



HAL
open science

δ -Risk: Toward Context-aware Multi-objective Privacy Management in Connected Environments

Karam Bou-Chaaya, Richard Chbeir, Mansour Naser Alraja, Philippe Arnould, Charith Perera, Mahmoud Barhamgi, Djamal Benslimane

► **To cite this version:**

Karam Bou-Chaaya, Richard Chbeir, Mansour Naser Alraja, Philippe Arnould, Charith Perera, et al.. δ -Risk: Toward Context-aware Multi-objective Privacy Management in Connected Environments. ACM Transactions on Internet Technology, 2021, 21 (2), pp.51. 10.1145/3418499 . hal-03259581

HAL Id: hal-03259581

<https://hal.science/hal-03259581v1>

Submitted on 14 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

δ -Risk: Toward Context-aware Multi-objective Privacy Management in Connected Environments

KARAM BOU-CHAAYA and RICHARD CHBEIR, Universite de Pau et des Pays de l'Adour, E2S UPPA, LIUPPA, Anglet, France

MANSOUR NASER ALRAJA, Dhofar University, Oman

PHILIPPE ARNOULD, Universite de Pau et des Pays de l'Adour, E2S UPPA, LIUPPA, Mont-de-Marsan, France

CHARITH PERERA, Cardiff University, United Kingdom

MAHMOUD BARHAMGI and 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 article, we address the second goal by proposing a user-centric 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 the user based on her preferences and contexts. Computed strategies are optimal in that they seek to closely satisfy user requirements and preferences while maximizing data utility and minimizing the cost of protection. We implemented our proposed approach and evaluated its performance and effectiveness in various scenarios. The results show that δ -Risk delivers scalability and low-complexity in time and space. Besides, it handles privacy reasoning in real-time, making it able to support the user in various contexts, including ephemeral ones. It also provides the user with at least one best strategy per context.

CCS Concepts: • **Security and privacy** → **Privacy-preserving protocols**; • **Theory of computation** → **Continuous optimization**

Additional Key Words and Phrases: User-centric privacy, privacy risk quantification, privacy by design, context-aware computing, semantic reasoning, Internet of Things

This work is supported by the Research Council (TRC), Sultanate of Oman (Block Fund-Research Grant).

Authors' addresses: K. Bou-Chaaya and R. Chbeir, Universite de Pau et des Pays de l'Adour, E2S UPPA, LIUPPA, Anglet, France; emails: {karam.bou-chaaya, richard.chbeir}@univ-pau.fr; M. N. Alraja, Dhofar University, Salalah 2509, Oman; email: malraja@du.edu.om; P. Arnould, Universite de Pau et des Pays de l'Adour, E2S UPPA, LIUPPA, Mont-de-Marsan, France; email: philippe.arnould@univ-pau.fr; C. Perera, Cardiff University, United Kingdom; email: charith.perera@ieee.org; M. Barhamgi and D. Benslimane, Université Claude Bernard Lyon 1, LIRIS lab, Lyon, France; emails: {mahmoud.barhamgi, djamal.benslimane}@univ-lyon1.fr.

ACM Reference format:

Karam Bou-Chaaya, Richard Chbeir, Mansour Naser Alraja, Philippe Arnould, Charith Perera, Mahmoud Barhamgi, and Djamel Benslimane. 2021. *δ -Risk*: Toward Context-aware Multi-objective Privacy Management in Connected Environments.

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), and so on.

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, and so on. 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, 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 two main limitations:

- (1) *lack of user awareness*. The user may not be completely aware of the direct and indirect privacy risks associated with sharing her data with providers to correctly specify her preferences in the first place. She may simply not know what sensitive information might be revealed from her data when data pieces are analyzed in isolation or combined with each other or/and with other side information acquired from external data sources (e.g., 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 may downgrade the accuracy of associated services; or under-protective, leading consequently to privacy breaches. Therefore, the user must be able to make dynamic adjustments to her privacy decisions according to the evolution of her context.

The objectives of our research work are to design suitable solutions that address the aforementioned two limitations, and to provide a complete context-aware privacy framework that meets the guidelines of the Privacy by Design (PbD) standard. Specifically, the framework needs to cope with: (i) raising user awareness of the privacy risks associated with sharing her data with consumers; (ii) assisting the user in optimizing her privacy decisions according to her contexts and preferences; and (iii) ensuring appropriate data protection in accordance with user decisions before transmitting it to data consumers. To overcome the first limitation, we proposed in 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 context evolves. 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 contexts. That is, over-protective measures limit the utility of shared data to eliminate the risks, but could also downgrade the accuracy of services. Likewise, 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 sometimes be fast (i.e., in real-time). Therefore, the solution must be: (i) user-friendly (i.e., not complex for the user); (ii) adaptive to the context and expressive in representing various contexts; (iii) scalable, to handle reasoning over an increasing number of risks and attributes. It must also maintain (iv) computational and storage efficiency, which makes it operational on various types of devices, including those with limited resources.

To cope with these challenges, this article proposes a new user-centric, context-aware and multi-objective privacy management approach, denoted δ -Risk. The proposed approach assists the user in optimizing her privacy decisions, by providing her with dynamic, fast, and optimal protection strategies that could be adopted in her context. Each of these strategies minimizes the risks inferred in the user context to meet her privacy requirements while maximizing the utility of data and minimizing the cost of protection. The strategy delivered consists of the best combination of protection levels to be assigned to shared attributes, i.e., the combination that best satisfies user 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 prototype in various scenarios, including worst-case ones. Then, we formally studied its effectiveness. The results show that δ -Risk delivers scalability and computational/storage efficiency. It handles reasoning in real-time, which makes it able to support the user in various contexts, including ephemeral ones (i.e., contexts with short time periods). It is always capable of: (i) identifying all possible appropriate strategies that answer the data utility/privacy ‘protection’ can be removed trade-off; (ii) delivering the best strategies to the user; and (iii) providing the user with at least one best strategy per context.

The remainder of this article is organized as follows. Section 2 introduces a scenario that motivates our proposal and identifies the challenges to tackle. Section 3 presents our CaPMan framework, and provides formal definitions of the key terms used in the article. Section 4 details the

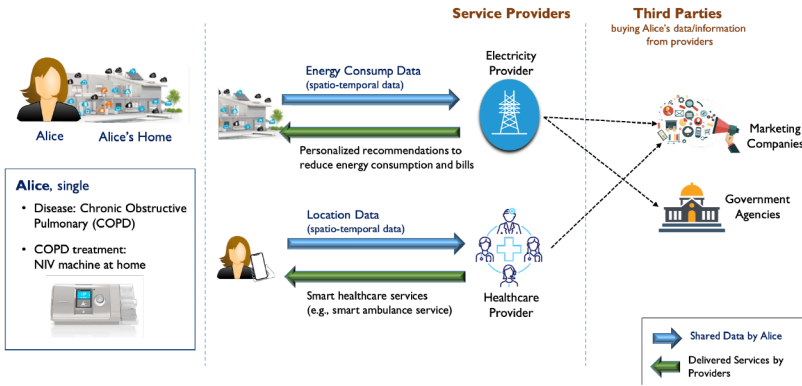


Fig. 1. Motivating Scenario.

δ -Risk approach. Section 5 outlines the experiments and tests performed. Section 6 highlights the Privacy by Design standard and reviews existing context-aware privacy preserving approaches in connected environments. Finally, Section 7 concludes the article and discusses future research directions.

2 MOTIVATING SCENARIO

To motivate our proposal, we investigate a real-life scenario that showcases some of the privacy risks entailed from sharing data with consumers, and highlights the need for dynamic/contextual adaptations of user-privacy decisions. Figure 1 illustrates the proposed scenario. Assume that Alice is a **Chronic Obstructive Pulmonary Disease (COPD)** patient. She pursues her medical treatment remotely using a 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 due to many factors such as the sensitivity of her context, or the third parties with whom the provider communicates her data. Assume that both providers have signed contracts with third parties interested in exploiting the data of customers (e.g., Alice) for different purposes, including marketing companies and government agencies. Marketing companies could be interested in exploiting consumption data to analyze the lifestyle of customers to send them targeted advertisements (e.g., advertisements about appliances that customers own or do not own). Government agencies could be interested in identifying customers involved in wrongdoing (e.g., fraud, crimes, etc.).

Even though Alice is notified, through agreed policies, of consumers who have access to her data, she may not necessarily be aware of the privacy risks involved with this sharing. The risks can be of two types: mono-source and multi-source risks. Mono-source risks arise from sharing data with a single data consumer. For instance, analyzing the energy consumption data (see the signature in Figure 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

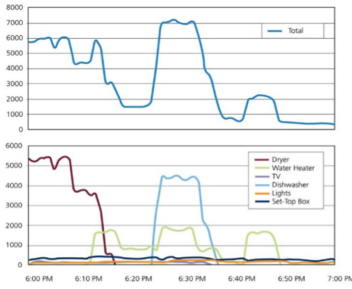


Fig. 2. Energy consumption signature.

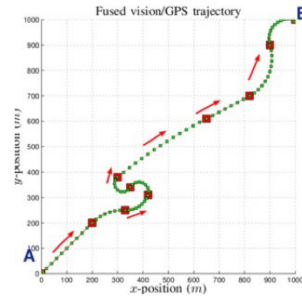


Fig. 3. Location data pattern.

(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 patterns (cf. Figure 3) can also entail significant mono-source risks for Alice such as the risks of disclosing her habits and routines, behaviors, political/religious affiliations, identity based on her locations in personal environments (e.g., home, office), and her health conditions based on frequent visits to hospitals. 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.

Multi-source risks are more complex risks that arise when customer data are communicated between data consumers (cf. Figure 1). For example, assume that Alice has unlawfully certified that she is living alone to be eligible for a welfare program when submitting her application. A marketing company having access to both location and consumption data can infer this fraud (it is enough to identify the use of particular devices, such as microwave and TV, while Alice is outside her home).

