Privacy in Pervasive Systems: Social and Legal Aspects and Technical Solutions

Sabrina De Capitani di Vimercati, Sara Foresti, Giovanni Livraga,
Stefano Paraboschi, Pierangela Samarati

Abstract We live today in a globally interconnected society characterized by growing availability of computational power and connectivity, enabling every citizen to carry out tasks, access services, and stay connected virtually anywhere anytime. Unfortunately, the downside of such convenience is an increased exposure of possibly sensitive information and new risks of privacy vulnerabilities. In this chapter, we survey the main issues related to privacy in emerging pervasive scenarios and discuss some approaches towards their solution.

1 Introduction

In today's society most actions we perform are recorded and the collected data are stored, processed, and possibly shared in a way that was impossible until few years ago, before the development of ubiquitous and pervasive technologies. Ubiquitous technologies represent one of the most significative revolution in the Information and Communication technologies. The term "ubiquitous computing" was introduced by Mark Weiser in the late 80's to describe a future world based on "the idea of spreading computers ubiquitously, but invisibly, throughout the environment" [48]. Pervasive and ubiquitous technologies are now present everywhere in our daily life. People may have several devices (e.g., smartphones, tablets) that can be used to access any kind of services everywhere everytime. There are also devices that can keep track and measure health conditions (e.g., blood pressure and heart rate) and send such information to different parties. The amount of data that is

Sabrina De Capitani di Vimercati, Sara Foresti, Giovanni Livraga, Pierangela Samarati
Università degli Studi di Milano, Via Bramante 65, 26013 Crema (Italy)
e-mail: {firstname.lastname}@unimi.it

Stefano Paraboschi
Università degli Studi di Bergamo, Via Marconi 5, 24044 Dalmine (Italy)
e-mail: parabosc@unibg.it

therefore generated everyday has grown exponentially and is expected to continue this growth in the coming years. Since the cost of data storage and processing has significantly decreased, all these data can be long-term stored and made accessible when needed.

While the technology advancements and the possibility of collecting, storing, processing, and accessing data everywhere in the world bring enormous benefits, users are becoming more and more concerned about their privacy. In fact, collected data can be used to identify individuals, or infer something that was not intended for disclosure. The location information generated by a cell phone, the pattern of walking as recorded by a surveillance camera, as well as the combination of seemingly innocuous information (e.g., the ZIP code and the date of birth) are all examples of data that can be exploited to identify the person to whom they refer, not to mention information like the biometrics that may raise some privacy concerns (e.g., [6, 20, 24, 42]). The main motivation behind these privacy issues is that when users, for example, subscribe to a new social networking service or provide some information to access a service, they immediately lose control over their data.

The users' abilities to manage their personal information and also to delete such information may then become difficult, if not impossible. This is a well recognized problem that research and development communities, governments, and public and private organizations are all trying to solve. In particular, at the European level, a proposal for a regulation was released in 2012 with the aim of unifying all the data protection laws within the European Union with a single General Data Protection Regulation (`http://ec.europa.eu/justice/data-protection/`). The main goal of this new regulation on data protection is to take into consideration the recent technological developments (e.g., cloud computing and social networks) and their security and privacy risks [21, 30] to build trust in the digital world and to empower users to keep the control over their data. There are several key aspects that are considered in the proposed regulation such as: the introduction of new concepts (e.g., encrypted data and genetic data), which address new privacy concerns; the right to be forgotten, which allows users to require the erasure of their personal data whenever, for example, such data are no more necessary for the purpose for which they have been collected; and the applicability of such a regulation also to companies based outside the EU that process the personal data of EU residents.

Clearly, privacy in the modern digital society is a complex concept that should be addressed from several points of view: legal, social, economical, and technological. The main focus of this chapter is on the technological aspect of privacy within today's ubiquitous and pervasive systems. In particular, we aim at analyzing the main privacy issues that can arise when collecting, processing, and sharing data in pervasive and ubiquitous environments, and then at presenting available technological solutions that can be put in place to counteract them. As a running example, we will consider a museum organization that manages a large cultural heritage. The museum aims at exploiting the pervasive availability of computing infrastructures to develop a framework for providing a cultural site (e.g., indoor museums, archaeological sites, historical archives, old town centers) with several *smart* services for assisting users (e.g., visitors or staff personnel) in the seamless exploration and management of the related environment. In this context, smart and pervasive solutions can be adopted, for example, from the electronic management of ticket purchases, to interactive and guided tours based on the sensed proximity of a visitor to a specific exhibition, to the continuous monitoring of the environmental conditions (e.g., humidity, temperature, pollutant concentrations) in the museum premises through environmental sensors.

The remainder of the chapter is organized as follows. Section 2 illustrates our reference scenario, and presents the main related privacy risks. Section 3 overviews the most well-known approaches to protect location information. Section 4 discusses some solutions that allow the privacy-preserving sharing of personal/sensitive data. Section 5 shows possible approaches that allow the secure storage of personal/sensitive data. Finally, Section 6 gives our final remarks and concludes the chapter.

2 Privacy in pervasive systems

We first introduce the reference scenario that will be considered in the remainder of this chapter (Section 2.1). We then illustrate the privacy issues that may arise in such a scenario (Section 2.2).

2.1 Reference scenario

Our scenario (Figure 1) refers to a museum with both indoor and outdoor exhibitions and facilities, distributed in a wide geographical area, such as a city or a region (e.g., Rome, Paris, New York). To be considered “smart”, the museum features a digital infrastructure providing digital services and pervasive solutions to both enhance the experience of the visitors, and efficiently manage the museum and its exhibitions. As illustrated in Figure 1, the considered scenario is characterized by the interaction of different subjects. In particular, users (i.e., visitors of the museum) interact with the museum system through an app that they can install on their smartphone, and by enabling GPS and location services. The app acts as a smart guide, as described in the following of this section. Also, the museum features a set of environmental monitoring stations to measure different environmental parameters (e.g., temperature, humidity, pollutants). Data about users and the environmental measures collected by the museum are stored at different servers (i.e., the registrations and payments server, the LBS provider, and the environmental measures server). The smart solutions adopted by the museum can be classified in the following three groups, depending on their objective.

- *Ticket purchases and visitor registrations.* Visitors of the museum can buy their tickets either in place or online. When buying online, users pay by credit card

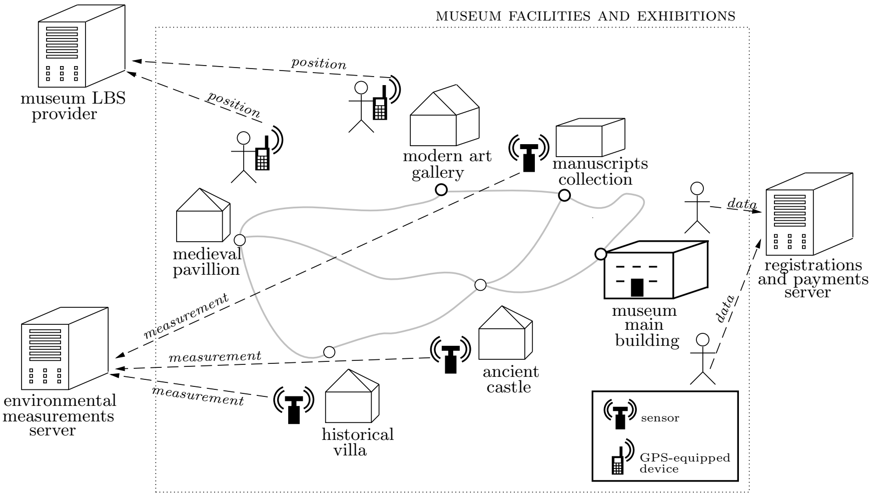


Fig. 1: Reference scenario

and can then collect their tickets presenting the credit card used for the payment at an automatic machine at any of the museum facilities. When purchasing a ticket, a visitor becomes a user of the museum systems. Users can also register to the museum web site and/or follow it on social networks to receive news, discounts, and other information. Data about users and registrations are stored at the registration and payments server, which is in charge of maintaining all the information provided by the users. Figure 1 illustrates the information flows caused by ticket purchases and registration activities by users as the dashed arrows, labeled *data*, from users to the registrations and payments server. Figure 2 illustrates an example of the relation stored and managed by this server. The relation stores personal data (attributes `Phone`, `Name`, `DoB`, and `Sex`), ticket information (attribute `TicketType`), and payment information (attribute `Payment`) about the visitors of the museum. Attribute `TicketType` represents the kind of ticket bought by a visitor, which can be either *Regular* (i.e., full price and no discount) or discounted/free if, for example, the visitor suffers from specific pathologies (*Health* ticket) or has a particular job (e.g., *Army* or *Government*) for which the museum adopts reduced fares.

