) *user preferences*, responsible for managing the preferences of the user; and (iii) *context modeling*, liable for modeling acquired information and the relationships that exist among them, which helps in better understanding the user context. We explored the *context modeling* component in previous work [2], where we proposed a generic and modular ontology for Semantic User Environment Modeling, entitled SUEM. In fact, adopting a semantic data model that maintains a flexible data structure becomes a fundamental requirement, especially as: (1) collected information can be heterogeneous (i.e., they have different data types and formats); (2) information can be captured from different types of data sources that could derive from both Connected environments (e.g., IoT sensor networks), and Web environments such as social networks, or any other public data source (e.g., public voting records, medical records); (3) gathered information may have different levels of granularity (i.e., different levels of precision); and (4) performing in a dynamic environment that cannot be controlled in advance makes the system unable to control or predict the knowledge to receive, nonetheless, it must be always capable of modeling it. The SUEM ontology introduces concepts and properties to represent the context information received about users, domains of interest, and environments. SUEM is extensible and can be adapted to various domain particularities. Full documentation of the SUEM ontology can be found at: <http://spider.sigappfr.org/SUEMdoc/index-en.html>.

3.1.2 Privacy Risk Inference. Responsible for inferring the risks involved in the user context. To achieve this, this module includes two components. First, the *privacy rules* component, which handles the definition/import of privacy rules that specify the risks to be detected by the system. The rules are defined according to the given syntax in Definition 8, and they are used as a reference schema for the reasoning process. This schema is regularly updated by the privacy community, and the rule updates are imported by the system when relaunching the risk inference process. It is important to state that the accuracy of the risk inference process depends from the quality of the defined rules. Consequently, we assume in this study that the privacy rules, defined by experts from the privacy community, are optimal. Second, the *privacy risk reasoner* component, which provides a semantic rule-based reasoning engine proposed in [2]. This engine performs continuous reasoning over modeled information to dynamically infer the risks involved in the user context.

3.1.3 Privacy Risk Management. The user might change progressively his preferences. This change can occur due to the risks incurred in his context or the sensitivity of the context (e.g., private meeting, located in a sensitive environment). Consequently, this module is responsible for: (1) managing the risks inferred based on the privacy requirements and interests of the user, and delivering optimized and meaningful strategies to the user; (2) protecting the spatio-temporal data values of attributes according to the protection strategy selected by the user. In order to achieve this, the module is comprised of three components: (i) *protection strategies*, in charge of identifying the best protection strategies to be proposed to the user in his context according to his preferences and context; (ii) *protection functions*, contains the list of available protection functions (cf. Definition 13); and (iii) *privacy protection service*, responsible for selecting the most appropriate protection functions to be linked to attributes in the relevant context, and to execute the selected functions on the attributes' data before being released to consumers. This paper explores the *protection strategies* component by introducing a new privacy risk management model detailed in the following section.

4 δ -Risk: TOWARDS CONTEXT-AWARE MULTI-OBJECTIVE PRIVACY MANAGEMENT IN CONNECTED ENVIRONMENTS

Empowering the user to make quick, effective and meaningful adaptation of his privacy decisions to cope with the evolution of his context remains challenging. In that regard, we propose in

the following a new user-centered, context-aware and multi-objective privacy risk management approach, denoted δ -Risk. δ is a privacy parameter specified by the user to express the maximum level of risk that he accepts to take in his context. The aim of this approach is to assist the user in optimizing his privacy decisions, so that to meet his requirements and preferences while maximizing data utility and minimizing the protection cost. To do so, δ -Risk provides the user with at least one best protection strategy to adopt in his context. In addition to his privacy preferences, the approach considers also the interests of the user (e.g., what services are important to him), thereby making the strategies provided not only optimal but also meaningful.

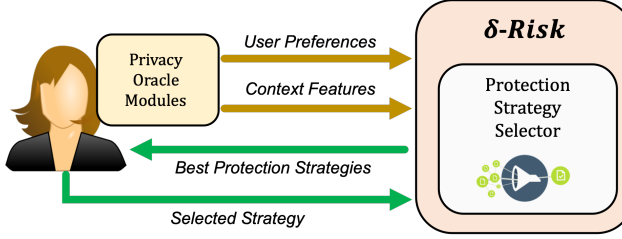


Fig. 5. δ -Risk Approach

Fig. 5 illustrates an overview of the solution. δ -Risk receives as input the preferences of the user and the context features, and outputs the best strategies that might be adopted in these circumstances. The user selects accordingly one protection strategy to be implemented on his data, and this strategy remains valid as long as there is no change in the entries. The δ -Risk principle is defined as follows: the global risk level to maintain in a user context should not bypass the threshold δ specified by the user (i.e. $\leq \delta$). We discuss in what follows the input parameters.

Context Features:

- The set of attributes shared by u in c , i.e., $SA_c = \{a_1, a_2, \dots, a_m\}$ (cf. Definition 4).
- The overview of privacy risks in c , represented by $R_c = \{\vec{r}; v\}$, where:
 - $\vec{r} = [r_1 \ r_2 \ \dots \ r_n]$ is a risk vector composed of the privacy risks inferred in c , where n denotes the number of risks inferred.
 - v denotes the global risk level in c , that is used to interact with the risk threshold specified by the user (i.e. δ). The challenge of how to quantify the global risk level is addressed in the following subsection.
- The costs of selected *protection functions* (cf. Definition 13), $cPF = \{c_1, c_2, \dots, c_m\}$, to be executed on attributes $\{a_1, a_2, \dots, a_m\}$ of SA_c . The selection process of protection functions is managed by the *privacy protection service* component, and their costs are provided accordingly.
- The impact matrix of shared attributes on the risks inferred, W_c .

Definition 11 (Impact Matrix). Let W_c be the *impact matrix* of attributes $\{a_1, a_2, \dots, a_m\}$ of SA_c on risks $\{r_1, r_2, \dots, r_n\}$ of R_c . W_c is automatically calculated by the *privacy risk reasoner* component during the risk inference process, such that:

$$W_c = \begin{bmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1m} \\ \omega_{21} & \omega_{22} & \dots & \omega_{2m} \\ \vdots & \vdots & \dots & \vdots \\ \omega_{n1} & \omega_{n2} & \dots & \omega_{nm} \end{bmatrix}, \text{ where } \omega_{ij} = \begin{cases} 0 & \text{if } r_i \leftrightarrow pr \in PR \text{ and } a_j \notin A_{pr} \\ 1 & \text{if } r_i \leftrightarrow pr \in PR \text{ and } a_j \in A_{pr} \end{cases}$$

The impact ω_{ij} of attribute a_j on risk r_i is equal to 1 only if a_j is included in the set of combined attributes (i.e. A) when defining the privacy rule pr to which r_i is linked. \square

User Preferences:

• **Privacy preferences:**

- (1) Risk threshold δ , which is the only mandatory parameter for u . δ has a value between 0 and 1, where 0 indicates that u does not accept to take any risk, and 1 means that u wants to share fine-grained data to preserve the full accuracy of related services.
- (2) Enforced protection levels for specific attributes, eP , which is an optional parameter for u . In fact, u might enforce specific protection levels to be assigned to particular attributes regardless of his context (e.g., resulting from agreements with service providers). These protection levels must be respected when computing the strategies.

• **Service preferences:**

The user can state which services are important to him (optional parameter). Accordingly, the system calculates the weights to assign to attributes. This process, detailed in the following, is managed by the *user preferences* component of the framework.

Let $\vec{wA} = [w_1 \dots w_m]$ be the vector of weights assigned to attributes $\{a_1, \dots, a_m\}$ of SA_c .

Let S be the set of available services s_1, s_2, \dots, s_n offered by the providers to u in exchange of his data, such that: $\forall s \in S, s : \langle A ; li \rangle$, where:

- A represents the set of attributes associated to s , such that $A \subseteq SA_c$
- li expresses the level of importance of s to the user. li is Boolean with a value of 1 if s is important, and 0 if not

Therefore, w_i of \vec{wA} is equal to the number of important services to which a_i is associated. This can be represented as follows:

$$\forall w_i \in \vec{wA} : w_i = \sum_{l=1}^n s_{l_i}.li \mid a_i \in s_{l_i}.A \tag{1}$$

Specifying a value for δ might be challenging as it depends on the level of expertise of the user. Therefore, as our objective is to keep the privacy management process simple to the user, we define three profiles that express the level of expertise of the user: beginner, intermediate, and advanced.