Once Alice is alerted of the risks involved in her current situation, 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 ones for Alice. Assume that the services offered by the healthcare provider are important to Alice. She may want to minimize her risks when being located in the pulmonary rehabilitation center, but without completely losing health 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 assist Alice in optimizing 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. Coping with user expertise:** People may have different levels of expertise to properly express their requirements/preferences and interact with the system. The proposed solution must therefore be user-friendly, allowing the guided assistance to be tailored to the user's expertise in order to maintain good quality of human-machine interactions.
- **Challenge 2. Making optimal context-based privacy decisions:** The user-privacy decisions depend on her situation (e.g., risks inferred) and preferences. Therefore, the proposed solution should always be able to provide the user with optimal and adaptive protection strategies to cope with the dynamicity of her contexts and preferences.
- **Challenge 3. Delivering scalability and efficiency:** The solution must be scalable, i.e., handles reasoning over an increasing number of sensed attributes and privacy risks. It should

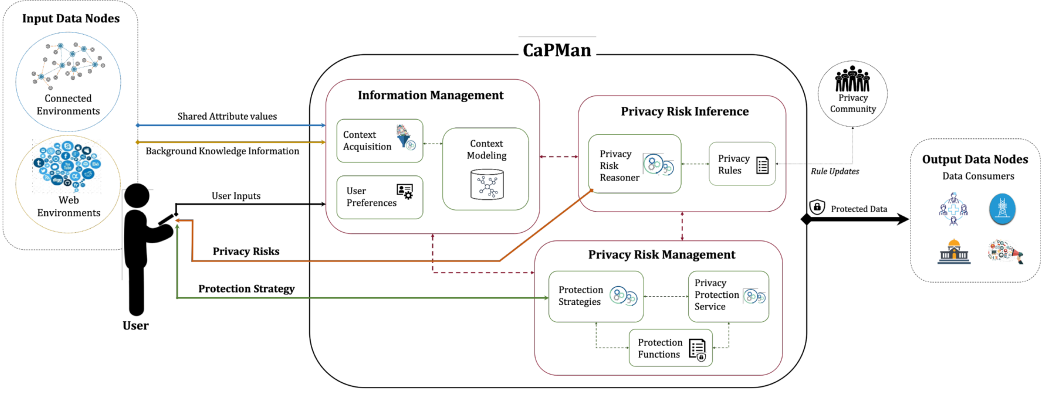


Fig. 4. CaPMan Framework.

also maintain computational and storage efficiency in order to support the user in various contexts and be operational on different types of devices, including resource-constrained ones.

3 CaPMan: FRAMEWORK FOR CONTEXT-AWARE PRIVACY MANAGEMENT IN CONNECTED ENVIRONMENTS

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, denoted **CaPMan**. We start by explaining the functioning of the framework, and formally defining the key terms used in the article. Then, we briefly describe the framework modules.

The aim of CaPMan is to provide a user-centric reasoning system that assists the user in managing her privacy protection (cf. Figure 4). This system can be embedded on user devices (e.g., mobile phone, computer, tablet) as middleware between the user and the connected consumers, so that it manages the user's data before being released to consumers. We consider in this study that all data consumers are not trusted by the user. Hence, the user starts by specifying the inputs: (1) the list of sensed attributes that are currently shared with data consumers; and (2) her preferences (detailed in Section 4). The system, on its side, collects the data values of sensed attributes, and collects additional background data describing the user and her surrounding environment from other external data nodes (e.g., social media platforms, public databases).

Let u denotes the *user of interest*.

Definition 1 (Data Node). Let DN be the set of input/output *data nodes* $\{dn_1, \dots, dn_n\}$. Input data nodes are data sources from which the data is collected (e.g., sensors, social networks). Output data nodes are data consumers with whom the data is shared (i.e., service providers and third parties). $dn \in DN$ is formalized as follows:

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

- *desc* is the textual description of dn (e.g., gps-sensor, facebook, healthcare-provider)
- *id* denotes the identity of dn , expressed as a uniform resource identifier (URI). All the definitions must end with a "black square": ■

Definition 2 (Physical Environment). Let E_u be the set of *physical environments* $\{env_1, \dots, env_n\}$ where the user u is/was located. $env \in E_u$ can be of two types: connected

(i.e., hosts smart systems) or unconnected environment.

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

- $desc$ denotes the textual description of env (e.g., home, office, mall)
- sz expresses the *spatial zone* of env (cf. Definition 3)
- $Sys \sqsubseteq DN$ is the set of systems (*data nodes*) deployed in env (e.g., sensors, actuators). For unconnected environments, $Sys = \emptyset$. ■

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 $loc : \langle long; lat; alt \rangle$, where $long$, lat , and alt denote, respectively, the longitude, latitude, and altitude of loc . ■

Example 1. The home of Alice is a physical environment, $home \in E_u$. It can be defined as follows:

```

home :
- desc : home of Alice
- sz : homeZone :  $\langle loc_1; loc_2; loc_3; loc_4 \rangle$ 
  loc1 :          loc2 :          loc3 :          loc4 :
- 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
- Sys = {sensor1 :  $\langle desc : energy-consump-sensor; id : 46.193.0.164 \rangle$ }

```

Physical environments may be dependent (e.g., spatial inclusion), making the associated information dependent. For instance, CaPMan may receive information describing the home and the city of the user, where home is inside the city, making the information collected on both environments dependent. However, in this study, we do not consider the dependency of environments, and the system reasons on each environment in isolation.

Definition 4 (Attribute). Let A be the *set of attributes* $\{a_1, a_2, \dots, a_n\}$ that include data describing u and her physical environments. $a \in A$ is formalized as follows:

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

- $desc$ denotes the textual description of a (e.g., location, energy-consump, age)
- $access \in \{r; r/w\}$ represents the access rights of the CaPMan system to the data of a , which can be read or read/write.
- $source \in DN$ is a *data node* expressing the data source from which a is captured. $source$ can derive from Connected environments (e.g., sensor, device) or Web environments (e.g., social network, public database)
- $D_{consumer}$ represents the set of data consumers with which a is shared, such that:

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

$-dc_i \in DN$ is a *data node* expressing a data consumer (i.e., service provider or third party)

- $D_{consumer} = \emptyset$ indicates that data consumers are unknown
- $D_{consumer} = \{\perp\}$ denotes that a is a public attribute (i.e., shared with everyone)
- $ent \in \{u, env\}$ denotes the entity described by a , which can be u or an environment $env \in E_u$
- $Log = \{\langle rv; M \rangle\}$ is the set of data values of a . Log can be seen as the log file of a , where:
 - rv denotes the raw data value
 - M is the set of metadata characterizing rv (e.g., time/location of capture, data-type).

Definition 4.1 (Sensed Attribute). Let $SA \sqsubseteq A$ be the set of **sensed attributes**, i.e., attributes that characterize sensed data by deployed/wearable sensors, and on which the CaPMan system has access to control and manage, such that:

$$\forall a \in SA : a.access = r/w. \quad \blacksquare$$

Definition 4.2 (Background-oriented Attribute). Let $BA \sqsubseteq A$ be the set of **background-oriented attributes**, i.e., attributes that characterize background data about the user and/or her environments, and on which the CaPMan system has read-only access, such that:

$$\forall a \in BA : a.access = r. \quad \blacksquare$$

Example 2. Alice has two sensed attributes that can be represented as follows:

$a_1 :$ - desc : energy consumption - access : r/w - 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 - access : r/w - 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 : $\langle (-33.0534, 16.3103); \{t_{capture} : 11:00:00\} \rangle$ $\langle (-36.0534, 17.4401); \{t_{capture} : 11:01:00\} \rangle$
--	--

Example 3. Assume that the system has captured background data describing the marital status of Alice, which is publicly shared on her Facebook profile:

$a_3 :$ - desc : marital-status - access : r - source : socialAccount1: $\langle desc : facebook; id : https://www.facebook.com/Alice \rangle$ - $D_{consumer} = \{\perp\}$ - ent : u - Log : $\langle single; \{t_{capture} : 12:00:00; d_{type} : String\} \rangle$

The CaPMan system models acquired contextual data and launches the risk reasoner that performs rule-based reasoning over context data, while relying on imported *privacy rules* (cf. Definition 6) to infer the *privacy risks* (cf. Definition 8) involved. If no risk is inferred, the system

continues to generate data values for consumers as received (i.e., without applying additional protection). Otherwise, it alerts the user about the risks involved in this data sharing, and recommends a list of best protection strategies that could be adopted in this situation. Meanwhile the system stops communicating data to consumers and waits for the user's response. When the user selects the strategy to implement, the system accordingly protects the pending data values of attributes and releases the protected version of data to consumers. The system continues to apply the same protection strategy to the received data values until a new context emerges, where the entire reasoning process is relaunched to consider the changes in context and their impact on the risk overview and strategies. By default, the system stores only consecutive contexts, which makes it low-complex in storage (cf. Theorem 1), and operational on various types of devices, including those with limited resources (cf. Challenge 3). Knowing that the number of historical contexts to be kept in the cache can be increased according to the storage capacity of the integrating device (this will be further discussed in future work). Therefore, the global process (i.e., risk inference and strategy identification process) is by default executed once per context.

