Privacy Enhancing Technologies for solving the Privacy-Personalization Paradox: Taxonomy and Survey

Nesrine Kaaniche[a,b,*], Maryline Laurent[b,c], Sana Belguith[d]

[a]*Department of Computer Science. The University of Sheffield, UK*
[b]*Member of the Chair Values and Policies of Personal Information, France*
[c]*SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, France*
[d]*School of Science, Engineering and Environment, University of Salford, Manchester, UK*

Abstract

Personal data are often collected and processed in a decentralized fashion, within different contexts. For instance, with the emergence of distributed applications, several providers are usually correlating their records, and providing personalized services to their clients. Collected data include geographical and indoor positions of users, their movement patterns as well as sensor-acquired data that may reveal users' physical conditions, habits and interests. Consequently, this may lead to undesired consequences such as unsolicited advertisement and even to discrimination and stalking. To mitigate privacy threats, several techniques emerged, referred to as **Privacy Enhancing Technologies**, PETs for short. On one hand, the increasing pressure on service providers to protect users' privacy resulted in PETs being adopted. One the other hand, service providers have built their business model on personalized services, e.g. targeted ads and news. The objective of the paper is then to identify which of the PETs have the potential to satisfy both usually divergent - economical and ethical - purposes. This paper identifies a taxonomy classifying eight categories of PETs into three groups, and for better clarity, it considers three categories of personalized services. After defining and presenting the main features of PETs with illustrative examples, the paper points out which PETs best fit each personalized service category.
Then, it discusses some of the inter-disciplinary privacy challenges that may slow down the adoption of these techniques, namely: technical, social, legal and economic concerns. Finally, it provides recommendations and highlights several research directions.

Keywords: Privacy Enhancing Technologies, recommendation services, web-search engines, pervasive applications, location-based services, profile-based services, cryptographic trends, secure communications,

*Corresponding author
Email address: n.kaaniche@sheffield.ac.uk (Nesrine Kaaniche)

anonymous certification, private information retrieval, secure multi-party computation, homomorphic encryption, trust models

1. Introduction

Technology's advances, namely the development of cloud infrastructures, Internet of Things (IoT) applications and social networks, have dramatically increased the amount of collected, stored and processed personal data. Indeed, the US International Data Corporation (IDC) predicts that worldwide data creation will increase to an enormous 163 zettabytes (ZB) by 2025 [1]. In fact, independently from users' attitudes towards technology, they inevitably release sensitive and personal information either by fulfilling a job application form, by purchasing a fidelity card, by passing through a toll-road or by using a credit card. In addition, the increasing number of connected objects is becoming trendy and is fiercely present in users' daily lives, from smartphones stuffed with applications exploiting location, time and other context parameters to home applications and from *smart* clothes to *connected* accessories (*i.e.,* self quantified devices) obtaining sensed data, such as wearable sensors monitoring fitness activities, sleep disorders as well as motion patterns.

In parallel, personalized services emerged. That is, users may enjoy a variety of targeted recommendations, such as points of interest, movies, books, doctors, practitioners, agencies, routes, $\cdots$, regrettably at the expense of their privacy. Thus, people today are living in the era where everything is connected, thus continuously generating, acquiring and processing a huge amount of data. These collected information may be improperly used, leading to undesired advertisement, identity theft or even discrimination, and several privacy breaches. Indeed, privacy, generally defined as *the right to be let alone* under the American law, is becoming more and more a common critical concern [2], mainly with successive revelations about the abuse of personal data collection and processing, starting from the US NSA spy program, revealed by E. Snowden in May 2013[1], until the last Cambridge Analytica and Facebook scandal disclosed in April 2018[2].

A growing number of users are concerned about the negative consequences of the massive collection of data, arising from the large-scale monitoring of individuals' life, in terms of human rights and societal values [3]. In 2014, a report titled *Big Data and Privacy*, is published by the White House to highlight the challenges for data protection, applied to the personal data collection by pervasive systems. In 2016, the European Union (EU) adopted a new General Data Protection Regulation (GDPR) which objective is to effectively ensure the protection of the data subject, *i.e.,* data owner [4]. In particular, the regulation clarifies the conditions under which it is compulsory to obtain the consent of the data subject before processing his personal data, especially for sensitive personal

[1] http://www.bbc.com/news/world-us-canada-23123964
[2] http://fortune.com/2018/04/10/facebook-cambridge-analytica-what-happened/

data and data relating to minors. The GDPR also introduces the new obligation of accountability for organizations (*i.e.,* data processors and data controllers). Indeed, each entity processing personal data must be able to demonstrate at any times that it is complying with the obligations laid down by the GDPR.

To meet these different requirements, several mechanisms, known as Privacy Enhancing Technologies (PETs), are gaining an expanding interest. They aim at proposing efficient privacy preserving algorithms, applications, systems and services in various domains and environments, following the Privacy by Design paradigm.

Objective of our Paper — This paper focuses on PETs to support personalized services. At first glance, privacy and personalization may look antagonistic, as privacy is in favor of the nondelivery of personal data while service personalization needs such data to fit users' profiles. Due to progress in technologies, *e.g.,* cryptographic functions, data mining and statistics, these apparently opposing needs can be satisfied through specific PETs.

Our paper classifies PETs into three different groups and eight categories, according to which entity is mainly involved in the privacy-preserving decision, *i.e.,* which entity is supporting the main cost for privacy and whether the channel between the client and the server is affected. As depicted in Figure 1, the first group, named as user-side techniques, requires end-user to manage his identity protection by himself by installing specific softwares to control attributes disclosure up-to the certification of attribute properties. User-side techniques include two main PETs categories namely, anti tracking technologies and privacy preserving certification, and two sub-categories, called data perturbation and Secure Multi-Party Computation (SMC), under the obfuscation and privacy preserving computation categories, respectively. The second group, referred to as server-side techniques, requires the server to be strongly involved in data processing either by anonymizing databases for data sharing or valorization, or by performing heavy computation over encrypted data at the request of customers. Server-side techniques contain two categories - the statistical disclosure control techniques and self-destructing systems, and two sub-categories, namely Private Information Retrieval (PIR) techniques and homomorphic encryption mechanisms, under the obfuscation and privacy preserving computation categories, respectively. It is worth noticing that the obfuscation and privacy preserving computation categories involve both user-side and server-side privacy enhancing techniques, deployed w.r.t. the system's design goals. The third group, named as channel-side techniques, relates to the quality of the channel between the user and the server - whether it is mediated and/or encrypted - or the quality of the exchanged data which can be voluntarily degraded. Channel-side techniques include secure communications and Trusted Third Party approaches. Each category is presented while defining the provided properties and supported techniques, tools and cryptographic mechanisms.

Existing Surveys — In the literature, a number of recent articles surveyed

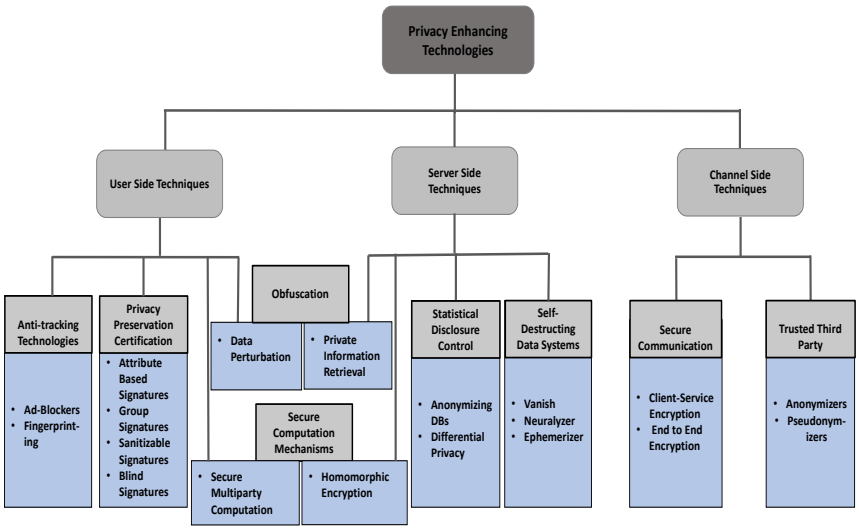

Figure 1: Taxonomy of Privacy Enhancing Technologies

privacy preserving solutions in diverse environments such as cloud computing [5, 6], fog computing [7] or the Internet of Things (IoT) [8] and more generally, in any type of computing [9, 10]. Others are focused on privacy challenges associated to specific domains, namely e-health applications [11], smart cities [12], pervasive systems [2], recommendation services [13] and cyber-security technologies [14].

Only few survey papers [15, 16, 17] covered privacy preserving techniques for personalized services. In 2007, Kobsa highlighted the tension between the privacy and personalization [17] and considered three main categories for privacy preserving personalized-web systems: the first is pseudonomous systems, the second is client-side personalization mechanisms while the third is called distribution, encrypted aggregation, perturbation and obfuscation solutions. [17] also presented useful information practices and discussed privacy laws, industry and company regulations that sparked a lot of debates during that period. Toch *et al.* [15] provided a detailed analysis of privacy risks while considering three different personalization techniques, namely social-based personalization, behavioral profile-based personalization and location-based personalization. The authors also reviewed privacy enhancing solutions for personalized information systems and proposed some research directions.
In [18], Parra *et al.* surveyed privacy metrics, mainly for the data-perturbation category, they showed that privacy preserving mechanisms generally impact the data utility and concluded that the problem of quantifying user privacy has to be more investigated. Later, the same authors presented a review on PETs and

metrics for personalized information systems [16]. The authors classified PETs into five groups, namely anti-tracking tools, cryptography-based techniques while emphasizing Private Information Retrieval (PIR) schemes, pseudonymous and anonymous certification systems, users collaboration approaches and data perturbation techniques.

Position of our paper — Compared to most closely related surveys, our paper provides a comprehensive overview of privacy challenges and requirements and highlights the main privacy-personalization trade-offs associated to different specific domains and use cases, namely recommendation systems, pervasive services and web search engines. In addition, a classification of PETs into eight categories is proposed, as shown in Figure 1. For each category, the paper describes the provided properties and reviews recent research proposals and industrial solutions for appropriate use cases. Furthermore, relying on a detailed comparison between reviewed PETs, we provide an accurate interdisciplinary discussion, including technical, legal, social and economic aspects, w.r.t. new emerging EU regulations. Finally,we identify several research directions and technical trends, namely *Artificial Intelligence* AI-based algorithms, known as data intensive algorithms, and *Blockchain-based systems* that might be of interest for the privacy preserving personalized-services, but with some adaptations to fit the privacy engineering principles.

Paper organization — Section 2 identifies privacy issues and requirements and defines three main trust models; and Section 3 gives a categorization of the main personalized services and applications while highlighting their features. Categorized services include recommendation services, pervasive applications and web-search engines. Section 4, Section 5 and Section 6 introduce and review user-side, server-side and channel-based privacy preserving techniques, respectively. Section 7 provides a detailed comparison between reviewed PETs w.r.t. involved approaches, supported trust and architectural models, main drawbacks and suitable applications. Section 8 discusses privacy open issues and research challenges, while considering the multi-disciplinary approach, before concluding in Section 9.

2. Privacy Issues and Requirements, and Trust Models for Personalized Services

Pervasive and mobile applications collect a large amount of personal information to provide customized services to their clients, hence they also carry a great potential of privacy risks w.r.t. gathered data usage and access. In the following, we first introduce privacy risks, in subsection 2.1, privacy requirements in subsection 2.2 and supported trust models in subsection 2.3.

2.1. Privacy Risks

Privacy risks are mainly related to environments, technologies and involved parties. In [15], Toch *et al.* introduced a detailed analysis of privacy risks that

are associated to prominent personalization techniques. Indeed, as pointed out in a recent report of the European Network and Information Security Agency (ENISA) [19], understanding privacy concerns from a technical point of a view, leads to identify:

(a) collected/processed data that are released and may be considered as sensitive, personal and identifying data,

(b) data that may be used to identify and/or revoke the anonymity of a user,

(c) possible adversaries (*i.e.,* entities that may gain access to personal information) which can rely on:

- data being transferred and processed that the adversary has access to,
- external and background knowledge of the adversary,
- possible collusion with some other actors.

For instance, for health care and well being applications, a large amount of commonly considered sensitive identifying information are collected, stored and processed. Obviously, the wearable IoT device has a limitless potential to improve the daily life, by allowing the collection of health information (*e.g.,* with new FitBit/Vivosport devices recording basic health information). The gathered data can be used to identify disease correlations and support new treatment options as well as remotely monitor the process of the treatment, however, these benefits are counterbalanced by the privacy challenges, as the precise identification of a particular user and his behavioral patterns is a growing concern.

In addition, with the help of big data analytics and the emergence of AI-based personalized applications, the accumulated raw data are highly valuable as specific patterns can be extracted, while eminently increasing the privacy risks of identifying users. In fact, adversaries' aim consists on obtaining private information about the model's training data or the target model [20, 21]. Attacks targeting users' privacy include (i) inferring whether input samples were used to train the target model (*i.e.,* membership inference attacks [21]), (ii) learning global properties of training data (*i.e.,* property inference attacks), or (iii) covert channel model training attacks. Attacks targeting the model privacy include uncovering the model details and inferring hyperparameters [22, 20].

2.2. Privacy Requirements

Referring to [23], privacy preservation requirements are defined as follows:

- **anonymity** — it means the ability of the user to access a resource or service without disclosing his identity to third parties. That is, the anonymity of a user means that he is not identifiable within a set of subjects, known as the anonymity set.
  Several levels of anonymity have been defined in the literature, ranging

from complete anonymity (*i.e.,* no one can reveal the identity of the user) to pseudo-anonymity (*i.e.,* the identity is generally not known but can be disclosed if necessary) to pseudonymity (*i.e.,* multiple virtual identities can be created and used in different settings).

- **data minimization** — it is a fundamental feature of privacy preservation and a GDPR requirement. It requires that service providers collect and process the minimum amount of information, needed for appropriate execution of a service or a particular transaction. The goal is to minimize the amount of collected personal information by service providers, for instance, to reduce the risk of profiling and tracking users.

- **unlinkability** — this property is essential for user privacy support and is closely related to the anonymity property. Unlinkability of two or more Items of Interest (IoIs, e.g., users, messages, actions, $\cdots$) from an attacker's perspective means that within the system (comprising these and possibly other items), the attacker cannot sufficiently distinguish whether these IOIs are related or not. Unlinkability is divided into two properties *issue-show unlinkability* and *multi-show unlinkability*, as follows:

  - the issue-show unlinkability ensures that any information gathered during users credentials' issuing cannot be used later to link the proof of identity to the original credential, during the authentication process;
  - the multi-show unlinkability guarantees that several presentation tokens derived from the same credential and transmitted over several sessions can not be linked by the service provider.

- **unobservability** — this property means the undetectability of a user[3] against all users uninvolved in an Item of Interest (IoI) and its anonymity even against the other user(s) involved in that IoI. That is, a user can use a resource or a service, without being noticed by others. Unobservability also requires that third parties cannot determine if an operation is running.

2.3. Trust Models

Within the sphere of information systems' security, the concept of trust is interpreted as a relation among the different entities that participate in various protocols. Trust relations are based on evidence established by the previous interactions of actors within a protocol.
In this survey, we consider three main trust models, focused on data owners w.r.t. their relations and interactions with other involved actors, defined as follows:

[3]Undetectability of an Item of Interest (IoI) from an attacker's perspective means that the attacker cannot sufficiently distinguish whether it exists or not.

- **trusted model** — users trust an external entity, generally known as a Trusted Third Party (TTP) which is in charge of protecting their sensitive data. Anonymizers and pseudonymizers are the main examples of trusted model architectures (*cf.* subsection 6.2). The idea behind these TTP-based approaches is conceptually simple such that all exchanged messages between the communicating entities should go through the central entity. Their main drawbacks are that they come at the cost of infrastructure and suppose that users are willing to trust other entities. Moreover, third parties might eventually be forced, for instance by authorities, to reveal the collected sensitive information they have access to.

- **untrusted model** — users mistrust any of the involved actors. As users just trust themselves, it is their own responsibility to protect their privacy. Examples of mechanisms relying on the assumptions of the untrusted model are SMC mechanisms and other techniques relying on data perturbation and operating on the user side. That is, users need not trust any entity but privacy protection comes at the cost of system functionality, performances and data utility.

- **semi-trusted model** — trust is distributed among the set of entities, involved in the execution of the protocols. In this model, the data owner does not totally trust other peers, *i.e.,* other users, the service provider, *etc.*. It is assumed that those peers are honest as they generate accurate inputs or outputs and perform calculations properly, during the different steps of the protocol. However, they may be curious to gain extra data from the protocol, such as obtaining credentials/attributes of a data user, retrieving an encrypted data content, or distinguishing the data owner based on several interactions. Several techniques are considered under the semi-trusted model, such as Private Information Retrieval (PIR) solutions and Statistical Disclosure Control techniques.

3. Categorization of Personalized Services

To better understand how PET techniques better fit services needs, we first categorize the main personalized services with some of their features, like the personalization techniques, the possible adversaries and the type of gathered data, as shown in Table 1. Subsection 3.1 below introduces different application categories, their characteristics, specific privacy threats and requirements. Subsection 3.2 details the personalization based approaches, namely location-based, profile-based and content-based techniques.

3.1. Personalized Services, Applications and Systems

As shown in Table 1, three main categories of personalized services are distinguished and presented below, *i.e.,* recommendation services, web search engines and pervasive applications.

Table 1: The Main Personalized Services Categories and Features

Category	Services	Techniques	Adversaries	Collected Data
recommendation services	online advertisement [25]	profile-based [24] location-based [26] content-based [27]	service providers merchants	profile location interests
web search	personalized web search [29]	profile-based [28] location-based [30]	service providers third parties	profile location queries
pervasive applications	geo-social network applications [31]	location-based [32] profile-based [33]	service providers other users	location profile, interests
	health care and well-being applications [35]	location-based [34] profile-based [36]	service providers city equipments	location, activities sensed data
	vehicular applications and smart cities [38]	location-based [37] profile-based [39]	other drivers city equipments road authorities	location, movements' trace driving behaviour users' habits

3.1.1. Personalized Recommendation Services

These services enable users to receive diverse personalized recommendations with respect to their close interests and profile's attributes. All along, personalized recommendation systems are becoming effective revenue for online business. That is, as discussed in [40], personalized recommendations are mainly based on personal information analysis and they are known to be efficient as 35% of the consumers are purchasing on Amazon thanks to targeted recommendations, and 75% of watched contents are recommended on Netflix.
The system model of recommendation services relies on two main entities: the user and the recommender [13]. User's personal data are generally stored on his local device (*i.e.,* smartphone). The recommender collects these data, analyses and processes the gathered data by creating users' profiles, interests and preferences, in order to provide accurate recommendations. These recommendations may be sent by messages or shown via pop-up windows.

Although these systems are very efficient and useful for each involved entity, they give rise to several privacy concerns:

(a) collected personal data are inconveniently exposed to the recommending entity, as they disclose their personal interests and profiles,

(b) collected personal data may be misused by the recommender, which can for instance, sell these data without the user's consent [41],

(c) provided data may be disclosed or gathered by attackers, taking advantage of some vulnerabilities at the recommender's side.

Because users need to reveal information in order to make use of desired functions of recommendation systems, the idea consists on resolving the hard -appropriate- balance between utility and users' privacy. Providing accurate recommendations remains the main objective of these systems, however massively collecting personal information may also lead to -irreversible- privacy breaches.

3.1.2. Personalized Web Search Services

Personalized web search aims at providing customized search results tailored to users' interests and general profiles, at the expense of personal users' data collection and analysis. That is, data analysis permits to understand the users' intent behind issued queries, for improving the search quality, known as the Quality of Experience (QoE) [42].
Personalized Web search solutions are generally categorized into two categories, namely click-log based methods and profile based methods. The methods based on the click-log are simple: they mainly rely on the clicked pages based on the user's query history. Although it has been shown that this strategy works consistently [43], it can only work on repeated requests from the same user. Conversely, the profile-based methods enhance the search experience with complex user-interest models generated from user profiling techniques.

Once again, privacy concerns arise as personal data are heavily and implicitly collected by Web Search Engines (WSEs):

(a) data leakage, for instance, as already happened with the American OnLine (AOL) query logs [44], may both regress users' trust in their providers impeding them from using several services, and also mitigate WSE providers' enthusiasm for providing personalized services, thus harming the whole business model,

(b) collected personal data may be misused by the service provider, for instance, this latter may sell these data without the consent of the user.

Privacy risks have become the main issue to a large proliferation of *personalized* search services and to the emergence of privacy-preserving WSE initiatives, *e.g.,* Qwant [4], DuckDuckGo [5], ... To protect users' privacy in personalized WSEs, it is important to address the trade-off between the improvement of the quality of the search with users' profile customization utility while hiding and preventing massive sensitive data collection by service providers.