- *Smart guides to artworks*. Besides traditional guided tours to the different exhibitions, in which an authorized guide escorts the visitors, the museum also offers location-based and automatic guided tours. To this aim, visitors can download an ad-hoc location-based app, provided by the museum, on their smartphone (*smartphone* icon close to users in Figure 1), which describes the artworks based on the position of the user. This indeed represents a great advantage for the visitors of the museum, who can decide their own itineraries without the need of reserving a guide in advance while, at the same time, enjoying professional illustrations of

Phone	Name	DoB	Sex	ZIP	TicketType	Payment
(800) 917-5551	Alice	1960/04/10	F	97401	Health	Credit card
(500) 234-5678	Bob	1970/05/12	M	98302	Army	Debit card
(541) 271-2136	Carol	1960/04/04	F	97467	Regular	Cash
(360) 474-4614	Daniel	1970/05/20	M	98245	Army	Cash
(360) 373-2030	Erik	1970/07/12	M	98312	Navy	Cash
(541) 946-1711	Fred	1960/04/11	F	97434	Professor	Credit card
(360) 435-3746	Greg	1970/07/25	M	98223	Government	Check
(253) 863-5555	Hal	1970/07/30	M	98389	Marines	Cash
(360) 794-7058	Ian	1970/05/12	M	98290	Army	Credit card
(503) 497-91 33	John	1950/12/01	M	97210	Air Force	Debit card

Fig. 2: Personal and payment data of the visitors of the museum

the exhibitions based on their position. Visitors can then walk around in the city and, as soon as they approach an artwork (such as the Trevi fountain, the Sistine Chapel, or the Colosseum in Rome), their location-based app will ring an alarm and a description of the artwork will start. The location-based service offered by the museum can also suggest to visitors the best itinerary to avoid queues that would delay their visit. To this aim, it collects and aggregates location data about the users who are using the location-based app, and takes them into consideration when determining the best itinerary to be suggested to a new visitor. For instance, it can re-arrange the itinerary of visitors (e.g., if a place is too crowded, an updated itinerary skipping that place can be suggested to a visitor). Figure 1 illustrates the communications from users to the Location-Based Service (LBS) provider of the museum to support the smart guide service as dashed arrows, labeled *position*.

- *Environmental monitoring.* To protect the artworks, the museum uses a pervasive environmental monitoring system (*sensor* icons distributed in the environment in Figure 1). This system analyzes and keeps under control different parameters that might harm the artworks such as: the temperature of a room to regulate air conditioning, the quality of the air (e.g., specific pollutants and humidity to enforce specific countermeasures to protect the artworks), and the number of visitors at an exhibition or in a given room to regulate further accesses to the same. Figure 1 illustrates the communication exchanges from sensors to the environmental measurements server as dashed arrows, labeled *measurement*. Figure 3 illustrates an example of the relation kept by this server storing the measurements of temperature (in Celsius degrees), humidity percentage, concentration of PM10 pollutant (in μg/m^3), and noise pollution (in dB).

The data about users and environment are *analyzed* (either at runtime, such as for the environmental sensing, or offline, such as for discovering statistics on the visitors based on the contact information provided at the time of ticket purchases), *stored*, and *maintained* for possible future use, possibly including disclosure to third parties. For instance, the ministry of arts and culture periodically asks the museum to provide all data related to visits and payments. Thanks to these data, the ministry

Sensor	**Temp** (^{o}C)	**Humidity**	**PM10** (μg/m^3)	**Noise** (dB)
int_A	25	40%	25	60
ext_B	28	60%	30	55
int_C	27	45%	40	57
ext_D	30	55%	50	62
ext_E	29	53%	55	58
int_F	22	42%	59	32
ext_G	30	59%	50	47
ext_H	28	60%	42	50
int_I	28	43%	58	30
int_J	22	51%	35	35
ext_K	32	63%	37	65

Fig. 3: Environmental measurements at the facilities of the museum

can study marketing strategies, and take knowledge-based decisions regarding, for example, special rates and discounts for specific groups of visitors, increasing the personnel working in the museum, adjusting the opening hours, planning special exhibitions and many other activities. The museum can also decide to *share* these data with third parties. For instance, the museum can provide its data to research organizations to study countermeasures for improving the quality of the air by reducing the concentration of specific pollutants. The paths of users among the different facilities of the museum can be shared with other museums, to suggest each user the most appropriate smart visit based on their previous ones.

2.2 Privacy issues

The main privacy issues that arise in the considered scenario are related to the fact that the data collected by the museum include sensitive information that can put at risk the privacy of the users to whom it refers. The collected data span from the *contextual information* generated by the pervasive infrastructure to the information released by the users themselves. Contextual information is needed to develop smart services that can react to the environment surrounding a user. A notable example of this kind of information is the *location information* that users continuously release during their visits. Such information can then be used to track the movements of users, which is considered intrusive and harmful of their privacy. The data released by the users are needed to take advantage of the museum services. The storage and processing of these data should always be performed in respect of the privacy of the users. For instance, information like phone numbers cannot be shared with an advertising company without the prior consent of the users. When accessing sensitive information, both *direct* and *indirect* privacy violations may occur, as illustrated in the following.

- *Direct violations.* Direct violations are caused by the presence in the collected data of sensitive information available to all parties accessing the data. For instance, all recipients accessing the data collection in Figure 2 can discover the phone numbers of the visitors of the museum.
- *Indirect violations.* Indirect violations are caused by the possibility of determining sensitive information that is not explicitly included in the collected data, but can be obtained from them. For instance, observing the discounts applied to the tickets purchased by the visitors of the museum in Figure 2, a recipient can infer that *Alice* suffers from a disease and that *Bob* is a military.

