Digital Infrastructure Policies for Data Security and Privacy in Smart Cities

Sabrina De Capitani di Vimercati, *Università degli Studi di Milano, sabrina.decapitani@unimi.it*
Sara Foresti, *Università degli Studi di Milano, sara.foresti@unimi.it*
Giovanni Livraga, *Università degli Studi di Milano, giovanni.livraga@unimi.it*
Pierangela Samarati, *Università degli Studi di Milano, pierangela.samarati@unimi.it*

Abstract

The availability of large amounts of data is vital for the working of smart cities, which need to process heterogeneous information on citizens and the surrounding environment for enabling smart services. Since those data collections can include personal/sensitive information, ensuring security and privacy of the data collected and produced in a smart city is a key problem to be addressed. Two of the main pillars for protecting data privacy and security are anonymization and encryption, which however need to be carefully designed and adopted for ensuring effective protection while not compromising the possibility of performing analysis, a central aspect in smart cities. In this chapter, we address the problem of protecting large data collections in the context of smart cities, and illustrate possible approaches for effectively anonymizing data and for encrypting them, while permitting to perform computations.

Keywords: Data security, data privacy, anonymization, encryption, collaborative computation.

Contents

1 Introduction

In the vocabulary of our society, a smart city is a place where –to the benefits of its citizens and of the urban environment they live in– new services are created, and traditional services are made more efficient, through the use of Information and Communication Technologies (ICTs). At their heart, smart cities and smart services are based on the knowledge that can be extracted from the analysis of large and heterogeneous data streams and collections. For example, intelligent transportation can leverage mobility data coming from connected vehicles, images coming from surveillance cameras, information on accidents and plans for road works to support commuters in finding the best way to move from home to work and vice-versa, with clear environmental benefits in terms of reduced traffic congestions and emissions. Similarly, intelligent healthcare can leverage medical records of patients, real-time sensing through wearable devices, and information on social contacts for providing citizens with smart healthcare, with benefits also for the environment in terms, for example, of better control of contagious diseases. It goes without saying that such a scenario, while unlocking new means for the common well-being and novel services that can be beneficial for all citizens, raises at the same time major concerns related to the security and privacy of the data used, analyzed, and produced [9, 52]. The collected data can include users' identities, their locations and movements (e.g., collected for intelligent transportation), health status and medical conditions (e.g., collected for intelligent healthcare), personal habits and lifestyle (e.g., inferred from smart living applications or intelligent surveillance systems), social circles and activities (e.g., determined from sensing devices and online social media platforms), just as mere examples. Improperly accessing, analyzing, modifying, or sharing those data (be them raw collected data, associations derived from the combination of different sources, or the result of analysis) may result in major privacy and security violations, with negative impacts on both the involved citizens, and the overall system of the smart city that would appear vulnerable and could be considered a threat to its citizens.

It is interesting to underline that guaranteeing protection to data related to individuals is not simply a recommendation from privacy advocates and researchers, but a concrete requirement also coming from legislators and governmental agencies. For example, the European General Data Protection Regulation (GDPR), through its revolutionary requirements for data protection *by design* and *by default*, requires the data controller (i.e., the subject or entity that determines the purposes and means of the processing of personal data) to adopt appropriate technical and organizational measures designed to implement data protection principles, and to ensure that, by default, only personal data which are necessary for the specific purpose of the processing are elaborated [1]. Violating these requirements is severely sanctioned. To this end, ensuring proper protection to data is actually an enabling factor for permitting smarter cities that make use, in a controlled way, of citizens' data.

The security and privacy problems that may arise in such a complex scenario are multiple and diverse [19, 20, 37, 38, 39]. Data in smart cities are collected, transmitted over the network, stored, processed, and possibly shared. In any of these steps data could be violated, improperly accessed, tampered without permission, abused for purposes different from what was the original purpose. The scenario can be further complicated by the (almost inevitable) use of external platforms (e.g., in the cloud) necessary for storing and processing the huge amounts of data needed for the working of a smart city. Cloud providers are typically considered honest-but-curious, meaning trusted for correctly managing data but not for accessing their contents, hence introducing another variable in the already complex scenario of data protection in smart cities. In addition, data used in a smart city can be under the authority of different entities, which may be entitled to specify access or usage restrictions and requirements. For instance, air quality measurements may be performed by a private company managing a sensor network, while healthcare data may be under the control of a public medical authority: combining and analyzing data coming from these two sources may be of interest for intelligent environmental monitoring, but inevitably requires to comply to all restrictions that those entities may have specified [24].

The scientific community has devoted major efforts towards the development of effective

[1] `http://data.europa.eu/eli/reg/2016/679/oj`

models and techniques for protecting data throughout the various steps of their lifecycle. Two of the main pillars on which more complex solutions, specifically tailored to different application scenarios, can be built are data *anonymization* and *encryption*. In the context of smart cities, anonymization can be adopted for different purposes, for instance whenever data should be shared or undergo analysis that, while needing access to part of its informative content, do not need (and/or cannot access) the identities of the individuals to which the data pertain. For instance, consider a dataset containing health information of a set of citizens. To the purpose of intelligent medicine, such dataset can be analyzed by the local hospital, and knowing who are the individuals behind the analyzed data can permit, for instance, to link those data to real-time measurements coming from wearable devices for monitoring patients and promptly intervening whenever a situation arises. The same dataset could however also be of interest to the local municipality to determine relationships between pollutant emissions (collected through a sensor network) and respiratory diseases: in this case, the medical dataset could (and/or should, depending on the existing restrictions and regulations) be anonymized, so to allow the municipality to study the correlations without exposing the individuals' identities. Encryption can instead be adopted to protect data from unauthorized access, ensuring that only subjects knowing the encryption key can access a dataset. In the context of smart cities, this can prove extremely useful when leveraging the cloud, or more in general external providers, for storing or analyzing the data. In principle, encryption could be enforced before or after outsourcing to the cloud. In the first scenario, if the storage provider is not given access to the encryption key, data would be protected also against it.