3.1.3. Personalized Pervasive Services

Pervasive systems include a large variety of applications and services, including geo-social network applications, vehicular applications, smart cities' services and health care and well-being applications. Most of these services are personalized w.r.t. the user movements and precise location. Indeed, end-users are interested in sharing their location data in terms of check-in based mechanisms, for example with Foursquare or Facebook places, or for continuous location reporting.

[4] qwant.com
[5] duckduckgo.com

Let us emphasize that location and time are the main parameters that permit to precisely determine the context of an interaction. That is, the association of a user with a specific location at a given time can reveal affiliations, trace movements, deduce habits, interests or even religious beliefs and/or health problems. Hence, several privacy threats have to be considered:

(a) collected location data permit service providers to trace users' movements and determine frequently attended places or even guess places that users are unlikely visiting (*i.e.,* not preferred places). The location data may provide interesting information to curious service providers, such that they may conclude associating events and when/where users have been together, possibly the frequency and the duration.

(b) releasing personal data (mainly collected from body sensors) to a third party, without the user consent, undesirably disclose a lot of information about an individual's physical condition or even lead to discrimination, for instance by insurance companies [45],

(c) provided data may be disclosed or gathered by external attackers or intruders, mainly while considering that several pervasive mechanisms collect more than strictly needed information for each specific application.

3.2. Personalization based approaches

Personalization techniques combine ideas from user profiling, information retrieval, artificial intelligence and user interface design, in order to provide customized services to end-users. Hereafter, we briefly discuss two main personalization techniques, namely user profile-based, location-based and content-based techniques.

3.2.1. Personalized profile-based approaches

Generally, a user submits queries to a WSE, clicks on news links in a personalized news recommendation system, and assigns tags to resources (*i.e.,* photos, bookmarks, $\cdots$), w.r.t. his profile and interests. These information (queries, clicks, tags) permit the service provider to extract a fine-grained user profile [18].
Users' profiles are generally constructed and modeled based on histograms [18]. For instance, for the Google News platform, news are classified into a predefined set of topics. Subsequently, users are modeled based on their distribution of clicks within the predefined sets.

3.2.2. Personalized location-based approaches

Based on spatial and temporal data, location-based techniques are becoming a leading factor for online business. Meanwhile, there exist two different categories for location-based approaches, namely for providing elementary services and derivative services [46].

Elementary services are typically the navigation and search of Point of Interest (POIs) services. They require continuous reporting of geographical positions for guiding users. This is the main function of several pervasive systems, like geo-social networks, and applications such as Google Maps[6], Baidu Maps[7], Waze[8] and AutoNavi Map[9].

Derivative services consist mainly of tracking, mobile commerce (*i.e.,* recommendation services), location-based games and social location-based mobile applications.

For tracking purposes, the location-based personalization approach records geographic positions of users and objects and provides trajectories and moves expressed by an ordered sequence of respective locations (*e.g.,* CityMapper[10]). Second, for mobile advertisement based on location, several examples emerged, such as location-sensitive advertising or billing (*e.g.,* Groupon[11] and AdNear[12] applications): location based advertisement pushes ads to users associated to their exact geographical position while location-based billing allows users to consume economically, based on different rating zones, hence, they are charged differently based on their location. Finally, new favorites of young people emerged, namely location-based games, such as PokemonGo[13] and social, location-based applications such as Tinder[14].

3.2.3. Personalized content-based approaches

As shown in Table 1, content-based approaches are mainly used for personalized recommendation systems. They generally consist on recommending items while comparing user's personal preferences and behaviours with the properties of items. For instance, to propose personalized ads, the advertisement company, *i.e.,* recommender, may compare the keywords associated with ads, with the keywords pointing out the user's preferences. Thus, to obtain these personalized recommendations, users are generally required to provide their personal information, *i.e.,* preferences, location, *etc.* to the recommender for analysis and processing.

4. User Side Techniques

User-side privacy preserving techniques include mechanisms that require the involvement of the end-user to protect his own identity and personal data. Indeed, installing anti-tracking softwares, cooperating with other users or adopting

[6]https://www.google.com/maps
[7]https://map.baidu.com/
[8]https://www.waze.com/fr/
[9]http://www.autonavi.com/
[10]https://citymapper.com/paris
[11]https://www.groupon.fr/
[12]https://near.co/
[13]https://pokemongolive.com/en/
[14]https://tinder.com/?lang=en

a privacy preserving certification system are the main user-oriented PETs, that permit users to protect their privacy. In the following, we first introduce basic anti-tracking technologies (subsection 4.1). Then, we present privacy preservation certification systems (subsection 4.2). Afterwards, data perturbation and secure multi party computation mechanisms are detailed in subsection 4.3 and subsection 4.4, respectively.

4.1. Basic Anti-tracking Technologies for Online Services

In this section, we present tracking tools and review anti-tracking technologies for the application level.

4.1.1. Tracking Tools

Web tracking is a technique that performs certain tasks such as content personalization, user authentication or online advertising. Its main goal is to improve the user experience and maintain an Internet economy as highlighted by most of the Internet advertising companies. Web tracking techniques can be stateful or stateless, depending on the type of data that are required by the website and stored at the client side.

- **Stateful tracking** — it mainly relies on cookies to store information on users' computers. That is, a cookie is a thin file containing user information such as hashed authentication information, the last visited pages, *etc.*. A cookie has an expiration date after which it is deleted from the user's computer. Its lifetime varies but generally is considered as sufficiently long. In addition, once a user visits a website, the lifetime of stored cookies is reset to the default duration. This way, websites can maintain the browsing state that otherwise would be lost. Figure 2 shows the main interactions between the user's browser and the web service provider, to manage first-party cookies.
  Nowadays, websites may abuse the stateful nature and advantages provided by cookies. In fact, websites involve several resources, each one able to generate cookies. Some of these resources are not necessarily owned by the hosting website, and these external resources may also include cookies. Indeed, they can certainly gather the user's browsing history and profile her browsing habits. For instance, a website *bogus.com* includes a resource like an image or a link-list imported from a third-party domain like *ad.com*. When a user visits *bogus.com*, the third-party resource generates a cookie that is stored on the user's computer. As such, when the same user visits another website *too-bogus.com* which uses the same *ad.com* resource as *bogus.com*, the third party *ad.com* recognizes him, as the related cookie is already stored in his computer.
- **Stateless tracking and fingerprinting methods** — this method does not require to store data on users' computers. The fingerprint-based mechanisms mainly rely on extracted properties of users' browsers. These

powerful tracking tools are increasingly used by websites, as they permit advertising companies to bypass the private browsing mode as well as cookie-related regulation in Europe [4]. Fingerprinting techniques are even more informative to advertising companies than third-party cookies. They can complete a user profile for identification purpose, even across several advertising services by sharing fingerprint data between different services. Thanks to the quick evolution of JavaScript and the huge number of devices that interpret this language, websites push scripts to the browser to inspect available and accessible resources of the browser.

4.1.2. Anti-tracking Tools

For the application layer, three aspects of tracking techniques can be counteracted: cookies in general, Java and Flash scripts, and fingerprinting methods.

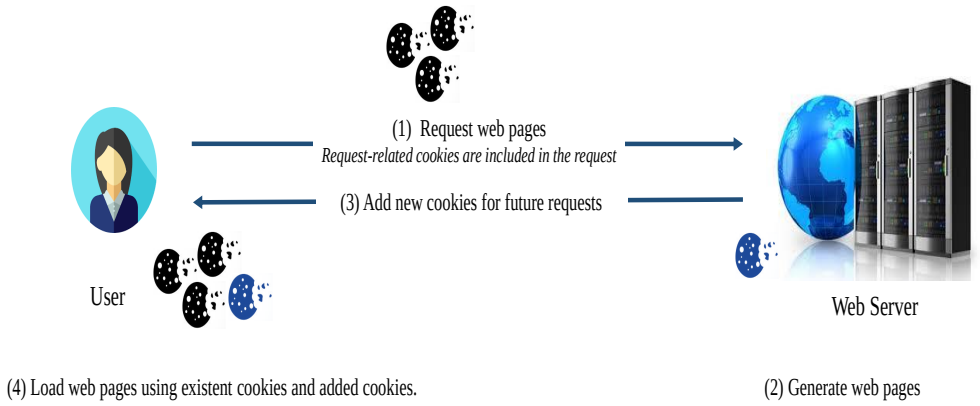

Figure 2: First-party Cookie Flow Diagram

- **Anti-cookies based techniques** — cookies can be directly managed at the user browser or by using a cookie management tool. For instance, a privacy mode can be activated in the browser leading to disabled browsing history and reduced local traces (Google Chrome: private mode, Firefox: private mode, IE: InPrivate). In addition, among browsers, the Do Not Track (DNT)[15] header can be deployed as a Hypertext Transfer Protocol (HTTP) header selected field. It ensures that a web application disables either its tracking or cross-site user tracking (the ambiguity remains unresolved) of a requesting end-user. As this technology is not yet standardized, there is no technical measure to prevent curious web service providers from maintaining end-users' tracking. In fact, it is up-to the

[15] https://donottrack-doc.com/

website provider to perform honest processing with regard to the DNT header. Therefore, more efficient techniques have to be used like browser extensions for preventing cookies, namely third party cookies, Flash cookies *etc.*. Hereafter, we review most-deployed tools.

First, the Privacy Badger[16], introduced by the Electronic Frontier Foundation (EFF)[17] prevents third party websites from tracking the user, based on the number of first-party websites embedding third party cookies. Once a third party domain is identified as tracking end-users across different sites, it may be blocked. Blocking is launched for instance based on the number of times the user meets the same tracking domain during his browsing activities.
Furthermore, AdBlock Plus[18] and uBlock[19] block all undesirable ads, pop-up ads that are embedded by default, for example, video ads on YouTube, Facebook ads or even flashy banners. Both of them are using regular expressions (RegExp for short) to match against the URLs to contact. If the match is positive, then the URL is blocked. AdBlock Plus is also able to identify acceptable ads. It can be configured to block domains known to spread malwares or to disable any known tracking tools. It also removes all the social media buttons from each website.

The Disconnect tracker[20] provides Private Browsing and Private Search browser extensions for several web browsers such as Firefox, Chrome, Opera, *etc.*. The Private Search feature prevents search engines from tracking users' requests, while Private Browsing is responsible for blocking trackers by classifying them in blacklists. The Disconnect tracker provides a user-friendly approach of blocking trackers by only enabling necessary requests used for loading content. Indeed, Disconnect detects trackers based on the number of requests they have queried for each user, and displays them in one of four categories: advertising, analytics, social and content. Users can re-enable a tracker or whitelist a website manually using the dashboard in the upper right corner of the Web browser. Disconnect maintains a database of trackers by crawling popular websites for third-party requests, and then categorizing those requests by type.
Finally, the Ghostery[21] tool blocks tracking cookies installed by default in the user's browser. It also enables its users to detect and control JavaScript "tags" and "trackers". JavaScript beacons are embedded in many web pages, largely invisible to the user, allowing collection of the user's browsing habits via HTTP cookies, as well as participating in more sophisticated

[16] https://www.eff.org/privacybadger
[17] https://www.eff.org/
[18] https://adblockplus.org/
[19] https://www.ublock.org/
[20] https://disconnect.me/
[21] https://www.ghostery.com/

forms of tracking such as Canvas fingerprinting[22].

- **Disabling Java and Flash scripts** — several anti-tracking techniques, mainly browser extensions have emerged to block JavaScript and Flash scripts. Let us emphasize that fewer than 5% of worldwide websites are still using Flash scripts and most websites are going ahead of its complete retirement[23].
  NoScript[24] and ScriptSafe are able to disable all scripts' types, such as JavaScript, Java, Flash, *etc.*, from running on visited pages, if they are not added manually to a whitelist, by the corresponding user. Instead, Flashblock[25] can only block Flash content.

- **Counteracting fingerprinting-based techniques** — the goal of these techniques is to share the same fingerprint among several users, in order to blend in with the masses. For this purpose, it is important to efficiently determine the fingerprint surface [47], to be able to define a shared fingerprint. That is, fingerprinting can be performed both in a passive and active form. On one side, the passive fingerprinting uses attributes deduced from the communication such as the order of HTTP headers. On the other side, the active fingerprinting run attribute-gathering scripts on the client-side (*e.g.,* JavaScript engine speed). Consequently, anti-fingerprinting tools have to take into consideration both forms. Note that the use of browser extensions such as Disconnect can block cookies but at the same time make users more unique, thus, more identifiable w.r.t. fingerprinting tracking techniques.
  First, using a tool to disable the execution of Javascripts like NoScript generally protects users against JavaScript-based fingerprinting techniques. Another technique against fingerprinting is to use a cookie-blocker like uBlock to prevent the fingerprint system from accessing some relevant information, such that attributes inherent in the communication. Finally, the best method to have an effective fingerprint without decreasing the anti-tracking status with less browser extensions, is to use The Onion Router (TOR for short) and NoScript with Tails, as Tails provides exactly the same environment to its users (with TOR and NoScript already installed).

[22] Canvas fingerprinting is a type of *browser fingerprinting* techniques for tracking online users. It allows websites to uniquely identify and track users using HTML5 canvas elements instead of browser cookies.
[23] https://www.alphr.com/software/1009184/adobe-flash-dead-websites-ditch-software
[24] https://noscript.net/
[25] https://flashblock.en.softonic.com/

Table 2: Summary of Main Tracker-Blocker Tools

Tool	Approach	Blocking	Effectiveness	Firefox Users
Ghostery	domain-based	javascript beacons tracking cookies	(+) good protection	1.1 M
Disconnect	request number-based whitelist	trackers	(+) acceptable protection (-) fail to block some very popular trackers	0.2 M
Privacy Badger	behaviour-based	third party trackers	(-) ineffective with fresh browser installation	0.5 M
AdBlock Plus	RegExp-based blacklist	ads pop-up ads	(+) acceptable protection	11 M
uBlock	RegExp-based	ads pop-up ads	(+) acceptable protection (-) fail to block some very popular trackers	4.9 M
Request Policy	cross-site blacklist	all third parties	(+) good protection (-) break of web-page rendering	0.09 M

where RegExp refers to Regular Expression and (+) and (-) represent the advantages and drawbacks of each tracker-blocker tool.

4.1.3. Comparison between Ad-blocker tools

This section gives a comparison between previously presented tracker blocker tools with respect to the effectiveness and Quality of Experience (QoE) criteria. Referring to the [48] benchmark study conducted in 2017, we consider both privacy-protection and performance requirements. In [48], different plug-ins and setups are used as a testbed to browse regular web pages while collecting navigation data. Each configuration contains one single tracker-blocker from AdBlock, Privacy Badger, Blur, Disconnect, uBlock, Ghostery and Request Policy[26] tools, while one configuration, referred to as *Plain* does not contain any modified setup. Measurement results show that the picture is very diverse, with no plug-in being able to guarantee complete protection while improving performance as promised.

By considering the different experimental setups, it is noticeable that the majority of considered web sites do ignore the European Directive [4]. The directive forbids web services from installing tracking and profiling cookies before explicit consent is given by the user, but as far as we know, this is not observed for most of the services. Based on [48], Table 2 presents a comparison between 6 different tracker-blocker tools, with respect to several quantitative and qualitative criteria. That is, by Approach, we mean the approach used to detect and block trackers, including domain-based, Regular Expression based (RegExp for short), behaviour-based and cross-site based blacklisting. By Blocking, we introduce resources that may be blocked by each tool, such as: trackers, ads and third parties. By Effectiveness, we show the advantages and drawbacks of each reviewed tool, w.r.t. trade-off between users' privacy preservation and efficiency. By Firefox Users, we provide the number of Firefox users installing the anti-trackers[27].

From Table 2, it is noticeable that Request Policy tool is offering a good protection. That is, it blocks *all* third party contents, indiscriminately, thus

[26] https://www.requestpolicy.com/

[27] https://addons.mozilla.org/fr/firefox/, February 2019

breaking the web-page rendering. Similarly, Ghostery considered as a handy Chrome extension/Firefox add-on permits to provide a good protection for end-users. As stated above, Ghostery removes intrusive ads and blocks trackers, resulting in a faster browsing experience.

On the other side, Privacy Badger, as an open source browser extension, blocks more generic behaviors. It ensures a balanced approach between end-users and content providers by blocking advertisements and tracking cookies that do not respect the Do Not Track setting in a user's web browser. That is, Privacy Badger only blocks those ads which come with embedded trackers. Unfortunately, its main blocking algorithm used to identify trackers is ineffective with fresh browser installation. Separately, AdBlock Plus, is the most widely used tool by Firefox users. It provides an acceptable protection, by using regular expressions to match against URLs. Whenever the match is positive, the request is blocked. uBlock and Disconnect also provide an acceptable level of privacy protection. However, they fail to detect known trackers.

4.2. Privacy Preserving Certification

Privacy preserving certification, also known by privacy preserving attribute based credentials (AC), are cryptographic mechanisms that allow users to obtain certified credentials for their attributes from trusted issuers, and later derive presentation tokens that reveal only required information satisfying service providers' predicates.

In 1982, David Chaum introduced the concept of privacy-preserving certification [49]. Later, this promoting idea has been fully formalized by Camenisch and Lysyanskaya (CL) [50] in 2001. Since then, different concrete constructions have been proposed and considered as essential elementary units in privacy-preserving identity-management systems. In fact, relying on AC, each honest user is able to prove to a requesting SP, that he holds authenticated attributes, known also as credentials, obtained from trusted issuing authorities. Moreover, AC techniques prevent SPs to -trace- users' activities based on successive authentication sessions. That is, the user derives a proof associated to each different access request, such that the SP is not able to link a single received proof to another or to any information relative to its owner, even in case of collusion between providers and with the credential issuer. This property, referred to as multi-show property, has first been formerly presented by CL in [51], allowing a user to unlinkably prove possession of a credential as many times as necessary. However, in some applications, this multi-show property is too flexible to be useful as it limits the fulfillment of required features in some systems such as in e-voting systems, electronic surveys, *etc.*

AC techniques attract a lot of interest and full attention of service providers, thanks to their ability to support the data minimization fundamental feature [52] which states that data collection should be strictly required compared to the provided requested service, as explained in subsection 2.2. This interest is today magnified as this principle is at the core of the European General

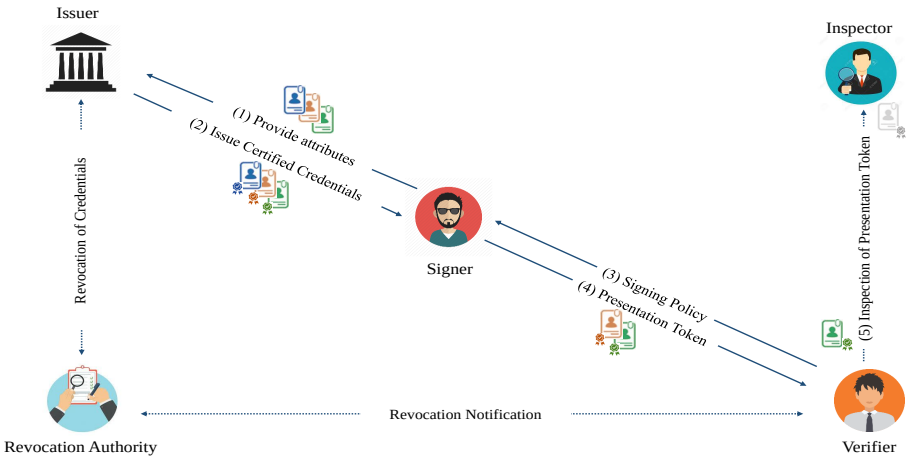

Figure 3: Traditional AC Entities Flow Diagram

Data Protection Regulation [4] and also the U.S. National Strategy for Trusted Identities in Cyberspace (NSTIC) [53].

The design of a privacy preserving certification scheme strongly relies on the use of malleable signature schemes, that provide several interesting properties, such as the selective disclosure feature and the unforgeability property [54]. In fact, the selective disclosure property refers to the ability provided to the user to present to the verifier partial information extracted or derived from his credential, for instance, to prove he is older than 18 to purchase liquors, while not revealing his birth date. The unforgeability property ensures that unless a user possesses a legitimate and certified credential, i.e, secret key, he is not able to generate a valid authentication proof, i.e, user's signature over the SP's access policy.

In the following, we first introduce main entities and general concepts of AC schemes. Second, we detail several signature schemes that are presented with their associated AC schemes.

4.2.1. Definitions

This section defines identified AC's entities (section 4.2.1) and procedures (section 4.2.1).

Entities. AC systems rely on some well identified entities. As shown in Figure 3, three main entities are considered as *mandatory* namely the issuer, the verifier and the user, while both a revocation authority and the inspector, designed as *optional* [55].