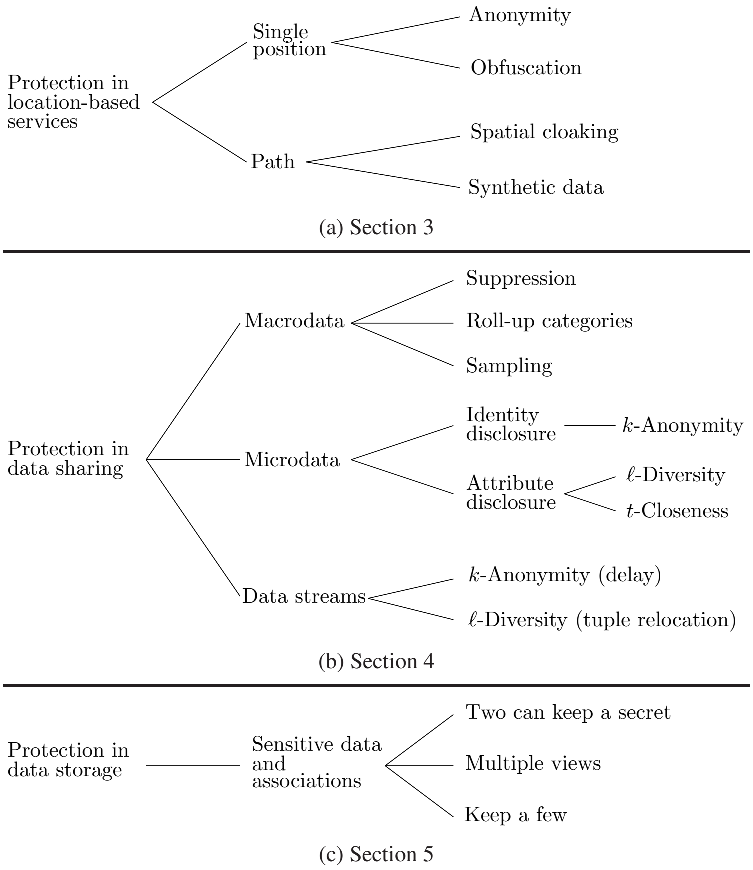
It is interesting to note that privacy violations might affect both individuals represented in the collected data (i.e., registered visitors) as well as individuals that are apparently not involved in the data release. For instance, the relation in Figure 2 includes personal and payment information of the visitors of the museum and its improper sharing or distribution can affect the privacy of the visitors. As another example, consider the table in Figure 3. An insurance company might increase the premium to individuals living in the areas close to the museum and for which the table reports a high PM10 concentration. This behavior clearly affects the privacy of individuals who are not necessarily visitors of the museum.

As it might be clear from the discussion above, privacy violations can occur for a variety of different reasons, including the presence of sensitive information in the data collection (e.g., attributes `Phone` and `TicketType` in the relation in Figure 2), the existence of *correlations* and *associations* among different datasets (e.g., correlations among pollutant concentrations and respiratory diseases), and the observation of data *evolution*. As an example of this latter aspect, suppose that environmental sensor *ext_B* of the museum (see Figure 3) be close to the city railway and record noise levels continuously at regular time intervals. Assume also that the schedule of freight trains be sensitive and therefore not publicly available. By observing peaks in the sensed noise levels, and linking them to the (public) timetables of passenger trains, it might be possible to deduce the schedule of freight trains. *Unusual* data can also leak sensitive information: for example, an individual paying the museum ticket with a very exclusive credit card makes her/his stand from others, and reveals that, with high probability, s/he enjoys a relatively high income.

In the remainder of this chapter, we survey some of the approaches that can be adopted to protect data and users from the privacy issues described above. To guide the reader through the chapter, Figure 4 illustrates a summary of the solutions that will be described in the remaining sections.

3 Protecting location information

The widespread adoption of mobile communication devices and the advancements made on location technologies have contributed to the development of a great variety of location-based services for business, social or informational purposes. As an effect of such innovative services, however, privacy concerns are increasing. In



(a) Section 3

(b) Section 4

(c) Section 5

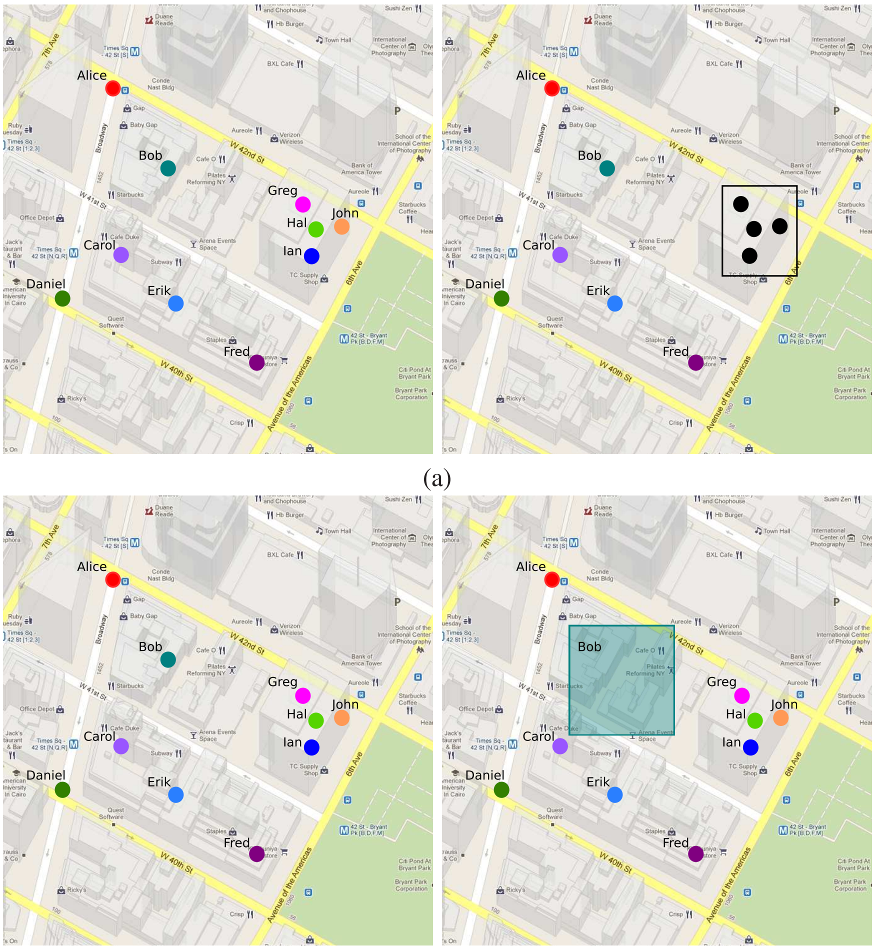
Fig. 4: Summary of the solutions illustrated in Sections 3–5

fact, location information is subject to a variety of privacy threats, including stalking or physical harassment. Location information can also be exploited for inferring sensitive information about users. As an example, consider a user running the location-based app provided by the museum for smart guides in Rome. Since the exhibitions and facilities of the museum are distributed over the city, while walking from the Colosseum to Trevi fountain, the user might stop by a pharmacy selling medicaments for a specific disease, still releasing her/his position to the LBS of the museum. While this might not be a problem when the service provider (the museum, in our example) is trusted by the user, this situation becomes problematic when location information is shared with (or managed by) third parties. Anyone accessing such location information can in fact infer that the user (or an individual close to her) suffers from that specific disease. The existing solutions for protecting privacy of location information can be classified based on whether they aim at protecting the single positions of a user, or her/his path whenever s/he continuously releases the trace of her/his movements to a provider. In our running example, the first class of solutions are important if the user decides to use the smart guide app in pull mode, that is, the app issues a query to the LBS of the museum when the user is close to an artwork. The second class of solutions can instead be useful when the user decides to use the smart guide in push mode. To this aim, the app continuously sends the user location to the LBS of the museum and, as soon as s/he reaches a point of interest, the app automatically receives the description for that art piece.

We will now illustrate the most well-known approaches for protecting location privacy, distinguishing between solutions tailored to protect the single position of a user, and those aimed at protecting her/his path.

Single position protection. *Anonymity-based* techniques (e.g., [4, 5, 8, 25, 28, 33, 39, 40]) aim at protecting the association between users' identities and their precise position to prevent re-identification by observing users' requests to the LBS. These techniques include solutions based on the concept of k-anonymity [16, 43] originally proposed in the database context (see Section 4.2). To protect users' identities, their explicit identifier is removed and the precision of their position is degraded in such a way that a user is indistinguishable by other $k-1$ users in a given location area or temporal interval. Figure 5(a) illustrates an example of the application of such protection techniques, where $k = 4$. In the figure, users are represented by a small circle, labeled with the user name. In the right-hand side of the figure, users *Greg*, *Hal*, *Ian*, and *John* are de-identified (i.e., their identities are not associated with their queries), and all queries are associated with the area represented by the gray rectangular in the figure. Therefore, every request can be indistinguishably generated by any of the four users. Whenever the identity of a user needs to remain attached to her/his location information (e.g., with reference to our scenario, when descriptions of exclusive art pieces should be available only to specific visitors who paid an additional ticket), *obfuscation-based* techniques (e.g., [2, 3, 22]) can be adopted, instead of anonymity-based solutions. Rather than anonymizing users, these techniques degrade the accuracy of their location. The main goal of these techniques is therefore to perturb the location information of the users, while still maintaining a binding with their identity. Figure 5(b) illustrates an example of the application of these techniques. The right-hand side of the figure shows a degradation of the released position of user *Bob*, represented by the shaded rectangle, so to protect his actual position. Note that, as opposite to Figure 5(a), the identity of *Bob* is not hidden to the LBS provider, and remains associated with his (degraded) position.