Both data anonymization and encryption, while representing promising solutions for protecting data in different scenarios, still represent open research fields. In particular, anonymizing data is a complex problem, which has received attention by the research community for decades now and is still far from having a definitive solution. Similarly, encrypting data, while representing a simple approach for protecting them against unauthorized accesses, can represent an obstacle for performing operations and analysis on the data, which is however a primary goal in smart cities. In this chapter, we address the problem of protecting large data collections in the context of smart cities, and focus on the issues of effectively anonymizing/encrypting them while permitting computations over them. We illustrate basic concepts and survey recent approaches, based on syntactic and semantic privacy requirements, for anonymizing data collections (Section 2), and discuss and overview recent approaches for data encryption permitting selective and fine-grained access without decryption (Section 3). We also overview the problem of allowing collaborative computations combining different data sources while obeying different access restrictions to the involved data (Section 4). We finally give our conclusions in Section 5.

2 Data anonymity

Ensuring data anonymity is a problem far from having a trivial solution. The straightforward solution of removing all identifying information (e.g., names or e-mail addresses) from a piece of data unfortunately does not offer any anonymity guarantee. Besides identifiers, there might in fact be other information, called *quasi-identifier* (QI), which may be used, possibly in combination with other data sources, to link de-identified (and only apparently anonymous) data to the individuals to whom such data pertain [21]. Examples of QI can include date of birth, gender, addresses, and other information whose combination may be peculiar to an individual. Completely removing or obfuscating also QI information along with the removal of identifiers, while possibly enhancing data privacy, may prevent useful analysis on the protected data, which is a key aspect in smart cities and must not be overlooked.

To illustrate how QI can be used to re-identify individuals, consider the dataset in Figure 1, produced by the public hospital `Hosp` of a smart city for intelligent healthcare and reporting, for a set of citizens, the most recent diagnosis they had along with demographic information. Figure 2(a) illustrates a de-identified version of the dataset in Figure 1, where social security numbers and names of patients have been removed. Still, the de-identified

SSN	Name	BirthDate	Gender	ZIP	Diagnosis
978-05-0111	Andrew	1955/07/05	M	12311	Stroke
978-06-0212	Barbara	1970/10/13	F	12333	Flu
978-04-0513	Carl	1955/07/10	M	12313	COVID-19
978-12-0114	Donna	1970/10/20	F	12331	Flu
978-07-0715	Elizabeth	1970/06/15	F	12332	Cancer
978-42-0116	Flynn	1955/07/20	M	12312	Asthma
978-56-0217	Gladys	1970/06/03	F	12324	Cancer
978-12-0118	Harriett	1970/06/28	F	12325	Dementia
978-00-0319	Isobel	1970/10/13	F	12326	Flu
978-63-0120	Janet	1970/12/01	F	12344	Depression

Figure 1: An example of a dataset including health information for a set of citizens.

dataset contains attributes BirthDate, Gender, and ZIP that may be linked to other non-de-identified data sources to restrict the uncertainty over some of the respondents (i.e., the individuals to whom data refer). Suppose that the de-identified dataset is shared by `Hosp` with private organization `Env`, in charge of maintaining a sensor network for intelligent mobility, and which has access to the smart city civil registry. If the registry includes only a female, born on 1970/12/01 and living in 12344, then `Env` (and any subject having access to the registry) can easily re-identify the last record of the de-identified dataset as pertaining to *Janet*, disclosing also the fact that *Janet*'s most recent diagnosis is depression. Such linking attack is unfortunately a concrete threat to the protection of individual's identities: a uniqueness analysis over the US 2000 Census data has estimated that more than the 60% of the entire US population is *uniquely identifiable* by the simple combination of their gender, ZIP, and full date of birth [33].

The scientific community has devoted major efforts towards the definition of effective approaches for anonymizing data while ensuring their utility, according to different privacy requirements and definitions. A first definition of privacy (*syntactic definition*) captures, with a numerical value, the degree with which a dataset is protected. An example of a syntactic privacy definition can state that each release of data should be indistinguishably related to no less that a threshold number of individuals in the overall population (Section 2.1). Syntactic privacy definitions are at the basis of different approaches aimed at protecting both the identities of the individuals represented in a dataset, and their sensitive

SSN	**Name**	**BirthDate**	**Gender**	**ZIP**	**Diagnosis**
		1955/07/05	M	12311	Stroke
		1970/10/13	F	12333	Flu
		1955/07/10	M	12313	COVID-19
		1970/10/20	F	12331	Flu
		1970/06/15	F	12332	Cancer
		1955/07/20	M	12312	Asthma
		1970/06/03	F	12324	Cancer
		1970/06/28	F	12325	Dementia
		1970/10/13	F	12326	Flu
		1970/12/01	*F*	*12344*	Depression

(a)

Name	**Address**	**City**	**ZIP**	**BirthDate**	**Gender**
...	...	...	...	...	...
Janet	1100 Main Street	Portland	*12344*	*70/12/01*	*female*
...	...	...	...	...	...

(b)

Figure 2: A de-identified version of the dataset in Figure 1 (a) and an example of publicly available non de-identified (b) dataset.

information. A second definition of privacy (*semantic definition*) captures the protection that should be ensured by the analysis carried out on a dataset that includes information to be protected. An example of a semantic privacy definition can state that the result of an analysis on a released dataset must be insensitive to the insertion or deletion of a record in the dataset (Section 2.2). Semantic privacy definitions are at the basis of different protection approaches aimed at hiding the actual original informative content of a dataset, while ensuring adequate statistical results.

Protection approaches enforcing syntactic privacy definitions assume a precise definition of what information can operate as QI, and of what information is sensitive. These approaches then typically satisfy the privacy desideratum by modifying the QI with non-perturbative techniques, in such a way to guarantee data truthfulness. On the other hand, protection approaches enforcing semantic privacy definitions do not need such strict classification of information as QI and sensitive. They typically perturb data and thus do not guarantee their truthfulness. In the remainder of this section, we overview some of the most well-known protection approaches pursuing both syntactic and semantic privacy definitions.