Definition 5 (User Context). A **user context**, c , is a spatio-temporal context during which the system has a fixed set of attributes describing u and her environment. c is formalized as follows:

$c : \langle t; s; A \rangle$, 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
- s expresses the *spatial zone* of c (cf. Definition 3)
- A represents the set of *attributes* characterizing c ($A = SA \cup BI$). It contains at least one *sensed attribute*, i.e., $\exists a \in c.A : a.access = r/w$

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

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

Definition 6 (Privacy Rule). Let PR be the **set of privacy rules** $\{pr_1, pr_2, \dots, pr_n\}$ that define the risks to be inferred by the system (i.e., mono-source and multi-source risks). $pr \in PR$ is a reasoning rule indicating which *attribute* or combination of *attributes*, if processed, leads to reveal which *privacy-sensitive information* about u . pr includes at least one *shared attribute*, such that

$pr : A' \rightarrow psi$, where:

- $A' = \langle a_1 \wedge \dots \wedge a_n \rangle \mid a_i \in A \forall i \in [1; n]$ denotes the sequence of *attributes* combined using the logical AND operator (i.e., \wedge), where $\exists a \in A' : a.access = r/w$
- psi expresses the *privacy-sensitive information* to be disclosed by A' (cf. Definition 7). ■

Definition 7 (Privacy-Sensitive Information). A **privacy-sensitive information**, $psi \in PSI$, is a piece of information that, if disclosed, can be harmfully used against u . psi might be revealed when processing/analyzing the knowledge acquired about u and her environment. 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: user-profile information (e.g., disease, salary), habits (e.g., daily activities), behaviors, preferences, presence/absence, sleep/wake cycles, appliances and medical devices used, and fraud. ■

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 $psi_1 = \text{“presence/absence at home”}$; $pr_1 : a_1 \rightarrow psi_1$.

Definition 8 (Privacy Risk). A **privacy risk**, r , is defined as the probability of satisfying the privacy rule $pr \in PR$ in the user context c . r can be seen as the probability of disclosing the privacy-sensitive information $(psi)_{pr}$ through the combination $(A')_{pr}$. It is generated when the relevant pr is satisfied in c , and remains valid for the entire time period of c . 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 can be represented as follows:

$$r = P(pr) \mid pr \in PR \text{ and } r \in [0, 1]. \quad \blacksquare$$

The number of risks to infer in a single process iteration is fixed, it depends on the number of privacy rules imported in this iteration. It is possible to have several risks r_i mapped to the same $psi \in PSI$ in case the psi is disclosed through various combinations of attributes defined by different privacy rules.

Example 5. Assume that when launching the process, the rule pr_1 is satisfied in the context of Alice. Therefore, a risk r_1 is inferred for Alice, such that: $r_1 = P(pr_1) = 1$.

3.1 CaPMan Framework Modules

As illustrated in Figure 4, CaPMan is a modular framework composed 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 Module. Inferring context-aware risks requires first to build up a global view of the user context. This is done by gathering attributes describing the user and her surrounding physical environment. Hence, this module is responsible for managing (i.e., capturing and modeling) user attributes and preferences. It includes the following components: (i) *context acquisition*, in charge of capturing attributes from the user and her Connected/Web environments; (ii) *user preferences*, responsible for managing the preferences of the user; and (iii) *context modeling*, liable for modeling acquired attributes 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. This is motivated by the fact that 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; 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 information received about the user, 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>.

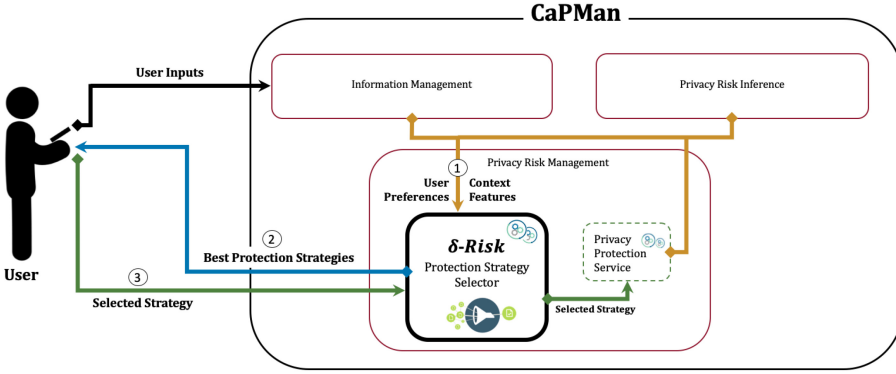
3.1.2 Privacy Risk Inference Module. Responsible for inferring the risks involved in the user context. To achieve this, this module includes two components. First, the *privacy rules* component, which handles the definition/import of privacy rules that specify the risks to be detected by the system. The rules are defined according to the given syntax in Definition 6, 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 on the quality of the defined rules. We assume in this study that the privacy rules, defined by experts from the privacy community, are pre-validated (this validation is out of scope of this study). Second, the *privacy risk reasoner* component, which provides a semantic rule-based reasoning engine proposed in Reference [2]. This engine reasons on modeled information to dynamically infer the risks involved in the user context.

3.1.3 Privacy Risk Management Module. Responsible for assisting the user in the management of her privacy by: (i) assessing and minimizing the risks inferred based on the privacy requirements and interests of the user; (ii) delivering optimized and meaningful strategies; and (iii) protecting sensor data streams according to the context-dependent protection strategy selected by the user. In order to do so, the module consists of three components. First, the *protection strategies* component, in charge of managing user risks and identifying the best protection strategies to be suggested to the user. Computed strategies are optimal in that they seek to closely satisfy user requirements and preferences while maximizing data utility and minimizing the cost of protection. The risk manager continuously adjusts the strategies provided to cope with the dynamic nature of the user context and preferences. In fact, the user might change progressively her preferences due to the sensitivity of the risks entailed, or the sensitivity of the context (e.g., private meeting, located in a hospital). Second, the *protection functions* component, which includes the list of available protection functions (e.g., random-noise function, generalization function) that the *protection strategies* and *privacy protection service* components can rely on during their computing processes. Finally, the *data protection* component, responsible for: (1) selecting the most appropriate protection functions, in terms of compatibility and computational cost, to be executed on sensor data streams to achieve required protection levels (i.e., the protection levels stated in the strategy chosen by the user); and (2) executing selected functions on data pieces in order to communicate protected data to consumers. This component provides therefore contextual data protection according to user decisions (i.e., in active mode).

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

Empowering the user to make quick, effective, and meaningful adaptation of her privacy decisions to cope with the evolution of her context remains challenging. In that regard, we propose in the following a new user-centric, 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 she accepts to take in her context. The aim of this approach is to assist the user in optimizing her privacy decisions, so that to meet her requirements and preferences while maximizing data utility and minimizing the cost of protection. To do so, δ -Risk provides the user with at least one best protection strategy to adopt in her context. In addition to her privacy preferences, the approach considers also the interests of the user (e.g., what services are important to her), thereby making the strategies provided not only optimal but also meaningful.

Figure 5 illustrates an overview of the solution. δ -Risk receives as input the preferences of the user and the context features (step 1), and outputs the best strategies that might be adopted in these

Fig. 5. Overview of the δ -Risk Approach.

circumstances (step 2). The user selects accordingly one protection strategy to be implemented on her shared attributes (step 3), 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 context should not bypass the threshold δ specified by the user. We detail in what follows the input parameters of the approach.

Context Features:

- (1) The set of attributes shared by u in c , i.e., $c.SA = \{a_1, a_2, \dots, a_m\}$ (cf. Definition 4.1).
- (2) 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 expresses the **global risk level** in c , that is used to interact with u (i.e., δ). How to measure the global risk level is addressed in the following subsection.
- (3) The **set of costs of protection functions** (cf. Definition 9) selected by the system, cPF , to be executed on attributes of $c.SA$.
- (4) The **impact matrix** of shared attributes on the risks inferred, W_c (cf. Definition 10).

Definition 9 (Protection Function). A **protection function**, $f \in PF$, is a protection method that can be executed on the data values of an attribute $a \in c.SA$ before being released to consumers. f is a local function stored in the system, such that:

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

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

$$\text{type} \in \{\text{noiseAddition}; \text{anonymization}; \text{accessControl}; \text{encryption}\}$$
- **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 ones:
 - $SA' \sqsubseteq c.SA$, denotes the set of attributes on which f will be executed
 - p , expresses the desired protection level to reach for the data values of all $a \in SA'$. ■

Definition 10 (Impact Matrix). Let W_c be the **impact matrix** of attributes $\{a_1, a_2, \dots, a_m\}$ of $c.SA$ on risks $\{r_1, r_2, \dots, r_n\}$ of $R_c.\vec{r}$. W_c is automatically calculated by the *privacy risk reasoner*

component during the risk inference process, such that

$$W_c = \begin{bmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1m} \\ \vdots & \vdots & \dots & \vdots \\ \omega_{n1} & \omega_{n2} & \dots & \omega_{nm} \end{bmatrix}, \text{ where } \omega_{ij} = \begin{cases} 0 & \text{if } r_i = P(\mathbf{pr}) \text{ and } a_j \notin A'_{pr} \\ 1 & \text{if } r_i = P(\mathbf{pr}) \text{ and } a_j \in A'_{pr} \end{cases}.$$

The impact ω_{ij} of an attribute a_j on a 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 \mathbf{pr} to which r_i is linked. ■

Example 6. Assume that the following $R_c.\vec{r}$ and W_c describe, respectively, the risk vector inferred for Alice in her current context, and the corresponding impact matrix:

- $R_c.\vec{r} = [r_1 \quad r_2 \quad r_3 \quad r_4]$ is the risk vector inferred for Alice in c .
 - $W_c = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}$ where a_1 impacts all risks, and a_2 impacts only r_2 and r_3 .
-

User Preferences:

(1) **Privacy preferences:**

- (i) The **risk threshold** δ , with 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. The system recommends possible values for δ to u according to her profile and context (see in Figure 7).
- (ii) The **set of enforced protection levels** for specific attributes, eP . These levels are extracted from pre-signed agreements with service providers. Only advanced users have the possibility to manually enforce protection levels to their attributes as shown in Figure 6.