Path protection. The protection of the trajectory information of a user is a critical aspect in our reference scenario. Suppose that the users adopt the museum app in push mode, meaning that the app on their smartphone continuously sends their positions to the location-based service offered by the museum. While walking around and sightseeing the city, a user might visit other places that can be considered sensitive as they can be exploited to infer personal information about her. For instance, the user can stop to a pharmacy selling drugs for rare diseases, hence making this information available to all parties observing her/his movements. In this scenario, it is possible to adopt path-protecting approaches. Figure 6 illustrates an example



(a)

(b)

Fig. 5: Protecting users' location through anonymity-based (a) and obfuscation-based (b) approaches

of the application of such protection techniques, in which the real paths followed by *Alice* and *Bob* in the left-hand side of the figure have been protected by appearing indistinguishable to the eyes of an observer. Traditionally, these solutions use *spatial cloaking* techniques: a cloaked spatial region must be shared by at least k users and, to protect user trajectories, all k users must appear as belonging to the same region as time passes (e.g., [11, 41, 49]). A different approach is based instead on the generation, and release to the LBS, of (partially) *synthetic* trajectories. For instance, the technique in [38] relies on mix-zones created over synthetic trajectories, obtained with first-order Markov chains from historical data. The release of

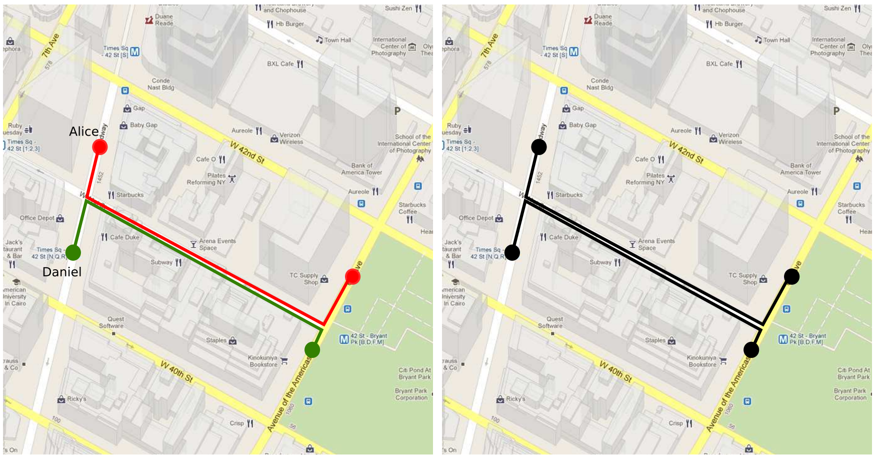


Fig. 6: Protecting users' path

fake paths is also at the basis of a recent technique aimed at counteracting the risk of sensitive information disclosure due to the observation of unusual paths. In fact, being unusual with respect to what is expected and considered common, these paths can leak information not intended for disclosure. The proposal in [6] introduces a framework, based on first-order Markov chains, to evaluate how "unusual" a path followed by a user is with respect to traditional trajectories (in our example, common itineraries followed by visitors), to reduce the risk of inferences by releasing a slightly modified and safe (i.e., less unusual) path.

4 Privacy-preserving data sharing

Data sharing and dissemination are becoming more and more common and, in some cases, even mandatory by law. Collected data can be disseminated in the form of *macrodata* or *microdata* [17]. Macrodata are *aggregate* values representing statistics of interests computed over a sample population. Such statistics are measures that summarize the values of one or more properties/attributes of *respondents* (i.e., individuals, organizations, associations, business establishments, and so on). Microdata are specific data related to single respondents (i.e., single visitors, in our example). The release of macrodata and/or microdata might cause leakage of sensitive information that was not intended for disclosure. In this section, we will then illustrate available solutions for protecting macrodata (Section 4.1) and microdata (Section 4.2), and for protecting data streams, which are common in pervasive scenarios since data are often collected by sensing devices in streams (Section 4.3).

	Cash	Check	Credit Card	Debit Card	Tot
M	3	1	1	2	7
F	1	0	2	0	3
Tot	4	1	3	2	10

(a) Number of male and female visitors purchasing tickets with a given payment method

	Cash	Check	Credit Card	Debit Card	Tot
M	30	10	10	20	70
F	10	0	20	0	30
Tot	40	10	30	20	100

(b) Percentage of male and female visitors purchasing tickets with a given payment method

	Cash	Check	Credit Card	Debit Card	Tot
M	0	0	12.5	4	16.5
F	0	0	11	3.5	14.5
Tot	0	0	23.5	7.5	31

(c) Average delay (number of days) between ticket purchase and collection

Fig. 7: Count (a), frequency (b), and magnitude (c) tables

4.1 Protecting macrodata

Macrodata are represented as tables where each cell of a table is the value of a quantity computed over the considered properties. A macrodata table usually includes marginal totals, that is, the aggregate computed over each row/column in the table. Depending on how macrodata tables are defined, they can be classified as: *i) count* and *frequency tables*, where each cell contains the *number* (*percentage*, respectively) of respondents that share the same value over all attributes of analysis reported in the table; and *ii) magnitude tables*, where each cell contains an *aggregate value* (e.g., sum) of a *quantity of interest* over all attributes of analysis reported in the table. Figures 7(a) and 7(b) illustrate an example of count and frequency tables, respectively, computed over the data in Figure 2, reporting the number and percentage of male and female visitors who purchased tickets with a given payment method. Figure 7(c) illustrates an example of magnitude table reporting the average delay between the purchase of a ticket and its collection. Columns in the tables represent the payment methods, while rows represent male and female visitors, respectively.

Although macrodata do not explicitly include information specifically related to single respondents, sensitive information can still be leaked. To counteract the risk of unintended information disclosure, it is then necessary to first identify and then protect cells that can be considered sensitive [17, 23].

Identifying sensitive cells. Sensitive cells can be identified according to different strategies [23]. In count and frequency tables, sensitive cells can be identified through the *threshold* rule, which classifies a cell as sensitive if its value is less than a given threshold. As an example, consider the macrodata table in Figure 7(a) and suppose that the threshold is set to 1. In this case, the second cell and the third cell in the first row, and the first cell in the second row should be considered sensitive. In magnitude tables, sensitive cells can be identified through different rules (e.g., (n,k)-rule, p-percent rule, $pq-$rule) all aimed at identifying cells whose value could be exploited to estimate too accurately the contribution of one specific respondent. As an example, according to the (n,k)-rule a cell is considered sensitive if less than n respondents contribute to more than k% of its value. For instance, the third cell of the first row in Figure 7(c) does not satisfy (2,90%)-rule as one respondent only contributes to 100% of the cell content.