SSN	**Name**	**BirthDate**	**Gender**	**ZIP**	**Diagnosis**
		1955/07/**	M	123**	Stroke
		1955/07/**	M	123**	COVID-19
		1955/07/**	M	123**	Asthma
		1970/10/**	F	123**	Flu
		1970/10/**	F	123**	Flu
		1970/10/**	F	123**	Flu
		1970/06/**	F	123**	Cancer
		1970/06/**	F	123**	Cancer
		1970/06/**	F	123**	Dementia

Figure 3: An example of 3-anonymous version of the dataset in Figure 2(a).

2.1 *k*-Anonymity

k-Anonymity is an anonymization approach pursuing a privacy requirement that demands that no release of data can be related to less than a certain number *k* of individuals [47]. *k*-Anonymity has been specifically designed to counteract the improper disclosure of individuals' identities through the QI-based linking attack illustrated in Figure 2. *k*-Anonymity is based on proper modifications to the QI (besides the complete removal of identifiers) to ensure that no piece of data in a dataset can be uniquely related to its respondent through her QI, and vice-versa. Practically, *k*-anonymity ensures that each combination of QI values in a dataset appears with at least *k* occurrences. In this way, each individual in any external data source could be mapped to 0 or at least *k* records in the anonymized dataset. This guarantee can be obtained in different ways. The original proposal of *k*-anonymity adopts data generalization (to the QI) and data suppression. Data generalization is a data protection technique, which replaces data values with other more general values. For instance, an individual's complete date of birth ⟨`year/month/day`⟩ can be generalized to ⟨`year/month`⟩, or just to ⟨`year`⟩. Since generalization (while maintaining data truthfulness) removes details from data, with reference to the QI-based linking attack in Figure 2, it is easy to see that the more generalized the QI, the less probable the risk of finding unique correspondences with external data sources. Data suppression is adopted by *k*-anonymity to reduce the amount of generalization that would otherwise be needed. For example, the dataset in Figure 3 represents a *k*-anonymous version of the dataset in Figure 2(a) with *k*=3. Note

SSN	Name	BirthDate	Gender	ZIP	Diagnosis
		1955/**/**	M	123**	Stroke
		1955/**/**	M	123**	COVID-19
		1955/**/**	M	123**	Asthma
		1970/**/**	F	123**	Flu
		1970/**/**	F	123**	Flu
		1970/**/**	F	123**	Flu
		1970/**/**	F	123**	Cancer
		1970/**/**	F	123**	Cancer
		1970/**/**	F	123**	Dementia
		1970/**/**	F	123**	Depression

Figure 4: An example of 3-anonymous version of the dataset in Figure 2(a) without suppression.

that the last record of the original dataset, pertaining to Janet, has been suppressed to reduce the amount of generalizaton adopted to ensure 3-anonymity: since Janet is the only respondent born in December 1970 in the original dataset, to obtain 3-anonymity (or, more precisely, k-anonymity with $k > 1$) while including her record it would have been necessary to generalize the date of birth removing also the month, as illustrated in Figure 4. Note also that such solution would collapse all the female records in a single equivalence class (i.e., the set of records sharing the same QI values). In these situations, it is better to remove a few outlier records if this can require less adoption of generalization.

The 3-anonymous dataset in Figure 3 has been obtained adopting suppression at the level of entire records, and generalization at the level of attributes (in such a way that all the values of a certain attribute are uniformly generalized). Clearly, both techniques could be adopted at different granularity levels also. Regardless of the granularity level, the more the generalization, the more the protection but also the more the information loss caused by the release of imprecise and incomplete information. The problem of computing a k-anonymous version of a dataset minimizing information loss is NP-hard, and several approaches have been proposed [21]. Alternative proposals achieve the k-anonymity requirement by using, instead of generalization, microaggregation [27], which selectively replaces data items with new ones and hence, unlike generalization, does not preserve data truthfulness. Microaggregation-based k-anonymity clusters records in groups of size at least k based on their similarity, and then replaces their QI values with a representative one computed through an aggregation operator (e.g., mean or median).

While a k-anonymous dataset effectively protects the identities of data respondents and is less vulnerable to the linking attack in Figure 2, it still may leak information not intended for disclosure on respondents' sensitive information. As an example, consider the 3-anonymous dataset in Figure 3, and suppose that a recipient knows that a target respondent is born on October 1970. The recipient can therefore see that the target respondent belongs to the second equivalence class in Figure 3, and that her most recent diagnosis is *flu*, even without discovering which is her actual record. This is due to the fact that all the records sharing QI value $\langle 1970/10/{*}{*}\rangle$ have the same value for the sensitive attribute Diagnosis. As another example, consider the last equivalence class. Respondents in this class have around 0.7 probability of having cancer, since 2 records out of 3 assume this value. If the actual probability of suffering from cancer in the overall population is (significantly) lower than 0.7, the simple fact that a target respondent is included in the last equivalence class can signal to a recipient an increase in the likelihood that such respondent is suffering from cancer. To counteract similar issues, the original definition of k-anonymity has been extended to the consideration of the sensitive values when clustering records in the equivalence classes for ensuring the k-anonymity requirement. Examples of well-known extended approaches are ℓ-diversity [44], which demands that each equivalence class contain at least ℓ well-represented values for the sensitive attribute (therefore counteracting the first issue mentioned above), and t-closeness [43], which demands that the distribution of the values of the sensitive attribute in equivalence classes be close to the distribution of the attribute in the overall table (therefore counteracting the latter issue mentioned above). For example, the dataset in Figure 5 is a 2-diverse (and 3-anonymous) version of the dataset in Figure 2(a), where each equivalence class counts at least ℓ=2 different values for the sensitive attribute Diagnosis. It is interesting to note that any approach devised for practically enforcing k-anonymity can be easily extended for ensuring ℓ-diversity and t-closeness, by considering the values assumed by the sensitive attribute.

SSN	Name	BirthDate	Gender	ZIP	Diagnosis
		1955/**/**	M	1231*	Stroke
		1955/**/**	M	1231*	COVID-19
		1955/**/**	M	1231*	Asthma
		1970/**/**	F	1233*	Flu
		1970/**/**	F	1233*	Flu
		1970/**/**	F	1233*	Cancer
		1970/**/**	F	1232*	Flu
		1970/**/**	F	1232*	Cancer
		1970/**/**	F	1232*	Dementia

Figure 5: An example of 3-anonymous and 2-diverse version of the dataset in Figure 2(a).