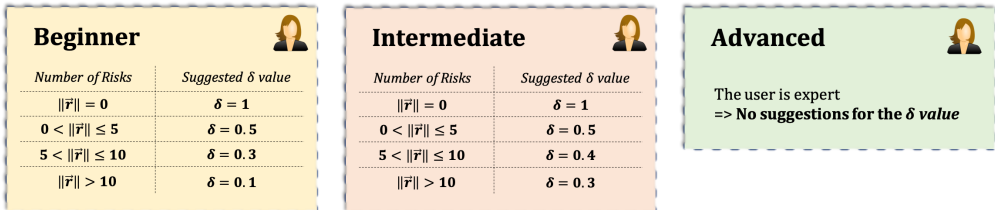


Fig. 6. User Profiles

The aim here is to assist the user in specifying the δ value. To do so, the user starts by selecting his profile according to his level of expertise. Consequently, the system dynamically suggests a value for δ based on his profile and context. Advanced users are experts in managing their privacy, which means they do not require assistance to specify δ . Beginner users are non-savvy users, i.e.,

they require more critical assistance than intermediate users. This makes the suggestion of δ more critical for beginners. The choice of δ depends on the number of risks inferred in the user context, i.e., the value of δ decreases with the increase in the number of risks as shown in Fig.6. If no risk is inferred, the suggested δ value is 1 (i.e. keep sharing fine-grained data). Once a risk is inferred (with a total number of risks less than 5), the suggested *delta* value is automatically reduced by half (i.e. 50% protection) for both profiles (i.e. beginner and intermediate). Then, the suggested δ decreases by 0.2 for beginner, and by 0.1 for intermediate, with the increase of the risk number. The lowest value to suggest by the system (i.e. 0.1 for beginner and 0.3 for intermediate) is achieved when the number of risks exceeds 10. It is important to mention that the system keeps the choice for the user to select the suggested value or to manually specify a value for δ .

Once δ is specified, the system calculates the best strategies to propose to the user in his context. To do so, the δ -Risk process consists of two operations, namely *protection strategy identification* and *best strategy selection*. Before detailing the process, we start by defining what constitutes a *protection strategy*, *protection function* and *best strategy*.

Definition 12 (Protection Strategy). A *protection strategy*, $\vec{p} \in P_c$, is a vector composed of an appropriate combination of protection levels p_1, p_2, \dots, p_m to be assigned to attributes a_1, a_2, \dots, a_m shared by the user in his context. Appropriate means a combination that meets the user's privacy requirements (i.e. δ and eP) while maximizing the data utility of attributes. A protection strategy can be represented as follows:

$$\vec{p} = [p_1 \quad p_2 \quad \dots \quad p_m] \quad | \quad p_j \in [0, 1] \quad \forall j \in [1, m]$$

A *protection level*, p_j of \vec{p} , is probabilistic with a value between $[0, 1]$, where 0 indicates that a_j is shared without any protection (default value), and 1 means stop sharing a_j . A value between 0 and 1 expresses the level of protection that must be reached when executing a *protection function* on a_j . Knowing that the way to achieve this level depends on the selected *protection function*. \square

Definition 13 (Protection Function). A *protection function*, $f \in PF$, expresses a selected protection method to be executed on the data flow of an attribute $a \in SA_c$. f is a local function stored in the system. A protection function is formalized as follows:

$f : \langle \textit{name} ; \textit{type} ; \textit{cost} ; \textit{Param} \rangle$, where:

- *name* denotes the name of f (e.g., random noise, differential privacy)
- *type* represents the protection type to which f belongs, such that:

$$\textit{type} \in \{\textit{noiseAddition} ; \textit{anonymization} ; \textit{accessControl} ; \textit{encryption}\}$$
- *cost* expresses the cost of f in terms of processing time and memory overhead
- *Param* represents the set of input parameters of f , including at least the following parameters:
 - a , denotes the attribute on which f will be executed
 - p , expresses the desired protection level to reach for the data values of a \square

Definition 14 (Best Protection Strategy). A *best protection strategy*, $\vec{b}p \in BP_c$, is an appropriate strategy $\vec{p} \in P_c$, that most satisfies user preferences (i.e. wA), and has the best combination of protection functions (i.e. lowest protection cost). These constraints are expressed by the *ranking score* assigned to \vec{p} . Therefore, \vec{p} is said to be a *best protection strategy* $\vec{b}p$ only if it has the highest ranking score. This can be formalized as follows:

$$\forall \vec{p}_i \in P_c : \vec{p}_i \models \vec{bP} \text{ only if } \forall \vec{p}_j \in P_c, \text{score}(\vec{p}_j) \leq \text{score}(\vec{p}_i)$$

where:

- **score** denotes the ranking score that is calculated and assigned to the protection strategy $\vec{p} \in P_c$ after executing the ranking function **Rank**. \square

As shown in Fig.7, the first δ -Risk operation consists of identifying all possible strategies that meet the user's privacy preferences (i.e. δ and eP) while maximizing the data utility of attributes $\{a_1, a_2, \dots, a_m\}$ of *DEN*. If no strategies result from this operation, this means that the combination of the δ value and the enforced protection levels eP is inconsistent (cf. Definition 15). In this case, the system asks the user to change one of his privacy preferences and assigns a timeout period for this query: (1) if the user fails to respond before the timeout expires, the system sets the value of δ to 0, which leads to stop sharing all attributes and thus eliminates all risks; (2) elsewhere, the system receives the adjustments and the process is re-launched. If the first operation generates *protection strategies*, the second operation focuses on ranking the resulting strategies in order to select the *K-best strategies* to be proposed to the user. The ranking function, **Rank**, considers the service preferences of the user (i.e. wA) and the costs of selected *protection functions* (i.e., cPF).

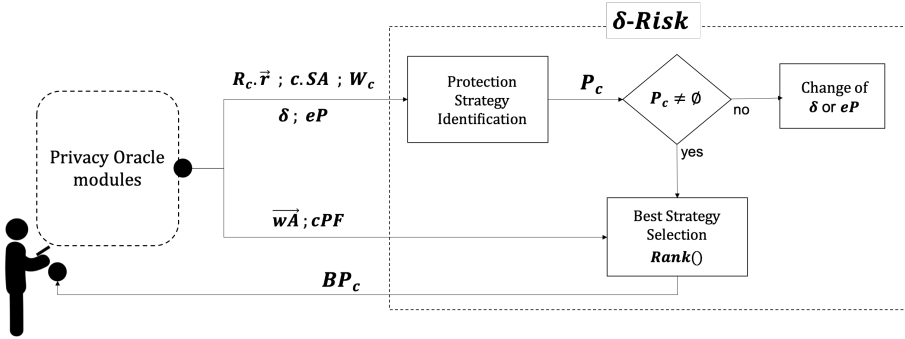


Fig. 7. δ -Risk Process

The δ -Risk process is by default executed once per consecutive similar contexts (cf. Definition 7). However, the user might change his preferences while being in the same context, which requires recalculating new best strategies. To handle this, the system locally stores the *protection strategies* identified from the first operation as long as the newly emerged contexts are similar. Therefore, if wA has been changed, only the ranking operation will be re-executed to select the new best strategies that meet these changes. Otherwise, the entire δ -Risk process will be re-launched.

Example 6. Assume that the first operation has generated the following 2 appropriate strategies:

$$P = \begin{bmatrix} \vec{p}_1 \\ \vec{p}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0.6 \\ 0.6 & 0 \end{bmatrix}$$

Assume that attributes a_1 and a_2 has the same weight, and the cost of the protection functions to execute on a_1 and a_2 are respectively 2 and 1. When executing the ranking function (detailed in Section 4.3), \vec{p}_2 will have a score higher than \vec{p}_1 , and thus will be selected as the best strategy. \vec{p}_2 suggests applying 60% protection on a_1 and sharing a_2 without any protection.

Determining appropriate combinations of protection levels requires first to quantify a privacy risk in order to study the impact of a protection level on the risk value (cf. Challenge 1). Then, to quantify the global risk level to ensure that the resulting strategies satisfy the risk threshold δ specified by the user. Therefore, we begin by formally quantifying a *privacy risk* and a *global risk level*, and then detail the two operations of the δ -Risk process.

4.1 Privacy Risk & Global Risk Level Quantification

A privacy risk is linked to one or more shared attributes. This means that protecting impacting attributes will lead to minimize the risk value. Therefore, a risk r_i of \vec{r} depends from the levels of protection $\{p_1, p_2, \dots, p_m\}$ assigned to attributes $\{a_1, a_2, \dots, a_m\}$ of SA_c having impact on r_i . This can be represented as follows:

$$\vec{r} = \mathcal{F}(W_c, \vec{p}) \text{ , where:} \quad (2)$$

- \mathcal{F} is a function that takes as parameters an impact matrix and a protection vector, and returns a risk vector comprised of the calculated risk values.