Protecting sensitive cells. Once detected, sensitive cells must be protected. Several protection techniques have been proposed for macrodata tables. For count and frequency tables, the easiest solution consists in suppressing sensitive cells (*primary suppression)*. Unfortunately, primary suppression might open the door to inferences: if marginal totals are published together with the released table, or are publicly known, it might still be possible to restrict the uncertainty about the missing values. To overcome this risk, additional cells need to be suppressed (*secondary suppression*), and linear programming techniques are typically adopted to minimize the number of cells undergoing secondary suppression. Besides suppression, rounding techniques can also be used, which consist in modifying the original value of a cell by rounding it to a near multiple of a chosen base number. The *roll-up categories* technique instead modifies the original table combining rows and/or columns to obtain a less detailed table. A widely used protection technique is *sampling*, which consists in computing the aggregate values in the macrodata table over a representative sample of the collected data (e.g., in our running example over a sample of the museum visitors). Protection is provided by uncertainty, since a recipient does not know whether a target respondent has been considered in the sampling. These protection techniques can be adopted to protect both count and frequency tables. We note however that also other, more sophisticated, approaches have been proposed to protect sensitive cells in macrodata release.

4.2 Protecting microdata

Many scenarios require that the specific stored data (microdata) be released. Figures 2 and 3 represent two examples of microdata tables. Although microdata provide higher flexibility and utility for final recipients than macrodata, they are subject to a greater risk of privacy breaches. In particular, a microdata table must be protected against both *identity disclosure* (i.e., disclosure of respondents' identities) and *attribute disclosure* (i.e., disclosure of respondents' sensitive information). In the remainder of this section, we present some well-known approaches to protect microdata tables against identity and attribute disclosures.

Phone	Name	DoB	Sex	ZIP	TicketType	Payment
		1960/04/10	F	97401	Health	Credit card
		1970/05/12	M	98302	Army	Debit card
		1960/04/04	F	97467	Regular	Cash
		1970/05/20	M	98245	Army	Cash
		1970/07/12	M	98312	Navy	Cash
		1960/04/11	F	97434	Professor	Credit card
		1970/07/25	M	98223	Government	Check
		1970/07/30	M	98389	Marines	Cash
		1970/05/12	M	98290	Army	Credit card
		1950/12/01	*M*	*97210*	*Air Force*	*Debit card*

(a) De-identified version of the relation in Figure 2

Name	Address	City	ZIP	DoB	Sex
...	...	...	...	...	...
John Jacob	1100 Garden State Parkway	Portland	*97210*	*50/12/01*	*male*
...	...	...	...	...	...

(b) Portland voters' list

Fig. 8: An example of de-identified microdata table (a) and of publicly available non de-identified dataset (b)

4.2.1 Identity disclosure

The attributes in a microdata table can be classified in four classes: *identifiers*, *quasi-identifiers*, *sensitive* attributes, and *non-sensitive* attributes. Identifiers are attributes whose values univocally identify respondents, such as social security numbers and phone numbers. Quasi-identifiers are attributes that can be linked to external sources of information to reduce the uncertainty over the identity of respondents, such as `ZIP`, `DoB`, and `Sex`. Sensitive (non-sensitive, resp.) attributes correspond to the remaining sensitive (non-sensitive, resp.) information of the microdata table. The first step for protecting a microdata table consists in removing (or encrypting) explicit identifiers. A de-identified microdata table, however, does not provide any guarantee of anonymity, since quasi-identifiers might be linked to publicly available information to re-identify respondents. For instance, the de-identified table in Figure 8(a) (computed from the table in Figure 2 removing attributes `Phone` and `Name`) can be linked with the public voters' list of Portland (Figure 8(b)), which includes a single tuple related to a male, living in the 97210 area, and born on 01 December 1950. This combination of values, if unique in the external world as well, uniquely identifies the corresponding tuple in the microdata table as pertaining to *John Jacob*, 1100 Garden State Parkway, revealing that he works in the Air Force and that he paid the visit to the museum with a debit card. It is interesting to note that a study performed on 2000 U.S. Census data showed that 63% of the U.S. population can be *uniquely identified* combining their gender, ZIP code, and complete date of birth [27].

To protect respondents' identities from the linking attack illustrated above, *k*-anonymity [43] requires that any released tuple be *indistinguishably related* to no less than a certain number k of respondents. Since re-identification through linking attacks exploits quasi-identifying attributes, this requirement is translated as follows: *Each release of data must be such that every combination of values of quasi-identifiers can be indistinctly matched to at least k respondents* [43]. Starting from the assumption that each respondent is represented by a tuple in a microdata table (and, vice versa, that each tuple is related to a single respondent), a microdata table satisfies the k-anonymity requirement iff: *i)* each tuple in the table cannot be related to less than k individuals in the population; and *ii)* each individual in the population cannot be related to less than k tuples in the table. Since it is not possible to take into consideration all possible external sources of information, the k-anonymity requirement is typically enforced by taking a safe approach and requiring each respondent be indistinguishable from at least $k-1$ respondents of the table itself (which represents a sufficient, though not necessary, condition for the k-anonymity requirement). A table is therefore said to be k-anonymous if each combination of values of the quasi-identifier appears with either zero or at least k occurrences in the released table.

k-Anonymity is traditionally enforced by adopting *generalization* and *suppression* techniques on the attributes composing the quasi-identifier, without modifying sensitive and non-sensitive attributes. Generalization substitutes the original values with more general values (e.g., the date of birth can be generalized by releasing only the year of birth). Suppression consists in removing information, and is particularly useful to reduce the amount of generalization necessary to guarantee k-anonymity whenever a limited number of outliers (i.e., quasi-identifying values with less than k occurrences) would require considerable generalizations. Generalization and suppression can be applied at different levels of granularity, and several approaches have been proposed combining them in different ways [7, 16, 34, 35, 43]. The majority of available solutions rely on attribute generalization and tuple suppression. Figure 9(a) illustrates a 3-anonymous microdata table obtained from the table in Figure 8, where attribute `Payment` has been projected out since not intended for release. Attributes `DoB`, `Sex`, and `ZIP` in the table are considered as the quasi-identifier, and `TicketType` is considered sensitive as the museum is not authorized to disclose such information. The 3-anonymous table has been obtained by generalizing attributes `DoB` (only the year and month of birth are released) and `ZIP` (only the first two digits are released). Also, the outlier tuple related to John Jacob has been suppressed not to force further generalization on the date of birth, since John is the only respondent born in 1950.

Reducing the details in the anonymized table, k-anonymity inevitably causes information loss. To find a good trade-off between data protection and utility for final recipients, it is necessary to compute a k-anonymous table minimizing the adoption of generalization and suppression. To this aim, both exact and heuristic algorithms can be adopted [16].

Phone	Name	DoB	Sex	ZIP	TicketType
		1970/05/**	M	98***	Army
		1970/05/**	M	98***	Army
		1970/05/**	M	98***	Army
		1960/04/**	F	97***	Health
		1960/04/**	F	97***	Regular
		1960/04/**	F	97***	Professor
		1970/07/**	M	98***	Navy
		1970/07/**	M	98***	Government
		1970/07/**	M	98***	Marines

(a)

Phone	Name	DoB	Sex	ZIP	TicketType
		1970/**/**	M	983**	Army
		1970/**/**	M	983**	Marines
		1970/**/**	M	983**	Navy
		1960/**/**	F	974**	Health
		1960/**/**	F	974**	Regular
		1960/**/**	F	974**	Professor
		1970/**/**	M	982**	Army
		1970/**/**	M	982**	Army
		1970/**/**	M	982**	Government

(b)

Fig. 9: An example of 3-anonymous table (a) and 3-anonymous and 2-diverse table (b)

4.2.2 Attribute disclosure

k-Anonymity, while effective for protecting respondents' identities, does not protect against attribute disclosure. To protect the association between respondents' identities and their values of sensitive attributes, alternative solutions extending k-anonymity have been proposed. In the following, we will illustrate ℓ-diversity and t-closeness, two well-known extensions that counteract attribute disclosure.

ℓ-Diversity. ℓ-Diversity has been proposed to counteract two specific attacks that might cause attribute disclosure in a k-anonymous table, namely the *homogeneity attack* [37, 43] and the *external knowledge attack* [37].