(2) **Service preferences:** u has the possibility to state which services are important to her based on her profile (cf. Figure 6). Accordingly, the system calculates the weights to assign to attributes as follows:

- Let $\vec{wA} = [w_1 \quad \dots \quad w_m]$ be the vector of weights assigned to attributes $\{a_1, \dots, a_m\}$ of $c.SA$.
- Let S be the set of available services s_1, s_2, \dots, s_n offered by the providers to u in exchange of her data, such that: $\forall s \in S, s : \langle SA; li \rangle$, where:
 - SA'' represents the set of sensed attributes associated to s , such that $SA'' \sqsubseteq c.SA$
 - li denotes the level of importance of s to u . li is Boolean where 1 states that s is important
 Therefore, the weight of an attribute a_i , 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:

$$\forall a_i \in c.SA : w_i = \sum_{l=1}^n s_l.li \mid a_i \in s_l.SA''. \quad (1)$$

Users might have different levels of expertise to manage their privacy and express correctly their requirements and preferences (cf. Challenge 1). As our objective is to keep the privacy management user-centric, the provided solution must be user-friendly, i.e., it assists u and facilitates the interaction with her according to her expertise. On this basis, we define three user profiles:

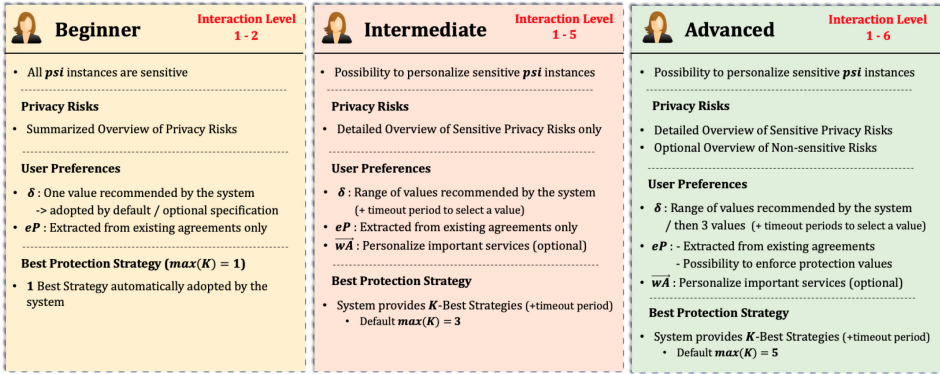


Fig. 6. User profiles.

- **Beginner:** u is not familiar with her privacy. She does not know how to interpret her risks and manage her privacy (e.g., specify her preferences, select protection strategies).
- **Intermediate:** u knows how to interpret her risks and use the system, i.e., she knows how to specify her preferences and understands the privacy measures to take. However, she does not know how to assess the implications of these measures.
- **Advanced:** u is expert in interpreting/analyzing her risks, and managing her privacy. She knows how to assess the direct and indirect implications of her privacy decisions.

Figure 6 details the defined profiles and their characteristics. The aim here is to limit the level of interaction with u according to her profile. This level is expressed in Figure 6 by min-max number of interactions for each profile. For beginner, the system: provides a summarized overview of the main risks involved in her context; recommends one value for δ based on her situation, which is adopted by default, while giving u the possibility to manually specify δ ; considers only enforced protection levels (i.e., eP) from pre-signed agreements if exist; and adopts automatically one of the best strategies without requiring user intervention. Hence, the level of interaction with a beginner is limited to specifying the list of sensed data, with the option of specifying δ . For intermediate, the system provides all beginner options plus: the possibility to personalize sensitive *psi* instances, and the importance of the services received (i.e., choose which *psi* are sensitive and which services are important to her); a detailed overview of sensitive risks only (i.e., according to the personalized list of *psi*). Instead of recommending one value for δ , the system suggests a range of values, and assigns a timeout period for the user to specify the desired value. As well, the system provides K -best strategies to select one of them, and assigns a timeout period for this task. If the user fails to respond for both cases before the timeout expires, then the system selects by default the lower boundary of the recommended range for δ , and selects one of the best strategies provided. For advanced, the system provides all intermediate options plus: an optional overview of non-sensitive risks; the possibility to manually enforce protection levels to certain shared attributes (considered in eP); longer time periods to choose δ and to select a strategy; and an additional recommendation of three values for δ if the first timeout expires (min/max/median values of the first recommended range). Therefore, the system is able to perform autonomously without requiring any mandatory interaction with u except specifying her sensed attributes.

Specifying δ might be challenging for the user as it may depend on her level of expertise. Accordingly, the system assists the user in this task based on her profile. Figure 7 details the process followed to recommend δ values. The choice of δ depends on the number of risks inferred in the

Beginner		Intermediate		Advanced	
Number of Risks	Recommended δ value	Number of Risks	Recommended range for δ	Number of Risks	Recommended range for δ
$\ \vec{r}\ = 0$	$\delta = 1$	$\ \vec{r}\ = 0$	$\delta = 1$	$\ \vec{r}\ = 0$	$\delta = 1$
$0 < \ \vec{r}\ \leq 5$	$\delta = 0.5$	$0 < \ \vec{r}\ \leq 5$	$\delta \in [0.4; 0.6]$	$0 < \ \vec{r}\ \leq 5$	$\delta \in [0.3; 0.7]$
$5 < \ \vec{r}\ \leq 10$	$\delta = 0.3$	$5 < \ \vec{r}\ \leq 10$	$\delta \in [0.3; 0.5]$	$5 < \ \vec{r}\ \leq 10$	$\delta \in [0.2; 0.6]$
$\ \vec{r}\ > 10$	$\delta = 0.1$	$\ \vec{r}\ > 10$	$\delta \in [0.2; 0.4]$	$\ \vec{r}\ > 10$	$\delta \in [0.1; 0.5]$

Fig. 7. Recommended δ values.

relevant user situation. If no risk is inferred, then the recommended δ is 1 (i.e., keep sharing fine-grained data). Otherwise, the system recommends one value for beginner, a range of values for intermediate with an amplitude of 0.2, and a larger range for advanced with an amplitude of 0.4 (cf. Figure 7). The recommended values/ranges can be updated by the system administrator based on user interactions.

Once δ is specified, calculating optimal and adaptive protection strategies becomes a challenging endeavor (cf. Challenge 2). To address this challenge, 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*, and a *best strategy*.

Definition 11 (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 $\{a_1, a_2, \dots, a_m\} \in c.SA$. Appropriate means a combination that meets the privacy preferences of u (i.e., δ and eP) while maximizing the utility of attribute values. A protection strategy can be represented as follows:

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

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* $f \in PF$ on a_j . Knowing that the way to achieve this level depends on the selected *protection function*. ■

Definition 12 (Best Protection Strategy). A **best protection strategy**, $\vec{bp} \in BP_c$, is an appropriate strategy $\vec{p} \in P_c$, that most satisfies the service preferences of u (i.e., \vec{wA}), and has the lowest cost of protection (i.e., based on the corresponding combination of protection functions). These constraints are expressed by the *ranking score* assigned to \vec{p} , which is computed as follows:

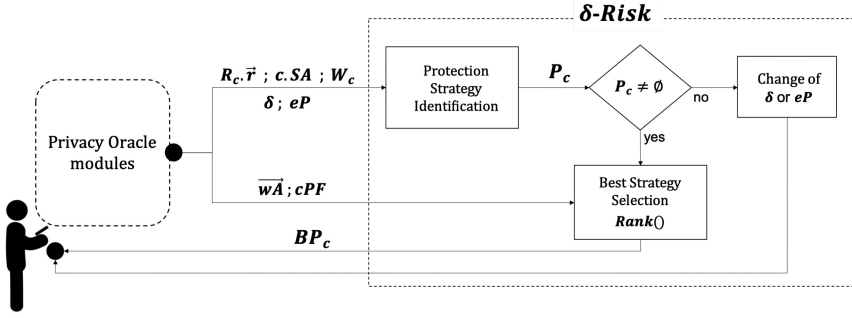
$$score(\vec{p}) = Rank(\vec{p}, \vec{wA}, cPF) \rightarrow \mathbb{N}, \text{ where:}$$

- **Rank()** expresses the ranking function. It takes as input a *protection strategy* $\vec{p} \in P_c$, the vector of weights (\vec{wA}), and the set of costs of selected protection functions (cPF). It outputs the *ranking score* of \vec{p} that is calculated according to the distance between \vec{p} and \vec{wA} , and the costs of the combined *protection functions*. Algorithm 2 details the **Rank()** function.

Therefore, \vec{p} is said to be one of the *best protection strategies*, $\vec{bp} \in BP_c$, only if it has the highest ranking score. This can be formalized as follows:

$$\forall \vec{p}_i \in P_c : \vec{p}_i \text{ only if } \vec{bp} \text{ only if } \forall \vec{p}_j \in P_c, score(\vec{p}_j) \leq score(\vec{p}_i). \quad \blacksquare$$

Figure 8 details the δ -Risk process. The first δ -Risk operation consists of identifying all possible appropriate protection strategies (i.e., P_c). If no strategies result from this operation, then the

Fig. 8. δ -Risk process.

combination of the privacy preferences (i.e., δ and eP) is inconsistent (cf. Definition 13). In this case, the system asks u to change one of these preferences and assigns a timeout period for this query: (1) if u 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 (i.e., full protection of the user's privacy); (2) elsewhere, the system receives the changes and the strategy identification process is re-launched. If the first operation generates several *protection strategies*, then the second δ -Risk operation focuses on ranking the resulting strategies to select the K -best strategies to be proposed to u . The ranking function, $Rank()$, considers the service preferences of u (i.e., \vec{wA}) and the costs of selected *protection functions* (i.e., cPF).

The δ -Risk process is by default executed once per context. However, when being in one of these contexts, u can change her service preferences or the system can select new protection functions, which requires recalculating new best strategies. To handle this, the system locally stores the *protection strategies* identified by the first operation (i.e., P_c) as long as the newly emerged contexts are similar. Therefore, if \vec{wA} or cPF has been changed within these contexts, only the second operation (i.e., $Rank()$) is re-executed to select new best strategies that meet these changes. Otherwise, the entire δ -Risk process is re-launched.