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} = \mathcal{F} \left(\begin{bmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1m} \\ \omega_{21} & \omega_{22} & \dots & \omega_{2m} \\ \vdots & \vdots & \dots & \vdots \\ \omega_{n1} & \omega_{n2} & \dots & \omega_{nm} \end{bmatrix}, \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix} \right)$$

Before exploring the risk quantification function (\mathcal{F}), we define the assumptions to consider:

- (1) A privacy risk has at least one impacting attribute $a_j \in SA_c$. This means that:

$$\forall \vec{w}_i \in W_c, \sum_{j=1}^m \omega_{ij} \neq 0$$

- (2) If no protection assigned to attributes impacting r_i , the risk value is 1 (i.e. highest level).
- (3) If the full protection is assigned to attributes impacting r_i , r_i is eliminated.
- (4) The higher is a protection level p_j applied on an attribute a_j impacting r_i , the lower is the value of r_i

Let \widetilde{W}_c denotes a normalized version of W_c such that:

$$\widetilde{W}_c = \begin{bmatrix} \widetilde{\omega}_{11} & \widetilde{\omega}_{12} & \dots & \widetilde{\omega}_{1m} \\ \widetilde{\omega}_{21} & \widetilde{\omega}_{22} & \dots & \widetilde{\omega}_{2m} \\ \vdots & \vdots & \dots & \vdots \\ \widetilde{\omega}_{n1} & \widetilde{\omega}_{n2} & \dots & \widetilde{\omega}_{nm} \end{bmatrix}, \text{ where } \widetilde{\omega}_{ij} = \frac{\omega_{ij}}{\sum_{j=1}^m \omega_{ij}} \quad \forall i \in [1, n], j \in [1; m] \quad (3)$$

A privacy risk is therefore quantified as follows:

$$\vec{r} = \mathcal{F}(W_c, \vec{p})$$

$$\vec{r} = 1 - (\widetilde{W}_c \times \vec{p}) \quad (4)$$

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} = 1 - \left(\begin{bmatrix} \widetilde{\omega}_{11} & \widetilde{\omega}_{12} & \dots & \widetilde{\omega}_{1m} \\ \widetilde{\omega}_{21} & \widetilde{\omega}_{22} & \dots & \widetilde{\omega}_{2m} \\ \vdots & \vdots & \dots & \vdots \\ \widetilde{\omega}_{n1} & \widetilde{\omega}_{n2} & \dots & \widetilde{\omega}_{nm} \end{bmatrix} \times \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix} \right)$$

Example 8. According to Example 7, the best strategy delivered to Alice in her context is $\vec{b}_p = [0.6 \ 0]$. Once adopted by the user, the risk values will be therefore minimized to:

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = 1 - \left(\begin{bmatrix} 1 & 0 \\ 1/2 & 1/2 \\ 1/2 & 1/2 \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0.6 \\ 0 \end{bmatrix} \right)$$

$$r_1 = 1 - 0.6 = 0.4 ; r_2 = 1 - 0.3 = 0.7 ; r_3 = 1 - 0.3 = 0.7 ; r_4 = 1 - 0.6 = 0.4$$

After quantifying a privacy risk, we now focus on how to measure the global risk level in a user context, i.e. $R_c.v$. Once the user specifies the value of δ , this means he does not accept taking any risk above the specified threshold. In that respect, the global risk level will be equal to the maximal risk value in the relevant context. $R_c.v$ is quantified as follows:

$$R_c.v = \max \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} \mid R_c.v \in [0, 1] \tag{5}$$

4.2 Protection Strategy Identification

We discuss in this section the first δ -Risk operation. This operation consists of identifying appropriate combinations of protection levels that meet the user's privacy requirements (i.e. δ , eP) while maximizing the data utility of attributes (cf. Challenge 3). To answer this challenge, we rely on the proposed risk quantification model and the δ -Risk principle, such that:

$$R_c.v \leq \delta$$

$$\Rightarrow \max \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} \leq \delta \Rightarrow \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} \leq \delta$$

Nonetheless, maximizing the utility of data requires assigning the lowest acceptable protection levels to related attributes, i.e., the lowest protection that satisfies the δ -Risk principle. This can be achieved by preserving the maximum possible risk values, which means $R_c.\vec{r} = \delta$. Consequently, the best-case scenario for data utility/privacy protection consists of identifying appropriate combinations of protection levels that satisfy $R_c.\vec{r} = \delta$. This gives rise to the following linear system of n equations with m unknowns:

$$\begin{aligned}
R_c \cdot \vec{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} = \delta \Rightarrow 1 - \left(\begin{bmatrix} \widetilde{\omega}_{11} & \widetilde{\omega}_{12} & \dots & \widetilde{\omega}_{1m} \\ \widetilde{\omega}_{21} & \widetilde{\omega}_{22} & \dots & \widetilde{\omega}_{2m} \\ \vdots & \vdots & \dots & \vdots \\ \widetilde{\omega}_{n1} & \widetilde{\omega}_{n2} & \dots & \widetilde{\omega}_{nm} \end{bmatrix} \times \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix} \right) = \delta \\
\Rightarrow \begin{cases} \widetilde{\omega}_{11} \cdot p_1 + \widetilde{\omega}_{12} \cdot p_2 + \dots + \widetilde{\omega}_{1m} \cdot p_m = 1 - \delta \\ \widetilde{\omega}_{21} \cdot p_1 + \widetilde{\omega}_{22} \cdot p_2 + \dots + \widetilde{\omega}_{2m} \cdot p_m = 1 - \delta \\ \vdots \\ \widetilde{\omega}_{n1} \cdot p_1 + \widetilde{\omega}_{n2} \cdot p_2 + \dots + \widetilde{\omega}_{nm} \cdot p_m = 1 - \delta \end{cases} \quad (6)
\end{aligned}$$

In order to solve the resulted system, we use the Gauss-Jordan Elimination (GJE) method, an implicit pivoting strategy that performs row operations to convert a matrix into a reduced row echelon form [14]. This method has been widely used in various domains such as traffic control management [15], image change and climate prediction [16, 17], cluster and grid computing [18, 19], and location privacy [20]. Solving the system using the GJE method can result in three possible cases: (1) system is inconsistent, i.e., the δ/eP combination is inconsistent, which generates no solution; (2) system independent, which generates exactly one solution; and (3) system dependent, which generates an infinite number of solutions.

In fact, the constraint presented in case (1) could result if the system contains an equation that includes only *enforced protection levels*. This will result in one possible δ value and will therefore entail an inconsistency if the user-specified δ value does not match the acceptable value. This can be formalized as follows:

Definition 15 (δ/eP Inconsistency). Let p_1, p_2 be the protection levels to be assigned to attributes $\{a_1, a_2\} \subseteq SA_c$. Assume that risk r_i of \vec{r} depends only from $\{a_1, a_2\}$. This means that the linear system will include the following equation:

$$\widetilde{\omega}_{11} \cdot p_1 + \widetilde{\omega}_{12} \cdot p_2 = 1 - \delta$$

Therefore, the combination δ/eP is said to be inconsistent only if:

$$\{p_1, p_2\} \subseteq eP \text{ and } \delta \neq 1 - (\widetilde{\omega}_{11} \cdot p_1 + \widetilde{\omega}_{12} \cdot p_2) \quad \square$$

In what follows, we detail the proposed reasoning algorithm for the first δ -Risk operation.

Algorithm 1. presents the protection strategy identification operation that takes as input the impact matrix Wc , the δ value, and the set of enforced protection levels eP . It outputs the set of identified strategies Pc . The process starts first by checking the value of δ . If equal to 0, this means that the user does not accept to take any risk and the protection levels must be at their highest levels. Hence, the process calls the *createFullProtStrategy* function that creates the full protection strategy $\vec{p} = [1 \ 1 \ \dots \ 1]$. If δ is 1, this means that the user wants to share fine-grained data and the protection levels must be at their default values. The process calls consequently the *createDefaultStrategy* function that assigns the enforced value to p_j if $p_j \in eP$, or a value of 0 if not. If δ is between 0 and 1, this means that the user wants to preserve the utility of the data but without taking any risk above the threshold δ . Hence, the process builds the linear system by calling the *buildSystem* function, and then calls the *checkInconsistency* function to check the δ/eP constraint (cf. Definition 15). This function returns a Boolean value stored in *inconsistency*.