- *Homogeneity attack*. k-Anonymity does not impose restrictions on the values that can be assumed by the sensitive attribute in an equivalence class (i.e., by the tuples sharing a same value for the quasi-identifier). As a consequence, it might happen that a given equivalence class includes tuples with the same sensitive value. If a data recipient knows the quasi-identifier value of an individual that is represented in the table, the data recipient can identify the equivalence class corresponding to the target respondent, and then infer the value of her/his sensitive attribute. For instance, consider the 3-anonymous table in Figure 9(a) and suppose that a recipient knows that *Daniel* born on 1970/05/20 is included in the table. Since all the tuple in the equivalence class with quasi-identifier value equal to (1970/05/**,M,98***) have *Army* as value for attribute `TicketType`, the recipient can infer that *Daniel* works in the army, which represents a sensitive information not intended for disclosure in our example.
- *External knowledge attack*. k-Anonymity assumes that the only external knowledge a recipient can have be represented by external sources linking respondents' quasi-identifier values to their identities. However, a recipient might exploit some additional external knowledge about some respondents to infer their associated sensitive information. For instance, consider a 3-anonymous equivalence class where two out of three tuples have *Army* as value for attribute `TicketType`, while the third tuple has value *Health*. Suppose now that a recipient knows that

a target respondent *Phil* is included in this equivalence class, and that *Phil* does not suffer from any specific disease. The recipient can easily infer that *Phil* is not likely to pay for a reduced ticket for medical conditions, hence discovering that he works in the army.

To counteract these two attacks, ℓ-diversity extends k-anonymity by requiring the existence of at least ℓ *well-represented* values for the sensitive attribute in each equivalence class [37]. A straightforward understanding of “well-represented” values requires each equivalence class to have *at least* ℓ different values for the sensitive attribute. For instance, the 3-anonymous table in Figure 9(b) is also 2-diverse. It is easy to see that an ℓ-diverse table is not vulnerable to the homogeneity attack as each equivalence class has at least ℓ different values for the sensitive attribute. Also, external knowledge attacks lose effectiveness as ℓ increases, since more external knowledge is necessary to associate a specific sensitive attribute value with a target respondent.

An ℓ-diverse table that minimizes the adoption of generalization and suppression to reduce information loss can be computed using any algorithm that computes an optimal k-anonymous table, by simply adding a control to check whether the condition on the diversity of the sensitive attribute values is satisfied by all the equivalence classes in the table [37].

***t*-Closeness.** An ℓ-diverse table might still cause improper disclosures of sensitive information, since it is vulnerable to the following two attacks [36].

- *Skewness attack*. This attack may occur when the distribution of values of the sensitive attribute within a given equivalence class differs from the general (demographic or in the whole table) one. Indeed, differences in these distributions highlight changes in the probability with which a respondent in the equivalence class is associated with a specific sensitive value. As an example, the 2-diverse table in Figure 9(b) leaks the information that respondents in the third equivalence class work in the army with 2/3 probability, compared to the 1/3 probability over the whole relation.
- *Similarity attack*. This attack may occur when the values of the sensitive attribute within a given equivalence class are (despite syntactically different as demanded by ℓ-diversity) semantically similar. For instance, all respondents in the first equivalence class of the 2-diverse table in Figure 9(b) work in the Armed Forces, as the values assumed by the three tuples are *Army*, *Marines*, and *Navy*.

To counteract these two attacks, t-closeness extends the k-anonymity requirement taking into account the distribution of sensitive values in equivalence classes [36]. t-Closeness requires that the frequency distribution of the sensitive values in each equivalence class be close (i.e., with distance smaller than a fixed threshold t) to the distribution of the same attribute values in the microdata table. Note that the distance between the frequency distribution of the sensitive attribute values in the released table and in each equivalence class can be evaluated adopting several metrics (e.g., Earth Mover Distance [36]). The enforcement of the t-closeness requirement makes the skewness attack harmless, as the knowledge of the quasi-identifier value for a target respondent does not change the probability of inferring the sensitive value associated with her. t-Closeness reduces also the effectiveness of the similarity attack: the presence of semantically similar values in an equivalence class can only be due to the presence of the same values in the whole microdata table.

4.3 Protecting data streams

The solutions proposed to provide k-anonymity, ℓ-diversity, and t-closeness, as well as the majority of microdata protection techniques, assume all data that need to be released to be available at initial time. Then, the chosen protection technique can be applied on the whole collection at once. In the context of pervasive systems, however, this assumption might be too strong as new data are continuously generated (and possibly need to be immediately released), forming a so-called *data stream*. In the context of data streams, timeliness usually assumes a paramount importance in the release process, as disclosing old or outdated data is likely to be of little interest for final recipients. Data streams can be protected by applying ad-hoc solutions to guarantee k-anonymity, which are typically based on generalization and on the introduction of a limited delay in data publication. The first solution in this direction has been proposed in [50], and consists in publishing all the tuples in an equivalence class at the same time. To this aim, a set of equivalence classes – all initially empty – is prepared. When a new tuple is generated by the stream, it is inserted into a suitable equivalence class, if such class exists; a new equivalence class suitable for the tuple is generated, otherwise. As soon as an equivalence class includes k tuples (which must be related to k different respondents), these tuples are generalized to the same quasi-identifier value and published.

Aiming at enforcing ℓ-diversity, rather than k-anonymity, an alternative approach has been proposed in [47], where data are assumed to be generated and published as "snapshots" (i.e., sets of records available at a given moment of time) of d tuples each. This technique combines traditional generalization and suppression techniques with *tuple relocation* to guarantee ℓ-diversity. In a nutshell, relocation consists in moving a tuple from one snapshot to a more recent one, if this delay in data publishing can be useful to satisfy ℓ-diversity.

5 Privacy-preserving data storage

Privacy concerns can arise also when data storage and management is delegated (for various reasons, such as economical costs) to external, possibly not fully trusted, storage providers. These scenarios present several challenging issues, ranging from fault tolerance, data protection, data and query integrity to private access (e.g., [31, 32, 44]). Relying on external providers is particularly appealing in the context of pervasive data, due to the high volume of data generated requiring large

$c_1 = \{$`Phone`$\}$
$c_2 = \{$`Name, TicketType`$\}$
$c_3 = \{$`Name, Payment`$\}$
$c_4 = \{$`TicketType, Payment`$\}$

Fig. 10: Confidentiality constraints for the relation in Figure 2

storage space that, for example, the museum is most likely not to have. In this scenario, to protect confidentiality of data to unauthorized users – including the external provider – a straightforward solution is represented by wrapping an encryption layer around the data to be protected. While effective for protecting data confidentiality, encryption inevitably complicates query execution that becomes possible only with the adoption of expensive ad-hoc encryption schemes [10, 18, 26, 46], or indexes [9, 19, 29, 45]. Moreover, in many scenarios, the sensitive information to be protected is represented by the association among data items, rather than the data themselves singularly taken. For instance, with reference to our running example, knowing that a user named *Alice* visited the museum, and that a user paid a reduced ticket for health reasons may not be sensitive. But discovering that *Alice*, who visited the museum, paid a reduce ticket because of her health problems might represent a confidential information.

Sensitive associations can be modeled as *confidentiality constraints*, which are set of attributes whose joint visibility (i.e., association) is sensitive. Attributes whose values are sensitive per se correspond to singleton constraints. For instance, with reference to our running example, Figure 10 represents an example of confidentiality constraints over the relation in Figure 2. Constraint c_1 states that the phone numbers of the visitors represent sensitive information to be protected, and constraint c_2 (c_3 and c_4, respectively) states that the association between visitors' name and type of ticket (name and payment, and type of ticket and payment, respectively) is sensitive and must be protected.

The adoption of encryption to satisfy confidentiality can be (partially) avoided storing the collected data through a set of *privacy-preserving views*, which are defined in such a way to satisfy confidentiality constraints [1, 12, 13, 14, 15]. To this aim, sensitive associations among attributes are broken (fragmented) by storing the attributes composing each of them in different views. Sensitive associations are then protected by restricting visibility over the views or by ensuring their unlinkability.