Example 7. 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 the dependent attributes a_1 and a_2 have the same weight, and the cost of the protection functions to execute on a_1 and a_2 are, respectively, 2 and 1. Hence, when executing the $Rank()$ 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 privacy risks to study the impact of these levels on risk values. Then, to quantify the global risk level (i.e., $R_c.v$) to ensure that the resulting combinations satisfy the δ -Risk principle. Therefore, we begin by formally quantifying a *privacy risk* and the *global risk level*, and then we detail the two δ -Risk operations.

4.1 Privacy Risk and Global Risk Level Quantification

A privacy risk is associated to one or more sensed attributes. This means that protecting impacting attributes will lead to minimize the risk value. Consequently, the risk vector \vec{r} depends on the protection levels assigned to shared attributes, \vec{p} , and the impact matrix of attributes on risks (i.e.,

W_c). This can be represented as follows:

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

- \mathcal{F} is the risk quantification function that takes as parameters an impact matrix and a protection vector, and returns a risk vector composed 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 sensed attribute $a_j \in c.SA$. 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 , then the risk value is 1 (i.e., highest level).
- (3) If the full protection is assigned to attributes impacting r_i , then r_i is eliminated.
- (4) The higher the protection level p_j impacting r_i , the lower the value of r_i .

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

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

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 Examples 6 and 7, the best strategy delivered to Alice in her context is $\vec{bp} = [0.6 \ 0]$. Once selected by Alice, the risk values will therefore be 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 the *privacy risks*, we now focus on how to measure the *global risk level* in the user context (i.e., $R_c.v$). This level is used to interact with u (i.e., δ). Once u assigns a value for δ , this means she 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 therefore quantified as follows:

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

4.2 Protection Strategy Identification

This section discusses the first δ -Risk operation, which consists of identifying appropriate *protection strategies*. To achieve this, we rely on the proposed risk quantification model and the δ -Risk principle, such that

$$R_c.v \leq \delta, \\ \Rightarrow \max \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} \leq \delta \quad \Rightarrow \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} \leq \delta.$$

Nonetheless, maximizing the utility of attributes' data requires assigning the lowest appropriate protection levels to these data. These levels are obtained when minimizing risks to the highest acceptable values, i.e., when $R_c.\vec{r} = \delta$. Therefore, the best-case scenario for data utility/privacy protection is to identify 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:

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

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, resulted when the δ/eP combination is inconsistent, which does not generate any solution; (2) system independent, resulted when attributes are independent, which generates exactly one solution; and (3) system dependent, resulted when attributes are dependent, which generates an infinite number of solutions.

In fact, the inconsistency problem presented in case (1) is resulted when the system contains at least one equation that includes only *enforced protection levels*. This leads to limiting the options for δ to one possible value, and will therefore entail an inconsistency if the user-specified value does not match the acceptable one. Definition 13 discusses this constraint.

Definition 13 (δ/eP Inconsistency). Let p_1, p_2 be the protection levels to be assigned to attributes $\{a_1, a_2\} \subseteq c.SA$. Assume that risk r_i of \vec{r} is impacted only by $\{a_1, a_2\}$. The linear system will therefore include the following equation: $\widetilde{\omega}_{11} \cdot p_1 + \widetilde{\omega}_{12} \cdot p_2 = 1 - \delta$, and the δ/eP combination is said to be inconsistent only if

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

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

ALGORITHM 1: Protection Strategy Identification

```

Input:  $Wc[][]$ ,  $\delta$ ,  $eP[]$ ; // impact matrix, risk threshold, and the enforced protection levels;
Output:  $Pc[][]$ ; // protection strategies;
1 Variables:  $System[][]$ ,  $M[][]$ , inconsistency, dependency;
2 begin
3   if ( $\delta = 0$ ) then
4     // user requests the full protection, i.e., stop sharing data;
5      $Pc \leftarrow createFullProtStrategy(1)$ ;
6   else if ( $\delta = 1$ ) then
7     // user accepts to share fine-grained data;
8      $Pc \leftarrow createDefaultStrategy(0, eP[])$ ; // strategy created with default values of protection levels;
9   else
10     $System \leftarrow buildSystem(Wc[], \delta, eP[])$ ; // build the linear system;
11     $M \leftarrow solveSystemGJE(System)$ ; // solves the system using the GJE method;
12    inconsistency  $\leftarrow checkInconsistency(M)$ ; // returns true if  $\delta/eP$  combination is inconsistent;
13    if (inconsistency = false) then
14      dependency  $\leftarrow checkDependency(M[])$ ; // returns true if system is dependent;
15      if (dependency = false) then
16        // attributes are independent (unique solution);
17         $Pc \leftarrow createIndependentStrategy(M[], eP[])$ ;
18      else
19        // attributes are dependent (infinite number of solutions);
20         $Pc \leftarrow createDependentStrategies(M[], eP[])$ ;
21    else
22       $notifyUserOfInconsistency()$ ; // user has to change either  $\delta$  or the relevant  $p \in eP$ ;
23 return  $Pc[][]$ 

```

Algorithm 1 presents the protection strategy identification algorithm 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 (line 3), then 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 $\vec{p} = [1 \ 1 \ \dots \ 1]$ (line 5). If δ is 1 (line 6), then 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 (line 8). If δ is between 0 and 1, then 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 solves it using the GJE method by calling the *solveSystemGJE* function (lines 10,11). This function returns a reduced row echelon form stored in $M[][]$. To check for inconsistency (i.e., δ/eP constraint), the process calls the *checkInconsistency* function, which returns a Boolean value stored in *inconsistency* (line 12).

If *inconsistency* is *false* (i.e., system is consistent), then the process checks attribute dependency in $M[][]$ by calling the *checkDependency* function, which returns a Boolean value stored in *dependency* (line 14). If *dependency* is *false* (i.e., attributes are independent), then the system has a unique solution that leads to create one protection strategy. This procedure is done by the *createIndependentStrategy* function (line 17), and the process is ended. If *dependency* is *true*, then attributes are dependent, and the system has an infinite number of possible solutions. The process calls accordingly the *createDependentStrategies* function (line 20), which starts by identifying existing dependencies among the unknown p_j items. Then, it performs two operations on each dependent p_j item. The first operation prioritizes the attribute of the selected p_j , by assigning a 0 value to p_j , which means that no protection is applied on a_j . The second operation assigns a value of 1 to p_j (i.e., stop sharing a_j), which gives priority to the associated dependent attributes. Next, both operations calculate the remaining p items that are dependent from p_j . This function consequently identifies several appropriate strategies, where each emphasizes at least one dependent attribute, and the process is ended.

If *inconsistency* is *true* (i.e., δ/eP combination is inconsistent), then the system notifies the user and asks her to either assign the acceptable value of δ , or to release one of the impacting $p \in eP$. The process is consequently re-launched either with updated δ/eP , or with $\delta = 0$ (if the assigned time period expires without user response).

It is important to note that this article only describes 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 (i.e., $|P_c| > 1$), ranking these strategies and selecting the K -best ones to be proposed to the user becomes a need. K expresses the number of best strategies, i.e., the ones with the highest ranking score (cf. Definition 12). Nonetheless, fixing the maximal value of K remains challenging, especially as many factors may contribute to the perceived choice overload, including number of options, time constraints, and user expertise [21]. In this study, user decisions need sometimes to be fast (i.e., in real-time), and the user's expertise is expressed by the selected profile (cf. Figure 6). Therefore, we assign the following default values to $\mathbf{max}(K)$ in accordance with the defined profiles: 1 for beginner, 3 for intermediate, and 5 for advanced. The value of $\mathbf{max}(K)$ can be manually changed by \mathbf{u} , and the default values could be updated by the system administrator based on user interactions.

The best strategies must best meet the preferences and interests of \mathbf{u} . 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 *Rank()* function, 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 detail the reasoning algorithm of the *Rank()* function.

Algorithm 2 outlines the ranking function, *Rank()*, takes as input the set of identified strategies ($Pc[][]$), the vector of weights assigned to attributes ($wA[]$), and the set of costs of selected

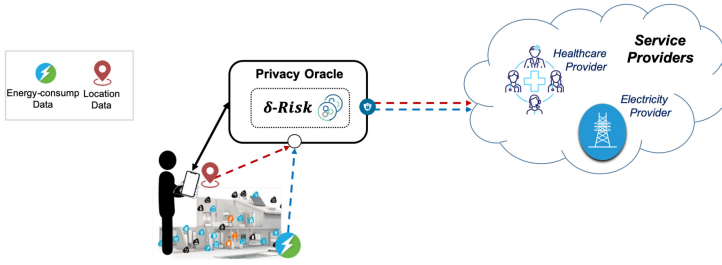
protection functions ($cPF[]$). It outputs the set of K -best protection strategies, $BPc[][]$. The function starts first by identifying the strategies with the shortest distance to $wA[]$ (lines 3–14). To do so, the first step is to identify the number of different weight values and sort them in a descending 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 (line 3). Then, for each distinct weight value, we check the number of attributes having this weight through the *attributesWithSimilarWeight* function (line 5). 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 (lines 9 and 10). Thereafter, we filter the resulting set of strategies to consider only strategies with the highest score (lines 11–14). 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 cost of protection of the resulting strategies (lines 15–18). The cost of a strategy is equal to the sum of costs of the protection functions that 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 (line 19). Only strategies with the highest ranking score are selected and added to the set $BPc[][]$ (line 20), which will therefore constitute the best strategies to be proposed to the user in her context.