The generalization-based approaches illustrated above ensure a certain degree of unlinkability between an individual and her identity and/or sensitive information by hiding each individual in a crowd of at least other $k-1$ individuals that could possibly have her identity. Such protection comes at the price of producing imprecise or incomplete QI information. A different strategy, pursuing the same rationale of breaking the correspondence between an individual and her information but without generalizing QI, is represented by data fragmentation. With fragmentation, the original dataset is split in a set of vertical fragments defined over non-overlapping subsets of the original attributes, to break the correspondence between data that should not be visible together, such as the QI and the sensitive data. The sub-records in the different fragments are then clustered, and information on the association among clusters is released, so to hide the original precise associations between sub-records in coarser-level associations among clusters. This approach can be effectively adopted for enforcing the protection approaches illustrated above without resorting to generalization. For instance, ℓ-diversity can be achieved by clustering records in groups containing at least ℓ well-represented sensitive values, and then splitting the dataset in two fragments so that one contains the QI, and the other the sensitive attribute. Each sub-record in the fragments is enriched with the identifier of the cluster to which it belongs, so to release associations among groups [51]. Figure 6 illustrates an example of a fragmented version of the medical dataset in Figure 2(a) (again, after the removal of the last record) that satisfies ℓ-diversity with ℓ=2: it is easy to see that each respondent in the left-hand-side fragment can be asso-

SSN	Name	BirthDate	Gender	ZIP	ID
		1955/07/05	M	12311	1
		1955/07/10	M	12313	
		1955/07/20	M	12312	
		1970/10/13	F	12333	2
		1970/10/20	F	12331	
		1970/06/15	F	12332	
		1970/06/03	F	12324	3
		1970/06/28	F	12325	
		1970/10/13	F	12326	

ID	Diagnosis	Count
1	Stroke	1
	COVID-19	1
	Asthma	1
2	Flu	2
	Cancer	1
3	Cancer	1
	Dementia	1
	Flu	1

Figure 6: An example of a 3-diverse version of the dataset in Figure 2(a) with data fragmentation.

ciated with at least 2 different values for the sensitive attribute Diagnosis. Note that in the figure, in line with the original fragmentation-based approach [51], sensitive attribute values are reported in their fragment only once, accompanied by their number of occurrences (attribute Count).

Data fragmentation has been investigated also for protecting, besides the association between respondents' identities and their information, generic associations among data items when such associations are considered sensitive [2, 10, 12]. Intuitively, the original dataset can be split in a set of fragments so to break all sensitive associations, while ensuring that fragments are unlinkable by unauthorized subjects (so to avoid the possibility of reconstructing the broken associations, either directly or indirectly). Sub-records in the fragments can then be clustered, and associations among the clusters can be released [13].

2.2 Differential privacy

Differential privacy is an example of a protection approach that pursues a semantic privacy definition [28]. Unlikely the generalization-based and fragmentation-based approaches illustrated in Section 2.1, differential privacy does not guarantee the truthfulness of the protected data, since its protection guarantees come from the perturbation of either the original data, or the results of a computation. Differential privacy permits to perform computations over a dataset, without exposing data of the single individuals. The requirement pursued by differential privacy is that, given a dataset T and an analysis over it, the impact that the insertion of a record in T (and, similarly, of the removal of a record from T) can have on the result of the analysis is limited. More precisely, given two datasets T and T' differing only for one record, a randomized function $\mathcal{K}$ is said to satisfy *ϵ-differential privacy* if and only if $P(\mathcal{K}(T) \in S) \leq \exp(\epsilon) \cdot P(\mathcal{K}(T') \in S)$, with S a subset of the outputs of $\mathcal{K}$ and ϵ a privacy parameter (clearly, the lower the value of ϵ, the greater the protection offered). This guarantees that the contribution of the specific data of an individual to the computation results remains negligible and, therefore, that the privacy of that individuals is not exposed by her contribution to the dataset: should her withdraw her data from the dataset, the results of the computations would not be altered significantly. In other words, differential privacy limits the information gain that a recipient can have on the specific data of an individual. Such protection is obtained through the controlled perturbation (e.g., controlled injection of noise) in the released data.

With respect to the moment in which protection is enforced (i.e., when data are distorted), differential privacy can be enforced in a centralized scenario, where individuals contribute their original (unprotected) data to a central entity in charge of data collection, and protection is then enforced, by the central entity, interactively (i.e., adding noise to the results of analysis carried out on the original dataset) or non-interactively (i.e., producing a sanitized version of the original dataset). This scenario assumes that the central entity be trusted for accessing the original and unprotected data, as also implicitly assumed also by the generalization-based and fragmentation-based approaches illustrated in Section 2.1. Differential privacy can however also be enforced in a local scenario, based on the randomization of the individual pieces of data directly at the respondents' side, that is, before being collected (e.g., [6, 29]).

3 Data encryption

Data anonymization permits to sanitize data collections to be shared, publicly or selectively. Since the amounts of data that need managed in the context of smart cities can be extremely large, also storage can represent an issue. The availability of a multitude of specialized cloud providers, offering storage capabilities at competitive prices, represents a promising solution for data management. However, cloud providers are typically not considered fully trusted to access the content of the data they store, especially in the context of smart cities where (citizens') data can be sensitive, proprietary, or simply subject to access restrictions. For instance, health data collected and produced by `Hosp` for intelligent healthcare are sensitive and should not be improperly disclosed, even if outsourced to the cloud to relieve `Hosp` storage and management burden. A possible solution leverages owner-side encryption, by means of which a data collection is encrypted by its owner before being outsourced to a cloud provider. While effectively protecting data from improper observations by the cloud provider and by unauthorized subjects that do not know the encryption key, encryption can have an impact when the outsourced data are to be retrieved by authorized subjects. In the remainder of this section, we overview the problems of ensuring selective and fine-grained access to outsourced data.