If *inconsistency* is *false* (i.e. system is consistent), the process solves the system using the GJE method by executing the *solveSystemGJE* function, which returns a reduced row echelon

form stored in M . To check the state of dependency of the resulted matrix, the process calls the *checkDependency* function that returns a Boolean value saved in *dependency*. If *dependency* is *false*, this means that the attributes are independent, and the system has a unique solution that leads to create one strategy composed of the resulting constant values, where each is linked to an unknown p_j item. This procedure is done by the *createIndependentStrategy* function, and the process is ended. If *dependency* is *true*, this means that the attributes are dependent, and the system has an infinite number of possible solutions. The process calls the *createDependentStrategies* function, which start first by identifying existing dependencies among the p_j items. Then, as our objective is to handle the data utility/privacy protection trade-off, the process executes a double iteration procedure on each dependent p_j item. The first iteration prioritizes the selected p_j item, and assigns it a 0 value, which corresponds to the minimal protection that could be assigned to the dependent attribute. The second iteration assigns a value of 1 to p_j (i.e. highest protection), which prioritizes the other dependent p items. Next, both iterations calculate the remaining p items that are dependent from p_j . The procedure identifies several appropriate strategies that satisfy the trade-off, where each emphasizes at least one dependent attribute, and the process is ended.

If *inconsistency* is *true*, the system notifies the user and asks him either to assign the acceptable value to δ (cf. Definition 15), or to release one of the impacting $p \in eP$. And the steps that follow are previously detailed in the process description.

Algorithm 1: Protection Strategy Identification

Input: Wc, δ, eP ;

Output: Pc ;

Variables: $System, M, inconsistency, dependency$;

begin

if ($\delta = 0$) **then**

 // user requests the full protection, i.e. stop sharing data;

$Pc \leftarrow createFullProtStrategy(1)$;

else if ($\delta = 1$) **then**

 // user accepts to share fine-grained data;

$Pc \leftarrow createDefaultStrategy(0, eP)$; // strategy created with default values of protection levels;

else

$System \leftarrow buildSystem(Wc, \delta, eP)$; // build the linear system;

$inconsistency \leftarrow checkInconsistency(System)$; // returns true if δ/eP combination is inconsistent;

if ($inconsistency = false$) **then**

$M \leftarrow solveSystemGJE(System)$; // solves the system using the GJE method;

$dependency \leftarrow checkDependency(M)$; // returns true if system is dependent;

if ($dependency = false$) **then**

 // attributes are independent (unique solution);

$Pc \leftarrow createIndependentStrategy(M, eP)$;

else

 // attributes are dependent (infinite number of solutions);

$Pc \leftarrow createDependentStrategies(M, eP)$;

else

$notifyUser()$; // user has to change either δ or the relevant $p \in eP$;

It is important to note that this paper describes only the pseudo-code of the main process due to space limitations. Nonetheless, the pseudo-codes of the aforementioned functions are detailed in the prototype source code provided in Section 5.1.

4.3 Best Strategy Selection

In case the number of strategies identified by the first operation is greater than 1 ($\sum \vec{p} \in P_c > 1$), ranking these strategies and selecting the K -best strategies to be proposed to the user becomes a need. According to Definition 14, K expresses the number of strategies with the highest ranking score. Nonetheless, fixing the maximal value of K remains challenging. Especially as many factors may contribute to perceived choice overload, including the number of options, time constraints, user expertise [21]. On this basis, as the user has to make quick decisions in real-time, the system fixes the following default values for $\max(K)$ with respect to the defined user profiles in Section 4: 1 for beginner, 3 for intermediate, and 5 for advanced user. The choice of $\max(K)$ can be manually changed by the user.

The best strategies must satisfy the most the user's preferences and interests. To achieve this, the second δ -Risk operation consists of ranking the resulting strategies (i.e. P_c) according to the service preferences (i.e. \vec{wA}) and the costs of selected protection functions (i.e. cPF). This process is provided through the function $Rank()$, which operates on the basis of the following principle: the highest ranking score corresponds to the strategy with the shortest distance to \vec{wA} and the lowest cost of protection. In what follows, we present the reasoning algorithm for the $Rank()$ function.

Algorithm 2: Best Strategy Selection - $Rank()$ function

Input: P_c, wA, cPF ;

Output: BPc ;

Variables: $sortedWA, A, minP, Score, maxScore, CostPc$;

begin

```

sortedWA ← sortAndFilter(wA); // sorts wA in a descending sequence and removes redundant values;
for i ← 0 to sortedWA.length - 1 do
  A ← attributesWithSimilarWeight(wA, sortedWA[i]);
  // the set A will include the indexes of attributes with same weight sortedWA[i];
  for j ← 0 to A.length - 1 do
    // for each attribute having the weight sortedWA[i];
    minP ← getMinP(Pc, A[j]); // the minimal protection level to be assigned to attribute aj;
    Score ← addScore(Pc, minP, A[j], wA); // add the weight of aj to strategies including minP
  maxScore ← getMaxScore(Score); // returns the maximal score assigned to strategies
  for k ← 0 to Score.length - 1 do
    if (score[k][1] != maxScore) then
      Pc ← deleteStrategy(k); // keep only strategies with the highest score
  for i ← 0 to Pc.length - 1 do
    for j ← 0 to Pc[0].length - 1 do
      if (Pc[i][j] != 0) then
        CostPc[i][1] = CostPc[i][1] + cPF[j]; // calculate the cost of protection of each strategy
  Score ← addCostToScore(Score, CostPc); // add the calculated cost to the score of strategies
  maxScore ← getMaxScore(Score);
  BPc ← selectBestStrategies(Pc, Score, maxScore);
  // BPc includes only strategies with the highest score

```

Algorithm 2. outlines the ranking function, $Rank()$, takes as input the set of identified strategies (P_c), the vector of weights assigned to attributes (wA), and the set of costs of selected protection functions (cPF). It outputs the set of K -best protection strategies, BPc . The function starts first by identifying the strategies with the shortest distance to wA . To do so, the first step is to identify the

number of different weight values and sort them in a descending sequence (i.e. from the most to the least important). This number will constitute the default number of iterations for this step. This step is done by calling the *sortAndFilter* function. Then, for each distinct weight value, we check the number of attributes having this weight through the *attributesWithSimilarWeight* function. In fact, having several attributes with the same weight requires considering strategies that prioritize each of them separately. Therefore, for each of these attributes, we check which strategy includes the corresponding minimal protection value, and we add the weight of the attribute to the score of the strategy. Thereafter, we filter the resulting set of strategies to consider only strategies with the highest score. This will ultimately lead to strategies that include the minimum possible protection levels assigned to attributes based on their level of importance. These strategies have therefore the shortest distance to wA . After, the function calculates the costs of the resulting strategies. The cost of a strategy is equal to the sum of costs of the protection functions which are only linked to the attributes protected by this strategy (i.e. attributes with protection levels higher than 0). The calculated costs are added consequently to the scores of relevant strategies, and only strategies with the highest ranking score are selected and added to the set BPC .

5 EXPERIMENTAL VALIDATION & EVALUATION

In this section, we illustrate the use of the proposed prototype, we evaluate the performance of the approach and we formally study its effectiveness.

5.1 Approach Validation: Java-based Prototype

In order to validate our approach and implement the δ -Risk mechanism, we developed a Java-based prototype, and we embed it on the user device as middleware between the user and the connected providers (cf. Fig.8). This prototype performs real-time reasoning on the user context and generates dynamic strategies according to the user's preferences and the context particularities. The source code of the proposed δ -Risk system is accessible on the following link: <http://spider.sigappfr.org/research-projects/delta-risk/>.

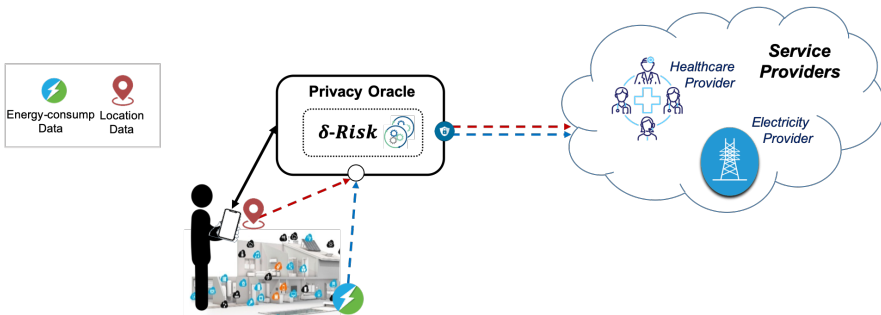


Fig. 8. δ -Risk implementation

The objective is to illustrate the use of the approach. Hence, we consider the scenario provided in Section 2 as the current context of Alice (cf. Fig.8).