Given a relation to be protected, privacy-preserving views can be defined according to different paradigms, differing on how data are fragmented to satisfy the confidentiality constraints. In the following, we briefly illustrate the three most important approaches that can be used in our scenario to create privacy-preserving views.

Two can keep a secret [1]. Given a data collection, this strategy produces two views V_1 and V_2, to be stored at two non-communicating providers. Sensitive attributes are protected by *obfuscating* (e.g., encrypting) them, while sensitive associations are protected by distributing the attributes in the confidentiality constraint between the two views. In addition to sensitive attributes, also some attributes appearing

V_1

tid	Name	DoB	Sex	Payment^k	Phone^k
t_1	Alice	1960/04/10	F	α	λ
t_2	Bob	1970/05/12	M	β	μ
t_3	Carol	1960/04/04	F	γ	ν
t_4	Daniel	1970/05/20	M	δ	ξ
t_5	Erik	1970/07/12	M	ε	o
t_6	Fred	1960/04/11	F	ζ	π
t_7	Greg	1970/07/25	M	η	ρ
t_8	Hal	1970/07/30	M	θ	σ
t_9	Ian	1970/05/12	M	ι	τ
t_{10}	John	1950/12/01	M	κ	υ

V_2

tid	TicketType	ZIP	Payment^k	Phone^k
t_1	Health	97401	ϕ	$\mathit{\Gamma}$
t_2	Army	98302	χ	$\mathit{\Delta}$
t_3	Regular	97467	ψ	$\mathit{\Theta}$
t_4	Army	98245	ω	$\mathit{\Lambda}$
t_5	Navy	98312	ε	$\mathit{\Xi}$
t_6	Professor	97434	ϑ	$\mathit{\Pi}$
t_7	Government	98223	ϖ	$\mathit{\Sigma}$
t_8	Marines	98389	ρ	$\mathit{\Upsilon}$
t_9	Army	98290	ς	$\mathit{\Phi}$
t_{10}	Air Force	97210	φ	$\mathit{\Psi}$

(a) Two can keep a secret

V_1

salt	enc	Name	DoB
s_{01}	α	Alice	1960/04/10
s_{02}	β	Bob	1970/05/12
s_{03}	γ	Carol	1960/04/04
s_{04}	δ	Daniel	1970/05/20
s_{05}	ε	Erik	1970/07/12
s_{06}	ζ	Fred	1960/04/11
s_{07}	η	Greg	1970/07/25
s_{08}	θ	Hal	1970/07/30
s_{09}	ι	Ian	1970/05/12
s_{10}	κ	John	1950/12/01

V_2

salt	enc	TicketType	Sex
s_{11}	λ	Health	F
s_{12}	μ	Army	M
s_{13}	ν	Regular	F
s_{14}	ξ	Army	M
s_{15}	o	Navy	M
s_{16}	π	Professor	F
s_{17}	ρ	Government	M
s_{18}	υ	Marines	M
s_{19}	ϕ	Army	M
s_{20}	χ	Assistant	M

V_3

salt	enc	Payment	ZIP
s_{21}	ψ	Credit card	97401
s_{22}	o	Debit card	98302
s_{23}	π	Cash	97467
s_{24}	ρ	Cash	98245
s_{25}	σ	Cash	98312
s_{26}	τ	Credit card	97434
s_{27}	υ	Check	98223
s_{28}	ϕ	Cash	98389
s_{29}	χ	Credit card	98290
s_{30}	ψ	Debit card	97210

(b) Multiple views

V_o

Phone	Name	TicketType
(800) 917-5551	Alice	Health
(500) 234-5678	Bob	Army
(541) 271-2136	Carol	Regular
(360) 474-4614	Daniel	Army
(360) 373-2030	Erik	Navy
(541) 946-1711	Fred	Professor
(360) 435-3746	Greg	Government
(253) 863-5555	Hal	Marines
(360) 794-7058	Ian	Army
(503) 497-91 33	John	Assistant

V_s

DoB	Sex	ZIP	Payment
1960/04/10	F	97401	Credit card
1970/05/12	M	98302	Debit card
1960/04/04	F	97467	Cash
1970/05/20	M	98245	Cash
1970/07/12	M	98312	Cash
1960/04/11	F	97434	Credit card
1970/07/25	M	98223	Check
1970/07/30	M	98389	Cash
1970/05/12	M	98290	Credit card
1950/12/01	M	97210	Debit card

(c) Keep a few

Fig. 11: Privacy-preserving views over the relation in Figure 2 satisfying the constraints in Figure 10

in sensitive associations might be obfuscated when two views are not sufficient to protect all sensitive associations. A common attribute `tid` is included in both views, to allow the data owner (and all authorized users) to reconstruct the original relation. Figure 11(a) illustrates two views defined over the relation in Figure 2 satisfying the constraints in Figure 10. Note that attribute `Payment`, although not sensitive per se, has been obfuscated in both the views: in fact, its plaintext representation in view V_1 would violate constraint c_3, and in view V_2 would violate c_4.

Multiple views [13, 15]. Given a data collection, this strategy produces a set $\{V_1,\dots,V_n\}$ of unlinkable views. The multiple views approach removes the limiting assumption of the existence of two non-communicating providers, hence resulting applicable to several real-world scenarios. According to this approach, sensitive attributes are protected with encryption, while sensitive associations are protected by distributing their attributes in different views. Views include disjoint sets of attributes, to guarantee their unlinkability. Note that, since the number of views that can be produced is not limited to two, no attribute that is not sensitive per se needs to be protected with encryption. To allow query execution over a single view, each view is *complete*, meaning that it stores all the attributes of the original relation in either encrypted or plaintext form. Attributes that are encrypted in a view are encrypted in a single encrypted chunk (at the level of tuple), which is properly salted not to expose the frequencies of values. Figure 11(b) illustrates three views defined over the relation in Figure 2 satisfying the constraints in Figure 10.

Keep a few [14]. Given a data collection, this strategy produces two views V_o and V_s only, one of which (i.e., V_o) is stored at a trusted party (e.g., the data owner). The *keep a few* approach completely departs from encryption: sensitive attributes are protected by storing them in V_o maintained at the trusted party, while sensitive associations are protected by storing at least one attribute, for each association, in V_o. The two views include a common attribute `tid` to allow the owner and authorized users to reconstruct the content of the original relation. Figure 11(c) illustrates the two views V_o and V_s defined over the relation in Figure 2 satisfying the constraints in Figure 10, where view V_s stores attribute `Phone`, which is sensitive per se, and one attribute for constraints c_2, c_3 and c_4.

6 Conclusions

The pervasive availability of computing infrastructures, often enriched with sensorial capabilities and context awareness to provide personalized services to users, causes unprecedented privacy risks that need to be carefully tackled. In this chapter, starting from a sample scenario, we have illustrated such privacy risks, and discussed some available solutions to counteract them when accessing, sharing, and storing information collected through pervasive systems.

Acknowledgements This chapter was supported in part by the EC within the 7FP project ABC4EU (grant agreement 312797) and within a H2020 project ESCUDO-CLOUD (grant agreement 644579), and by the Italian Ministry of Research within PRIN project “GenData 2020” (2010RT-FWBH).