3.1 Enabling selective access

The main problem of enforcing selective access to data that have been outsourced to the cloud is caused by the fact that neither the cloud provider (for security and/or trust reasons) nor the data owner (for performance/convenience reasons) can evaluate access requests. A possible solution consists in the outsourced data to *self-enforce* access restrictions, through *selective owner-side encryption* [16]. With selective owner-side encryption, different data items are encrypted by their owner (before being outsourced to the cloud) with different keys, which are then distributed to users in such a way that they can decrypt all and only the data items for which they are authorized. With such an approach, data are protected also against the cloud provider itself, which is not communicated any encryption key. To reduce the burden of key management for the users, selective owner-side encryption is typically accompanied by *key derivation* techniques, by means of which the value of an encryption key can be computed starting from the value of another encryption key and of a piece of

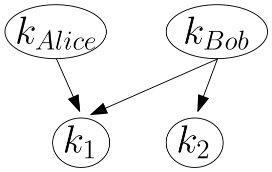
	r_1	r_2
Alice	1	0
Bob	1	1

Figure 7: An example of access matrix.

public information [4].

The derivation of a key k_y from another key k_x is enabled by a public token $t_{x,y}$ computed as $k_y \oplus h(k_x, l_y)$, with $\oplus$ the bitwise `xor` operator, h a deterministic non-invertible cryptographic function, and l_y a public information associated with k_y. A careful adoption of key derivation permits each user to manage a single key only, from which she can then retrieve all the keys of all the data items she is authorized to access. The overall key derivation process, regulating which user can derive which keys, is controlled by the data owner through the definition of appropriate key derivation structures correctly modeling the authorization policies [22]. For example, consider two users Alice and Bob and two resources (e.g., two data items) r_1 and r_2, and an authorization policy such that Alice can access r_1, and Bob can access r_1 and r_2 (as illustrated by the access matrix in Figure 7). With selective owner-side encryption and key derivation, a possible solution can require to encrypt r_1 and r_2 with keys k_1 and k_2, respectively, and to communicate to Alice and Bob encryption keys k_{Alice} and k_{Bob}, respectively. Figure 8 illustrates and example of a key derivation structure along with the catalog $\mathcal{T}$ storing the tokens enabling a possible derivation for enforcing the authorization policy. Nodes in the structure correspond to keys, and edges to tokens. An edge from k_a to k_b models the existence of a token enabling the derivation of k_b from k_a. For example, with reference to the structure in Figure 8, it is easy to see that Bob can derive both keys k_1 and k_2 for accessing the two resources he is authorized to access starting from his own key k_{Bob}, that is, managing a single key only.

Since access to a resource is granted with the knowledge of (or the possibility to derive) the key adopted for its encryption, changes in authorization policies should be reflected by a re-encryption of the involved resources, so that encryption correctly reflects the updated policy. Since the cloud provider is typically not trusted to access plaintext data, such an



$\mathcal{T}$

From	**To**	**Value**
k_{Alice}	k_1	$k_1 \oplus h(k_{Alice}, l_1)$
k_{Bob}	k_1	$k_1 \oplus h(k_{Bob}, l_1)$
k_{Bob}	k_2	$k_2 \oplus h(k_{Bob}, l_2)$

Figure 8: An example of key derivation structure and token catalog.

operation would in principle require the data owner to locally perform the re-encryption herself, and to upload to the cloud the re-encrypted resources. To reduce the burden for the owner, a possible solution is to adopt two independent layers of encryption over data items: one for enforcing the original policy, and a second one for enforcing updates to the policy [16]. If the cloud provider is trusted to correctly manage data (as commonly assumed in the *honest-but-curious* trust model), the management of such second layer of encryption can be delegated to the provider itself, which can then easily manage policy updates without accessing plaintext data (which remain always wrapped by the first layer of encryption). The token catalog is then updated to allow authorized users to derive the two keys used at the two levels for the resources they can access.

Selective owner-side encryption can also support dynamic scenarios of digital data markets, when data owners wish to make their data selectively available to others possibly receiving an incentive. Incentivizing owners to share, in a controlled way, their data is indeed relevant to smart cities, where analysis and computations of different data sources can fuel novel services to citizens. For examples, intelligent healthcare can benefit from more individuals sharing their health data and measurements. In these scenarios, selective encryption (adopted for protecting data against the storage provider as well as unauthorized external subjects) can be coupled with blockchain [23]. A possible solution is to publicly store on-chain the structure to be used for key derivation, in such a way to securely log the interactions among the subjects, and the possible incentives that the owners should receive for making their data selectively available [31].

The owner-side selective encryption approaches illustrated above typically use symmetric encryption. An alternative solution for enabling selective access with asymmetric encryption is represented by *attribute-based encryption* (e.g., [34, 40, 42, 49]). Similarly to selective encryption, also with attribute-based encryption authorizations are enforced by permitting a user to decrypt different data items. In particular, attribute-based encryption is based on the definition of *attributes*, relevant for the authorizations, associated with users and data items. The decryption of a data item by a certain user is permitted if that user's attributes match the attributes of the data item. Encryption keys are then generated accordingly so to allow the correct enforcement of the authorization policy.

3.2 Enabling fine-grained access

When data are outsourced to a cloud provider in encrypted form, and the cloud provider is not given access to the encryption keys, enabling fine-grained access (e.g., retrieving the name and age of the patients whose most recent diagnosis is flu and who live in a certain area) can represent an issue. In fact, the cloud provider cannot access plaintext data to retrieve only those data items that satisfy the conditions in the access request. Clearly, permitting only the retrieval of the entire dataset and perform the filtering at the requesting side is not a viable solution, due to its unacceptable overhead. The research community has therefore investigated the problem and proposed different solutions for (partly) delegating the evaluation of fine-grained access conditions directly at the provider side.