ALGORITHM 2: Best Strategy Selection - *Rank()* function

```

Input:  $Pc[][]$ ,  $wA[]$ ,  $cPF[]$ ; // protection strategies, vector of weights, and the costs of protection functions;
Output:  $BPc[][]$ ; // best protection strategies;
1 Variables:  $sortedWA[]$ ,  $A[]$ ,  $minP$ ,  $Score[][]$ ,  $maxScore$ ,  $CostPc[][]$ ;
2 begin
3    $sortedWA \leftarrow sortAndFilter(wA[])$ ; // sorts  $wA$  in a descending sequence and removes redundant values;
4   for  $i \leftarrow 0$  to  $|sortedWA|$  do
5      $A \leftarrow attributesWithSimilarWeight(wA[], sortedWA[i])$ ;
6     // the set  $A$  will include the indexes of attributes with same weight  $sortedWA[i]$ ;
7     for  $j \leftarrow 0$  to  $|A|$  do
8       // for each attribute having the weight  $sortedWA[i]$ ;
9        $minP \leftarrow getMinP(Pc[], A[j])$ ; // the minimal protection level to be assigned to attribute  $a_j$ ;
10       $Score \leftarrow addScore(Pc[], minP, A[j], wA[])$ ; // updates the score of strategies having  $minP$ 
11       $maxScore \leftarrow getMaxScore(Score[][])$ ; // returns the maximal score assigned to strategies
12      for  $k \leftarrow 0$  to  $|Score|$  do
13        if  $(Score[k][1] \neq maxScore)$  then
14           $Pc \leftarrow deleteStrategy(k)$ ; // keeps only strategies with the highest score
15      for  $i \leftarrow 0$  to  $|Pc|$  do
16        for  $j \leftarrow 0$  to  $|Pc[0]|$  do
17          if  $(Pc[i][j] \neq 0)$  then
18             $CostPc[i][1] = CostPc[i][1] + cPF[j]$ ; // calculate the cost of protection of each strategy
19       $Score \leftarrow addCostToScore(Score[], CostPc[])$ ; // adds the calculated cost to the score of strategies ;
20       $maxScore \leftarrow getMaxScore(Score[][])$ ;
21       $BPc \leftarrow selectBestStrategies(Pc[], Score[], maxScore)$ ; //  $BPc$  includes then the best strategies
22 return  $BPc[][]$ 

```

Fig. 9. δ -Risk implementation.

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

Fig. 10. Risks inferred for Alice.

\overline{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. 11. Best strategy proposed to Alice.

5 EXPERIMENTAL VALIDATION AND EVALUATION

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

5.1 Approach Validation: Java-based Prototype

To implement and validate the δ -Risk solution, we developed a Java-based prototype, and we embed it on the user device as middleware between the user and the connected providers (cf. Figure 9). 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/>.

The goal here is to illustrate the functioning of the solution. We consider the scenario provided in Section 2 as the actual situation of Alice (cf. Figure 9). We execute the privacy risk inference prototype¹ proposed in our previous work [2] to infer the risks involved in this situation. Figure 10 describes the overview of risks provided to Alice, and Figure 11 illustrates the impact of *energy-consump* and *Location* attributes on these risks. Once alerted, assume that Alice has specified a value of 0.6 for δ . The δ -Risk process is consequently executed, and generates the following best strategy that suggests applying 40% protection on *Energy-consump* and 40% protection on *Location* (cf. Figure 11).

5.2 Experimental Protocol

The objective of our experimental protocol is to prove that δ -Risk: (1) is scalable; (2) maintains low-complexity in space (i.e., in memory overhead and storage) and time (cf. Challenge 3);

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

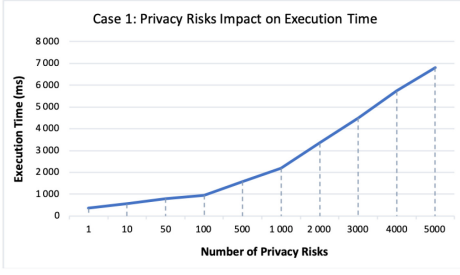


Fig. 12. Execution time.

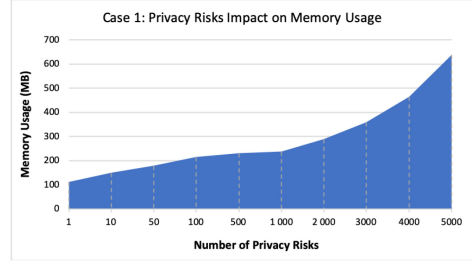


Fig. 13. Memory usage.

(3) handles reasoning in real-time; (4) always identifies all possible appropriate strategies that answer the data utility/privacy protection trade-off; (5) always delivers the best strategies to the user; and (6) provides the user with at least one best strategy per context. To achieve this, we evaluate the performance of the proposed solution, and we formally study its effectiveness.

5.2.1 Performance Evaluation. To evaluate the performance of the δ -Risk solution, we consider four use cases to study the impact of the following five metrics on the system’s performance: (i) number of risks inferred ($R_c \cdot \vec{r}$); (ii) number of attributes sensed ($c \cdot SA$); (iii) the dependency level of attributes (W_c); and (iv) the variation of the user’s service preferences ($\vec{w}A$) and the costs of protection functions (cPF). To test the performance in more real scenarios, we consider having both dependent and independent attributes (with a maximum dependency level of 4) in cases i, ii and iv. 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 sensed attributes at 4 (we consider that the four attributes are dependent in the impact matrix W_c), the δ value at 0.6, the vector of weights $\vec{w}A = [1 \ 2 \ 1 \ 2]$, and the costs of four selected protection functions $cPF = \{1, 3, 1, 1\}$. We execute the process ten times, taking into account the following number of risks for each iteration: 1; 10; 50; 100; 500; 1,000; 2,000; 3,000; 4,000; and 5,000. Figure 12 shows the impact of the risk number 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 execution time of less than 2 s up to 1,000 risks, and less than 7 s up to 5,000 risks. When considering RAM usage (Figure 13), it follows a linear evolution up to 5,000 risks, with an average of less than 200 MB up to 1,000 risks. It is important to note that, in real scenarios, the number of risks inferred in a given context will not practically exceed 1,000 for the user. This highlights the importance of using the GJE method to solve the linear system.

Case 2: We vary the number of attributes shared by the user. We fix the number of risks at 100, the δ value at 0.6, the vector of weights $\vec{w}A = [1 \ 2 \ 1 \ 2 \ 0 \ \dots \ 0]$, and the costs of the four selected protection functions $cPF = \{1, 3, 1, 1\}$. We execute the reasoning process twelve times, taking into account the following number of sensed attributes for each iteration: 1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100. We consider four dependent attributes in W_c when the number of shared attributes is 5, and three different dependency levels for the other iterations (2, 3, and 4). According to Figure 14, the evolution of the execution time is quasi-linear up to 100 attributes with an average of less than 4s. The evolution is also similar for the memory usage (cf. Figure 15)

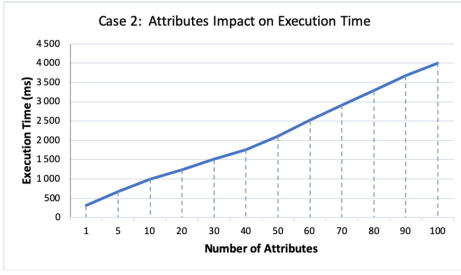


Fig. 14. Execution time.

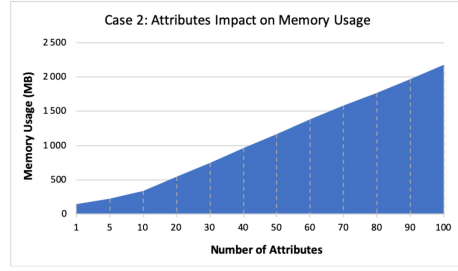


Fig. 15. Memory usage.

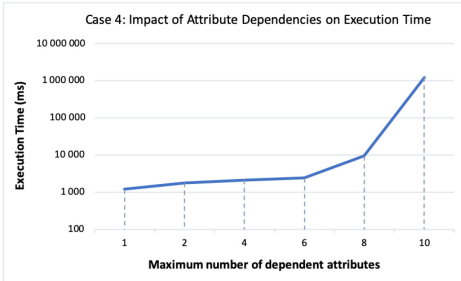


Fig. 16. Execution time.

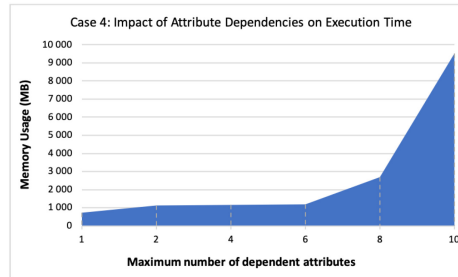


Fig. 17. Memory usage.

with an average of less than 2,000 MB. It is important to note that, in real scenarios, the number of attributes shared by the user in her context will not practically exceed 50.

Case 3: We vary the dependency level of attributes shared by the user (i.e., W_c). We fix the number of risks at 100, the number of attributes at 50, the vector $\vec{wA} = [1 \ 2 \ 1 \ 2 \ 0 \ \dots \ 0]$, the δ value at 0.6, and the costs of the four selected protection functions $cPF = \{1, 3, 1, 1\}$. We execute the reasoning process six times, taking into account the following dependency level for each iteration: 1 (i.e., attributes are independent), 2 (i.e., two attributes of the 50 are dependent), 4, 6, 8, and 10. According to Figure 16, the execution time remains quasi-constant with an average value of 1s until the dependency level exceeds 6, where the execution time tends to be exponential (e.g., reaches a value of 1,227 s for a dependency level of 10). Same evolution for RAM usage as shown in Figure 17. However, it is important to note that combining more than six *shared attributes* to reveal specific *privacy-sensitive information* about the user that cannot be revealed otherwise is quasi-impossible.