We execute the privacy risk inference prototype¹ proposed in our previous work [2], to infer the privacy risks involved in this context. Fig.9 describes the overview of risks provided to Alice. Assume that after being alerted, she decided to decrease the value of δ to 60%. The *Energy-consumption* attribute (a_1) impacts risks 1,2,3,4,6,7; the *Location* attribute (a_2) impacts risks 5,6. Consequently,

¹The source code of the risk inference prototype is available here: <http://spider.sigappfr.org/research-projects/privacy-oracle/>

the δ -Risk process is executed, and generates the following best strategy that suggests applying 40% protection on *Energy-consump* and 40% protection on *Location* (cf. Fig.10).

Date: avr. 06,2020 11:08:30 ms
Date: avr. 06,2020 11:08:30 ms

Number of privacy risks inferred is: **7 Risks**

- **Risk 1:** Inferring Appliances and Devices used at home
- **Risk 2:** Inferring waking and sleeping patterns
- **Risk 3:** Inferring presence and absence hours at home
- **Risk 4:** Inferring Performed Activities in the living environment
- **Risk 5:** Inferring habits, behaviors, and preferences
- **Risk 6:** Inferring appliances turned on when you are not at home
- **Risk 7:** Inferring your disease from shared consumption data

\vec{W}_c	r_1	r_2	r_3	r_4	r_5	r_6	r_7
a_1	1	1	1	1	0	0.5	1
a_2	0	0	0	0	1	0.5	0

Delta value: 0.6
The Best Protection Strategies in c are:
p1 = [0.4, 0.4]
Full process execution time: 300 ms

Fig. 9. Risks inferred for Alice

Fig. 10. Best strategy proposed to Alice

5.2 Experimental Protocol

The objectives of our experimental protocol are to (i) evaluate the ability of the approach to reason in real-time, and to prove that the approach: (ii) always identifies all possible appropriate strategies that answer the data utility/privacy protection trade-off; (iii) always delivers the best strategies to the user; and (iv) always provides the user with at least one best strategy per context. To achieve the first objective, we evaluate the performance of the δ -Risk mechanism. Then, we formally study the effectiveness of our approach in order to prove the aforementioned three concepts.

5.2.1 Performance Evaluation.

To evaluate the performance of the δ -Risk mechanism, we consider four use cases to study the impact of the following five metrics on the system's performance: (i) number of risks inferred (R_c, \vec{r}); (ii) number of attributes shared ($c.SA$), while considering complex scenarios (i.e. dependency level of attributes is equal to 4); (iii) the dependency level of attributes (\vec{W}_c); and (iv) the variation of the user's service preferences (\vec{wA}) and the costs of protection functions (cPF). The system's performance is evaluated by considering two evaluation criteria: (1) total execution time of one iteration; and (2) memory overhead. The tests were conducted on a machine equipped with an Intel i7 2.80 GHz processor and 16 GB of RAM. The chosen execution value for each scenario is an average of 10 sequenced values.

Case 1: We vary the number of privacy risks inferred in a user context. We fix the number of shared attributes at 4, the δ value at 0.6, the vector $\vec{wA} = [1 \ 2 \ 1 \ 2]$, and the costs of the selected protection functions $cPF = \{1, 3, 1, 1\}$. We aim in these experiments to consider complex scenarios, so we consider that the four attributes are dependent and impact all risks. We ran the process seven times such that: the first run reasons over 10 risks, the second 50, the third 100, the fourth 500, the fifth 1000, the sixth 5000, and the last one reasons over 10000 risks. Fig. 11 shows the impact of increasing the number of risks on the algorithm's execution time. We notice that the total execution time is quasi-linear. The system can handle real-time reasoning with an average execution time of less than 2s per iteration up to 1000 risks, and less than 13s up to 10000 risks. When considering RAM usage Fig. 12, it remains quasi-constant up to 1000 risks, then follows a linear evolution until achieving 5000 risks. Therefore, the system is not significantly impacted by the number of risks considered during the reasoning process. Especially as in real scenarios, the number of risks inferred will not practically exceed 1000 for a single user. This highlights the importance of using the GJE method to solve the linear system.

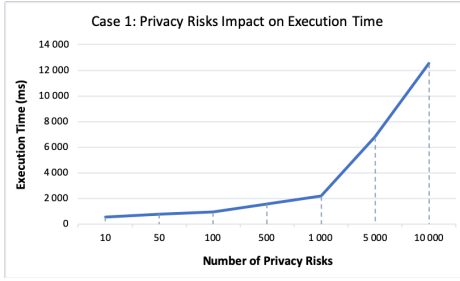


Fig. 11. Execution Time

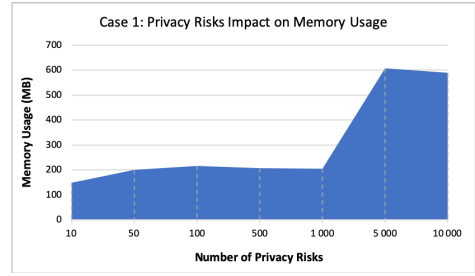


Fig. 12. Memory Usage

Case 2: We vary the number attributes shared by the user. We fix the number of risks at 30, the δ value at 0.6, the vector $\vec{w}A = [1 \ 2 \ 1 \ 2]$, and the costs of the selected protection functions $cPF = \{1, 3, 1, 1\}$. We also fix the dependency level of attributes at 4. We ran the reasoning process eight times such that: the first run reasons over 5 attributes, the second 10, the third 15, the fourth 20, the fifth 25, the sixth 30, the seventh 50, and finally 100 in the last run. According to Fig. 13, the evolution of the execution time remains quasi-linear until 50 attributes with an average of less than 5s, and then tends to be exponential where it reaches a value of 940s for 100 attributes. The evolution is also similar for the memory usage (cf. Fig. 14). We notice that even in complex scenarios where the dependency level of attributes is high, the system maintains good performance and support reasoning in real-time for a number of attributes, shared at once, less than 50, which practically corresponds more to real scenarios.

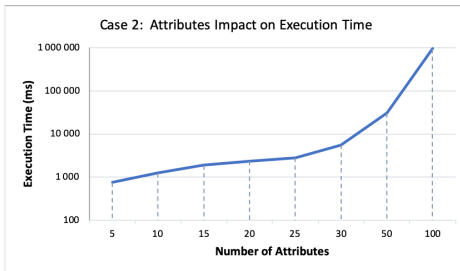


Fig. 13. Execution Time

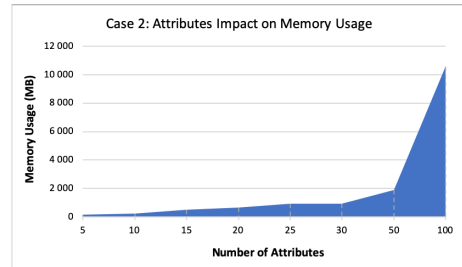


Fig. 14. Memory Usage

Case 3: We vary the dependency level of attributes (i.e. W_c). We fix the number of risks at 30, the δ value at 0.6, the number of attributes at 12, the vector $\vec{w}A = [1 \ 2 \ 1 \ 2]$, and the costs of the selected protection functions $cPF = \{1, 3, 1, 1\}$. We ran the reasoning process six times such that: in the first run, the dependency level is 1 (i.e. attributes are independent), 2 in the second run, 4 in the third, 6 in the fourth, 8 in the fifth, and 10 in the last run. As shown in Fig. 15, the execution time remains quasi-constant with an average value of less than 1s until the dependency level exceeds 6, where the execution time tends to be exponential (e.g., reaches a value of 1227s for a dependency level of 10). Same evolution for RAM usage as illustrated in Fig. 16. However, having a combination of more than six attributes that leads to reveal a psi about a user that cannot be revealed otherwise does not seem a possible scenario. Hence, for a dependency level less or equal to 6, the system supports real-time reasoning with good performance.