A first family of solutions is based on the definition of *indexes*. Indexes are metadata that can be associated with data by their owner before being outsourced. In the context of relational databases, indexes are represented as additional attributes added to the outsourced relation. Indexes are defined for attributes expected to be involved in query evaluation, and reflect the values assumed by the attributes for which they are defined without disclosing their values. Given a relation, the owner then outsources to the cloud an encrypted and indexed version. Figure 9 represents an encrypted and indexed version of the medical dataset of Figure 1. In this case, encryption is applied at the granularity level of records (i.e., each record is encrypted as a whole), and two indexes have been defined for attributes ZIP (I_Z)

encr_record	I_Z	I_D
j5as?$	ι	α
%grT6	λ	β
z12#	ι	γ
f*grTi_6	λ	β
lF=+	λ	ε
6#R_u	ι	ζ
1wp(yQ	κ	ε
yKu8$	κ	η
nfP*r;	κ	β
a%g_6	ι	ϑ

Figure 9: An example of an encrypted and indexed version of the dataset in Figure 1.

and Disease (I_D) Queries are translated for operating on indexes: since indexes are not sensitive, queries on them can be executed by the cloud provider itself, and the final user can possibly discard from the result only those records that have been included due to the imprecise evaluation over indexes (e.g., when more plaintext values are mapped to a same index value). Indexes can be defined in different ways, and can support different kinds of operations (e.g., to support range conditions over indexes, it is necessary that the mapping between original and index values correctly reflects the original ordering) [18, 36, 48]. For example, index I_Z is defined as a *bucket-based* index (a type of index that splits the domain of an attribute in buckets and associates each bucket with a label), such that ZIP values from 12310 to 12319 are mapped to index value ι, from 12320 to 12329 are mapped to index value κ, and from 12330 to 12339 are mapped to index value λ. Index I_D is instead an encryption-based index, which associates each attribute value v with an index value $i(v)$ obtained by applying a deterministic encryption function to v.

Alternative solutions are represented by the adoption of *searcheable* or *homomorphic* encryption schemes. Searcheable encryption schemes support the search, over encrypted data, of data whose decrypted version satisfies a certain condition (e.g., [25, 41]). To increase the efficiency of the search process, searcheable encryption schemes can tolerate a certain amount of information leakage [8, 26]. Homomorphic encryption schemes support the evaluation of operations over encrypted data without the need of decrypting them (e.g., [7]). Homomorphic encryption schemes can be classified depending on whether they support only a specific kind of operation (*partially* homomorphic schemes, such as El-Gamal scheme for multiplications, or Pailler scheme for additions), generic computations over data an arbitrary number of times (*fully* homomorphic schemes), or generic computations for a limited number of times (*somewhat* homomorphic schemes) [1, 32, 50].

Different kinds of encryption may also be used in combination in an onion-like approach, where each data item is encrypted with different layers of encryption, and each layer supports a specific operation on data. Inner layers support more operations, and are expected to provide less protection [45]. When an authorized user requires a given operation on a data item, then the encryption layers are removed (i.e., through decryption) until the one supporting the required operation is reached and the operation can be executed.

4 Collaborative computations

Combining different data sources and performing computations and analysis over them can generate knowledge. This is particularly relevant in the context of smart cities, characterized by the collection and production of huge amounts of heterogeneous and diverse data about citizens and their environments. For instance, health information collected through a variety of sensors may be usefully combined with mobility data and social interactions extracted from social media services to monitor and possibly control the diffusion of infectious diseases. Since computations involving different data sources can be intensive, a promising solution for mitigating their economic costs consists in delegating (parts of) the computation to cloud providers offering computation capabilities at competitive prices. However, the data involved in a computation can be sensitive or subject to access restrictions, and it is therefore necessary to ensure that the involvement of external subjects (i.e., the cloud providers) in the computation does not expose information not intended for disclosure. In the context of relational databases, several approaches have been proposed for enabling secure collaborative computations over data. Traditional solutions can be based on the definition of *views* (e.g., [35, 46]), representing portions of a dataset for which different subjects are authorized, or on *access patterns* (e.g., [3, 5]), restricting the data that can be accessed based on the input that is provided by a subject. Such solutions however only capture the information that is explicitly involved in the computation, while a computation may also leak information not explicitly visible. For instance, consider a relational query of the form SELECT Name, Surname FROM MedicalData WHERE Diagnosis=‘Cancer’. Its result, despite containing only patients’ names and surnames, implicitly contains information on their diseases as well, in terms of the fact that all returned patients suffer from cancer. For this reason, a subject not authorized to access patients’ diagnosis should not access the result of the query.

To permit collaborative computations with the selective involvement of non-fully trusted providers, while controlling both explicit as well as implicit information flows, recent proposals have built on the idea of keeping track, as the computation proceeds, of the operations that are being executed and of the implicit information flows they cause. This permits, for instance, to maintain information on the join conditions evaluated in the computation of a relation, which reveal the fact that the tuples surviving the join appeared in both the joined relations [17]. To enlarge the spectrum of the possible subjects that can collaborate for the computation including, for instance, providers that offer computational resources at competitive prices but are not fully trusted to access plaintext data, a recent proposal permits the data owners to specify different levels of visibility over their data: the classical yes/no visibility over a data item, and the *encrypted visibility*, meaning that the subject can only access an encrypted version of the data item [14]. The proposal in [14] enforces authorizations regulating the visibility over data by dynamically injecting, in the computation, on-the-fly encryption and decryption operations. In this way, depending on the subjects to which the different parts of the computation are delegated, visibility over plaintext data is dynamically disabled and enabled. A computation can then be distributed to the most convenient providers, for instance based on economic factors, while enforcing the authorization policies set by data owners. To counteract improper flows of information, the proposal in [14] evaluates the information flow enacted by the computation, considering for each step