Case 4: We focus on varying the vector of weights \vec{wA} and the selection of protection functions (i.e., varying the corresponding set of costs cPF). The aim here is to highlight the importance of storing the appropriate strategies identified by the first δ -Risk operation (i.e., P_c) while being in consecutive similar contexts. We illustrate the variation of both metrics in the same use case as they both produced the same performance results. Hence, we fix the number of risks at 100, the number of attributes at 50, and the δ value at 0.6. We consider three different dependency levels for the attributes (2, 3, and 4). We execute only the second δ -Risk operation (i.e., $Rank()$ function) while considering several changes in the weights and costs. As shown in Figure 18, the execution time remains quasi-constant with an average execution time of less than 500 ms. Similar for the RAM usage (cf. Figure 19). Therefore, if within consecutive similar contexts the user changes her service preferences or the system varies the selection of protection functions, then the solution will be able to select new best strategies in less than 500 ms.

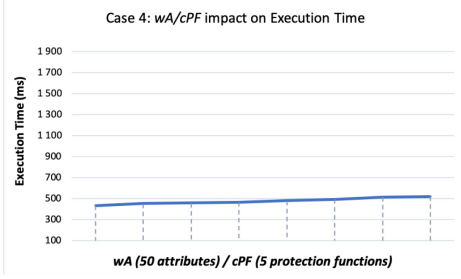


Fig. 18. Execution time.

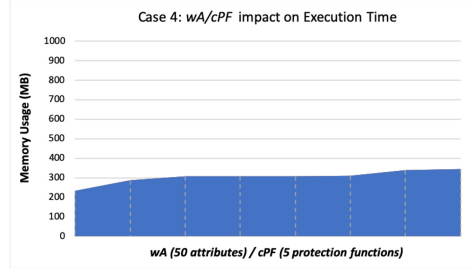


Fig. 19. Memory usage.

Discussion. The experiments conducted show that δ -Risk is scalable, i.e., it handles an increasing number of risks and attributes with good performance. In fact, if we consider the worst case scenario of 1,000 privacy risks, 50 shared attributes, and simultaneous dependency levels of 2, 4, and 6, then the solution is able to respond and provide strategies within an average time of 3s and an average RAM space of 1,200 MB. If we consider a more quasi-real case scenario of 20 risks, 5 attributes, and a dependency level of 3, then the solution responds within an average time of 550 ms and an average RAM space of 180 MB. Accordingly, δ -Risk is capable of handling real-time reasoning. It also maintains low computational complexity in execution time and memory overhead, which makes it operational even on devices with limited computational resources (cf. Challenge 3).

5.2.2 Approach Effectiveness. We present in this section a formal study to prove the effectiveness of the proposed approach.

THEOREM 1. *The δ -Risk solution maintains low storage complexity.*

PROOF. Let n denotes the maximum number of *attributes* that could be collected by the CaPMan system about u and her environment in a context. As previously mentioned in Section 3, the system stores only consecutive contexts by default, resulting in a linear storage complexity of $O(2n)$. However, the number of *attributes* stored for a single context (i.e., $c.A$) will not practically exceed 100, which makes the storage complexity of the solution low. \square

THEOREM 2. *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 u shares only one attribute, such that $c.SA = \{a_1\}$. According to Assumption 1 stated in Section 4.1, all risks are inevitably associated to the attribute a_1 , i.e., W_c is composed of a single vector with values equal to 1. Consequently, the resulting linear system consists of a single equation $p_1 = 1 - \delta$ (cf. Equation (6)), generating therefore one protection strategy $\vec{p} = [p_1] = [1 - \delta]$, which will constitute the best strategy to be delivered, $b\vec{p} = \vec{p} = [1 - \delta]$.

GENERIC CASE. Assume that u shares m attributes in her context c , i.e., $c.SA = \{a_1, \dots, a_m\}$, and the number of risks inferred is n ($R_c \cdot \vec{r} = [r_1 \dots r_n]$). W_c will therefore be a $n \times m$ matrix of $\{0,1\}$ values expressing the impact of attributes $a_j \in c.SA$ on risks r_i of $R_c \cdot \vec{r}$. According to Equation (6), this leads to build a linear system of n equations with m unknowns (i.e., $[p_1 \dots p_m]$):

- If $\delta = 0$, then u does not accept to take any risk, i.e., all risks inferred in c must be eliminated, such that $R_c \cdot \vec{r} = [r_1 \ \dots \ r_n] = [0 \ \dots \ 0]$. Hence, the protection levels $[p_1 \ \dots \ p_m]$ to assign to attributes must be at their highest level, i.e., leading, according to Equation (4), to the full protection strategy $\vec{b}p = \vec{p} = [1 \ \dots \ 1]$.
- If $\delta = 1$, then no protection is to be applied on shared attributes and that u wants to share fine-grained data to preserve the full accuracy of the services she receives in return. Consequently, no additional protection is needed, and the protection levels must be left 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 $\delta \in]0; 1[$, then u wants to use specific services but without taking any privacy risk above the threshold δ . Consequently, the solution identifies all possible appropriate strategies that satisfy the best-case scenario $R_c \cdot \vec{r} = \delta$ 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 three possible cases:

- (1) System is inconsistent, i.e., the δ/eP combination is inconsistent (cf. Definition 13). Therefore, u is requested either to choose the acceptable value of δ or to release one of the impacting $p \in eP$. The process is consequently re-launched with updated δ/eP or with $\delta = 0$ (if the assigned time period expires without user response).
- (2) Attributes are independent, and the system has a unique solution:

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

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

- (3) Attributes are dependent, and the system has an infinite number of solutions:

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

Nonetheless, as our goal is to address the data utility/privacy protection trade-off, we only focus on assigning the minimum acceptable protection to dependent attributes (i.e., a_j/a_k). Hence, the process performs a double iteration on each dependent p_j/p_k item. The first iteration gives priority to the selected p_j/p_k item by giving it a value of 0, which means that no protection will be applied on attribute a_j/a_k . The second iteration assigns a value of 1 to p_j/p_k (i.e., highest protection level), which gives priority to the other dependent p

items. Then, both iterations calculate the remaining dependent p items based on the matrix of dependencies M . When completed, the process identifies several appropriate strategies $\vec{p} \in P_c$ 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. \square

THEOREM 3. *The δ -Risk process always delivers the best strategies to the user.*

PROOF. After identifying all possible appropriate strategies, the process executes the ranking function, $\mathbf{Rank}()$, to select only the best strategies to deliver to u . This function ranks the strategies according to the service preferences of u (i.e., \vec{wA}) and the 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 $|P_c| = 1$, the identified strategy is automatically selected as the best one. If $|P_c| \geq 1$, then the process ranks the strategies and always selects the K -best ones to deliver to u . \square

THEOREM 4. *δ -Risk provides the user with at least one best strategy per context.*

PROOF. It is easy to see that in all circumstances, the process is able to provide at least one best strategy per context to u . \square

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 effective and meaningful involvement of the user in the management of her 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 is an essential component of the approach. This principle is not selected as a criterion as it is by default satisfied in all privacy approaches.
- (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 manages this trade-off in a positive-sum.
 - (b) *Hybrid Protection.* The approach handles combination of several protection functions.
- (5) *End-to-End Security.* The approach guarantees a full life-cycle protection. We focus here on three sub-criteria, namely, *real-time protection*, *context-aware protection*, and *scalability*.

Table 1. Comparative Study of Existing Context-aware Privacy Preserving Works

PbD Principles	Proactive and Preventative	Privacy as Default Setting	Full Functionality		Full Life-cycle Protection			Visibility and Transparency	User-centric Privacy Management			
			Privacy vs. Utility	Hybrid Protection	Scalability	Context-aware Protection	Real-time Protection		User Awareness	User-Friendly	Privacy Preferences	User Interests
Nesse et al. [25]	Yes	Yes	Yes	No	Yes	Yes	No	Yes	No	No	No	No
Matos et al. [26]	Yes	Y/N	Y/N	No	Y/N	Yes	Yes	No	No	No	No	No
Gheisari et al. [27]	Yes	Yes	Yes	No	Yes	Yes	Y/N	Y/N	No	No	No	No
Sylla et al. [28]	Yes	Y/N	Y/N	No	Yes	Yes	Yes	Y/N	Yes	Y/N	Yes	No
Alagar et al. [29]	Yes	Yes	Yes	No	Y/N	Yes	Yes	Y/N	No	Y/N	Yes	No
CaPMan	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

¹ Y/N means that the referenced work did not approach this concept.

- (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.
- (7) *Respect for User Privacy: Keep it User-Centric.* The approach empowers user-friendly options. We divide this criterion into five sub-criteria to cover all user-centric privacy dimensions:
- User Awareness: Informed Decision-making.* The approach empowers the user to make informed decisions by sensitizing her to the risks taken through appropriate notifications.
 - User-friendly.* The approach is user-friendly, i.e., assists the user to correctly manage her privacy according to her level of expertise.
 - Privacy Preferences.* The approach considers the privacy preferences of the user.
 - User Interests.* The approach considers the interests of the user (e.g., important services).

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. [28] presented a global vision of their context-aware security and privacy as a service 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 her 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.

Discussion. Table 1 summarizes the evaluation of existing context-aware privacy preserving approaches based on the aforementioned criteria. The majority of these works, i.e., References [25, 26, 28, 29], have only presented an overview of their proposed frameworks and briefly explained how they work. Besides, none of the existing works covers all aforementioned criteria, and they are also tailored to specific application domains. Therefore, we introduce in this article CaP-Man, 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. CaPMan answers the defined criteria as follows: (1) it implements proactive reasoning process to infer the risks before being materialized, and assists the user in adapting her privacy decisions before releasing her data to consumers; (2) once shared attributes are specified, the CaPMan system can work autonomously without requiring any mandatory user intervention; (3) CaPMan is fully functional, i.e., it treats the privacy protection/data utility trade-off in a positive-sum, and supports the combination of multiple protection functions in the delivered strategies to minimize the cost of protection; (4) User data is protected before being released to data consumers. This makes the user's privacy protected for the entire data lifecycle. The solution is scalable, and supports real-time and context-aware protection; (5) CaPMan performs continuous adaptation of privacy policies according to the adopted protection strategies; (6) CaPMan raises user awareness regarding her context-dependent risks, and provides her with fast, optimal, and adaptive strategies that consider her privacy preferences and interests.