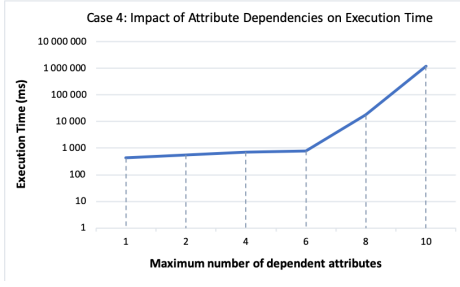


Fig. 15. Execution Time

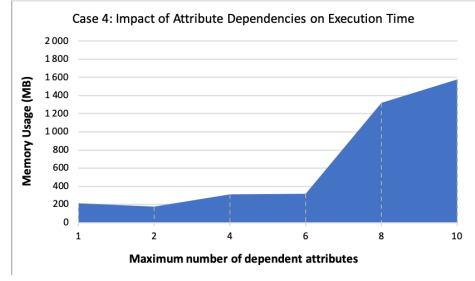


Fig. 16. Memory Usage

Case 4: In this case, we focus on varying the vector of weights $\vec{w}A$ and the costs of protection functions. The aim here is to highlight the importance of storing the appropriate strategies identified by the first δ -Risk operation while being in the same context. We considered the variation of both metrics in the same use case as they both produced the same performance results. Hence, we fixed the number of risks at 30, the δ value at 0.6, the number of attributes at 20, and the dependency level at 4. We ran only the second δ -Risk operation while considering several changes in the weights and costs. As shown in Fig. 17, the execution time remains quasi-constant with an average execution time of less than 500ms. Similar for the RAM usage (cf. Fig.18). Therefore, if within the same context, the user tends to adjust his service preferences, or the availability/cost of protection functions vary, the process will be able to respond quickly and select new best strategies in less than 500ms.

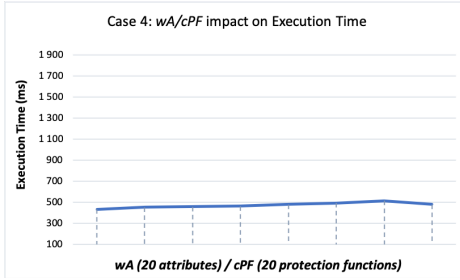


Fig. 17. Execution Time

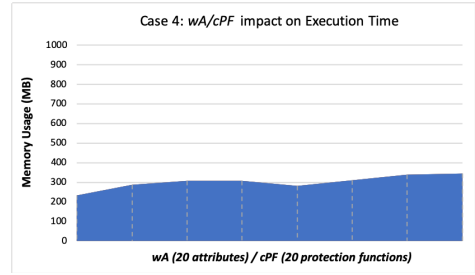


Fig. 18. Memory Usage

Discussion. The experiments conducted show that, within an average time of 2s, δ -Risk can handle up to 1000 risks, 20 attributes (with a dependency level of 4), and a maximum attribute-dependency level of 6. Moreover, δ -Risk can manage, in the same context, variations in service preferences and availability/costs of protection functions within an average time of 500ms.

5.2.2 δ -Risk Effectiveness.

We present in the following a formal study of the approach's effectiveness.

THEOREM 1. The δ -Risk process is always able to identify all possible appropriate strategies \vec{p} that meet the best-case scenario for data utility/privacy protection (i.e. $R_c \cdot \vec{r} = \delta$).

PROOF. The proof consists of two cases, namely a simple and a generic case.

SIMPLE CASE. We consider that the user shares only one attribute, such that $c.SA = \{a_1\}$. Whatever the number of inferred risks is, according to Assumption 1 stated in Section 4.2.1, all risks are linked to attribute a_1 , i.e., W_c is a vector with values equal to 1. Consequently, the system formed will consist of a single equation $p_1 = 1 - \delta$ (cf. Equation 21). This generates one protection strategy $\vec{p} = [p_1] = [1 - \delta]$, which will constitute the best strategy to be delivered, $\vec{b}p = \vec{p} = [1 - \delta]$.

GENERIC CASE. Assume that the user shares m attributes in his context, i.e., $c.SA = \{a_1, \dots, a_m\}$, and the number of risks inferred is n ($R_c.\vec{r} = [r_1 \dots r_n]$). W_c will therefore be a $n \times m$ matrix of $\{0,1\}$ values expressing the impact ω_{ij} of attributes $a_j \in c.SA$ on risks r_i of $R_c.\vec{r}$. According to Equation 21, this will lead to build a linear system of n equations with m unknowns (i.e., $[p_1 \dots p_m]$):

- If $\delta = 0$, this means that the user does not accept to take any risk, i.e., all risks inferred in the present context must be eliminated, such that $R_c.\vec{r} = [r_1 \dots r_n] = [0 \dots 0]$. Hence, the protection levels to apply on attributes must be at their highest level, i.e., leading, according to Equation 19, to the full protection strategy $\vec{b}p = \vec{p} = [1 \dots 1]$.
- If $\delta = 1$, this means that no protection is to be applied on shared attributes and that the user wants to share fine-grained data to preserve the full quality of the received services in exchange. Consequently, no additional protection is needed, and the protection levels must be at their default values. The output will therefore consist of the following strategy:

$$\vec{b}p = \vec{p} = [p_1 \dots p_m] \mid p_j = \begin{cases} 0 & \text{if } p_j \notin eP \\ v & \text{if } p_j \in eP, \text{ where } v \text{ is the enforced value} \end{cases}$$

- If $\delta \in]0;1[$, this means that the user wants to preserve the utility of the data but without taking any risk above the threshold δ . According to Equation 21, the approach will identify all possible appropriate strategies that satisfy the best-case scenario $R_c.\vec{r} = \delta$ using the Gauss Jordan Elimination method to solve the linear system, such that:

$$\left[\begin{array}{cccc|c} \widetilde{\omega}_{11} & \widetilde{\omega}_{12} & \dots & \widetilde{\omega}_{1m} & 1 - \delta \\ \widetilde{\omega}_{21} & \widetilde{\omega}_{21} & \dots & \widetilde{\omega}_{2m} & 1 - \delta \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \widetilde{\omega}_{n1} & \widetilde{\omega}_{n1} & \dots & \widetilde{\omega}_{nm} & 1 - \delta \end{array} \right] \rightarrow M = \left[\begin{array}{cccc|c} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1m} & v_1 \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2m} & v_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \dots & \alpha_{nm} & v_m \end{array} \right]$$

The process results in two possible cases:

- (1) Attributes $\{a_1, \dots, a_m\} \sqsubseteq c.SA$ are independent, and the system has a unique solution, such that:

$$M = \left[\begin{array}{cccc|c} 1 & 0 & \dots & 0 & v_1 \\ 0 & 1 & \dots & 0 & v_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & v_m \end{array} \right]$$

This leads to identify only one strategy that satisfies the best-case scenario, which will therefore constitute the best strategy to deliver, $\vec{b}p = \vec{p} = [v_1, v_2, \dots, v_m]$.

- (2) Attributes $\{a_1, \dots, a_m\} \sqsubseteq c.SA$ are dependent, and the system has an infinite number of solution, such that:

$$M = \left[\begin{array}{cccc|c} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1m} & v_1 \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2m} & v_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \dots & \alpha_{nm} & v_m \end{array} \right] \quad | \quad \exists \vec{\alpha}_l \in M, \{j, k\} \in [1, m] : \alpha_{lj} \times \alpha_{lk} \neq 0$$

Nonetheless, as our goal is to address the data utility/privacy protection trade-off, we focus only in this step on assigning dependent attributes (i.e. a_j/a_k) with the minimum acceptable protection. Hence, the process executes a double iteration process on each dependent p_j/p_k item. The first iteration prioritizes the selected p_j/p_k item, by assigning it a value of 0, which corresponds to the minimum protection to be implemented on attribute a_j/a_k . The second iteration assigns a value of 1 to p_j/p_k (i.e. highest protection), which prioritizes the other dependent p items. Then, both iterations calculate the remaining dependent p items based on the matrix of dependencies M . This procedure identifies several appropriate strategies $\vec{p} \in P$ that meet the trade-off, where each emphasizes at least one dependent attribute.

Therefore, for all δ values, the process is always capable of calculating all possible appropriate strategies that satisfy the best-case scenario for data utility/privacy protection ■

THEOREM 2. The δ -Risk process always delivers the best strategies to the user.

PROOF. The process calculates always all possible appropriate strategies. However, in case the first operation generates more than one strategy, the process executes the ranking function, **Rank**, to select only the best strategies. This function ranks the resulting vectors according to user preferences (i.e. \vec{wA}) and costs of selected protection functions (i.e., cPF). It assigns the highest ranking score to the strategy with the shortest distance to \vec{wA} and the lowest cost of protection. Therefore, for every δ value, if the first operation outputs only one appropriate strategy, this latter constitutes the best strategy. If not, the process always selects the best strategies that most closely satisfy the user's preferences while maximizing data utility and minimizing the protection cost ■

THEOREM 3. For every δ value, the approach always provides the user with at least one best strategy per context.