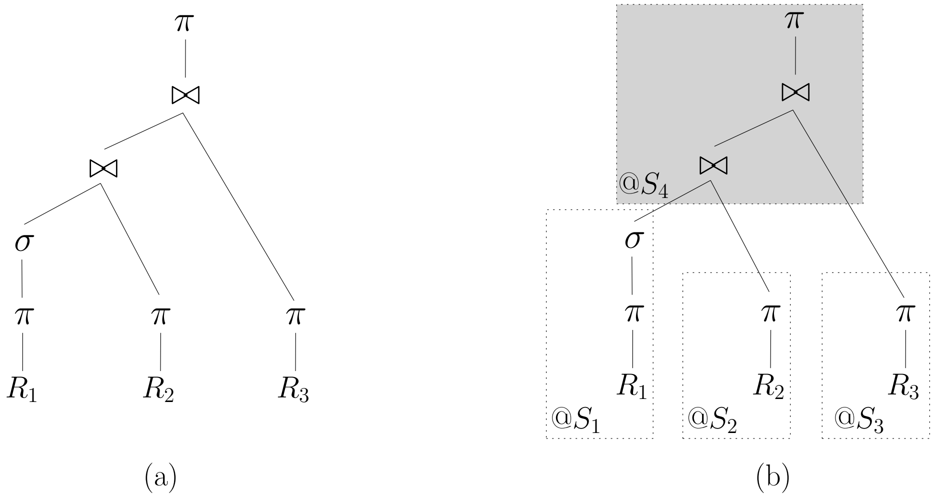


Figure 10: An example of a collaborative query execution with dynamic encryption

of the computation both its explicit and its implicit informative content, depending on the specific operations that have been executed until then. Authorizations are then enforced against both explicit and implicit information flows. Figure 10(a) illustrates and example of an execution plan for a query over three relations R_1, R_2 and R_3 (owned by three subjects S_1, S_2 and S_3 respectively), where nodes represent traditional operators of relational algebra (projection π, selection σ, join $\bowtie$). Assume the existence of a computational providers S_4, which offers computational resources at competitive prices but cannot access all involved data in plaintext. Figure 10(b) illustrates an example of assignment of query operations to the different subjects in the system (owners and the computational provider S_4), assuming that the computation of the joins and of the final projection is executed, by S_4, on encrypted data. This requires the injection of on-the-fly encryption operations on the input relations to the join operations. The encryption operations are performed by subjects S_1, S_2 and S_3 (which produce the relations to be encrypted). Note that if S_4 were not authorized to access the attribute(s) over which the selection over R_1 is executed, then it would not be allowed for the join computations, since such a selection would cause an implicit information flow leaking the value of the attribute(s) over which it is performed [14]. The proposal in [14] has been also extended to scenarios where the original relations are outsourced to non fully-trusted storage providers and hence stored in encrypted form (Section 3) [15].

Other aspects that have been investigated in distributed query evaluations include the possibility of permitting users to specify constraints on how the query should be evaluated, for instance, to specify that all join operations should be executed by a specific subject only (e.g., [30]), and cooperative executions of pre-determined computations with the help of trusted hardware components (e.g., [11]).

5 Summary

Anonymization and encryption are two of the main pillars that can be adopted in smart cities for protecting the privacy and security of data whenever they are shared or outsourced for storage or computation. In this chapter, we have considered the problems of anonymizing datasets and of encrypting them while permitting computations over them, and we have surveyed possible approaches for addressing them. We also have presented recent approaches for collaborative computations that can be enabled by data encryption for protecting visibility over sensitive data.

Acknowledgments

This work was supported in part by the EC within the H2020 Program under projects MOSAICrOWN and MARSAL, by the Italian Ministry of Research within the PRIN program under project HOPE, and by JPMorgan Chase & Co under project "k-anonymity for AR/VR and IoT/5G".

References

[1] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (CSUR)*, 51(4), 2018.

[2] 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.

[3] A. Amarilli and M. Benedikt. When can we answer queries using result-bounded data interfaces? In *Proc. of PODS 2018*, Houston, TX, USA, June 2018.

[4] M. Atallah, M. Blanton, N. Fazio, and K. Frikken. Dynamic and efficient key management for access hierarchies. *ACM Transactions on Information and System Security (TISSEC)*, 12(3):18:1–18:43, 2009.

[5] M. Benedikt, J. Leblay, and E. Tsamoura. Querying with access patterns and integrity constraints. *Proc. of the VLDB Endowment (PVLDB)*, 8(6):690–701, 2015.

[6] D. Bernau, J. Robl, P. W. Grassal, S. Schneider, and F. Kerschbaum. Comparing local and central differential privacy using membership inference attacks. In *Proc. of DBSec 2021*, Calgary, Canada, July 2021.

[7] D. Boneh, C. Gentry, S. Halevi, F. Wang, and D. J. Wu. Private database queries using somewhat homomorphic encryption. In *Proc. of ACNS 2013*, Banff, AB, Canada, June 2013.

[8] C. Bösch, P. Hartel, W. Jonker, and A. Peter. A survey of provably secure searchable encryption. *ACM Computing Surveys (CSUR)*, 47(2), 2014.

[9] T. Braun, B. C. Fung, F. Iqbal, and B. Shah. Security and privacy challenges in smart cities. *Sustainable Cities and Society*, 39:499–507, 2018.

[10] 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 Transactions on Information and System Security (TISSEC)*, 13(3):22:1–22:33, 2010.

[11] A. Dave, C. Leung, R. A. Popa, J. E. Gonzalez, and I. Stoica. Oblivious coopetitive analytics using hardware enclaves. In *Proc. of EuroSys 2020*, Heraklion, Greece, April 2020.

[12] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, G. Livraga, S. Paraboschi, and P. Samarati. Fragmentation in presence of data dependencies. *IEEE Transactions on Secure and Dependable Computing (TDSC)*, 11(6):510–523, 2014.

[13] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, G. Livraga, S. Paraboschi, and P. Samarati. Loose associations to increase utility in data publishing. *Journal of Computer Security (JCS)*, 23(1):59–88, 2015.

[14] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, G. Livraga, S. Paraboschi, and P. Samarati. An authorization model for multi-provider queries. *Proc. of the VLDB Endowment (PVLDB)*, 11(3):256–268, 2017.

[15] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, G. Livraga, S. Paraboschi, and P. Samarati. Distributed query evaluation over encrypted data. In *Proc. of DBSec 2021*, Calgary, Canada, July 2021.

[16] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Encryption policies for regulating access to outsourced data. *ACM Transactions on Database Systems (TODS)*, 35(2):12:1–12:46, 2010.

[17] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Authorization enforcement in distributed query evaluation. *Journal of Computer Security (JCS)*, 19(4):751–794, 2011.

[18] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. On information leakage by indexes over data fragments. In *Proc. of PrivDB 2013*, Brisbane, Australia, April 2013.

[19] S. De Capitani di Vimercati, S. Foresti, G. Livraga, V. Piuri, and P. Samarati. A fuzzy-based brokering service for cloud plan selection. *IEEE Systems Journal (ISJ)*, 13(4):4101–4109, 2019.