Indeed, in an AC system, each user represents a pivotal entity, who aspires a privacy preserving access to requested services, afforded by service providers, referred to as verifiers. Each verifier imposes an access control policy, called

presentation policy, to its resources and services, while enforcing a set of credentials that have to be owned by the users. To do, each user has first to obtain credentials from a trusted issuing authority, known by issuer. Then, he selects the appropriate information (*i.e.,* a sub-set of certified attributes) from the credentials and shows the selected information to the requesting service provider, under a *presentation token*. To effectively generate and accurately verify presentation tokens, the most recent revocation information have to be gathered from the revocation authority, by the user, respectively the verifier. That is, the revocation authority is responsible for revoking issued credentials and maintaining a list of valid credentials. When revoked, a credential can no longer be used to derive presentation tokens. The inspector is a trusted auditing entity, which is able to effectively remove the anonymity of a user when needed.

Procedures. An AC system relies on three main procedures and six randomized algorithms defined as follows:

- *Initialisation Procedure* — It includes the following two algorithms:
  - SETUP — given a security parameter ξ, the algorithm SETUP outputs the public parameters $params$ and the public-private key pair of the issuer (pk_{is}, sk_{is}).
  - USERKG — it takes as input $j \in \mathbb{N}$ (*i.e., j* corresponds to the user index) and outputs the public and private keys' pair $(pk_u, sk_u)_j$ of the user j.
- *Issuance Procedure* — This procedure is an interactive protocol combining the ISSUE and the OBTAIN algorithms as follows:
  - ISSUE $\leftrightarrow$ OBTAIN: the ISSUE $\leftrightarrow$ OBTAIN protocol corresponds to the issuance procedure conducted between the issuer and the user. The ISSUE algorithm, executed by the issuing entity, takes as input the public parameters $params$, the public key of the user pk_u, the secret key of the issuer sk_{is}, and the set of attributes $\{a_i\}_{i=1}^{N}$ (*i.e., N* presents the number of attributes). To receive a credential C, the user performs the OBTAIN algorithm which takes as input the public key of the issuer pk_{is} and the secret key of the user sk_u.
- *Presentation Procedure* — This procedure is an interactive protocol combining the SHOW and the VERIFY algorithms as follows:
  - SHOW $\leftrightarrow$ VERIFY: the SHOW $\leftrightarrow$ VERIFY interactive protocol corresponds the presentation procedure conducted between the user and the verifier. The SHOW algorithm takes as input the issuer's public key pk_{is}, the user's private key sk_u, the set of required attributes $\{a_i\}_{i=1}^{N'}$ and a credential C, w.r.t. the presentation policy, and it outputs a presentation token. The VERIFY algorithm, executed by the verifier, takes as input the public key of the issuer pk_{is}, the set

of attributes $\{a_i\}_{i=1}^{N'}$, and the presentation token. It outputs a bit $b \in \{0, 1\}$ for success or failure of the verification.

As introduced above, AC systems mainly rely on malleable signatures, namely Attribute based Signatures (ABS), introduced in subsection 4.2.2, group signatures, presented in subsection 4.2.3, sanitizable signatures, detailed in subsection 4.2.4 and blind signatures, introduced in subsection 4.2.5.

4.2.2. Attribute-based Signatures

Attribute-based Signatures, introduced by Maji *et al.* in 2010, is a flexible primitive that enables a user to sign a message with fine grained control over identifying information [56]. Indeed, each user, holding a set of attributes, has to obtain a secret key associated with his attributes from a trusted issuing authority. Thus, he is able to sign a message w.r.t. a predicate satisfied by any subset of his attributes. In the following, we denote an attribute based signature scheme by $\mathcal{ABS}$.

Definitions. Generally, an $\mathcal{ABS}$ scheme involves several entities, namely a Signature Trustee (ST), the Attribute Authority (AA), and potentially several signers and verifiers [57]. The ST is represented by a global entity that is responsible for generating global system parameters, while the AA is in charge of issuing the signing keys associated with the set of attributes that each user (signer) possesses, as depicted in Figure 4. Even though AA knows the signing keys and users' associated attributes, it is impossible for a curious AA to identify which attributes have been used in a given valid signature. Therefore, AA is unable to assign the signature to his originating user and/or to link several signatures as generated by the same user. Up to now, several ABS schemes appeared in the cryptographic literature, based on different design directions. Indeed, (i) the attribute value can be a binary-bit string [56, 57, 58, 59, 60], or relies on a particular data structure [61], (ii) access structures may be defined as threshold policies [58, 59, 60], monotonic policies [56, 61] or non-monotonic policies [57], and (iii) the attributes' secret keys issuance can be ensured by a single authority [59, 56, 61], or a group of authorities [57, 56].
The first security model for $\mathcal{ABS}$ schemes was proposed by Shahandashti *et al.* [59]. The authors pointed out the main procedures and security properties, such as correctness, unforgeability and signer-attribute privacy. Afterwards, Maji *et al.* [56] and El Kaafarani *et al.* [62] introduced and formalized the *perfect privacy* property which states that a signature should not reveal neither the identity of the signing user nor the set of attributes used for the signing procedure.

From Attribute Based Signatures to Anonymous Credential Systems. Several schemes have been proposed in literature [63, 64, 65, 66]. Most of the proposed schemes often rely on $\mathcal{ABS}$ schemes to anonymously authenticate users with

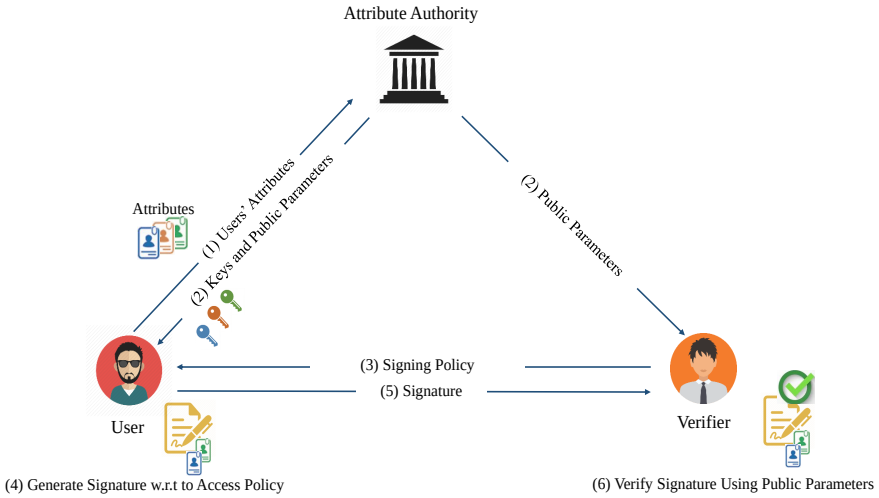

Figure 4: Attribute Based Signature Flow Diagram

third parties, *i.e.,* merchant web-site, e-assessment platform, dating web-site, cloud provider, *etc.* In [63], a new AC framework is presented, based on $\mathcal{ABS}$ while identifying additional requirements for $\mathcal{ABS}$ to meet AC features. For instance, a doctor (acting as user) obtains a certified credential (*i.e.,* professional card) by the Ministry of Health (which plays the role of the issuer) over the set of his attributes $\mathcal{S} = \{a_1 := \mathsf{name}; a_2 := \mathsf{Rob}, a_3 := \mathsf{city}, a_4 := \mathsf{NewYork}, a_5 := \mathsf{doctor}, a_6 := \mathsf{carcinology}\}$. The whole set of attributes is committed to a single value using the user public key, and it is signed using the secret key of the issuer, to derive the resulting credential, referred to as C.
Afterwards, the doctor can, for instance, prove that he is a doctor living in New York, without revealing his name nor his field of activity. To do so, the signing access structure is defined as $\Upsilon = (\mathsf{doctor} \vee \mathsf{cargeiver}) \wedge (\mathsf{city} \wedge (\mathsf{NewYork} \vee \mathsf{Tokyo} \vee \mathsf{Paris}))$. The doctor whose attributes satisfy the access structure is able to use his credential C to extract the related keys associated to the requested attributes a_3, a_4 and a_5. As such, the doctor remains anonymous among the group of doctors living in New York. That is, he can successfully prove the requested features since the signature of the Health's Ministry over the doctor's attributes is valid.
To meet AC requirements, the construction introduced in [63] ensures the traceability of signatures such that a new inspection procedure has been introduced to remove anonymity and identify the user originating an $\mathcal{ABS}$ signature. Second, the unlinkability between issuers has been addressed, relying on a new issuance procedure. In fact, in $\mathcal{ABS}$ schemes, when a user requests multiple authorities to issue private keys with respect to his attributes, issued credentials can be easily associated to one user through its public key.

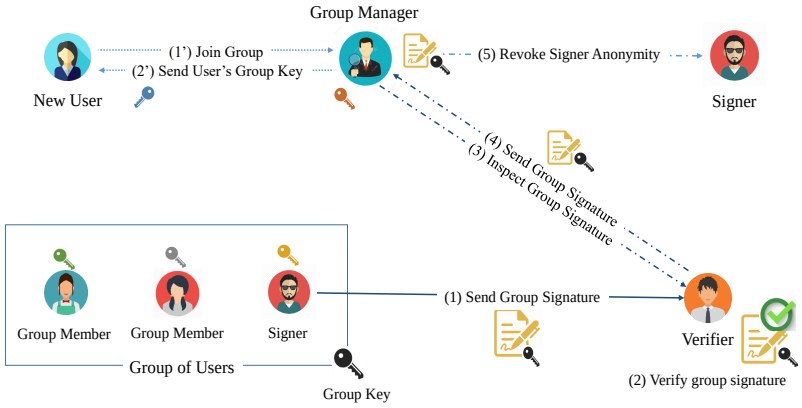

Figure 5: Group Signature Flow Diagram

4.2.3. Group Signatures

Generally, a group signature is a public-key primitive that allows members of a group to generate signatures on behalf of the group they are belonging to. As such, any verifying entity can be convinced that the generated signature has been produced by a legitimate group member, without inferring his real identity or being able to identify him. In the following, we denote a group signature scheme by $\mathcal{GS}$.

Definitions. As shown in Figure 5, a $\mathcal{GS}$ scheme involves four different roles, namely the group manager, group members (*i.e.,* signers), an opener and the verifier. The group manager is responsible for managing the group, by adding and revoking group members. Note that the group manager can either be a single entity or a number of coalitions of several entities (*e.g.,* group members) that can also play the role of an opener. In fact, a user has first to interact with the group manager to join the group and receive related secret information. Once registered, each group member can sign messages on behalf of the group. The generated signature can then be verified by any entity, called verifier, that is able to check the correctness of a group signature without identifying the actual signer. In case of dispute, a designated entity is able, when needed, to open a signature to identify the actual signer. Thus, group signatures permit to ensure two main privacy properties, namely, users' anonymity and accountability.

Group Signatures Applications. Group signatures have been applied in several use-cases and different settings. For instance, a variant of group signatures constitutes a building block for AC systems, namely Camenisch-Lysyanskaya (CL)

signature [50, 51] that was used to build the Idemix system, as presented above. Generally speaking, a group signature permits anonymous access control. It may be applied to electronic voting [67, 68, 69, 70] or even anonymous electronic cash [71, 72, 73]. That is, a voter can anonymously access the polling place and a payer can anonymously have the right to pay.

As detailed in subsection 4.2.1, an anonymous credential scheme, on the contrary, allows to manage several groups at once. For example, a service provider wanting to offer some special rights to certain categories of its clients, such as students under 25 or seniors over 60. A group signature would not be relevant in this case by its own, however, when applied as an anonymous credential mechanism, it may provide the desired service. Indeed, this service provider does not necessarily need to obtain all the information about the user (name, address, exact age, *etc.*) but needs to be sure that these pieces of information (age and student status) are correct and that they have been certified by a trusted entity (*e.g.,* the university or the country's authority). With anonymous credentials systems, a user needs to register only once in order to belong to several groups (such as the group of persons being less than 25, the group of French people, *etc.*).

Several schemes have been presented in the literature to support anonymous access [74, 75, 72, 69]. Zheng *et al.* [75] have proposed a general construction of linkable group signature ($\mathcal{LGS}$) to achieve anonymity, auditing and tracing functions for later auditing anonymous communications. ($\mathcal{LGS}$) is differentiated from group signatures by enabling an authority to reveal whether two signatures have been generated by the same group member without accessing the signer's identity.

Helbach *et al.* [74] have presented an extension to code voting approach in order to resist vote selling attack by combining vote update feature and linkable group signature. Note that code voting consists on using a separate channel from the voting authority to the voter, *e.g.,* using snail mail. To mitigate against vote selling attack, a voter is allowed to update his vote as many time as he wants while the last vote is counted. This makes the vote buyer never sure that the voter will not update his vote, after he has proven his choice to the vote buyer. The use of $\mathcal{LGS}$ enables to link different votes issued by the same voter in order to update his vote.

Yan *et al.* [72] have designed an e-cash system based on a certificate-less group signature. The proposed $\mathcal{GS}$ signature achieves unforgeability, money tampering and money double-spending requirements. The proposed e-cash solution supports *Fair Off-Line Multi-Bank System* that allows the interaction of multiple banks in off-line mode.

Malina *et al.* [69] have proposed a combination of a group signature scheme and the probabilistic ElGamal encryption scheme to preserve the anonymity of voters during the voting and tallying processes. To achieve the accountability feature, this solution relies on the cooperation of a manager and the election authority to reveal a user identity and revoke her from the voting system.

4.2.4. Sanitizable Signatures

Sanitizable signatures are malleable signatures, introduced in 2005, by Ateniese *et al.* [76]. Sanitizable signatures, denoted by $\mathcal{S}t\mathcal{S}$, enable a designated party, referred to as a sanitizer, to modify some parts of the original message, in a controlled way initially defined by the original signer, while resulting in a still valid signature, as depicted in Figure 6.

Definitions. To enable the authorized sanitizer to modify parts of the original message m, the signer provides a description of the admissible modifications ADM for each message-sanitizer pairs. That is, the signer divides the message $m \in \{0,1\}^*$ into N blocks $m_1, \cdots, m_N$, defines the set $\mathcal{S}_{adm} \subseteq \{1, N\}$ of admissible blocks and signs the whole message using a key related to the sanitizer. Using this key, the sanitizer is next able to modify the admissible parts of the given message so that the resulting signature is still valid under the signer public key.

Fundamentally, two additional functions ADM and MOD are considered to define the admissible modifications, such as:

- ADM maps each message m to the admissible blocks. It outputs the rank of every admissible block and the corresponding the length l_i of the related i^{th} block, such that $l = \sum_{i=1}^{N} l_i$ is the length of the message m. The $\mathcal{S}_{adm} \subseteq \{1, N\}$ represents the set of the ranks of admissible blocks.
  We note that there exists an optional function, called FIX. It maps a message m to its fixed parts. The FIX function outputs the set of fixed blocks, called $\mathcal{S}_{fix} \subseteq \{1, N\}$. That is, $\mathcal{S}_{fix}$ is the concatenation of all blocks not appearing in $\mathcal{S}_{adm}$.

- MOD maps a message m_j to the modified message m'_j, where j is the rank of the block m_j. It is defined as a set of pairs (j, m'_j), such that the block m_j has to be replaced by m'_j.

A sanitizable signature scheme has to ensure the correctness property. It states that honestly generated (*signing correctness*) and sanitized (*sanitizing correctness*) signatures have to be accepted by the verifier, and that honestly generated proofs on valid signatures (*proof correctness*) have to be accepted by the verifying entity.
Several features have been proposed to $\mathcal{S}t\mathcal{S}$. Indeed, in 2010, Canard and Jambert integrated some extensions to sanitizable signature schemes [77]. They showed how to limit the set of possible modifications on one single block and how to enforce the same modifications on different messages blocks. Then, Canard *et al.* extended sanitizable signatures to the setting of multiple signers and sanitizers [78].

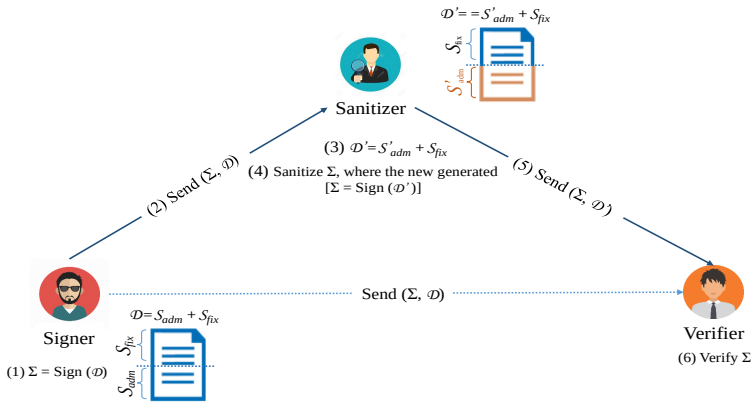

Figure 6: Sanitizable Signature Flow Diagram

From Sanitizable Signatures to Anonymous Credential Systems. As stated above, $\mathcal{S}t\mathcal{S}$ schemes are considered as powerful malleable primitives that allow computations on authenticated data. Thus, this class of signatures is widely adopted in the design of several AC systems [54, 79].

In [80], Chow *et al.* proposed a privacy-preserving distributed identity management mechanism for cloud environments. The proposed mechanism combines both group signatures and sanitizable signatures to authenticate to different cloud providers. The main purpose of using $\mathcal{S}t\mathcal{S}$ is to hide some parts of the messages which are not the concerns of a particular cloud service provider. Later, in [81], Canard and Lescuyer formalized the usage of $\mathcal{S}t\mathcal{S}$ for anonymous credential systems, while providing a concrete construction based on standard assumptions. They extended sanitizable signatures' features as follows:

- *verifying without sanitizing key* — the signature should be verified without the sanitizer's public key, in order to ensure user's anonymity.
- *traceability of signatures* — this algorithm should be performed by a separate authority in order to recover the actual designated sanitizer of a given message-signature pair.
- *no proof algorithm* — there is no need for a proof algorithm, in the sense that the above tracing procedure is not designed to decide whether a given signature has been sanitized or not.
- *restrictions on admissible values* — modifications are carried out in a controlled way.

- *replaying sessions* — the sanitizer has to modify the admissible parts of the message with respect to a random value sent by the verifier in each session.

Recently, in 2018, Pamies-Estremes *et al.* proposed a new security mechanism that addresses the issue of protecting the identity of end-users and their sensitive personal data generated by Internet-based personal devices like Google Home[28] and Amazon Echo[29] [82]. The proposed scheme combines the use of anonymization techniques, introduced in subsection 5.1 and sanitizable signatures. The signing process permits to hide authenticated identifying information while transferring data stream from the local device to remote storage servers.

4.2.5. Blind Signatures

Blind signatures allow a user to obtain a signature from a signer where this latter can not learn information about the message she signed and the user can get only one valid signature after one interaction with the signer [49, 71]. In the following, we denote by $\mathcal{BS}$, a blind signature scheme.

Definitions. A $\mathcal{BS}$ scheme is a form of a digital signature that enables the receiver to get a message signed by the signer without revealing any information about the message. As depicted in Figure 7, in the blind version of the Schnorr signature scheme [83], the signer proves knowledge of his secret x based on his public key $h = g^x$. Then, the receiver creates a challenge, such that the resulting signature is computed based on the commitment and the response of the signer. The resulting signature (c', r') is verified by checking if c' is equal to $H(m||g^{r'}h^{-c'})$. The verifier entity can forward her message and signature to a third party which will be able to verify the signature as being generated by a legitimate signer.

Blind Signatures Applications to Anonymous Authentication. $\mathcal{BS}$ schemes can conveniently be used to issue randomized presentation tokens, as required by AC system. That is, the validity of the signature, *i.e.,* correctness of verification, ensures that the user generating the token is allowed to access a service. The blindness of the signature ensures that the service provider is not able to either recognize the actual signing user nor detecting a previously issued token so as associate it with a particular user.

$\mathcal{BS}$ schemes have been applied in several use-cases and different settings. Indeed, a variant of blind signatures is a main building block for AC systems, *i.e.,* Brands' signature [84] that was used to build the U-Prove system. That is, issuing a credential is performed using an interactive protocol in which the

[28] https://store.google.com/product/google_home
[29] https://www.amazon.com/Amazon-Echo-And-Alexa-Devices/b?ie=UTF8&node=9818047011

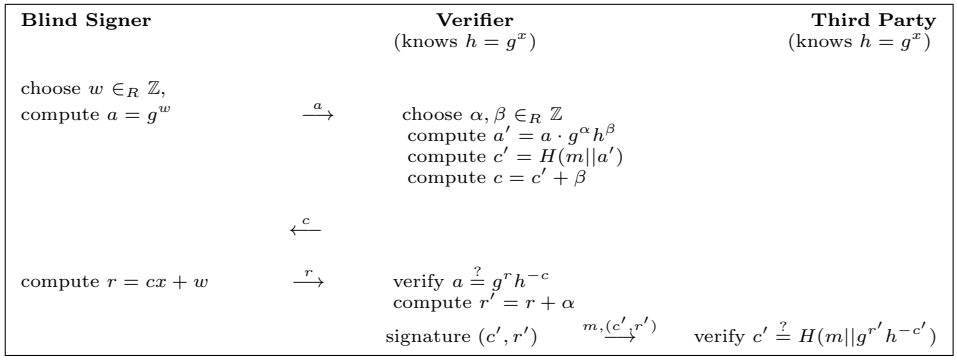

Figure 7: Schnorr Blind Signature Protocol

issuer blindly signs a commitment. The attributes are assumed to be common input to the user and the issuer, the resulting credential and the corresponding secret key are only known to the user. Thus, even in case of a collusion between an issuer and a verifier, it is impossible to relate a credential to a particular user.
In [85], Verheul proposed a new privacy preserving scheme, known as self-blindable credentials, that is based on Chaum-Pederson signature [86]. The idea behind the self-blindable credentials is that every time a credential is used it is blinded such that two occurrences of the same credential cannot be recognised. This is different than the U-Prove token which is the same in each transaction, and hence serves as a pseudonym. The benefit of this approach is that the use of such credentials is untraceable.