PROOF. It is easy to see that for every δ value, and for any context particularities, the process always delivers at least one best strategy to the user ■

6 LITERATURE

6.1 Privacy by Design

Privacy by Design (PbD) has brought a new vision for privacy protection to cope with the increasing complexity and interconnectedness of information technologies. Instead of reactively addressing privacy breaches after-the-fact, PbD approaches privacy proactively and tends to prevent privacy-invasive events before they happen by making privacy the default setting [22]. In 2010, PbD has been unanimously adopted as an international privacy standard in the 32nd International Conference of Data Protection and Privacy [23]. Nowadays, PbD is incorporated as a legal requirement in the General Data Protection Regulation (GDPR) [4], and globally recognized as an ISO standard, being developed by ISO/PC 317 Committee for Consumer Protection [24]. Since our global objective is to ensure an effective and meaningful involvement of user in the management of his privacy, we adopt the foundational PbD principles as criteria to compare the referenced works:

- (1) *Proactive not Reactive; Preventative not Remedial*. The approach includes proactive measures to anticipate and prevent privacy violations, i.e., to prevent privacy risks from materializing.
- (2) *Privacy as the Default Setting*. The approach protects the user's privacy by default without requiring user intervention.
- (3) *Privacy Embedded into Design*. Privacy must be an essential component of the core functionality provided by the approach.
- (4) *Full Functionality: Positive-Sum, not Zero-Sum*. The approach seeks to accommodate all interests and objectives in a positive-sum (i.e. win-win manner). We focus here on two sub-criteria:
 - (a) *Privacy Protection vs. Data Utility*. The approach is able to manage the privacy protection/data utility trade-off in a positive-sum.
 - (b) *Hybrid Protection*. The approach supports combination of several protection functions.
- (5) *End-to-End Security: Full Lifecycle Protection*. The approach guarantees privacy protection throughout the entire data lifecycle. Nonetheless, as the sensitivity of data may vary from one context to another, ensuring full protection requires considering context-awareness.
- (6) *Visibility and Transparency: Keep it Open*. The approach aims to ensure that the data/service exchange is operating according to the stated promises and objectives. The operations must remain visible and transparent to users and providers alike.
- (7) *Respect for User Privacy: Keep it User-Centric*. The approach empowers user-friendly options by considering user preferences, and offering measures such as strong privacy defaults and appropriate notice. Hence, we divide this criterion into five sub-criteria in order to cover all user-centered privacy dimensions:
 - (a) *User Awareness: Informed Decision-making*. The approach raises the user's awareness of his privacy risks through appropriate notifications, which helps him making informed privacy protection decisions.
 - (b) *User Privacy Requirements*. The approach takes into account the privacy requirements of the user (e.g., desired protection level, risk level to maintain).
 - (c) *User Interests*. The approach considers the interests of the user (e.g., important services).
 - (d) *User Privacy Management*. The approach empowers the user by enabling him to control and manage his privacy protection.

6.2 Related Work

6.2.1 Context-aware Privacy Preserving in Connected Environments.

Several works were proposed in the literature to address the challenges of security and privacy in connected environments and secure context awareness. Neisse et al. [25] introduced a context-aware security and privacy approach for smart city applications. This approach defines the context by relying on four parameters, namely time, location, network, and speed. It provides a context-based security policy management to control access to the data of users based on a set of Event-Condition-Action (ECA) rules. It also provides a privacy mechanism based on pseudonymization and delayed message delivery. Hence, the access to data could be accepted, denied, modified (using pseudonymization), or delayed. Matos et al. [26] presented an overview of their context-aware security approach, that provides authentication, authorization, access control, and privacy-preserving to fog and edge computing environments. However, the authors did not detail the components of their architecture, as they did not explain how privacy is approached in their work. Gheisari et al. [27] proposed a context-aware privacy-preserving approach for IoT-based smart city

using Software Defined Networking. The authors showed that the privacy is preserved through splitting sensitive data and sending split parts via a secure route. The decision made by the SDN controller is based on data sensitivity (context) and routes credits. Sylla et al. presented in [28] a global vision of their context-aware security and privacy as a service (CASPaas) architecture by briefly discussing the role of each module. They mentioned that the privacy module will be able to continuously analyze the user context and inform the user if there is a proven risk to his privacy. However, they have not yet explored any of the architecture modules. Alagar et al. [29] introduced a Context-Sensitive Role-based Access Control (CRBAC) scheme for healthcare application. This approach defines two types of access control: open access, for authenticated clients/medical devices; and closed Access, for non-member clients/devices. CRBAC is user-centric, where the privacy requirements are included as context-sensitive rules to be enforced whenever patient health information are shared by things.

PbD Principles	Proactive & Preventative	Privacy as Default Setting	Privacy in the Design	Full Functionality		Context-aware Security	Visibility & Transparency	User-friendly Options			
				Privacy vs. Utility	Hybrid Protection			User Awareness	User Privacy Requirements	User Interests	User Privacy Management
Nesse et al. [25]	Yes	Yes	Yes	Yes	No	Yes	Yes	No	No	No	No
Matos et al. [26]	Yes	Y/N	Yes	Y/N	No	Yes	No	No	No	No	No
Gheisari et al. [27]	Yes	Yes	Yes	Yes	No	Yes	Y/N	No	No	No	No
Sylla et al. [28]	Yes	Y/N	Yes	Y/N	No	Yes	Y/N	Yes	Yes	No	Yes
Alagar et al. [29]	Yes	Yes	Yes	Yes	No	Yes	Y/N	No	Yes	No	Yes

^a Y/N means that the referenced work did not approach this concept.

Table 1. Privacy preserving approaches comparison

Discussion. As shown in Table 1, none of the existing approaches cover all PbD principles. Moreover, the proposed frameworks are dedicated to specific application domains. Therefore, we introduce in this paper a new Privacy by Design framework for context-aware privacy management in connected environments. Our framework is generic and can be re-usable in different application domains. Table 2 details how PbD principles are satisfied by our framework.

PbD Principles		Privacy Oracle Framework
Proactive & Preventative		Privacy Oracle implements a proactive reasoning process to infer the risks involved before being materialized, and thus to adapt the protection measures before releasing data to consumers.
Privacy as the Default Setting		Privacy Oracle protects the user's privacy by default. In fact, the system keeps transmitting the user's data to consumers at their default protection condition as long as no risk occurs. However, once a risk is inferred, the system switches automatically to the highest level of protection (i.e. it stops sharing data) until one protection strategy is selected by the user, which leads to optimizing the protection.
Privacy in the Design		The core functionality of the framework is the context-aware privacy management.
Full Functionality	Privacy vs. Utility	Privacy Oracle is fully functional, it treats the privacy protection/data utility trade-off in a positive-sum. The protection strategies are always optimized in such a way that closely satisfy user's privacy requirements and preferences while maximizing data utility.
	Hybrid Protection	Computed strategies support the combination of multiple protection functions in order to achieve the desired privacy level while minimizing the protection cost.
Context-aware Security		The user data are protected before being transmitted to data consumers. This makes the user's privacy guaranteed for the entire data lifecycle. The system handles context-awareness, it performs a dynamic adaptation of protection measures to cope with context-sensitivity.
Visibility & Transparency		Privacy Oracle ensures continuous monitoring and dynamic updating of privacy policies according to the context-based strategies adopted by the user.
User-friendly Options	User Awareness	Privacy Oracle improves the awareness of the user by providing him with a dynamic overview of privacy risks according to the evolution of his context.
	User Privacy Requirements	Privacy Oracle considers the following user privacy requirements: the maximum level of risk to retain in his context, the enforced protection levels to maintain for specific attributes.
	User Interests	Privacy Oracle considers the service preferences of the user.
	User Privacy Management	Privacy Oracle enables the user to control his privacy protection and make informed and optimal protection decisions based on his preferences and contexts.

Table 2. Framework responsiveness to PbD principles

6.2.2 Privacy Risk Inference & Quantification.

Alerting users about their privacy risks constitutes a key step towards improving their privacy decision-making. Hence, the privacy risk inference and quantification fields have received extensive

attention over the last decade. Christin et al. [30] investigated mechanisms to warn users about potential privacy risks of sharing personal information. Their results show that more than 70% of the participants would change their settings after experiencing picture-based warnings. Important to underline that this approach did not incorporate any privacy risk inference system. Similarly, Bal et al. [31] introduced a novel privacy risk communication system that provides the user with more meaningful privacy information based on the actual behavior of smartphone apps. Hatamian et al. [32] proposed an informed decision-making supporter, called beacon alarming, to inform users of the data accessed by different smartphone applications. They also suggested expanding the functionality of the alarming system by employing fuzzy logic in order to assess the privacy risk scores of installed applications. Zhang et al. [33] provided a formal privacy quantification model for location-based services (LBS) that uses the Bayes conditional risk as a privacy metric. This model employs a general definition of conditional privacy regarding the adversary's estimation error to compare the LBS privacy metrics. Banerjee et al. [34] studied the privacy threats resulting from the deviations of data collectors practices from what they promise in their privacy policies, as opposed to the user's needs. Ngoc et al. [35] introduced a new metric to quantify privacy for users in social networking sites, based on probability and entropy theory.