[20] S. De Capitani di Vimercati, S. Foresti, G. Livraga, V. Piuri, and P. Samarati. Supporting user requirements and preferences in cloud plan selection. *IEEE Transactions on Services Computing (TSC)*, 14(1):274–285, 2021.

[21] S. De Capitani di Vimercati, S. Foresti, G. Livraga, and P. Samarati. Data privacy: Definitions and techniques. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (IJUFKBS)*, 20(6):793–817, 2012.

[22] S. De Capitani di Vimercati, S. Foresti, G. Livraga, and P. Samarati. Practical techniques building on encryption for protecting and managing data in the cloud. In P. Ryan, D. Naccache, and J.-J. Quisquater, editors, *The New Codebreakers: Essays Dedicated to David Kahn on the Occasion of His 85th Birthday*. Springer, 2016.

[23] S. De Capitani di Vimercati, S. Foresti, G. Livraga, and P. Samarati. Empowering owners with control in digital data markets. In *Proc. of IEEE CLOUD 2019*, Milan, Italy, July 2019.

[24] S. De Capitani di Vimercati, A. Genovese, G. Livraga, V. Piuri, and F. Scotti. Privacy and security in environmental monitoring systems: Issues and solutions. In J. Vacca, editor, *Computer and Information Security Handbook, 2nd Edition*. Morgan Kaufmann, 2013.

[25] I. Demertzis, D. Papadopoulos, and C. Papamanthou. Searchable encryption with optimal locality: Achieving sublogarithmic read efficiency. In *Proc. of CRYPTO 2018*, Santa Barbara, CA, USA, August 2018.

[26] J. Domingo-Ferrer, O. Farràs, J. Ribes-González, and D. Sánchez. Privacy-preserving cloud computing on sensitive data: A survey of methods, products and challenges. *Computer Communications*, 140–141:38–60, 2019.

[27] J. Domingo-Ferrer and V. Torra. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Mining and Knowledge Discovery*, 11(2):195–212, 2005.

[28] C. Dwork. Differential privacy. In *Proc. of ICALP 2006*, Venice, Italy, July 2006.

[29] Ú. Erlingsson, V. Pihur, and A. Korolova. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Proc. of CCS 2014*, Scottsdale, AZ, USA, November 2014.

[30] N. Farnan, A. Lee, P. Chrysanthis, and T. Yu. PAQO: Preference-aware query optimization for decentralized database systems. In *Proc. of ICDE 2014*, Chicago, IL, March–April 2014.

[31] S. Foresti and G. Livraga. Selective owner-side encryption in digital data markets: Strategies for key derivation. In *Proc. of SECRYPT 2021*, virtual, July 2021.

[32] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proc. of STOC 2009*, Bethesda, MA, USA, May 2009.

[33] P. Golle. Revisiting the uniqueness of simple demographics in the us population. In *Proc. of WPES 2006*, Alexandria, VA, USA, October 2006.

[34] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proc. of CCS 2006*, Alexandria, VA, USA, October/November 2006.

[35] M. Guarnieri and D. Basin. Optimal security-aware query processing. *Proc. of the VLDB Endowment (PVLDB)*, 7(12):1307–1318, 2014.

[36] H. Hacigümüs, B. Iyer, S. Mehrotra, and C. Li. Executing SQL over encrypted data in the database-service-provider model. In *Proc. of SIGMOD 2002*, Madison, WI, USA, June 2002.

[37] R. Jhawar and V. Piuri. Fault tolerance and resilience in cloud computing environments. In J. Vacca, editor, *Computer and Information Security Handbook, 2nd Edition*. Morgan Kaufmann, 2013.

[38] 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.

[39] R. Jhawar, V. Piuri, and M. Santambrogio. Fault tolerance management in cloud computing: A system-level perspective. *IEEE Systems Journal*, 7(2):288–297, 2013.

[40] A. Lewko and B. Waters. Decentralizing attribute-based encryption. In K. G. Paterson, editor, *Proc. of EUROCRYPT 2011*, Tallin, Estonia, May 2011.

[41] J. Li, Y. Huang, Y. Wei, S. Lv, Z. Liu, C. Dong, and W. Lou. Searchable symmetric encryption with forward search privacy. *IEEE Transactions on Secure and Dependable Computing (TDSC)*, 18(1):460–474, 2021.

[42] J. Li, Y. Zhang, J. Ning, X. Huang, G. S. Poh, and D. Wang. Attribute based encryption with privacy protection and accountability for cloud IoT. *IEEE Transactions on Cloud Computing (TCC)*, 2020. pre-print.

[43] N. Li, T. Li, and S. Venkatasubramanian. t-Closeness: Privacy beyond k-anonymity and ℓ-diversity. In *Proc. of ICDE 2007*, Istanbul, Turkey, April 2007.

[44] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. *ACM TKDD*, 1(1):3:1–3:52, 2007.

[45] R. Popa, C. Redfield, N. Zeldovich, and H. Balakrishnan. CryptDB: Protecting confidentiality with encrypted query processing. In *Proc. of SOSP 2011*, Cascais, Portugal, October 2011.

[46] S. Rizvi, A. Mendelzon, S. Sudarshan, and P. Roy. Extending query rewriting techniques for fine-grained access control. In *Proc. of SIGMOD*, Paris, France, June 2004.

[47] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 13(6):1010–1027, 2001.

[48] H. Wang and L. Lakshmanan. Efficient secure query evaluation over encrypted XML databases. In *Proc. of VLDB 2006*, Seoul, Korea, September 2006.

[49] B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Proc. of PKC 2011*, Taormina, Italy, March 2011.

[50] A. Wood, K. Najarian, and D. Kahrobaei. Homomorphic encryption for machine learning in medicine and bioinformatics. *ACM Computing Surveys (CSUR)*, 53(4), 2020.

[51] X. Xiao and Y. Tao. Anatomy: Simple and effective privacy preservation. In *Proc. of VLDB 2006*, Seoul, Korea, September 2006.

[52] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. S. Shen. Security and privacy in smart city applications: Challenges and solutions. *IEEE Communications Magazine*, 55(1):122–129, 2017.