4.2.6. Comparative Industrial Solutions

As stated above, two main industrial solutions emerged, namely Idemix[30] [87] and U-Prove[31] [88]. An Idemix credential is a Camenisch-Lysyanskaya (CL) signature, a variant of group signatures, generated by the issuer over the user's secret key and the attribute values [50]. To transform a credential into a presentation token, the user creates a zero-knowledge proof showing that he knows a valid CL signature on a committed value. Interested readers may refer to [89] for more details about cryptographic primitives, namely zero-knowledge proofs and commitments schemes.
On the other side, U-Prove is based on Brands signature which is a variant of blind signatures [84].

Table 3 presents an informal comparison between Idemix and U-Prove, based on the expected security and functional properties in AC systems. Both Idemix

[30] https://www.zurich.ibm.com/identity_mixer/
[31] https://www.microsoft.com/en-us/research/project/u-prove/

Table 3: Comparison between Industrial Solutions Idemix and U-Prove

Attribute based Credential Schemes			Idemix	U-Prove
Properties	anonymity		✓	✓
	unlinkability	issue-show unlinkability	✓	✓
		multi-show unlinkability	✓	×
	revocation		✓	✓
	inspection		✓	✓
	selective disclosure		✓	✓
Performances (in sec)*	issuing a credential (5 attributes)		2.6	5.5
	selective disclosure	empty proof	1.5	0.9
		a proof hiding two attributes among the five	< 1	0.6

* Implementation results are based on a MULTOS smart-card platform [90, 91].

and U-Prove preserve the anonymity of users during the authentication process, thanks to the use of CL and Brands signatures respectively, and zero-knowledge proofs [92, 89]. Additionally, these two anonymous credentials' systems inherit the unforgeability property from their related signature schemes, such that, an entity that does not belong to the set of authorized users cannot successfully run the authentication protocol with the service provider.
Furthermore, both credentials can be abstractly seen as signed commitments [89]. A commitment in this case is basically the product of (algebraic) group generators with attributes and a secret key as their exponents. As a result of this resemblance, selective disclosure can be done similarly. Having received some attributes, the verifier can reproduce a partial product from the commitment. Then, he can combine it with the presented proof of knowledge about all the other attributes. Only a user having a valid credential can provide such a proof.

However, unlike Idemix, U-Prove does not support multi-show unlinkability. This means that the U-Prove credential is revealed every time it is used, and as such, for better privacy support, it requires the issuer to generate as many different U-Prove credentials as their verification instances. This is due to the U-Prove issuance protocol relying on un-randomisable signatures. For Idemix provided with randomisable signatures, the user can randomize by himself the same Idemix credential as many times as needed, for presentation to the verifier, thus naturally ensuring multi-show unlinkability.

As explained in [90, 91], the implementation on a MULTOS smart-card platform of both U-Prove and Idemix solutions for five attributes shows that the issuance of U-Prove credentials is much more resource consuming than Idemix. Indeed, a the issuance procedure of a credential containing 5 attributes takes only 2.6 sec, using the Idemix issuance algorithm, compared to 5.5 sec, for U-Prove. This is mainly due the number of interactions during the issuance process, resulting on heavy communication overhead. On the other side, the presentation procedure of Idemix is much more resource-consuming compared to U-Prove. That is, the presentation of an empty proof takes around 1.5 sec, using the Idemix issuance algorithm, compared to only 0.9 sec. This is mainly due to the heavy computation overhead of CL signatures, performed in large

RSA groups.

4.2.7. Comparison between Privacy Preserving Certification Schemes based on Several Signatures Schemes

Table 4 gives a quantitative comparison between the reviewed signature schemes' properties w.r.t. security and functional security and functional features expected from anonymous certification schemes. First, we have to emphasize that all reviewed signature schemes, *i.e.,* $\mathcal{ABS}$, $\mathcal{StS}$, $\mathcal{GS}$ and $\mathcal{BS}$ schemes, are considered as cryptographic primitives while AC refers to a complete system, involving several procedures and algorithms. Consequently, while all signatures' schemes ensure the unforgeability and anonymity/privacy properties, it is fair and justifiable that these signatures do not support *all* AC's expected security and functional features.

Table 4: Comparative Signature Schemes Analysis to Fulfil the Security and Functional Requirements of Anonymous Credential Systems

ACS Expected Properties	**Sanitizable Signatures**	**Attribute Based Signatures**	**Group Signatures**	**Blind Signatures**
		Security Properties		
Unforgeability	unforgeability immutability *	unforgeability	unforgeability	unforgeability
Anonymity	privacy verifying without sanitizing key [81]	perfect privacy [62] public verifiability	anonymity public verifiability	privacy (blindness)
Multi-show unlinkability	unlinkability	unlinkability	unlinkability	unlinkability
		Functional Properties		
Selective Disclosure	sanitizing feature	predicate-based signing feature		
Traceability (inspection procedure)	traceability [81]	accountability [62]	accountability (open feature)	traceability

* It is not possible for the sanitizer to modify non admissible blocks of a signed message. That is, no modification is allowed on fixed parts or on an admissible part with disallowed values.

Then, as detailed above, all reviewed signatures' schemes have been extended and several algorithms were adapted and/or added to support AC's security properties.

Indeed, constructions [81] and [63], introducing two different AC systems based on $\mathcal{StS}$ and $\mathcal{ABS}$ respectively, adapted these primitives and provided a new feature, *i.e.,* the traceability. For instance, [81] proposed to design a new algorithm that is performed by a separate authority, *i.e.,* the inspector, in order to recover the sanitizer of a given message-signature pair. Another extension toward the support of multiple signers, *i.e.,* issuers for an AC system, is equally proposed [78]. Additionally, the sanitizing algorithm is adapted to limit modifications on admissible values/blocks, in order to ensure that these modifications are executed in a controlled way. Similarly, [63] introduced a new algorithm that is performed by a dedicated trusted entity, *i.e.,* the inspector, responsible for removing the anonymity of the user originating an $\mathcal{ABS}$ signature. In addition, [63] proposed an interesting extension to the credentials' issuance algorithm that supports the issuance of credentials from different issuers, associated to one user through its public key.

From Table 4, it is noticeable that contrary to *i.e.,* $\mathcal{ABS}$, $\mathcal{StS}$ and $\mathcal{GS}$, $\mathcal{BS}$ schemes do not support the multi-show unlinkability property, as explained in section 4.2.6, due to their intrinsic properties.

Another main difference between reviewed signatures' schemes consists in computation and communication overheads, induced from the execution of the different algorithms. Indeed, the bandwidth consumption is measured through the exchanged quantity of data during protocols' execution w.r.t. the number of attributes supported by a credential or a presentation token. On the other hand, the computation cost at the user, issuer and verifier sides is measured w.r.t. the number and complexity of performed operations during protocols' running. Thus, performances' criterion is very important while selecting the appropriate signature scheme. Nevertheless, $\mathcal{ABS}$, $\mathcal{StS}$, $\mathcal{GS}$ and $\mathcal{BS}$ schemes are direct signatures, thus, their issuance algorithm's overhead is rather interesting, compared to complete industrial solutions, *i.e.,* Idemix [87] and U-Prove [88] solutions.

4.3. Obfuscation: Data Perturbation

Obfuscation techniques aim at intentionally making information difficult to understand or perceive for security and privacy reasons. In fact, the speed of dissemination of information, the technical progress and the global nature of Internet make it difficult to delete data that may be too personal, embarrassing or confidential. Thus, obfuscation consists mainly in publishing large amounts of information that are false, imprecise, irrelevant and/or organized in such a way that the information that one wishes to protect is hidden, *i.e.,* embedded in a large volume of data.

Obfuscation mechanisms may be deployed at both the user and server sides. Indeed, data perturbation, considered as user-side technique is detailed hereafter, while private information retrieval, considered as a server-involving technique, is introduced in subsection 5.3.

From an end-user point of view, an alternative to hinder an attacker in its efforts to precisely profile users consists in perturbing the information they explicitly or implicitly disclose when communicating with a personalized information system. The submission of false data, together with the user genuine data, is an illustrative example of data-perturbation mechanism. In this kind of mechanisms, the perturbation itself typically takes place at the user side. This means that users do not need to trust any external entity.

Data perturbation techniques are used for enhancing privacy in location based services. Indeed, to protect queries, Pingley *et al.* introduced DUMMY-Q [93]. They proposed to generate dummy queries that will be sent to the LBS server along with the real query. That is, based on a query-perturbation mechanism, DUMMY-Q requires the installation of a software at the client-side in order to produce a series of fake queries with different query attributes, except the same location as the real one. As such, the LBS server cannot determine in which attributes the user has interest. Niu *et al.* proposed a dummy location selection (DLS) algorithm, that permits to hide the real location of users [94]. Contrary to the traditional dummy location schemes, [94] considers that an adversary may have side information about the real position of the user. Thus, the proposed algorithm consists in an entropy-based measure to achieve K-anonymity. Data perturbation is also used to ensure private web browsing. For instance,

in 2006, Elovici *et al.* presented a privacy-preserving web search system, called PRAW [95]. It ensures the privacy of a group of users that share the same access point to the Web. That is, the authors propose to hide the real user profile by generating fake transactions, for instance, accesses to a fake web page, *etc.* Similarly, in [96], Ye *et al.* proposed a private web browsing solution. Based on the PRAW scheme, the authors propose to send a real query with a certain probability and a dummy one with the complement of that probability. As such, it is not feasible for an attacker to figure out the real query. Recently, Masood *et al.* introduced Incognito, a privacy-aware obfuscation method for Web data [97]. The main idea behind this solution is to first rely on probabilistic methods to measure the privacy risk of generated Web data. Then, data with high predicted risk are obfuscated to minimize the privacy risk using semantically similar data.

Data perturbation of user profiles for privacy preservation purposes may be achieved not only through the injection of false/dummy information, but also via suppression. For instance, in [98], Parra *et al.* propose to eliminate tags from queries, in order to enhance privacy in semantic web scenarios. However, this solution comes at the expense of the semantic functions of the web. Subsequently, the authors investigate, in [99], the privacy utility trade-offs induced by the suppression technique.

A combined usage of both methods, namely the addition and suppression, is discussed for personalized recommendation systems [100]. Thus, the main idea is to analytically study the impact of the adoption of both methods. That is, users may submit false ratings that do not reflect their preferences, and/or refrain from ratings other items. Later, in 2017, Polatidis *et al.* presented a privacy-preserving collaborative recommendation scheme, through the perturbation of each rating [101]. The proposed approach relies on multiple levels and different ranges of random values for each level. Before transferring each rate to the remote server, the privacy level and the perturbation range have to be selected randomly from a fixed range of privacy levels.

4.4. Privacy preserving Computation: Secure Multiparty Computation (SMC)

Privacy preserving computation techniques aim at protecting users' privacy and the secrecy of data contents during processing over these data. These techniques involve several powerful cryptographic primitives that may be deployed at the user side as Secure Multi-party Computation (SMC) or at the server side as Homomorphic Encryption (HE) schemes presented in subsection 5.4.

4.4.1. Definitions

Secure Multi-party Computation (SMC) are considered under the untrusted model [102]. Indeed, the goal of SMC techniques is to enable distributed computing tasks among participating entities in a secure manner. That is, SMC considers that a group of participants wants to carry a joint computation of

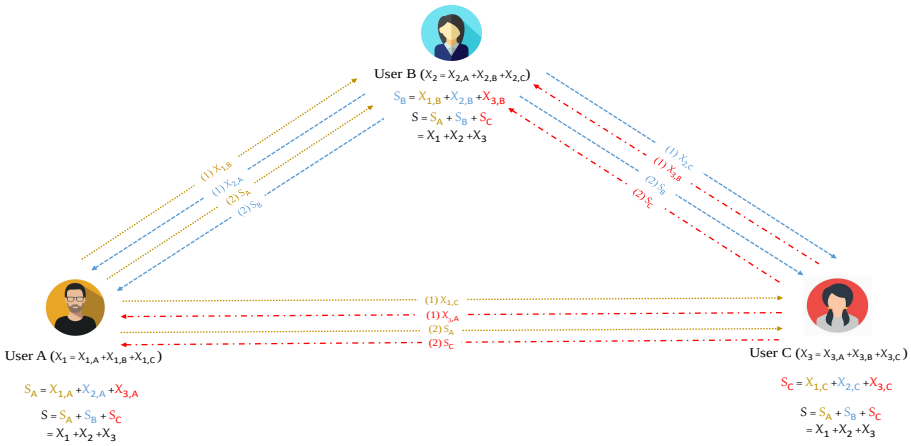

Figure 8: Secure Multiparty Computation (SMC) Flow Diagram: Addition Scenario

a given function while keeping secret the input data of each party. SMC has been used to solve several privacy-preserving problems such as private database queries, secret voting, privacy preserving data mining and privacy preserving intrusion detection tools and mechanisms [103, 104, 105, 106, 107].

SMC was first introduced and formalized in 1982 by Yao, relying on the millionaire problem [108]. The Yao's problem considers two entities that want to determine who is the richest, but without revealing their actual wealth. Indeed, a set of participants have to compute the value of a function f (*i.e.,* f represents the max function) using their private inputs, without revealing anything about them. Hence, they run an interactive protocol in between. Several SMC mechanisms rely on a circuit evaluation protocol [109]. In fact, each party's input value is represented by a boolean circuit, or an arithmetic circuit, while communication exchanges between circuit gates are based on the Oblivious Transfer protocol using randomly selected variables [110].

Figure 8 depicts an illustrative example of an SMC algorithm executed between three end-users, where f represents the addition function. Let us assume that each user U_i, where $i \in \{A, B, C\}$, has a secret X_j, such that $j \in \{1, 2, 3\}$. The group of users performs an interactive SMC algorithm to compute the sum of their private inputs, denoted by S. For this purpose, each user U_i first divides his secret X_j into three different pieces $X_{j,i}$. Then, it sends one share to each corresponding peer and keeps secret one share locally, as shown in Figure 8. Upon receiving the different shares from all peers, each user computes $S_i = \sum_{j=1}^{3} X_{j,i}$ and sends the result to his peers. Finally, each user is able to calculate the sum of different shares, such as $S = \sum_{i=A}^{C} S_i$.

Although SMC mechanisms ensure privacy-preserving features, they still suffer from a heavy processing and high communication cost, mainly burdensome for resource-constrained devices. Generally, the security of SMC schemes is de-

fined with respect to the ideal model, where f is executed by a trusted party. That is, it is required that during the execution of a protocol, parties cannot learn more information about the inputs of the other entities than they would learn if f was computed by the trusted parties. SMC schemes have to fulfill the following additional properties:

- **correctness** — a malicious party, or a set of malicious entities, should not be able to influence the output of the honest parties more than they could in the ideal model by modifying their own inputs.
- **inputs' independence** — considering that in the ideal model, all inputs are sent to the trusted party before any output is received. Thus, malicious parties do not know anything about the inputs of honest parties when they send their entries. Roughly speaking, each party's entry is chosen independently from others.
- **fairness** — this property ensures that honest parties have always access to the correct output, even if a minority of parties is malicious. However, the lack of fairness is unavoidable, when malicious entities can terminate the protocol after learning the output, thus preventing honest entities from learning it. Several works proposed to overcome this impossibility by relaxing the security requirements.

Three different approaches are generally deployed to provide secure multi-party computation functionalities, namely oblivious transfer [110], homomorphic encryption [111] and secret sharing [112, 103] techniques.
The oblivious transfer protocol generates high processing and communication overheads. The secret sharing approach gives better results in terms of computation cost, thanks to the usage of primitive operations [113]. However, it requires the existence of secure channels between different participating entities, hence generating a high bandwidth consumption, due to the involved interactions between users. The homomorphic encryption does not require the existence of secure channels and assures high level of privacy. However, it necessitates several processing operations to ensure homomorphism properties, thus generating high computation complexity.

4.4.2. SMC Applications

Several works have been proposed in order to ensure privacy preserving collaborative computations for pervasive applications [104, 114, 105, 106, 115**?**]. In [104], Reiter *et al.* introduced a novel user collaboration protocol, called Crowds. The proposed protocol enables a group of users to collaborate in order to submit their queries/messages to an untrusted web server. That is, before sending a message, the user flips a coin to decide whether to submit it directly to the service provider, or to send it to another peer, who will then repeat the randomized decision. Note that each user belongs to a pre-defined group of users, that collaborate together to preserve their mutual privacy. The main inconvenience of Crowds consists on the heavy communication overhead, mainly

due to additional hops.
Following Crowds protocol, Barba *et al.* proposed a privacy preserving scheme to report traffic violations in vehicular applications [116]. Rather, their construction introduced acceptable computation and communication costs compared to Crowds. In 2009, Pathak *et al.* proposed a scalable protocol to perform secure multi-party computations on encrypted data for banking applications [105]. In their scheme, modifier tokens are generated along with the encryption process. Thus, the enciphering algorithm is based on the acquired data and modifier tokens to compute the ciphertext without revealing the output data. Therefore, users' privacy is preserved. Jaydip Sen discusses research directions of SMC usage for IoT applications [117] and states that most of the proposed constructions are domain-specific and are not suitable in ubiquitous environments. As stated above, these schemes produce important computation and communication overheads for resource-constrained devices, due the usage of oblivious transfer, homomorphic encryption as well as secret sharing techniques. Recently, in 2017, Tonyali *et al.* [115] presented a privacy-preserving data aggregation scheme adapted to IoT applications. That is, each participating device uses a pseudo-random number generator to locally compute the shares that are calculated by the other nodes. Hence, they do not need to exchange the shares before each data collection round.

For personalized web-search services, Erola *et al.* propose a new privacy preserving web-search scheme, following the same philosophy introduced by Crowds, while grouping users with respect to their interests [118]. In the same vein, Rebollo *et al.* presented a new scheme for privacy preserving web search applications [119]. In their proposal, two users or more exchange a portion of their queries before submitting to the web-search engine, in order to hide their interests, w.r.t. the service provider.

5. Server Side Techniques

Server-side privacy preserving techniques include mechanisms where service providers are required to perform additional processing on their clients' hosted data. That is, service providers may be involved in anonymizing their databases, for further use by third parties, removing identifying traces or executing heavy computations on encrypted data contents at the request of their clients.
Server-side techniques include Statistical Disclosure Control mechanisms detailed in subsection 5.1, self-destructing systems presented in subsection 5.2, Private Information Retrieval techniques introduced in subsection 5.3 and homomorphic encryption schemes as discussed in subsection 5.4.

5.1. Statistical Disclosure Control

Statistical Disclosure Control (SDC) mechanisms are mainly used to protect data within statistical databases. They permit to resolve the trade-off between data usability and users' privacy preservation, as revealed results, either the

databases or a specific result over the databasis do not permit to reveal information related to a specific user.
SDC techniques include database anonymizing techniques and Differential Privacy mechanisms. Anonymization techniques are relevant for various use-cases, namely applications that do not require to learn the original user's identity, but only context information. Anonymization techniques mainly refer to database privacy preservation. That is, for statistical purposes, database privacy refers to the privacy of respondents to which the database records correspond. Even so, for cooperative applications where the database belongs to several corporations, it comes to the privacy protection of the various collaborating entities, *i.e.,* data owner privacy. For several pervasive applications, for instance health care systems, both respondent and data owner's privacy is required: the patient wants to ensure his privacy preservation and the medical records have to be protected. Finally, for personalized web search engines, the growing concern is about users' privacy.

In the following, we first review techniques derived from database anonymity (subsection 5.1.1). Then, we detail anonymization techniques for data mining processes, and mainly Differential Privacy (subsection 5.1.2).

5.1.1. Anonymizing databases

Main techniques for anonymizing databases w.r.t. respondent, owner and users' privacy include *k-anonymity*, *t-closeness* and *l-diversity* [120]. Note that these techniques that are originally used over statistical databases have extended usage to dynamic data, as presented below.