6.2.2 Privacy Risk Inference and Quantification. Alerting users about their privacy risks constitutes a key step toward 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. Hatamian et al. [31] 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 to assess the privacy risk scores of installed applications. Zhang et al. [32] provided a formal privacy quantification model for **location-based services (LBS)** that uses the Bayes conditional risk as a privacy metric. This model focuses on conditional privacy regarding the adversary estimation error to compare the LBS privacy metrics. Banerjee et al. [33] studied the privacy risks that may arise from the deviations of data collectors' practices from what they promise in their policies, as opposed to the user's needs.

7 CONCLUSION

This article presents a user-centric multi-objective approach for context-aware privacy management in connected environments, denoted δ -Risk. This approach features a new privacy risk quantification model to dynamically calculate and select the best protection strategies for the user based on her preferences and contexts (e.g., involved risks). Computed strategies are optimal in that they seek to closely satisfy user requirements and preferences while maximizing data utility and minimizing the cost of protection. We implemented our approach and evaluated its performance and effectiveness based on several use cases. Results show that δ -Risk delivers scalability and low-complexity in time and space. Besides, it handles privacy reasoning in real-time, making it able to support the user in various contexts, including ephemeral ones. It also provides 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 CaPMan system reasons on each context apart without considering historical contexts/risks. Nonetheless, contexts can be connected, which may have impact on the levels of protection to be assigned to sensed attributes in the present context to avoid privacy breaches. 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 related research problems, including: how to select the most appropriate protection functions to be executed on attributes, and how to measure system vulnerabilities accordingly.

REFERENCES

- [1] Betsy George, James M. Kang, and Shashi Shekhar. 2009. Spatio-temporal sensor graphs (stsg): A data model for the discovery of spatio-temporal patterns. *Intell. Data Anal.* 13, 3 (2009), 457–475.
- [2] Karam Bou Chaaya, Mahmoud Barhamgi, Richard Chbeir, Philippe Arnould, and Djamel Benslimane. 2019. Context-aware system for dynamic privacy risk inference: Application to smart IoT environments. *Future Gen. Comput. Syst.* 101 (2019), 1096–1111.
- [3] Mikhail A. Lisovich, Deirdre K. Mulligan, and Stephen B. Wicker. 2010. Inferring personal information from demand-response systems. *IEEE Secur. Privacy* 8, 1 (2010), 11–20.
- [4] Nicholas Vollmer. 2018. Table of contents EU General Data Protection Regulation (EU-GDPR). <https://www.privacy-regulation.eu/en/>.
- [5] State of California Department of Justice. 2018. California Consumer Privacy Act (CCPA). <https://oag.ca.gov/privacy/ccpa>.
- [6] C. Castelluccia, M. Cunche, D. Le Metayer, and V. Morel. 2018. Enhancing transparency and consent in the IoT. In *Proceedings of the IEEE European Symposium on Security and Privacy Workshops (EuroSPW'18)*. 116–119. DOI: <http://dx.doi.org/10.1109/EuroSPW.2018.00023>
- [7] I. D. Addo, S. I. Ahamed, S. S. Yau, and A. Buduru. 2014. A reference architecture for improving security and privacy in Internet of Things applications. In *Proceedings of the IEEE International Conference on Mobile Services*. 108–115. DOI: <http://dx.doi.org/10.1109/MobServ.2014.24>
- [8] Santosh Kumar, Sanjay Kumar Singh, Amit Kumar Singh, Shrikant Tiwari, and Ravi Shankar Singh. 2018. Privacy preserving security using biometrics in cloud computing. *Multimedia Tools Appl.* 77, 9 (2018), 11017–11039.
- [9] David W. Chadwick and Kaniz Fatema. 2012. A privacy preserving authorisation system for the cloud. *J. Comput. Syst. Sci.* 78, 5 (2012), 1359–1373.
- [10] Akber Dato. 2018. Data in the post-GDPR world. *Computer Fraud and Security* 9 (2018), 17–18.
- [11] Tim Collins. 2018. Marketing firm exactis leaks 340 million files containing private data. *Mail Online* (2018). <https://www.dailymail.co.uk/sciencetech/article-5900071/Marketing-firm-Exactis-leaks-340-million-files-containing-private-data.html>.
- [12] Mahmoud Barhamgi, Charith Perera, Chirine Ghedira, and Djamel Benslimane. 2018. User-centric privacy engineering for the Internet of Things. *IEEE Cloud Comput.* 5, 5 (2018), 47–57.
- [13] Victoria Y. Pillitteri and Tanya L. Brewer. 2014. *Guidelines for Smart Grid Cybersecurity*. Technical Report NISTIR 7628 Revision 1. National Institute of Standards and Technology. DOI: <http://dx.doi.org/10.6028/NIST.IR.7628r1>
- [14] Alston S. Householder. 2013. *The Theory of Matrices in Numerical Analysis*. Courier Corporation.
- [15] D. Nagarajan, T. Tamizhi, M. Lathamaheswari, and J. Kavikumar. 2019. Traffic control management using Gauss Jordan method under neutrosophic environment. In *AIP Conference Proceedings*, Vol. 2112.
- [16] L. Shang, S. Petiton, and M. Hugues. 2009. A new parallel paradigm for block-based Gauss-Jordan algorithm. In *Proceedings of the 8th International Conference on Grid and Cooperative Computing*. 193–200.
- [17] L. M. Aouad and S. G. Petiton. 2006. Parallel basic matrix algebra on the Grid'5000 large scale distributed platform. In *Proceedings of the IEEE International Conference on Cluster Computing*. 1–8.
- [18] Ling Shang, Zhijian Wang, Serge G. Petiton, Yuansheng Lou, and Zhizhong Liu. 2008. Large scale computing on component based framework easily adaptive to cluster and grid environments. In *Proceedings of the 3rd ChinaGrid Annual Conference*. IEEE, 70–77.
- [19] Lamine M. Aouad, Serge G. Petiton, and Mitsuhsa Sato. 2005. Grid and cluster matrix computation with persistent storage and out-of-core programming. In *Proceedings of the IEEE International Conference on Cluster Computing*. IEEE, 1–9.

- [20] Mingqiang Xue, Panos Kalnis, and Hung Keng Pung. 2009. Location diversity: Enhanced privacy protection in location based services. In *Proceedings of the International Symposium on Location-and Context-Awareness*. Springer, 70–87.
- [21] Alexander Chernev, Ulf Böckenholt, and Joseph Goodman. 2015. Choice overload: A conceptual review and meta-analysis. *J. Consum. Psychol.* 25, 2 (2015), 333–358.
- [22] Ann Cavoukian and Michelle Chibba. 2018. Start with privacy by design in all big data applications. In *Guide to Big Data Applications*. Springer, 29–48.
- [23] Ann Cavoukian. 2012. Privacy by design [leading edge]. *IEEE Technol. Soc. Mag.* 31, 4 (2012), 18–19.
- [24] 2018. ISO/PC 317 Consumer Protection: Privacy by Design for Consumer Goods and Services. <https://www.iso.org/committee/6935430/x/catalogue/>.
- [25] Ricardo Neisse, Gary Steri, Gianmarco Baldini, Elias Tragos, I. Nai Fovino, and Maarten Botterman. 2014. Dynamic context-aware scalable and trust-based IoT security, privacy framework. *Internet of Things Applications: From Research and Innovation to Market Deployment, IERC Cluster Book*.
- [26] Everton de Matos, Ramão Tiago Tiburski, Leonardo Albernaz Amaral, and Fabiano Hessel. 2018. Providing context-aware security for IoT environments through context sharing feature. In *Proceedings of the 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'18)*. IEEE, 1711–1715.
- [27] Mehdi Gheisari, Guojun Wang, Wazir Zada Khan, and Christian Fernández-Campusano. 2019. A context-aware privacy-preserving method for IoT-based smart city using software defined networking. *Comput. Secur.* 87 (2019), 101470.
- [28] Tidiane Sylla, Mohamed Aymen Chalouf, Francine Krief, and Karim Samaké. 2019. Towards a context-aware security and privacy as a service in the Internet of Things. In *Proceedings of the International Conference on Information Security Theory and Practice (IFIP'19)*. 240–252.
- [29] Vangalur Alagar, Alaa Alsaig, Olga Ormandjiva, and Kaiyu Wan. 2018. Context-based security and privacy for healthcare IoT. In *Proceedings of the IEEE International Conference on Smart Internet of Things (SmartIoT)*. IEEE, 122–128.
- [30] Delphine Christin, Martin Michalak, and Matthias Hollick. 2013. Raising user awareness about privacy threats in participatory sensing applications through graphical warnings. In *Proceedings of the International Conference on Advances in Mobile Computing and Multimedia*. 445–454.
- [31] Majid Hatamian and Jetzabel Serna-Olvera. 2017. Beacon alarming: Informed decision-making supporter and privacy risk analyser in smartphone applications. In *Proceedings of the IEEE International Conference on Consumer Electronics*. IEEE, 468–471.
- [32] Xuejun Zhang, Xiaolin Gui, Feng Tian, Si Yu, and Jian An. 2014. Privacy quantification model based on the Bayes conditional risk in Location-based services. *Tsinghua Sci. Technol.* 19, 5 (2014), 452–462.
- [33] Mishtu Banerjee, Rosa Karimi Adl, Leanne Wu, and Ken Barker. 2011. Quantifying privacy violations. In *Proceedings of the Workshop on Secure Data Management*. Springer, 1–17.