7 CONCLUSION

This paper presents a user-centered context-aware approach for privacy management in connected environments, denoted as δ -Risk, that assists users in optimizing their privacy decisions. To do so, δ -Risk features a new privacy risk quantification model to dynamically calculate and select the best protection strategies for users based on their preferences and contexts. Computed strategies are optimal in that they seek to closely satisfy user's requirements and preferences while maximizing data utility and minimizing the protection cost. We implemented our approach and evaluated its performance and effectiveness based on several use cases. Results show that δ -Risk: (1) handles privacy reasoning in real-time, which makes it able to support the user in any context, including ephemeral contexts; and (2) provides always the user with at least one best strategy per context.

As future work, we would like to study the dependencies between contexts. In fact, at this stage, the privacy oracle reasons on each context apart without considering historical contexts/risks. Nonetheless, contexts can be connected, which can affect the current risks or even generate new risks for users. Therefore, we want to tackle the challenges of cross-context dependencies while considering both logical and spatio-temporal aspects. We also want to address the challenge of measuring the impact of attributes on risks. In fact, this impact is probabilistic and attributes may have different impact values on risks. Finally, we aim to explore the *privacy protection service* component of our framework and study the related research problems, including: how to select the most appropriate protection functions to be executed on attributes' data, what metrics to consider, and how to measure system vulnerabilities in accordance with this selection.

REFERENCES

- [1] B. George, J. M. Kang, and S. Shekhar, "Spatio-temporal sensor graphs (stsg): A data model for the discovery of spatio-temporal patterns," *Intelligent Data Analysis*, vol. 13, no. 3, pp. 457–475, 2009.
- [2] K. B. Chaaya, M. Barhamgi, R. Chbeir, P. Arnould, and D. Benslimane, "Context-aware system for dynamic privacy risk inference: Application to smart iot environments," *Future Generation Computer Systems*, vol. 101, pp. 1096–1111, 2019.
- [3] M. A. Lisovich, D. K. Mulligan, and S. B. Wicker, "Inferring personal information from demand-response systems," *IEEE Security & Privacy*, vol. 8, no. 1, pp. 11–20, 2010.
- [4] N. Vollmer, "Table of contents EU General Data Protection Regulation (EU-GDPR)," May 2018.
- [5] State of California Department of Justice, "California Consumer Privacy Act (CCPA)," October 2018.
- [6] C. Castelluccia, M. Cunche, D. L. Metayer, and V. Morel, "Enhancing transparency and consent in the iot," in *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, pp. 116–119, April 2018.

- [7] I. D. Addo, S. I. Ahamed, S. S. Yau, and A. Buduru, "A reference architecture for improving security and privacy in internet of things applications," in *2014 IEEE International Conference on Mobile Services*, pp. 108–115, June 2014.
- [8] S. Kumar, S. K. Singh, A. K. Singh, S. Tiwari, and R. S. Singh, "Privacy preserving security using biometrics in cloud computing," *Multimedia Tools and Applications*, vol. 77, no. 9, pp. 11017–11039, 2018.
- [9] D. W. Chadwick and K. Fatema, "A privacy preserving authorisation system for the cloud," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1359–1373, 2012.
- [10] "Data in the post-gdpr world," *Computer Fraud & Security*, vol. 2018, no. 9, pp. 17 – 18, 2018.
- [11] "Marketing firm exactis leaks 340 million files containing private data," *Mail Online*, 2018.
- [12] M. Barhamgi, C. Perera, C. Ghedira, and D. Benslimane, "User-centric privacy engineering for the internet of things," *IEEE Cloud Computing*, vol. 5, no. 5, pp. 47–57, 2018.
- [13] V. Y. Pillitteri and T. L. Brewer, "Guidelines for smart grid cybersecurity," Tech. Rep. NISTIR 7628 Revision 1, National Institute of Standards and Technology, 2014.
- [14] A. S. Householder, *The theory of matrices in numerical analysis*. Courier Corporation, 2013.
- [15] D. Nagarajan, T. Tamizhi, M. Lathamaheswari, and J. Kavikumar, "Traffic control management using gauss jordan method under neutrosophic environment," in *AIP Conference Proceedings*, vol. 2112, 2019.
- [16] L. Shang, S. Petiton, and M. Hugues, "A new parallel paradigm for block-based gauss-jordan algorithm," in *2009 Eighth International Conference on Grid and Cooperative Computing*, pp. 193–200, 2009.
- [17] L. M. Aouad and S. G. Petiton, "Parallel basic matrix algebra on the grid'5000 large scale distributed platform," in *2006 IEEE International Conference on Cluster Computing*, pp. 1–8, 2006.
- [18] L. Shang, Z. Wang, S. G. Petiton, Y. Lou, and Z. Liu, "Large scale computing on component based framework easily adaptive to cluster and grid environments," in *The Third ChinaGrid Annual Conference*, pp. 70–77, IEEE, 2008.
- [19] L. M. Aouad, S. G. Petiton, and M. Sato, "Grid and cluster matrix computation with persistent storage and out-of-core programming," in *2005 IEEE International Conference on Cluster Computing*, pp. 1–9, IEEE, 2005.
- [20] M. Xue, P. Kalnis, and H. K. Pung, "Location diversity: Enhanced privacy protection in location based services," in *International Symposium on Location-and Context-Awareness*, pp. 70–87, Springer, 2009.
- [21] A. Chernev, U. Böckenholt, and J. Goodman, "Choice overload: A conceptual review and meta-analysis," *Journal of Consumer Psychology*, vol. 25, no. 2, pp. 333–358, 2015.
- [22] A. Cavoukian and M. Chibba, "Start with privacy by design in all big data applications," in *Guide to big data applications*, pp. 29–48, Springer, 2018.
- [23] A. Cavoukian, "Privacy by design [leading edge]," *IEEE Technology and Society Magazine*, vol. 31, no. 4, pp. 18–19, 2012.
- [24] "ISO/PC 317 consumer protection: Privacy by design for consumer goods and services," 2018.
- [25] R. Neisse, G. Steri, G. Baldini, E. Tragos, I. N. Fovino, and M. Botterman, "Dynamic context-aware scalable and trust-based iot security, privacy framework," *Chapter in Internet of Things Applications-From Research and Innovation to Market Deployment, IERC Cluster Book*, 2014.
- [26] E. de Matos, R. T. Tiburski, L. A. Amaral, and F. Hessel, "Providing context-aware security for iot environments through context sharing feature," in *17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (TrustCom)*, pp. 1711–1715, IEEE, 2018.
- [27] M. Gheisari, G. Wang, W. Z. Khan, and C. Fernández-Campusano, "A context-aware privacy-preserving method for iot-based smart city using software defined networking," *Computers & Security*, vol. 87, p. 101470, 2019.
- [28] T. Sylla, M. A. Chalouf, F. Krief, and K. Samaké, "Towards a context-aware security and privacy as a service in the internet of things," in *IFIP International Conference on Information Security Theory and Practice*, pp. 240–252, 2019.
- [29] V. Alagar, A. Alsaig, O. Ormandjiva, and K. Wan, "Context-based security and privacy for healthcare iot," in *2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pp. 122–128, IEEE, 2018.
- [30] D. Christin, M. Michalak, and M. Hollick, "Raising user awareness about privacy threats in participatory sensing applications through graphical warnings," in *Proceedings of International Conference on Advances in Mobile Computing & Multimedia*, pp. 445–454, 2013.
- [31] G. Bal, K. Rannenberg, and J. Hong, "Styx: Design and evaluation of a new privacy risk communication method for smartphones," in *IFIP International Information Security Conference*, pp. 113–126, Springer, 2014.
- [32] M. Hatamian and J. Serna-Olvera, "Beacon alarming: Informed decision-making supporter and privacy risk analyser in smartphone applications," in *2017 IEEE International Conference on Consumer Electronics*, pp. 468–471, IEEE, 2017.
- [33] X. Zhang, X. Gui, F. Tian, S. Yu, and J. An, "Privacy quantification model based on the bayes conditional risk in location-based services," *Tsinghua Science and Technology*, vol. 19, no. 5, pp. 452–462, 2014.
- [34] M. Banerjee, R. K. Adl, L. Wu, and K. Barker, "Quantifying privacy violations," in *Workshop on Secure Data Management*, pp. 1–17, Springer, 2011.
- [35] T. H. Ngoc, I. Echizen, K. Komei, and H. Yoshiura, "New approach to quantification of privacy on social network sites," in *24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 556–564, IEEE, 2010.