K-anonymity. In 2001, Samarati introduced the notion of k-anonymity, to prevent the conflict between information loss and disclosure risk.
For defining the k-anonymity approach, we first distinguish three types of attributes [121], for a microdata set $\mathcal{S}$:

- identifiers: attributes that exactly identify the respondent, such as, his social security number or tax number. Generally, it is assumed that during a pre-processing step, identifiers in $\mathcal{S}$ have to be removed or encrypted.
- key attributes: attributes of $\mathcal{S}$ that are useful to the application and which combination with external information can serve to re-identify respondents of the database. Examples of these attributes are : gender, age, ZIP code, $\cdots$ Unlike identifiers, these attributes cannot be removed from $\mathcal{S}$.
- confidential outcome attributes or sensitive attributes: Attributes which values are of high interest for the adversary. They usually include religion, salaries, *etc.*

Intuitively, to enforce anonymity, it is necessary to identify which attributes are considered as key attributes, called also quasi-identifiers [122]. That is to

say, k-anonymity is able to prevent identity disclosure, *i.e.,* a record in the k-anonymized set $\mathcal{S}_k$ cannot be mapped back to the corresponding record in the original $\mathcal{S}$, hence, by ensuring that each record is indistinguishable by at least other $k-1$ records based on the value of key attributes. However, k-anonymity is not resistant to attribute disclosure attacks, as illustrated by the following example: Let us suppose that a patient's health record is k-anonymized into a group of k-patients, while considering three different key attributes, namely Age = 42, Height = 160 and Weight = 75. Thus, if all patients share the same confidential attribute Disease = Cancer, k-anonymization may be useless as an attacker may link an external record with the above group of patients, based on the key attributes. Thus, the attacker can successfully perform an attribute disclosure attack by inferring that Mariana Ju suffers from Cancer.

K-anonymity techniques originally designed for statistical database have been applied to more dynamic context-aware systems [123, 124, 125, 126]. Indeed, for Location-Based Services (LBS), an attacker, having access to users' location, may be able to identify the requesting user, relying on its spatio-temporal parameters. Consequently, several research works propose to expand the precise location of the user to involve several potential requesting issuers. This leads to generalizing several context-data to ensure anonymity, thus resulting in the context information released to the service provider being sometimes too large and imprecise to provide an acceptable quality for the service. To alleviate the trade-off between privacy and personalization, Shin *et al.* propose several personalized anonymization profiles, based on different desired levels of privacy [127]. For instance, a user may choose to generalize the area with 1 km^2, without requiring generalization in his profile data.

Several services, such as well-being applications, or recommendation services do not generally rely on users' location, but other context-data, such as users' activities, users' habits and interests, *etc.* In [128], Riboni *et al.* pointed out that even when k-anonymity is enforced, an attacker may identify an actual requesting user, while monitoring the behaviour of the set of potential requesting entities w.r.t. services' responses. [128] proposes to deploy an intermediary trusted entity that is responsible for computing the level of privacy violation (based on some privacy metrics), associated with possible responses suggested by the service, thus the user is notified before authenticating with the service provider.

l-diversity. In 2006, Machanavajjla *et al.* introduced the notion of l-diversity as an improved version of k-anonymity [129] resistant to attribute disclosure attacks. Definition 5.1 defines l-diversity as follows:

Definition 5.1. l-diversity [121] A data set is said to satisfy l-diversity if, for each group of records that shares the same combination of key attributes (*i.e.,* equivalence class), there exist at least l well represented values for each confidential attribute.

Well represented means that key attributes should have at least l-distinct values or their entropy is greater than $\log l$.
Note that l-diversity approaches offer better resistance to attribute disclosures than k-anonymity, but still remain vulnerable [130] in case values of a sensitive attribute within a group are l-diverse, but semantically similar. Indeed, let us consider that some patients records are available in a 3-diverse data set where the confidential attribute Disease belongs to the set { breast cancer, liver cancer, bladder cancer }. The adversary knowing that a specific user is part of that group, can infer that the user has cancer. If the sensitive attribute is numerical and values within a group are l-diverse but very similar, the attacker can then estimate the sensitive attribute value.

The l-diversity technique has been used in several privacy-preserving location-based approach [131, 132, 133, 134, 135]. In [131], Liu *et al.* defined the query l-diversity concept for location-based services. In fact, the main idea consists in ensuring that queries' contents have to be different enough for all queries that share the same cloaked area. As such, it is impossible to link a query with its original issuer, with a probability less than a pre-defined threshold value. Later, in [132], Bamba et al presented a privacy preserving location-based framework for mobile applications, called PrivacyGrid. The proposed framework provides a location privacy preference profile model, which allows mobile users to explicitly define their preferred location privacy requirements in terms of both location hiding measures (*i.e.,* location k-anonymity and location l-diversity) and location QoS measures. Concretely, each mobile user communicates with a number of LBS servers, via a proxy anonymizing-location server. Indeed, each proxy server tracks the location updates of the mobile users and performs location anonymization based on publicly available data which can be used for ensuring location l-diversity for user requests. In 2009, Xue *et al.* formally defined the location diversity concept [133]. They assume that this concept improves the spatial k-anonymity by ensuring that each query can be associated with at least l different semantic locations (*e.g.,* parking, university, drug-store, *etc.*). Consequently, the authors presented a new algorithm that permits to construct a group of l-different semantic locations.

Wang *et al.* introduced a new scheme to protect queries' attributes [134]. The proposed construction permits to generate a cloaking area for continuous LBSs and the anonymizing server refines the results w.r.t. l-diversity and k-anonymity processes. In 2017, Ye *et al.* introduced a new l-diversity algorithm that relies on road networks in order to enhance trajectory secrecy [135]. The proposed algorithm pre-processes a set of similar trajectories to hide the actual real trajectory of a user. In addition, to de-personalize the user's trajectory, each location reported to LBS server is a cloaking region that contains at least other l-1 different trajectories, which are generated in advance from road maps.

t-closeness. In 2007, Li *et al.* introduced t-closeness to mitigate attacks against k-anonymity and l-diversity approaches [130]. It is defined as follows:

Definition 5.2. t-closeness [130] A data set is said to ensure the t-closeness property, if for each group of records that shares a combination of key attributes (*i.e.,* equivalence class), the distance between the distribution of a sensitive attribute in this class and the distribution of the attribute in the original table is no more than a threshold t.

While improving resistance to known attacks, t-closeness is known to damage the data utility due to some noise introduction through data perturbation and generalization techniques. Moreover, as identified in [136], even for the same level of t-closeness privacy referred to as "t", the effective achieved privacy measurement is actually more relying on the classification of attribute values, which is very much subjective, than the "t" value.

To ensure queries privacy, t-closeness algorithms are generally combined with k-anonymity techniques. For instance, in [137], Riboni *et al.* proposed an anonymisation technique for location based queries, relying on the generalization of spatio-temporal information. That is, the authors noticed that both several concurrent requests may occur and similar requests issued by the same users may be repeated. Thus, to encounter such kind of privacy attacks, the proposed framework is based on the association of k-anonymity and t-closeness techniques.

5.1.2. Differential Privacy

Differential privacy (DP) is gaining an expanding interest, mainly to ensure privacy preserving data mining [138, 139, 140, 141]. In a nutshell, differential privacy ensures that the removal or addition of a single database item does not (substantially) affect the outcome of any analysis (*i.e.,* the probability distribution of released items does not significantly change), as presented in Definition 5.3. This property is enforced by adding random noise, to the exact outcome. Moreover, note that differential privacy addresses data leakage attacks as even if a user has removed his data from the data set, no outputs would become significantly more or less likely.

Definition 5.3. Differential Privacy [138] A randomized function $\mathcal{F}$ ensures ϵ-differential privacy if for all data sets $\mathcal{D}_1$ and $\mathcal{D}_2$ differing on at most one element, and all $\mathcal{S} \subseteq Range(\mathcal{F})$,

$$Pr[\mathcal{F}(\mathcal{D}_1) \in \mathcal{S}] \leq exp(\epsilon) \times Pr[\mathcal{F}(\mathcal{D}_2) \in \mathcal{S}]$$

The probability is taken over the coin tosses of $\mathcal{F}$.

Dwork *et al.* [138] discussed Definition 5.3 as follows:

- The public parameter ϵ is not fixed and is defined at the convenience of the anonymizer, depending on processing contexts. That is, if the probability that an event is very small, it might be tolerable to increase ϵ by 2 or 3, while if the probability is considered to be close to unacceptable, then an increase by a factor of $e^{0.001}$ might be tolerable.

- Definition 5.3 is independent of any auxiliary knowledge the adversary, or user, may have about the database.

- Definition 5.3 extends to group privacy as well (and to the case in which a user contributes more than a single row to the database). A collection of several users might be concerned that their shared data might leak information, even when a single user does not.

When DP techniques are applied at the data owner side, without the need for a third party, it is called Local Differential Privacy (LDP) [142]. The main idea behind LDP is to allow users to locally perturb their input data. LDP algorithms have been applied in several contexts, and three main practical realizations are introduced, namely:

- Google's RAPPOR solution — it permits to identify popular visited websites (URLs) without revealing any individual user's browsing habits and interests. That is, RAPPOR combines Randomized Response with Bloom Filters to compactly encode massive sets [143].

- Apple's DP technique — it combines a set of algorithms and functions to ensure a perfect differentially private large data-sets. For instance, it relies on the Fourier transform to spread out signal information, and sketching techniques to reduce the dimensionality of the massive domain. This technique was introduced in 2016, at the keynote address of Apple's Worldwide Developers' Conference[32], where the company's senior vice president of software engineering Craig Federighi emphasized that Apple does not assemble user profiles. Afterwards, a patent of Apple's DP technique was filed [144].

- Microsoft's Telemetry collection — it applies an efficient LDP algorithm adapted to repeated collection of counter data, such as daily applications' usage statistics. This solution relies on fixed random numbers to collect data over time [145].

In the literature, DP has also been investigated and applied to several use-cases. In [146], Riboni *et al.* propose a differentially private mechanism to release check-in data to an untrusted recommender. They assume that there is a trusted check-in collector responsible for applying the differential privacy technique before releasing data to the untrusted recommender. This proposal permits to counteract both the recommender and malicious users, that attempt to send fake queries in order to construct locations visited by a target user.
In the same vein, Chen *et al.* consider the problem of publishing useful network structures while preventing an attacker from learning the existence of any single edge, even when the underlying edges are correlated, thanks to the differential privacy technique applied over data streams [147].

[32]https://www.wired.com/2016/06/apples-differential-privacy-collecting-data/

In 2017, To *et al.* proposed a framework for protecting privacy of worker locations, whereby the Spatial Crowdsourcing-server (SC-server) only has access to data sanitized according to differential privacy [148]. Workers subscribe to a cellular service provider (CSP) that provides Internet connectivity. The CSP already has access to the worker locations (e.g., through cell tower triangulation), but as opposed to the SC-server, the CSP signs a contract with its clients, stipulating the terms and conditions of location disclosure. The CSP collects user locations and releases them to SC-servers in a noisy form, according to DP. Recently, Asghar *et al.* presented an application of a differentially private algorithm to release public transport usage [149]. Advantages are multifold. Data can serve to design new value-added services. Users privacy is preserved by avoiding that valuable information about transport usage is extracted from the user's transport card with high privacy disclosure risks like users tracking over one trip or a series of trips.

5.2. Self-Destructing Data Systems

Earlier this year, Kromtech[33] security researchers have discovered that the logistics company, FedEx has stored a huge amount of customers' sensitive data such as scanned documents including passports, driving licenses and security IDs as well as home addresses, postal codes and phone numbers, in a publicly available Amazon S3 bucket. This research has revealed that these data have been collected between 2009 and 2012 which means that customers data have been online for years.

One solution to mitigate this kind of private data breaches is to apply *self-destructing data systems*. These systems are mainly designed to ensure data deletion from storage services and services after a pre-defined timeout.

More generally, self-destructing data can be widely applied to web applications where user's sensitive data can persist in cloud servers indefinitely. The application of self-destructing data mechanisms allows users to regain control over their data such as Facebook private messages, Google docs, private photos on Instagram, *etc.*

Vanish [150] is a self-destructing data solution based on a combination of cryptographic techniques, P2P infrastructures and Distributed Hash Tables (DHTs). Vanish encrypts users' data locally using a randomly generated secret key. Vanish generates secret shares from the key then destroys the local copy of the secret key while storing the generated secret shares across random nodes in the DHT. This latter discards the key shares after a certain time, therefore, keys are lost forever and data are permanently unreadable. On top of Vanish, a Firefox plugin is built for Gmail to self-delete emails while a browser extension is provided to integrate destroyable elements on a website. However, this mechanism suffers from Sybil attacks [151] which consist in compromising the system by collecting secret shares from DHT before reaching their time-out.

[33] https://kromtech.com/blog/security-center

To improve Vanish resistance toward Sybil attacks, several versions of Vanish (Vanish 0.1 and Vanish 0.2) have been introduced [152].

Ephemerizer [153] is a cryptographic protocol proposed by Perlam [153] to ensure the deletion of data after an expiration time. It is based on the use of an ephemerizer which is an entity responsible for generating a set of ephemeral keys while deleting the expired ones. Data are encrypted using these ephemeral keys which makes them non-accessible after the expiration time.

Neuralyzer [154] introduces a data deletion scheme based on the flexible expiration time concept. Instead of being based on a pre-defined expiration time, setting a suitable time for data deletion can be based on a suitable revocation model. For example, after interest in data has been dropped or even after detecting an excessive access to these data. Neuralyzer uses the cache mechanisms of the Domain Name System (DNS). It enciphers data and then splits the decryption key into parts and stores them on different DNS entries. The key is recoverable by all the parties who know which DNS entries are storing the key parts. Obviously, data access automatically extends the lifetime of the key in the DNS entries because the cache is not deleted. However, once the cache is not accessed, it is deleted which implies the destruction of the decryption keys stored in.

5.3. Obfuscation: Private Information Retrieval (PIR)

As introduced in subsection 4.3, obfuscation refers to hiding the intended meaning of a communication process, by making queries and/or responses difficult to understand. This category is mainly considered under a semi-trusted threat model, and it involves the Private Information Retrieval (PIR) as a server-side technique.

5.3.1. Definitions

Private Information Retrieval (PIR) enables users to request data items from a remote database server with no need to reveal which item is retrieved [155, 6, 156, 157]. A trivial solution to that problem is for the user to request the entire database, thus ensuring perfect privacy for the requesting entity, but at the price of high communication overhead. Private Information Retrieval (PIR) schemes are efficient solutions adapted for large databases. They support good privacy level with regard to curious servers' threat models, and as a consequence they are not compatible with personalization of services [6].

Generally, PIR algorithms are interactive schemes between a client and a remote database server, usually classified w.r.t. the privacy guarantees as follows:

- **Computational PIR (cPIR)** – provides privacy against computationally bounded servers. That is, the privacy of the cloud client depends on the computational intractability assumptions.
- **Information-theoretic PIR (itPIR)** – ensures privacy against computationally unbounded servers.

A major issue with computationally-private information retrieval schemes is that they are resource consuming in terms of computation costs. In fact, in order to answer a query, a remote service provider must process all of its entries. Consequently, as pointed out in [158], if the processing algorithm does not process some records, the server may easily deduce that the requesting user is not interested in them. This would reveal to the hosting data server partial information on which record the user is interested in. In the sequel, it is worth noticing that cPIR schemes are not as private as downloading the whole database and retrieving locally the desired entry.

PIR schemes are also classified based on the number of hosting server(s). As such, two mùain categories are distinguished, *i.e., single-server PIR* [159] and *multi-server PIR*, where a set of servers host a distributed replicated databases, and collaborate to answer clients' queries.

5.3.2. PIR Applications

PIR schemes have applications in many proposed privacy-sensitive applications, hence, several works have been proposed [158, 160, 161, 162, 163, 164, 165]. For instance, PIR schemes are widely applied for different location-based services. The main idea is to retrieve nearest neighbor (NN) point of interest w.r.t. the user position at the time of his query. In [166], Attallah and Frikken proposed a privacy preserving location-dependent query processing solution. That is, at the server side, location data are represented through a directed acyclic graph, w.r.t. points of interest stored by the service provider. However, in order to resolve a NN query, the user needs to send a number of queries that is proportional to the depth of the graph. Later, Mittal *et al.* proposed a PIR-Tor, an architecture for the Tor network in which users obtain information about only a few onion routers using private information retrieval techniques [160]. Clients use PIR techniques to download information about only a few relays. PIR prevents untrusted servers from learning any information about the clients' choices of relays. Hence, based on this approach, clients' anonymity is still preserved while ensuring scalability.

Paulet *et al.* introduced a novel privacy preserving query-content framework for location-based services [167]. The proposed framework ensures the privacy of both the LBS server and the end-user, while relying on two main procedures. First, an Oblivious Transfer algorithm is performed to map the private location of the user to a public cell. Second, a PIR algorithm is executed to privately obtain all the POI records in the designated public cell.
Recently, Ullah *et al.* introduced Poshida, a private information retrieval protocol for WSE applications [168]. They propose to obfuscate the user profile collected and maintained by WSE. Several experiments have been conducted to measure the profile exposure level with regards to the collected information by WSE. The authors show that the proposed Poshida protocol is able to hide up-to 85% of the user profile/attributes from curious entities.

5.3.3. Searcheable Encryption (SE) Schemes

SE schemes, considered as a server-side privacy enhancing technique, enable users to delegate keyword search capabilities over encrypted data contents to remote storage servers without disclosing any plaintext keyword [169, 170]. Any SE scheme needs to fulfill the following requirements:

- **keyword secrecy** — the keywords associated with a search query and encrypted data should not be revealed to any unauthorized entities.
- **search pattern secrecy** — search pattern is induced by a search query. The search pattern of a query for keyword W is defined as the information whether a data file contains this keyword.
- **access pattern secrecy** — access pattern refers to the information of search result. By identifying the receiver from the ciphertext, one could guess the purpose of the ciphertext that would leak important information. For instance, if a teacher is sending a ciphertext to group of students then by seeing student category and course names as attributes one has enough information to determine whether the ciphertext is related to examination, grading *etc.* The problem becomes bigger for applications like electronic health record system and some electronic commerce applications.

Two main SE categories can be distinguished, namely *single-owner and multi-user* and *multi-owner and multi-user* scenarios. On one side, the *single-owner and multi-user* category enables a single owner manage his encrypted data and search capabilities over them [169, 171, 172, 173]. On the other side, the *multi-owner and multi-user* scenario enable multiple data owners outsource their encrypted data to remote servers, such that outsourced data contents are searchable by multiple users [174, 175, 176].

Some of the *single-owner and multi-user* based schemes need to grant search access rights to the users via a shared secret key [169, 170, 173]; while the others need to issue search access rights in the form of *trapdoors* [171]. Recently, Attribute-based Encryption (ABE) has been used to design authorized keyword search schemes for multi-owner and multi-user distributed environments. In [176], the authors use plaintext access policies to encrypt keywords and attributes for authorizing the users. This may reveal sensitive information, like access and search patterns of a user to the curious remote servers, such as these information can be used for statistical analysis to break access pattern privacy.

Recently, two main challenging SE design goals should be considered, namely the support of both multi-keyword queries and results' ranking for effective data retrieval, instead of returning undifferentiated results. Indeed, several schemes have been proposed [161, 162, 163, 164, 165]. In [161], Cao *et al.* present a multi-keyword ranked search over encrypted cloud data (MRSE). Among various multi-keyword semantics, they are based on a similarity measure, called *coordinate matching* [177], *i.e.,* as many matches as possible, to capture the relevance of data documents to the search query. Specifically, [161] used the *inner product similarity* [177], *i.e.,* the number of query keywords appearing

in a document, to quantitatively evaluate such similarity measure of that document to the search query. During the index construction, each document is associated with a binary vector as a sub-index where each bit represents whether corresponding keyword is contained in the document. The search query is also described as a binary vector where each bit means whether corresponding keyword appears in this search request, so the similarity could be exactly measured by the inner product of the query vector with the data vector. However, directly outsourcing the data vector or the query vector will violate the index privacy or the search privacy. To meet the challenge of supporting such multi-keyword semantic without privacy breaches, Cao *et al.* propose using secure inner product computation, which is adapted from a secure k-Nearest Neighbor (kNN) technique [178].

5.4. Privacy preserving Computation: Homomorphic Encryption (HE)

As presented in subsection 4.4, privacy preserving computation techniques permit to protect users' privacy while processing over their outsourced data contents. Homomorphic Encryption (HE) schemes, considered as a server-side privacy technique, allow computation operations to be directly performed on ciphertexts by mirroring the corresponding operations on the plaintexts. That is, the decrypted value of computation results on ciphertexts corresponds to the value of operations on the plaintexts.

Conventionally, privacy preserving techniques can be identified under the following common setting. Let Alice and Bob be two semi-trusted communicating parties. Bob holds a private function f where Alice has a set of inputs $\mathcal{S}$, such as $\mathcal{S} = \{x_1, \cdots, x_n\}$. Alice needs to learn the result of $\mathsf{f}(\mathcal{S})$, without disclosing any information from $\mathcal{S}$.
If the secret function f is designed as a homomorphic function, then, a homomorphic encryption scheme can be considered. Indeed, as shown in Figure 9, Alice first encrypts her inputs' set and submits it to Bob. In turn, Bob performs the necessary homomorphic operations upon encrypted inputs and randomizes the resulting ciphertext before sending it back to Alice. Once the resulting ciphertext is received, Alice decrypts it and retrieves $y = \mathsf{f}(x_1, \cdots, x_n)$.