References

1. G. Aggarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, R. Motwani, U. Srivastava, D. Thomas, and Y. Xu. Two can keep a secret: A distributed architecture for secure database services. In *Proc. of CIDR 2005*, Asilomar, CA, USA, January 2005.
2. C.A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, and P. Samarati. Location privacy protection through obfuscation-based techniques. In *Proc. of DBSec 2007*, Redondo Beach, CA, USA, July 2007.
3. C.A. Ardagna, M. Cremonini, S. De Capitani di Vimercati, and P. Samarati. An obfuscation-based approach for protecting location privacy. *IEEE TDSC*, 8(1):13–27, January-February 2011.
4. C.A. Ardagna, S. Jajodia, P. Samarati, and A. Stavrou. Providing mobile users' anonymity in hybrid networks. In *Proc. of ESORICS 2010*, Athens, Greece, September 2010.
5. C.A. Ardagna, S. Jajodia, P. Samarati, and A. Stavrou. Providing users' anonymity in mobile hybrid networks. *ACM TOIT*, 12(3):1–33, May 2013. article 7.
6. C.A. Ardagna, G. Livraga, and P. Samarati. Protecting privacy of user information in continuous location-based services. In *Proc. of CSE 2012*, Paphos, Cyprus, December 2012.
7. R. J. Bayardo and R. Agrawal. Data privacy through optimal *k*-anonymization. In *Proc. of ICDE 2005*, Tokyo, Japan, April 2005.
8. C. Bettini, S. Jajodia, P. Samarati, and X. Sean Wang, editors. *Privacy in Location-Based Applications: Introduction, Research Issues and Applications*. LNCS 5599, Springer, 2009.
9. A. Ceselli, E. Damiani, S. De Capitani di Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati. Modeling and assessing inference exposure in encrypted databases. *ACM TISSEC*, 8(1):119–152, February 2005.
10. Y.C. Chang and M. Mitzenmacher. Privacy preserving keyword searches on remote encrypted data. In *Proc. of ACNS 2005*, New York, NY, USA, June 2005.
11. C.-Y. Chow and M.F. Mokbel. Enabling private continuous queries for revealed user locations. In *Proc. of SSTD 2007*, Boston, MA, USA, July 2007.
12. V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Fragmentation and encryption to enforce privacy in data storage. In *Proc. of the 12th European Symposium On Research In Computer Security (ESORICS 2007)*, Dresden, Germany, September 2007.
13. V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Combining fragmentation and encryption to protect privacy in data storage. *ACM TISSEC*, 13(3):22:1–22:33, July 2010.
14. V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Selective data outsourcing for enforcing privacy. *JCS*, 19(3):531–566, 2011.
15. V. Ciriani, S. De Capitani di Vimercati, S. Foresti, G. Livraga, and P. Samarati. An OBDD approach to enforce confidentiality and visibility constraints in data publishing. *JCS*, 20(5):463–508, 2012.
16. V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. k-Anonymity. In T. Yu and S. Jajodia, editors, *Secure Data Management in Decentralized Systems*. Springer, 2007.
17. V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. Microdata protection. In T. Yu and S. Jajodia, editors, *Secure Data Management in Decentralized Systems*. Springer, 2007.
18. R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. In *Proc. of CCS 2006*, Alexandria, VA, USA, October - November 2006.
19. E. Damiani, S. De Capitani di Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati. Balancing confidentiality and efficiency in untrusted relational DBMSs. In *Proc. of ACM CCS 2003*, Washington, DC, USA, October 2003.
20. S. De Capitani di Vimercati, S. Foresti, G. Livraga, and P. Samarati. Protecting privacy in data release. In A. Aldini and R. Gorrieri, editors, *Foundations of Security Analysis and Design VI*. Springer, 2011.

21. S. De Capitani di Vimercati, S. Foresti, and P. Samarati. Managing and accessing data in the cloud: Privacy risks and approaches. In *Proc. of CRiSIS 2012*, Cork, Ireland, October 2012.
22. M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. In *Proc. of PERVASIVE 2005*, Munich, Germany, May 2005.
23. Federal Committee on Statistical Methodology. Statistical policy working paper 22, May 1994. Report on Statistical Disclosure Limitation Methodology.
24. M. Gamassi, V. Piuri, D. Sana, and F. Scotti. Robust fingerprint detection for access control. In *Proc. of RoboCare 2005*, Rome, Italy, May 2005.
25. B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE TMC*, 7(1):1–18, January 2008.
26. E.-J. Goh. Secure indexes. Technical Report 2003/216, Cryptology ePrint Archive, 2003. http://eprint.iacr.org/.
27. P. Golle. Revisiting the uniqueness of simple demographics in the US population. In *Proc. of WPES 2006*, Alexandria, VA, USA, October 2006.
28. M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. of MobiSys 2003*, San Francisco, CA, USA, May 2003.
29. H. Hacigümüş, B. Iyer, and S. Mehrotra. Providing database as a service. In *Proc. of ICDE 2002*, San Jose, CA, USA, February 2002.
30. R. Jhawar, V. Piuri, and P. Samarati. Supporting security requirements for resource management in cloud computing. In *Proc. of CSE 2012*, Paphos, Cyprus, December 2012.
31. R. Jhawar, V. Piuri, and M. Santambrogio. A comprehensive conceptual system-level approach to fault tolerance in cloud computing. In *Proc. of SysCon 2012*, Vancouver, BC, Canada, March 2012.
32. R. Jhawar, V. Piuri, and M. Santambrogio. Fault tolerance management in cloud computing: a system-level perspective. *IEEE Systems Journal*, 7(2):288–297, June 2013.
33. P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE TKDE*, 19(12):1719–1733, December 2007.
34. K. LeFevre, D.J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proc. of SIGMOD 2005*, Baltimore, MD, USA, June 2005.
35. K. LeFevre, D.J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *Proc. of ICDE 2006*, Atlanta, GA, USA, April 2006.
36. N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In *Proc. of ICDE 2007*, Istanbul, Turkey, 2007.
37. A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. *ACM TKDD*, 1(1):3:1–3:52, March 2007.
38. J. Meyerowitz and R. Roy Choudhury. Hiding stars with fireworks: location privacy through camouflage. In *Proc. of MobiCom 2009*, Beijing, China, September 2009.
39. M.F. Mokbel, C.-Y. Chow, and W.G. Aref. The new Casper: Query processing for location services without compromising privacy. In *Proc. of VLDB 2006*, Seoul, Korea, September 2006.
40. K. Mouratidis and M.L. Yiu. Anonymous query processing in road networks. *IEEE TKDE*, 22(1):2–15, January 2010.
41. X. Pan, X. Meng, and J. Xu. Distortion-based anonymity for continuous queries in location-based mobile services. In *Proc. of ACM GIS 2009*, Seattle, WA, USA, November 2009.
42. V. Piuri and F. Scotti. Fingerprint biometrics via low-cost sensors and webcams. In *Proc. of BTAS 2008*, Washington, D.C., USA, October 2008.
43. P. Samarati. Protecting respondents' identities in microdata release. *IEEE TKDE*, 13(6):1010–1027, November 2001.
44. P. Samarati. Data security and privacy in the cloud. In *Proc. of ISPEC 2014*, Fuzhou, China, May 2014.
45. P. Samarati and S. De Capitani di Vimercati. Data protection in outsourcing scenarios: Issues and directions. In *Proc. of ASIACCS 2010*, Beijing, China, April 2010.
46. C. Wang, N. Cao, K. Ren, and W. Lou. Enabling secure and efficient ranked keyword search over outsourced cloud data. *IEEE TPDS*, 23(8):1467–1479, August 2012.

47. K. Wang, Y. Xu, R. Wong, and A. Fu. Anonymizing temporal data. In *Proc. of ICDM 2010*, Sydney, Australia, December 2010.
48. M. Weiser, R. Gold, and J.S. Brown. The origins of ubiquitous computing research at parc in the late 1980s. *IBM Systems Journal*, 38(4):693–696, 1999.
49. T. Xu and Y. Cai. Location anonymity in continuous location-based services. In *Proc. of ACM GIS 2007*, Seattle, WA, USA, Nov. 2007.
50. B. Zhou, Y. Han, J. Pei, B. Jiang, Y. Tao, and Y. Jia. Continuous privacy preserving publishing of data streams. In *Proc. of the EDBT 2009*, Saint Petersburg, Russia, March 2009.