In 1978, Rivest, Adleman and Dertouzous have introduced the concept of privacy homomorphism [179]. This promising idea was motivated by the possibility of querying encrypted databases, *i.e.,* the main setting aimed at allowing each data owner to execute a set of computations (queries, updates, $\cdots$), while ensuring that database' records remain perfectly protected from data leakage attacks.
Afterwards, several homomorphic constructions emerged and numerous algorithms, mainly designed over algebraic groups or rings appeared [180]. Generally, systems, defined over groups, support a single operation. Thus, for over 30 years, cryptographic systems, defined over groups, were presented to enable simple computations over encrypted database, such as Goldwasser and Micali [181], ElGamal [182], RSA [183] and Paillier [184] schemes. These schemes support either adding or multiplying over encrypted ciphertexts, *but not both operations*

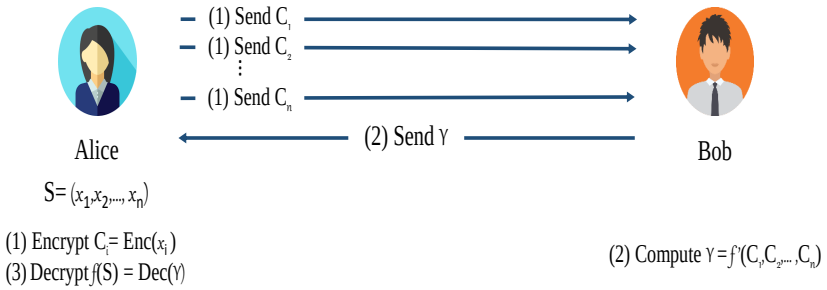

Figure 9: Homomorphic Encryption Flow Diagram

at the same time.

Concurrently, cryptographic schemes defined over rings usually support two operations, *i.e.,* addition and multiplication. As such, a cryptographic mechanism is considered as homomorphic, if the encryption algorithm is a homomorphism, where both the plaintext and ciphertext spaces are groups. In 2005, Boneh *et al.* introduced the first concrete construction of a (fully) homomorphic scheme [185]. Although the proposed scheme permits to perform both operations at the same time, it is limited to an arbitrary number of additions and only *one* multiplication.

5.4.1. Construction of Fully Homomorphic Encryption Schemes

A homomorphic encryption scheme includes four probabilistic algorithms, namely KeyGen, Encrypt, Decrypt and Eval. The KeyGen algorithm generates the public parameters pp and a pair of public and secret keys (pk, sk). The Encrypt algorithm takes as input the public key pk and the message m and outputs a ciphertext CT, while the Decrypt algorithm takes as input the ciphertext CT and the secret key sk and outputs a message m. The Eval algorithm takes as input an evaluation key evl (*i.e.,* note that the evl key can be considered as part of the public key pk), a function $f : \{0,1\}^n \longrightarrow \{0,1\}$, a set of n ciphertexts $\{CT_1, \cdots, CT_n\}$ and it outputs a ciphertext CT_f. We note that the representation of the function f is an important issue, and it is different from one scheme to another.

Fully Homomorphic Encryption (FHE) schemes are categorized into ideal FHE and leveled FHE. An ideal FHE scheme ideally supports calculations for any circuit, even circuits with an unlimited number of operations on ciphertexts, while a leveled FHE scheme only supports a limited number of calculations, based on a given polynomial size circuit. In 2009, Gentry proposed the first Fully Homomorphic Encryption (FHE), performing an arbitrary number of

additions and multiplications [111]. Later, further fully homomorphic schemes were presented [186] [187] [188], [189] following Gentry's framework. Indeed, the ideal lattice-based schemes rely on the hard problem of ideal lattice [190], while schemes defined over integers [186], [188] make use of the hard problem of the approximate greatest common divisor.

5.4.2. Homomorphic Encryption for Personalized Services

HE schemes have numerous applications [13, 191, 192, 193, 6], based on two essential properties, namely *circuit privacy* and *multi-hop homomorphism*. That is, the *circuit privacy* property guarantees that the cloud server inputs, *e.g.,* the f function, remain privacy protected from the client. A number of variants of circuit privacy have been introduced in order to ensure data secrecy against malicious users, where users are allowed to know only few elements about function f, *e.g.,* the size of the circuit. The *multi-hop homomorphism* property permits the output of the Eval function to be used by another homomorphic evaluation function.

HE schemes have widely been applied to recommendation services. In [191], Erkin *et al.* proposed a recommendation system based on HE schemes. The proposed scheme requires encrypting users' private data, *e.g.,* ratings, items, *etc.* relying on Partially Homomorphic Encryption, PHE for short [111]. As such, the recommender is able to process encrypted inputs without access to originally enciphered users' data. In addition, [191] proposal introduces a semi-trusted third entity to perform some extra-computation on encrypted data, in order to assist the recommender and prevent collusion between malicious users and the recommender.
Later, Badsha *et al.* presented a privacy preserving recommendation scheme [192], relying on ElGamal encryption scheme [182], without assistance. However, all entities are obliged to collaborate with the recommender to generate private recommendation for a target user. Afterwards, the authors proposed an extended version relying on the Boneh-Goh-Nissim (BGN) HE scheme [193]. In their proposal, they introduced a third party, to assist users decrypting the enciphered result returned by the recommending entity.

HE based techniques was also applied to enable private search queries. That is, the user submits an encrypted query and the remote untrusted server computes an encrypted answer without ever looking at the query in the clear [194, 195].

6. Channel Side Techniques

Channel-side techniques encompass mechanisms that may act on the security and privacy properties of the set-up communication channel between the server and end-users. They may result in the involvement of a mediating entity, the encryption or a voluntary degradation of exchanged data.

Channel-side techniques include secure communications introduced in subsection 6.1 and Trusted Third Parties (TTP) presented in subsection 6.2.

6.1. Secure Communications

Generally, physical network links do not ensure sufficient security and privacy guarantees. Nowadays, in the era of all-connected, it becomes even impossible to physically prevent pervasive surveillance. Indeed, users' generated data should be protected, namely personal information or sensitive user inputs. However, even access to public resources should be protected through encryption to prevent an attacker from deducing users' patterns of browsing, profiling, service use or extracting identifiers that may be used for future tracking.

6.1.1. Client-Service Secure Communications

To secure communications against pervasive surveillance, several service providers propose to deploy encrypted communication channels. It is important to emphasize that encrypted channels need to be implemented and configured correctly, to ensure a sufficient security level.

Several technologies and protocols have been introduced, namely the well-known Transport Layer Security 1.2 protocol (TLS 1.2)[196] and the Secure Shell (SSH) protocols [197]. Those technologies provide a confidential and conceivably authenticated channel between users and service providers. Indeed, deploying TLS 1.2, or an equivalent secure channel, for every network interaction should consider currently recommended cipher-suites [198]. Both TLS and SSH rely on public key cryptography techniques. Consequently, users and servers are able to set up an encrypted channel with no need to share secrets. TLS 1.2 is based on a public key certificate infrastructure to ensure the authenticity of the server involved in the communication. Conversely, compromised authorities may lead to the security of channels being compromised. On the other side, SSH relies on manual user verification of the service provider's key, which is more resource-consuming for end-users. The SSH protocol also requires to ensure whether end-users are able to perform periodical verification.

A number of technologies may be used for communication within an organisation to ensure the security of transmitted data, as detailed in subsection **??**. For instance, the Internet Protocol Security (IPSec) [199] permits to create secure communication tunnels between networked machines, or between networks connected by public network links. It is recommended that traffic internal to an organisation (local-area network) is encrypted if it may contain user information, such as for performing back-ups or communications between application and database servers [19].

6.1.2. End-to-End Secure Communications

End-to-end encrypted services refer to encrypted communications between end-users, meaning that the encryption layer is added at one end-user and is only cleared at the other end-user. Hence, transmitted data cannot be read by any third party including the service provider. Several services emerged, such as Voice-over-Internet-Protocol (VoIP), electronic mail or instant messaging and social networking that settle communications between end-users.

Service providers usually need to assist users to authenticate them, in order to create an end-to-end encrypted channel. However, it is preferable that the keys used to subsequently ensure the confidentiality and integrity of data never be available to the service providers, but derived on the end-user devices. Meanwhile, several service providers may require having some visibility for either routing data contents to the correct destination, or providing value added services, *i.e.,* w.r.t. users' experience. As such, the minimum amount of information should be exposed to service providers.

Web-sites and applications can approximately identify the location of devices, for instance, based on users' IP addresses. Most IP addresses permit to report the city or metropolitan area, while others may even refer to more specific places. Although most tracking mechanisms are mainly deployed at the application layer, hiding the original IP address is a well-known solution to avoid the simple technique of IP addresses' tracking. The most common method for hiding the remote site's IP address is to use virtual private networks (VPNs) or TOR (The Onion Router), which are introduced hereafter.

- **Virtual Private Networks (VPNs)** — a VPN is defined as a network technology that creates a secure (encrypted) network connection over a public network (for example, the Internet) and a private network of a service provider [200]. The information transmitted between the two communicating sites through the encrypted tunnel cannot be read by anyone else because the system relies on several factors to secure the service provider's private network and the public network through which the remote user connects.
  There are three main network protocols for using VPN tunnels, namely IPsec, the Point-to-Point Tunneling Protocol (PPTP) and Layer Two (2) Tunneling Protocol (L2TP) [199, 201, 202].
  Using VPNs, users' online activities (visited websites, for instance) are protected against disclosures to third parties. VPNs also avoid the geographical restrictions of certain services offered over the Internet. However, it should be noted that the VPN provider may still be able to identify both the user as well as his online activity.
  Several mobile VPN applications have emerged. For instance, Meddle, which is a mobile tool, has been designed to control the network traffic from mobile devices by blocking, filtering and modifying it [203]. Using Meddle, the user can track all the connections established by the mobile

phone. Meddle enables users to get informed about any access to their personal data via a dedicated mobile application. This enables the user to decide whether a particular information should be blocked from being sent over the network or whether this information needs to be changed.
CyberGhost[34] is a full-suite VPN platform, that permits to anonymize and encrypt users' online activity while hiding their IP addresses. With an extensive strategically distributed servers, CyberGhost is a good choice for Small and Medium Business (SMB), freelancers, and end-users to secure their online transactions. CyberGhost introduces a user-friendly interface and offers numerous features, such as IP hiding, a kill switch, malicious website blocking, *etc.*.
NordVPN[35] is a personal VPN service provider. It offers fast connection speed, worldwide connectivity and safe internet access. With over 4.800 servers worldwide, it offers a robust and dependable online protection, with ad blocking and most services expected in a VPN provider.
Later, in 2009, the first release of ExpressVPN[36] is presented, as a virtual private network mobile service that encrypts users' web traffic and hides their IP addresses. ExpressVPN presents one of the most used mobile VPNs, since it provides an easy-to-use interface as well as a wide choice of countries and servers, solid support and lots of security features, *e.g.,* network lock kill switch feature, to ensure that users' activities remain secure and private.

- **The Onion Router (TOR)** — anonymization networks are designed to anonymize Internet communications, *i.e.,* to make it mainly impossible to identify the communication channel between the end-user and the corresponding web-server. To provide this functionality, anonymization networks often rely on a distributed overlay network and onion routing to anonymize TCP-based applications, such as web browsing or Peer to Peer (P2P) networks. The most well known anonymizing networks is The Onion Router, TOR[37] for short.
  TOR is a network of virtual tunnels designed for anonymous web browsing. In onion routing, a proxy servers' chain is used to forward user data to the destination while the proxy link is encrypted. The first proxy in the chain is called the TOR proxy. It is in charge of randomly selecting at least 3 different intermediary TOR nodes, called onion routers upto the destination. Afterwards, the TOR proxy encrypts the message several times relying on symmetric keys, so that each intermediary TOR node has to remove one encryption layer for the destination to get the cleartext. Each intermediary node only knows the address of the previous and

[34] https://www.cyberghostvpn.com/en_US/
[35] https://nordvpn.com/
[36] https://www.expressvpn.com/
[37] https://www.torproject.org/projects/torbrowser.html.en

succeeding nodes involved in the routing path. Thus, the identity of the original source remains hidden to anyone who observes the communication, except over the first hop.
Anonymity protection is the primary scope and functionality of anonymous networks. Indeed, they hide the user's original IP address and/or other (direct or indirect) identifiers, thus masking his business activity.
There is a fundamental difference between anonymized networks and VPNs. Indeed, although VPNs can hide the user's original IP address, they do not provide anonymity (as the VPN provider is able to identify users of the service).

A number of technologies have been proposed, implemented and standardised to different extents to provide end-to-end confidentiality. For example, the Pretty Good Privacy (PGP)[38] software as well as S/MIME standards [204] may be used to protect email correspondences end-to-end.
Recently, an end-to-end encrypted e-mail technology, called PreVeil, has been introduced and patented by Popa *et al.* [205]. PreVeil is built based public-key cryptographic primitives. The gist is that every user has a pair of private and public keys. That is, the sending user has to first encrypt the e-mail under the receiver's public key to ensure that it is the only entity allowed to decrypt it. In addition, PreVeil permits users to access emails on multiple devices by securely transferring the private key, and also appoint so-called approval groups to securely recover the private key when lost. In the same vein, several mobile applications such as Crypto Phone[39] and Signal[40] have been proposed to provide end-to-end encrypted communications [206].

In [206], Ermoshina *et al.* reviewed around 30 end-to-end encrypted messaging protocols, w.r.t. security and privacy guarantees. The analyzed projects propose several solutions to secure remote data storage. Indeed, despite the guarantees of *no personal information collection*, some projects still store important amounts of data on the servers (*i.e.,* usage statistics, device information, keys, user-names or friend relations). Hence, it is always explained by emphasizing the necessity to propose better user experience based on the collected usage statistics). A related issue is the powerful double narrative on end-to-end encryption [206]. On one side, the discourse on empowerment and better protection of fundamental civil liberties is very strong. On the other side, several service providers show a need to defend themselves from the *encryption is used by jihadists*-type allegations. This narrative is sustained by previous and current ones about decentralized technologies and peer-to peer, w.r.t. *empowering-yet-illegal* tools. These issues are taking place in the broader context of discussions about governance by infrastructure and users' privacy preservation, as the Apple vs. FBI case and WhatsApp proposing, since April 2016, encryption by

[38] https://www.openpgp.org/
[39] http://www.cryptophone.de/
[40] https://signal.org/

default.

6.2. Trusted Third Party Mechanisms

For a while, the usage of unique identifiers across wide system, raised a critical issue. That is, it enables to easily correlate records with no need for data owners' consent, introducing several security and privacy issues. Indeed, a set of curious entities may be made powerful over users with the capability to trace them, combine their datasets and infer some information. Even worse, any data breach reveals fully identifiable and linkable personal information. Literally, several mechanisms, relying on the use of anonymizers and pseudonymisers, emerged. They are known as Trusted Third Party (TTP) techniques [207, 208, 209, 210, 211] and are generally considered under the trusted model.

Most well known TTP-based techniques are proxy servers, acting as anonymizers. Proxy servers are intermediary entities placed between communicating parties. Two main types of proxies may be considered (*cf.* Figure 10):

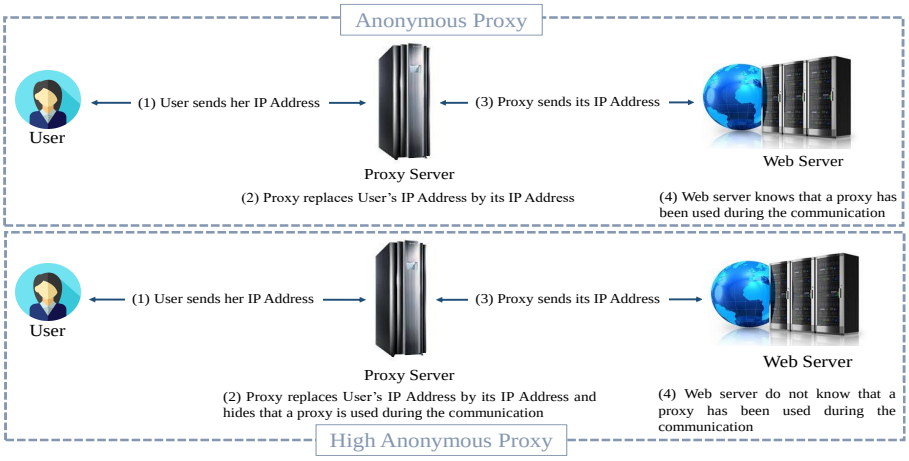

Figure 10: Proxy Anonymizing Servers Flow Diagram

- **anonymous proxy** — it hides the user's IP address by replacing it by its own address. However, other entities can conclusively figure out that a proxy server is set up in the communication.
- **high anonymity proxy** — similarly to the anonymous proxy, it hides the client's IP address by presenting its own address. Furthermore, it also hides that a proxy is being used in the communication.

The main issues raised by both TTP-anonymizing based techniques is that personalized services cannot be provided, as TTPs forward data such as queries, tags, ratings associated to several users, to the corresponding service providers.

In addition, the tracking problem is shifted to the proxy servers' level, as the matching of all pseudonyms, queries and results is performed at this level.

Concurrently, several approaches, called pseudonym systems, have been proposed. In fact, they rely on cryptographic methods to generate different context-specific pseudonyms that are unlinkable. In order to enable associating different outsourced data sets, a central entity, can later be asked to associate pseudonyms to a specific user on a case per case basis [212].
A pseudonym system is an interesting solution that permits to guarantee auditability and ensure confidentiality of data w.r.t. involved entities. In fact, each user is able to generate a number of pseudonyms, derived from his private key and arbitrary scope strings. He is, then, able to prove that he knows the secret key used to derive this pseudonym. As detailed in [212], a pseudonym system has to fulfill three main security properties: the key extractability, collision resistance and unlinkability. First, key extractability ensures that service providers are guaranteed that a honest user effectively knows the used secret key to derive such pseudonym (respectively proof). Second, the collision resistance property guarantees that for each context, any two users should have different pseudonyms with probability. Finally, unlinkability ensures that users cannot be linked/identified across contexts. That is, similar to several TTP-anonymizing techniques, relying on pseudonyms systems, the service provider cannot deduce the identity of the user, but merely the identity of the third party involved in the communication. However, as the trusted entity is the main central entity, it can easily correlate data with their respective users, while tracing users' activities, based on data exchanges.

Recently, Camenisch and Lehmann [207] presented a blind-convertor system (*i.e.,* the convertor is the central entity), where the conversion process, executed by the convertor, is performed in a privacy-preserving (*i.e.,* anonymous and unlinkable) fashion, with no need for learning the pseudonyms or the identity of the concerned user. The authors extended their work by presenting in [210] a system where user-centric audits are set-up by an oblivious convertor, which is still the main central entity. This appended auditability feature is a very important property, for compliance with legal and business requirements [4].

While several systems permit to satisfy transparency and auditability requirements based on a user-centric fashion, a Blockchain-based Data Usage Auditing system, BDUA, is proposed in [211]. The proposed system ensures a controlled yet privacy preserving exchange of distributed data, such that a set of authorized auditing entities are able to conduct an accurate auditing activity relying on registered blockchains' transactions . The BDUA scheme is an unlinkable pseudonym system where a selected convertor obviously collaborates with each user to derive individual pseudonyms for each server. That is, the convertor cannot learn the derived pseudonyms, but is still the only entity that can link pseudonyms together without learning the particular user or pseudonym for which such translation is requested. It also provides an accurate

privacy-preserving auditing process based on blockchain transactions.

Table 5 presents a comparison between several pseudonym schemes, w.r.t. functional requirements.

Table 5: Comparison between pseudonym schemes

Scheme	blind conversion	multi-convertor	user auditing	public auditing
[208]	×	×	✓	✓
[209]	×	×	✓	✓
[207]	✓	×	×	×
[210]	✓	×	✓	×
[211]	✓	✓	✓	✓

✓ and × indicate respectively whether the requirement is achieved or not.

To ensure privacy-preserving personalized identity-management systems for healthcare applications, another approach introduced by Verheul *et al.* in [213] consists on the development of Polymorphic Encryption and Pseudonymisation (PEP for short). Based on the PEP technology, end-users send their data contents in polymorphically encrypted form to service providers. When needed, each end-user is able to make (parts of) these data available (decryptable) for some particular/selected entities, for specific analysis purposes. Consequently, users remain in control, and can monitor which parts of their outsourced data are used where, by whom and for which purposes. To resolve the privacy-personalization trade-off, the authors introduce the polymorphic pseudonym system that permits each end-user to automatically have different pseudonyms at different contexts. These pseudonyms can only be de-pseudonymised by authorized entities (*e.g.,* medical doctors) who know the original identity of the end-user. An open source prototype implementation of the PEP framework is made available[41]. Several experiments are actually conducted in a real-life medical research project at the Radboud University Medical Center in Nijmegen, The Netherlands.

7. Comparison between PETs

Table 6 provides a detailed comparison between the reviewed PETs, w.r.t. involved approaches, supported trust and architectural models, main drawbacks, suitability for personalized services and supported use cases. In the following, we discuss summarized approaches w.r.t. two factors: trust model based classification, detailed in subsection 7.1 and use-case based classification, detailed in subsection 7.2.

[41] https://pep.cs.ru.nl/

Table 6: Comparison between Privacy Enhancing Technologies and Approaches

PET groups	Categories	Approaches	Trust Model	Architectural Model	Drawbacks	Adapted to Personalized Services	Use-Cases
User Side	anti tracking		semi-trusted	client-server	non-personalized services	×	web services
	privacy preserving certification	anonymous credentials group based signature attribute based signature	semi-trusted	client-server	computational overhead infrastructure requirements	✓	e-health applications e-voting smart-cities applications
	obfuscation	data perturbation	untrusted	distributed	traffic overhead privacy-utility trade-offs	✓	pervasive applications recommendation services
	privacy preserving computation	SMC	untrusted	distributed	several users must collaborate all participants need to be present collusion attacks	✓	pervasive applications web search smart-cities applications
Server Side	Statistical Disclosure Control	anonymization Differential Privacy	semi-trusted	client-server	privacy-utility trade-offs	✓	vehicular applications well-being applications geo-social applications Intelligent Transport Systems web search
	self-destructing data systems		semi-trusted	distributed	Sybil attacks	×	
	obfuscation	PIR	semi-trusted	client-server	non-personalized services computational overhead	×	
	privacy preserving computation	homomorphic encryption (HE)	semi-trusted	client-server, distributed	computational overhead	✓	recommendation services pervasive applications
Channel Side	secure communications	C-S communications	trusted	client-server	key management (e.g. PKI)	✓	web search e-banking messaging applications
		end to end	trusted	client-server (VPN)	activity identification by provider (VPN)		
			untrusted	distributed (TOR)	traffic overhead (TOR)	×	
	TTP	high anonymous proxy anonymous proxy convertors	trusted	TTP	users must trust an external entity collusion attacks infrastructure requirements	×	

7.1. Trust Model based Discussion

From Table 6, we note that three prominent approaches are considered under the trusted model, namely the TTP schemes, client-service secure communications and VPNs as a network-level anti-tracking tool.

Note that a VPN-driven communication can also blend under client-service secure communications' approaches [18] as both of them are relying on a secure channel enabling to mitigate external adversaries that try to get extra-knowledge from exchanged information. However, while a VPN-driven communication ends at a VPN provider, a client-service secure communication ends at the service provider. This means that a VPN-driven solution requires the user to trust the VPN provider of his choice and to manage the secure channel in between. For the client-service secure communications, a user is required to trust each service provider he is interacting with, and to establish a secure channel relying on external PKI entities he is also required to trust. Like VPN solutions, the TTP approaches consider a trustworthy intermediary server which, unlike VPNs, is anonymizing or pseudonymizing the traffic. Note that TTP schemes and client-service communications are still widely deployed techniques to provide some privacy-enhancing services [18, 210, 198, 19]. As stated in Section 6.2, TTP-anonymizing based techniques cannot provide personalized services, as intermediate servers are usually deployed to mix data associated to several data *i.e.,* queries, ratings, *etc.* before forwarding to service providers. Meanwhile, new emerging convertor-based systems relying on pseudonym systems can be deployed for some pervasive applications to resolve the privacy-personnalization trade-off, namely for data transfer scenarios [211]. Conversely, client-server secure communications enable personalized services, mainly for private web-search services and/or a set of pervasive applications *e.g.,* e-commerce and e-banking services. Based on the trust relationship between customers and their service providers, they support services that are highly adapted to users' usages and preferences, but offer no protection against curious providers.

Semi-trusted models include six different techniques, deployed either on the client side or the server side. They involve application-level anti-tracking tools and privacy-preserving certification schemes for the user-side techniques and Statistical Disclosure Control, PIR solutions, self-destructing data systems and homomorphic encryption schemes for the server-side techniques.

All these different techniques assume that service providers are honest in the sense that they genuinely perform the computations expected from protocols. However, they are curious as they try to infer any extra-information to closely identify and/or profile users [6]. Specifically, obfuscation and homomorphic encryption are adapted to applications requiring search over data stored on third party servers such as cloud storage services [214] or content delivery networks [215]. Statistical Disclosure Control techniques and privacy preserving certification schemes are mostly adapted to well-being applications such as e-health applications. For instance, data can be collected from authenticated users while preserving their personal data.

PETs, considered under the semi-trusted model, mostly rely on the client-server architectural model. Consequently, each client, at his level, goes to tech-

niques that disclose the minimum amount of identifying information to their service providers, thus, preventing them to build precise users' profiles. In the sequel, these mechanisms generally suffer from privacy-utility trade-offs, such that they may not be adapted to several personalized services (*i.e.,* web search applications). However, they are still promising solutions for pervasive applications, namely vehicular and geo-social applications [45, 139].

Finally, four different techniques are considered under the untrusted model namely network-level anti-tracking tools *i.e.,* TOR, secure multi-party computation schemes, end-to-end secure communications and perturbation techniques. Most of these techniques rely on distributed architectures, except for anti-tracking tools. In fact, anti-tracking are deployed under the client-server model, where each end-user is in charge of setting up appropriate tools to prevent the disclosure of extra-personal data [48]. Although providing perfect privacy, they are not adapted to personalized services, as they hide any useful information that help to build users' profiles, from service providers. Similarly, privacy preserving computation and end-to-end communications, based on distributed architectures, provide non-personalized services to end-users. In addition, they generate high computational and communication overheads, due to the number of participating parties that need to cooperate. Thus, they are suitable to many pervasive applications and services, involving resource-constrained devices [19, 104, 106, 115]. SMC techniques are suitable in smart cities applications which do not involve resource constrained devices. For instance, these techniques can be applied to collect data from smart homes in order to monitor energy consumption without leaking private data about users [106]. End-to-end communications are widely applied in messaging applications to ensure private communications between participants [206].

7.2. Use-Case based Discussion

From Table 6, we note that PETs can be classified into two main families, w.r.t. their ability to support personalization features. That is, the first family is not adapted to personalized services while the second one can be adapted to several settings, applications and environments permitting to provide privacy-preserving personalized services.

On one side, the first family, non-adapted to personalized services, involve four different approaches, namely self-destructing systems, PIR solutions, end-to-end secure communications and a set of anti-tracking tools, mainly TOR and some anti-cookies tools. Most of these techniques are deployed, in case end-users are concerned about their privacy and do not trust their service providers. Thus, they choose to gain privacy preservation while giving up personalized services ease. While some techniques, like self-destructing systems or PIR solutions, require the honest processing and collaboration of the service providers to guarantee full privacy, they are not adapted to personalized services.

On the other side, the second family includes techniques that are adapted to different categories of personalized services.

First, PETs adapted to web search include three prominent approaches, *i.e.,* SMC schemes, client-server secure communications and SDC approaches. Although the two first techniques are based on different architectural and trust models, they both guarantee the privacy-preserving search feature. Thus, they counteract data leakages by curious providers and external attackers respectively. For instance, in a secure collaborative computing setting, a set of users co-operate to send an aggregated query to the web-search provider, instead of sending individual queries. The main issues of this approach comprise the high computation and communication overhead and the relevance of returned results that mainly depend on the homogeneity of the profiles of group's members.
Note that application-level anti-tracking tools, *i.e.,* anti-cookies, are also widely installed by end-users. However, as explained in section 4.1, tracker-blockers do not necessarily provide privacy of end-users against curious providers. They are used in most-cases to block undesirable ads and pop-up advertisements.
Differently, SDC approaches are extensively deployed by web-search providers to ensure the privacy of their clients, by anonymizing users' stored queries. These databases are very essential for web-search providers and are used for statistical and economic purposes. Thus, SDC presents the main deployed server-side techniques, as they permit to resolve the privacy-utility trade-off and they afford providers to escape the GDPR [4] regulation as anonymized data are no longer processed as personal data.

Second, techniques adapted to pervasive applications mainly involve five approaches, *i.e.,* privacy preserving certification mechanisms, data perturbation solutions, SDC techniques and privacy preserving computation including SMC and HE schemes. In ubiquitous environments, the choice of the most efficient PET depends on both trust and architectural models. For instance, smart-cities applications, considered under opportunistic networks, generally rely on users' collaborative secure computation schemes, while other e-commerce applications, considered under a client-server architecture, generally rely on privacy preserving certification or secure C-S communications.

Third, techniques adapted to recommendation systems include three main approaches namely data perturbation solutions, secure multi-party computation techniques and homomorphic encryption schemes. As detailed in section 3.1, recommendation services enable users to receive diverse personalized recommendations, w.r.t. their close interests, mainly via an entity or a set of entities, called recommender(s). Thus, to preserve their privacy while ensuring an acceptable processing and communication overhead, data perturbation is considered as one of the leading PETs. Indeed, it is achieved through the injection of dummy data records/ratings or the suppression of redundant/non-genuine ones. While data perturbation is a very interesting approches to enhance privacy of users, it suffers from the privacy-utility trade-off, and several works proposed to analytically study the impact of the adoption of those methods [100].

Even if most of the reviewed techniques and approaches permit to provide privacy-preserving personalized services in various domains and environments, it is almost unpredictable which technologies would raise new privacy issues in the future. Thus, it is important to consider efficient privacy metrics to envision the privacy-personalization intersection set [15, 16]. In addition, the emerged new data analysis technologies may provide new burdens between efficient personalization techniques and privacy (further discussion is provided in Section 8). For instance, data centers are now able to process large amounts of data, thus increasing the possibilities to infer new data through data crossing with other services and applications. Considering powerful personalization techniques, deployed by Google, Facebook or even Amazon, enhanced by the huge databases of users' profiles they are holding, it is trivial that these companies may provide relevant and accurate contents and services to their clients. As predictions on users' preferences are more and more precise, data leakage attacks will consequently drive critical concerns (more details are discussed in Section 8).

8. Open Issues and Research Challenges

From the recently mediated privacy violations, it is to be noted that both research results are still at their early-stage (*i.e.,* not yet providing operational products) and solutions to privacy violations are seemingly beyond effective technical implementations. Indeed, efficient privacy preserving techniques need to meet a set of transversal legal and economic requirements while improving end-users' experience.

8.1. Technical Challenges

Considering identified privacy properties, presented in section 2, several technical challenges need to be further investigated. In this section, we point out some technical open issues and we give various research directions.

8.1.1. Privacy-sensitive Auditing Tools

A minority of works addressed transparency, *a.k.a.* auditing concerns, which have been emphasized by recent regulations and laws [210, 211, 216, 217]. For instance, in [210], Camenisch *et al.* introduce a controlled yet privacy preserving exchange of distributed data scheme, such that each data owner is able to conduct an accurate auditing relying on a public logging system. Nowadays, blockchain applications are becoming increasingly prevalent, to several use-cases and scenarios namely for highly distributed and decentralized settings. The philosophy underlying the blockchain technology is that records are *shared by all network nodes, updated by miners, monitored by everyone, and owned and controlled by no one* [218, 219]. In 2018, in [211], an accurate privacy-preserving auditing process based on blockchain transactions is proposed. The auditing scheme may be conducted by the data owner himself and/or a set of authorized entities. The auditing process is achieved thanks to transactions being

registered by each of the involved participating entity, and enciphering of linking information with respect to a multi-level attribute based encryption. Despite the expressiveness and power of the blockchain and smart contracts, the present form of these technologies lacks **transactional privacy**. The whole sequence of actions provided at the smart contract are propagated across the network and/or recorded on the blockchain. Thus, they are publicly visible. Even though entities are able to generate new pseudonymous public keys to increase their anonymity, the values of all transactions and balances for each (pseudonymous) public key are publicly visible. Further, recent works have also demonstrated de-anonymization attacks by analyzing the transactional graph structures of cryptocurrencies [220, 221]. The lack of privacy is a major concern towards the wide adoption of blockchains, mainly for financial transactions that are considered by many individuals and organizations as being highly secret. To mitigate these concerns, several approaches emerged [222, 223, 224, 225]. On one side, privacy-preserving crypto-currencies and confidential transactions, appeared [224, 223]. On the other side, privacy preserving smart contracts have been introduced [222, 225]. To bring privacy to smart contracts, non-interactive zero-knowledge (NIZK) proofs have been proposed as a tool to enable complex smart contracts that do not leak the user inputs [222, 226].

8.1.2. Privacy-sensitive Data Collection and Inference

Machine Learning (ML) techniques have been widely considered as an efficient and powerful AI approach to support contents and services' personalization. That is, ML techniques enable to collect and infer accurate information about users and thus building representative user's profiles, based on their interests, movements and habits. Hence, various works applied ML mechanisms for personalized recommendation services, web-search engines and a lot of pervasive applications [227, 228, 229, 230]. Several solutions mainly stressed on systems' performances and omitted privacy issues that may be raised due the massive collection of users' data.

The main idea in proposed private Machine Learning solutions consists in two steps. First, a ML model is urged to collect users' data in a privacy preserving fashion. Commonly, the process of inductive learning (by the machine) can be explained as a search process for general descriptions aiming at predicting output data based on the analyzed input data. Second, the designed model is used to generate personalized services [231, 232]. Thus, the main concern consists in defining privacy-enhancing cryptographic methods to meet an agreement between privacy, efficiency and quality of experience. Indeed, very limited works have presented interesting results to achieve a good trade-off. Indeed, recent advances in cryptographic systems have to be applied while considering both efficiency and effectiveness [233, 234, 235, 236, 237, 217].

8.1.3. Privacy-sensitive Techniques for Ubiquitous Environments

Mobile personalized services are gaining an expanding interest, following the ubiquity ability of mobile devices. However, due to the resource-constrained environments under consideration, the design of security and privacy-enhancing techniques require lightweight mechanisms, in order to achieve satisfying user-experience in mobile personalized applications and services [238, 236].

Therefore, it is important to investigate new research directions for developing and designing lightweight privacy preserving personalized applications, w.r.t. processing and communication overheads, adapted to limited-resources devices. One possible direction involves advanced trusted hardware, namely Intel Software Guard Extensions, *i.e.,* Intel SGX[42]. This trusted hardware is able to ensure secure computations with minimal processing overhead.
In addition, energy consumption has to be taken into consideration during the design of security mechanisms for mobile services. It is agreed that battery energy is one of the most critical resource of mobile devices [13]. Indeed, battery energy represents the main functional factor that straightly affects the user experience. Furthermore, there is a need for an accurate prediction of adversaries' knowledge. In fact, very protective techniques, based on conservative assumptions, can engender high computational costs and communication overhead while leading to inefficient quality of service. In [239], Canetti formalized the concept of Universally Composable (UC) security for proving the security of a complex system provided with many interactions between entities and algorithms. In a nutshell, when relying on UC security, the execution of a designed protocol is compared to the execution of an ideal protocol. The designed protocol is considered as UC-secure, if its execution is stable w.r.t. the ideal protocol execution instance. Based on this definition, [239] formulates a composition theorem that states that any UC-secure protocol is also secure if it is composed of other UC-secure protocols [240].

8.2. User Experience Challenges

User experience is the main pillar to both defining the perimeter of private information and the utility over the adoption of PETs. Regarding the first concern, several mediated cases in the past, such as Kodak cameras[43], Google glasses[44] or LG-TV[45], revealed how much data can be intrusively collected about people, with a vague but threatening picture related to possible subsequent exploitation and consequences over peoples' life. The perception of which data can be released, to which entities, differs from one user to another, depending on different factors, namely social status, culture, context, *etc.* In 2017, a

[42] https://software.intel.com/en-us/sgx

[43] https://timeline.com/how-the-first-mass-market-camera-led-to-the-right-to-privacy-and-roe-v-wade-4fb4cd87df7a

[44] https://www.theguardian.com/technology/2013/mar/06/google-glass-threat-to-our-privacy

[45] http://www.trustedreviews.com/news/smart-tv-privacy-problems-vizio-samsung-lg-sony-panasonic-2952175

survey, covering around 2,000 representative French Web users, was conducted in view of analyzing the impact of abusive data collection practices on users' attitudes [241]. The survey showed that Web users tend to become more vigilant, willing to share part of personal data according to the service category (bank, social media, purchasing web site, mobile operator, *etc.*). Moreover, the survey revealed that 10% of web users would prefer not to share any of their data if they had the choice. As a consequence, the first challenge is to be able to conceive solutions that permit to provide flexible protection adaptable to each user's preferences and contexts. The second challenge is to find a compromise between users' privacy, efficient users' experience and data utility [233, 234, 235]. For instance, in [234], privacy is modeled relying on negative utility associated with each released information. Thus, based on each user's knowledge, Massaguer *et al.* [234] proposed a distributed solution that permits to maximize the utility of released information while meeting the privacy requirements of users. Recently, Wang *et al.* studied dependencies between sensitive data and useful data [235]. They proposed closed-analytic formal expressions for the privacy-utility tradeoffs for dependent sensitive and useful information w.r.t. mutual data and Hamming distortion as the respective privacy and utility metrics.

Another key aspect for better user experience is to provide ergonomic and user-friendly applications, so the users do not go through complex settings for adapting the technology to their own preferences. In addition, as discussed in subsection 8.1, the provision of transparency and monitoring tools are of important interest, for helping users verifying whether their data are processed according to their expressed privacy preferences, and for getting users' confidence.

8.3. Legal Challenges

The GDPR is not the only legal text to take into consideration. Several regulations are currently under discussion at the European (EU) level, mainly the proposal for EU regulations on the free circulation of data (non-personal data[46]) and the e-Privacy regulation proposal[47] [242]. Indeed, [242] raises vigorous negotiations between the stakeholders on which data can be qualified as *non-personal* (vs personal), and the definition of a *mixed data set* (both personal and non-personal). The objective of the e-Privacy proposal is to extend the Directive on Privacy and Electronic Communications (Directive 2002/58/EC) to Over-The-Top services like WhatsApp and to metadata. In an undetermined future due to the vigorous negotiations, e-Privacy Regulation is expected to be a lex specialis added to the new GDPR, enforced since May, 25^{th}, 2018. However,

[46] Proposal for a regulation of the European Parliament and of the Council on a framework for the free flow of non-personal data in the European Union, Brussels, 13.9.2017, COM(2017) 495 final.

[47] Proposal for a regulation of the European Parliament and of the Council concerning the respect for private life and the protection of personal data in electronic communications and repealing Directive 2002/58/EC (Regulation on Privacy and Electronic Communications), Brussels, 10.1.2017.

it seems that this goal will not be achieved, mainly because of intense lobbying and the upcoming elections of Members of the European Parliament (MEPs).

With regard to the privacy-preserving personalized services, one has to understand which text should apply to which category of approaches. For instance, approaches like anonymization and obfuscation (*i.e.,* data perturbation) leading to revealing anonymized data to the providers are within the scope of the GDPR, such as requirement and compliance of the providers to respect privacy-preserving computation, privacy-preserving certification providing only encrypted personal data. Finally, whatever the solution, as metadata are revealed, providers are required to conform to e-Privacy. E-Privacy also adds a PET per default to internet browser and to electronic communications devices.

With the perspective of conducting auditing activities, it is of interest and challenging from a technical point of view, to conceive useful auditing tools. These tools permit to guarantee users, in a user-friendly way, that their consent is respected during data collection, processing, storage and analysis and organizations that they comply with their accountability obligation. Moreover, according to the use cases, some technical inspection procedures are required for re-identifying a malicious user, which raise challenging issues due to the multitude of auditing parties (*i.e.,* data owners, authorized authorities, $\cdots$), and fulfill several GDPR requirements, mainly the obligation of integrity and confidentiality, the principles of data protection by default and the *right to be forgotten* or in general *privacy by design* enforced by the GDPR.

8.4. Economic Challenges

Several challenges to adopt privacy-preserving technologies by economic actors, are identified. Analyzing the survey conducted on French web users [241], the authors stated that from an economic point of view, users' behaviors reflecting an outright refusal to share their personal data represent a risk which cannot be ignored by data driven firms. However, it is important to note that individual protection strategies (management of digital identities, ad blockers' settings, accounts' privacy parameters) may also stimulate economic activities. For instance, the widespread use of PETs, seen as a threat by publishers, is in fact accompanied by an increase in economic online activities such as e-commerce. Indeed, studies such as [243] shows that Internet users with privacy concern that have installed PETs are purchasing more frequently on Internet. Giving people a means of protecting themselves could strengthen their subjective feeling of individual empowerment. First, in order to meet new regulations' requirements, it may be necessary to re-design or at least revise actual information systems by data processors and collectors, thus leading to extra costs. Indeed, they may eventually be required to adjust their Privacy Impact Assessment (PIA) [244], or bear labels' costs. Even higher costs are required to actually conceive and deploy the privacy-by-design principle.

Second, several privacy preserving techniques rely on reduced amounts of collected personal information. Thus, mainly Small and Medium Enterprises (SME) do suffer from the incompatibility between their business models and data minimisation requirements, pushing them to review their planning for current and

future projects [245].
Third, SMEs mainly suffer from the problem of collecting users' consents because they do not directly interact with customers.

9. Conclusion

This paper reviews PETs technologies under concrete considerations of some updated applications and services which require user-based personalization. First, a classification of such services is proposed, along with their respective privacy threats, adversaries and requirements, after a short analysis of their inherent privacy issues. Second, a taxonomy of Privacy Enhancing Technologies (PETs) is presented where three different groups and eight categories have been identified, according to which entity is mainly involved in the privacy-preserving decision, which entity is supporting the main cost for privacy, and whether the channel between the client and the server is involved. Indeed, user-side, server-side and channel side techniques are detailed w.r.t. to eight families namely: anti-tracking, anonymous certification, privacy preserving computation, self-destructing systems, secure communications, trusted third parties and obfuscation techniques. The paper reviews each of these families with the main approaches available in the literature or in industrial solutions, and it establishes a comparison among them, based on the trust models being supported, and the services of interest that could implement such PET technologies. Afterwards, the remaining challenges about the users' privacy preservation are discussed under the multi-disciplinary prism, considering technical, social (*i.e.,* user experience), legal and economic concerns. Finally, several research directions are identified, namely privacy sensitive data collection mechanisms, with a focus on the usage of ML techniques, privacy-preserving auditing platforms and the ubiquity-efficiency trade-offs.

PETs are promising technologies that fulfill users' requirements regarding privacy especially with the emergence of privacy legislation while enabling service providers and third parties (*e.g.,* advertising companies) to provide optimal user experience and quality of service. Nevertheless, as the trade-off between privacy preservation and personalized services has not been solved yet, PETs are slowly adopted by companies. Thus, further research efforts have to be made in order to apply PETs in real world applications.

Acknowledgements

This paper is supported by the chair Values and Policies of Personal Information, Institut Mines-Télécom, France, and European Union's Horizon 2020 research and innovation programme under grant agreement No 830892, project SPARTA. Our many thanks go as well to Antoine Dubus, Jonathan Keller, Claire Levallois-Barth, and Martin Quinn for their useful and relevant comments on the multi-disciplinary discussion.

References

[1] D. Reinsel, J. Gantz, J. Rydning, Data age 2025 - the digitization of the world, International Data Corporation White Paper.

[2] C. Bettini, D. Riboni, Privacy protection in pervasive systems: State of the art and technical challenges, Pervasive and Mobile Computing 17 (2015) 159–174.

[3] J. Čas, Ubiquitous computing, privacy and data protection: Options and limitations to reconcile the unprecedented contradictions, in: Computers, privacy and data protection: An element of choice, Springer, 2011, pp. 139–169.

[4] Regulation(EU), 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation), ojeu l 119/1 of 4.05.2016., 2016.

[5] A. N. Toosi, R. N. Calheiros, R. Buyya, Interconnected cloud computing environments: Challenges, taxonomy, and survey, ACM Computing Surveys (CSUR) 47 (1) (2014) 7.

[6] N. Kaaniche, M. Laurent, Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms, Computer Communications 111 (2017) 120–141.

[7] P. Zhang, M. Zhou, G. Fortino, Security and trust issues in fog computing: A survey, Future Generation Computer Systems 88 (2018) 16–27.

[8] J. H. Ziegeldorf, O. G. Morchon, K. Wehrle, Privacy in the internet of things: threats and challenges, Security and Communication Networks 7 (12) (2014) 2728–2742.

[9] J. Heurix, P. Zimmermann, T. Neubauer, S. Fenz, A taxonomy for privacy enhancing technologies, Computers & Security 53 (2015) 1–17.

[10] X. Shen, B. Tan, C. Zhai, Privacy protection in personalized search, in: ACM SIGIR Forum, Vol. 41, ACM, 2007, pp. 4–17.

[11] D. Slamanig, C. Stingl, Privacy aspects of ehealth, in: Availability, Reliability and Security, 2008. ARES 08. Third International Conference on, IEEE, 2008, pp. 1226–1233.

[12] D. Eckhoff, I. Wagner, Privacy in the smart city—applications, technologies, challenges, and solutions, IEEE Communications Surveys & Tutorials 20 (1) (2017) 489–516.

[13] C. Wang, Y. Zheng, J. Jiang, K. Ren, Toward privacy-preserving personalized recommendation services, Engineering.

[14] E. Toch, C. Bettini, E. Shmueli, L. Radaelli, A. Lanzi, D. Riboni, B. Lepri, The privacy implications of cyber security systems: A technological survey, ACM Computing Surveys (CSUR) 51 (2) (2018) 36.

[15] E. Toch, Y. Wang, L. F. Cranor, Personalization and privacy: a survey of privacy risks and remedies in personalization-based systems, User Modeling and User-Adapted Interaction 22 (1-2) (2012) 203–220.

[16] J. Parra-Arnau, D. Rebollo-Monedero, J. Forné, Privacy-enhancing technologies and metrics in personalized information systems, in: Advanced Research in Data Privacy, Springer, 2015, pp. 423–442.

[17] A. Kobsa, Privacy-enhanced personalization, Communications of the ACM 50 (8) (2007) 24–33.

[18] J. Parra-Arnau, D. Rebollo-Monedero, J. Forné, Measuring the privacy of user profiles in personalized information systems, Future Generation Computer Systems 33 (2014) 53–63.

[19] G. Danezis, J. Domingo-Ferrer, M. Hansen, J.-H. Hoepman, D. L. Metayer, R. Tirtea, S. Schiffner, Privacy and data protection by design-from policy to engineering, arXiv preprint arXiv:1501.03726.

[20] L. Song, R. Shokri, P. Mittal, Privacy risks of securing machine learning models against adversarial examples, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 241–257.

[21] F. Farokhi, M. A. Kaafar, Modelling and quantifying membership information leakage in machine learning, arXiv preprint arXiv:2001.10648.

[22] M. Al-Rubaie, J. M. Chang, Privacy-preserving machine learning: Threats and solutions, IEEE Security & Privacy 17 (2) (2019) 49–58.

[23] A. Pfitzmann, M. Hansen, A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management.

[24] H. De Silva, P. Jayasinghe, A. Perera, S. Pramudith, D. Kasthurirathna, Social media based personalized advertisement engine, in: Software, Knowledge, Information Management and Applications (SKIMA), 2017 11th International Conference on, IEEE, 2017, pp. 1–6.

[25] J. Estrada-Jiménez, J. Parra-Arnau, A. Rodríguez-Hoyos, J. Forné, Online advertising: Analysis of privacy threats and protection approaches, Computer Communications 100 (2017) 32–51.

[26] J. Chen, K. He, Q. Yuan, M. Chen, R. Du, Y. Xiang, Blind filtering at third parties: An efficient privacy-preserving framework for location-based services, IEEE Transactions on Mobile Computing.

[27] J. N. Gross, System and method of presenting content based advertising, uS Patent 9,754,280 (Sep. 5 2017).

[28] F. Zhao, F. Yan, H. Jin, L. T. Yang, C. Yu, Personalized mobile searching approach based on combining content-based filtering and collaborative filtering, IEEE Systems Journal 11 (1) (2017) 324–332.

[29] J. M. Saji, K. Bhongle, S. Mahajan, S. Shrivastava, A. Jarali, Advancement in personalized web search engine with customized privacy protection, in: Progress in Intelligent Computing Techniques: Theory, Practice, and Applications, Springer, 2018, pp. 405–413.

[30] C. Bothorel, N. Lathia, R. Picot-Clemente, A. Noulas, Location recommendation with social media data, in: Social Information Access, Springer, 2018, pp. 624–653.

[31] A. M. Doorey, G. B. Wilcox, M. S. Eastin, Consumer privacy and the new mobile commerce, The Dark Side of Social Media: A Consumer Psychology Perspective.

[32] H. Li, H. Zhu, S. Du, X. Liang, X. S. Shen, Privacy leakage of location sharing in mobile social networks: Attacks and defense, IEEE Transactions on Dependable and Secure Computing 15 (4) (2018) 646–660.

[33] P. J. Wisniewski, B. P. Knijnenburg, H. R. Lipford, Making privacy personal: Profiling social network users to inform privacy education and nudging, International Journal of Human-Computer Studies 98 (2017) 95–108.

[34] T.-H. Wen, C.-S. Hsu, C.-H. Sun, J.-A. Jiang, J.-Y. Juang, A location-based client-server framework for assessing personal exposure to the transmission risks of contagious diseases, in: Human Dynamics Research in Smart and Connected Communities, Springer, 2018, pp. 133–148.

[35] S. Al-Janabi, I. Al-Shourbaji, M. Shojafar, S. Shamshirband, Survey of main challenges (security and privacy) in wireless body area networks for healthcare applications, Egyptian Informatics Journal 18 (2) (2017) 113–122.

[36] P. Nedungadi, A. Jayakumar, R. Raman, Personalized health monitoring system for managing well-being in rural areas, Journal of medical systems 42 (1) (2018) 22.

[37] Q. Hu, S. Wang, C. Hu, J. Huang, W. Li, X. Cheng, Messages in a concealed bottle: Achieving query content privacy with accurate location-based services, IEEE Transactions on Vehicular Technology.

[38] L. A. Martucci, S. Fischer-Hübner, M. Hartswood, M. Jirotka, Privacy and social values in smart cities, in: Designing, Developing, and Facilitating Smart Cities, Springer, 2017, pp. 89–107.

[39] X. Zheng, Z. Cai, J. Yu, C. Wang, Y. Li, Follow but no track: Privacy preserved profile publishing in cyber-physical social systems, IEEE Internet of Things Journal 4 (6) (2017) 1868–1878.

[40] I. MacKenzie, C. Meyer, S. Noble, How retailers can keep up with consumers, McKinsey & Company.

[41] D. Li, Q. Lv, L. Shang, N. Gu, Efficient privacy-preserving content recommendation for online social communities, Neurocomputing 219 (2017) 440–454.

[42] A. Krause, E. Horvitz, A utility-theoretic approach to privacy in online services, Journal of Artificial Intelligence Research 39 (2010) 633–662.

[43] Z. Dou, R. Song, J.-R. Wen, A large-scale evaluation and analysis of personalized search strategies, in: Proceedings of the 16th international conference on World Wide Web, ACM, 2007, pp. 581–590.

[44] K. Hafner, Researchers yearn to use aol logs, but they hesitate, New York Times 23.

[45] W. Sun, Z. Cai, Y. Li, F. Liu, S. Fang, G. Wang, Security and privacy in the medical internet of things: A review, Security and Communication Networks.

[46] S. Wang, Q. Hu, Y. Sun, J. Huang, Privacy preservation in location-based services, IEEE Communications Magazine 56 (3) (2018) 134–140.

[47] C. F. Torres, H. Jonker, S. Mauw, Fp-block: Usable web privacy by controlling browser fingerprinting, in: European Symposium on Research in Computer Security, Springer, 2015, pp. 3–19.

[48] S. Traverso, M. Trevisan, L. Giannantoni, M. Mellia, H. Metwalley, Benchmark and comparison of tracker-blockers: Should you trust them?, in: Network Traffic Measurement and Analysis Conference (TMA), 2017, IEEE, 2017, pp. 1–9.

[49] D. Chaum, Blind signatures for untraceable payment, in: Advances in Cryptology: Proceedings of Crypto'82, 1982.

[50] J. Camenisch, A. Lysyanskaya, An efficient system for non-transferable anonymous credentials with optional anonymity revocation, in: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques: Advances in Cryptology, EUROCRYPT '01, Springer-Verlag, London, UK, UK, 2001.

[51] J. Camenisch, A. Lysyanskaya, A signature scheme with efficient protocols., in: SCN, Vol. 2576 of Lecture Notes in Computer Science, Springer, 2002.

[52] M. Langheinrich, Privacy by design - principles of privacy-aware ubiquitous systems, UbiComp '01, 2001.

[53] W. House, Enhancing online choice, efficiency, security, and privacy, in: National Strategy for Trusted Identities in Cyberspace, April 2011, 2011.

[54] M. Chase, M. Kohlweiss, A. Lysyanskaya, S. Meiklejohn, Malleable signatures: New definitions and delegatable anonymous credentials, in: Proceedings of the 2014 IEEE 27th Computer Security Foundations Symposium, CSF '14, IEEE Computer Society, Washington, DC, USA, 2014.

[55] J. Camenisch, S. Krenn, A. Lehmann, G. L. Mikkelsen, G. Neven, M. O. Pederson, Scientific comparison of abc protocols: Part i – formal treatment of privacy-enhancing credential systems (2014).

[56] H. K. Maji, M. Prabhakaran, M. Rosulek, Attribute-based signatures, Cryptology ePrint Archive, Report 2010/595 (2010).

[57] T. Okamoto, K. Takashima, Efficient attribute-based signatures for non-monotone predicates in the standard model, in: Proceedings of the 14th International Conference on Practice and Theory in Public Key Cryptography Conference on Public Key Cryptography, PKC'11, 2011.

[58] J. Li, M. H. Au, W. Susilo, D. Xie, K. Ren, Attribute-based signature and its applications, ASIACCS '10, 2010.

[59] S. F. Shahandashti, R. Safavi-Naini, Threshold attribute-based signatures and their application to anonymous credential systems, AFRICACRYPT '09, 2009.

[60] J. Herranz, F. Laguillaumie, B. Libert, C. Rafols, Short attribute-based signatures for threshold predicates, in: Topics in Cryptology – CT-RSA 2012, 2012.

[61] Y. Zhang, D. Feng, Efficient attribute proofs in anonymous credential using attribute-based cryptography, in: Proceedings of the 14th International Conference on Information and Communications Security, ICICS'12, 2012.

[62] A. El Kaafarani, E. Ghadafi, D. Khader, Decentralized traceable attribute-based signatures, in: Topics in Cryptology – CT-RSA 2014, 2014.

[63] N. Kaaniche, M. Laurent, Attribute-based signatures for supporting anonymous certification, in: European Symposium on Research in Computer Security, Springer, 2016, pp. 279–300.

[64] S. Belguith, N. Kaaniche, A. Jemai, M. Laurent, R. Attia, Pabac: a privacy preserving attribute based framework for fine grained access control in clouds, in: SECRYPT 2016: 13th International Conference on Security and Cryptography, Vol. 4, Scitepress, 2016, pp. 133–146.

[65] S. Belguith, N. Kaaniche, M. Mohamed, G. Russello, Coop-daab: Cooperative attribute based data aggregation for internet of things applications, in: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", Springer, 2018, pp. 498–515.

[66] C. Kiennert, N. Kaaniche, M. Laurent, P.-O. Rocher, J. Garcia-Alfaro, Anonymous certification for an e-assessment framework, in: Nordic Conference on Secure IT Systems, Springer, 2017, pp. 70–85.

[67] N. El Mrabet, M. Joye, Guide to Pairing-Based Cryptography, CRC Press, 2017.

[68] A. Schneider, C. Meter, P. Hagemeister, Survey on remote electronic voting, arXiv preprint arXiv:1702.02798.

[69] L. Malina, J. Smrz, J. Hajny, K. Vrba, Secure electronic voting based on group signatures, in: Telecommunications and Signal Processing (TSP), 2015 38th International Conference on, IEEE, 2015, pp. 6–10.

[70] G. Ateniese, G. Tsudik, Some open issues and new directions in group signatures, in: International Conference on Financial Cryptography, Springer, 1999, pp. 196–211.

[71] A. Lysyanskaya, Z. Ramzan, Group blind digital signatures: A scalable solution to electronic cash, in: International Conference on Financial Cryptography, Springer, 1998, pp. 184–197.

[72] L. Yan, Z. Xiao, Z. Zhi-ming, An electronic cash system based on certificateless group signature, International Journal of Security and Its Applications 10 (2) (2016) 287–300.

[73] G. Maitland, C. Boyd, Fair electronic cash based on a group signature scheme, in: International Conference on Information and Communications Security, Springer, 2001, pp. 461–465.

[74] J. Helbach, J. Schwenk, S. Schäge, B. EVOTE08, Code voting with linkable group signatures, Electronic Voting 50.

[75] H. Zheng, Q. Wu, B. Qin, L. Zhong, S. He, J. Liu, Linkable group signature for auditing anonymous communication, in: Australasian Conference on Information Security and Privacy, Springer, 2018, pp. 304–321.

[76] G. Ateniese, D. H. Chou, B. de Medeiros, G. Tsudik, Sanitizable signatures, in: Proceedings of the 10th European Conference on Research in Computer Security, ESORICS'05, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 159–177.

[77] S. Canard, A. Jambert, On extended sanitizable signature schemes, in: Proceedings of the 2010 International Conference on Topics in Cryptology, CT-RSA'10, Springer-Verlag, Berlin, Heidelberg, 2010.

[78] S. Canard, A. Jambert, R. Lescuyer, Sanitizable signatures with several signers and sanitizers, in: Proceedings of the 5th International Conference on Cryptology in Africa, AFRICACRYPT'12, Springer-Verlag, Berlin, Heidelberg, 2012.

[79] A. Bilzhause, H. C. Pöhls, K. Samelin, Position paper: The past, present, and future of sanitizable and redactable signatures, in: Proceedings of the 12th International Conference on Availability, Reliability and Security, ACM, 2017, p. 87.

[80] S. S. Chow, Y.-J. He, L. C. Hui, S. M. Yiu, Spice–simple privacy-preserving identity-management for cloud environment, in: International Conference on Applied Cryptography and Network Security, Springer, 2012, pp. 526–543.

[81] S. Canard, R. Lescuyer, Protecting privacy by sanitizing personal data: A new approach to anonymous credentials, in: Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, ASIA CCS '13, ACM, New York, NY, USA, 2013.

[82] D. Pàmies-Estrems, N. Kaaniche, M. Laurent, J. Castellà-Roca, J. Garcia-Alfaro, Lifelogging protection scheme for internet-based personal assistants, in: Data Privacy Management, Cryptocurrencies and Blockchain Technology, Springer, 2018, pp. 431–440.

[83] C.-P. Schnorr, Efficient identification and signatures for smart cards, in: Conference on the Theory and Application of Cryptology, Springer, 1989, pp. 239–252.

[84] S. A. Brands, Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy, MIT Press, Cambridge, MA, USA, 2000.

[85] E. R. Verheul, Self-blindable credential certificates from the weil pairing., in: C. Boyd (Ed.), ASIACRYPT, Vol. 2248 of Lecture Notes in Computer Science, Springer, 2001.

[86] D. Chaum, T. P. Pedersen, Wallet databases with observers, in: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '92, Springer-Verlag, London, UK, UK, 1993, pp. 89–105.
URL http://dl.acm.org/citation.cfm?id=646757.705670

[87] J. Camenisch, E. Van Herreweghen, Design and implementation of the idemix anonymous credential system, in: Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS '02, ACM, New York, NY, USA, 2002.

[88] Microsoft, U-prove community technology, 2013.

[89] Y. Lindell, J. Katz, Introduction to modern cryptography, Chapman and Hall/CRC, 2014.

[90] P. Vullers, G. Alpar, Efficient selective disclosure on smart cards using idemix.

[91] W. Mostowski, P. Vullers, Efficient u-prove implementation for anonymous credentials on smart cards.

[92] A. Mohr, A survey of zero-knowledge proofs with applications to cryptography, Southern Illinois University, Carbondale (2007) 1–12.

[93] A. Pingley, N. Zhang, X. Fu, H.-A. Choi, S. Subramaniam, W. Zhao, Protection of query privacy for continuous location based services, in: Infocom, 2011 Proceedings IEEE, IEEE, 2011, pp. 1710–1718.

[94] B. Niu, Q. Li, X. Zhu, G. Cao, H. Li, Achieving k-anonymity in privacy-aware location-based services, in: INFOCOM, 2014 Proceedings IEEE, IEEE, 2014, pp. 754–762.

[95] Y. Elovici, B. Shapira, A. Meshiach, Cluster-analysis attack against a private web solution (praw), Online Information Review 30 (6) (2006) 624–643.

[96] S. Ye, F. Wu, R. Pandey, H. Chen, Noise injection for search privacy protection, in: Computational Science and Engineering, 2009. CSE'09. International Conference on, Vol. 3, IEEE, 2009, pp. 1–8.

[97] R. Masood, D. Vatsalan, M. Ikram, M. A. Kaafar, Incognito: A method for obfuscating web data, in: Proceedings of the 2018 World Wide Web Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2018, pp. 267–276.

[98] J. Parra-Arnau, D. Rebollo-Monedero, J. Forné, A privacy-preserving architecture for the semantic web based on tag suppression, in: International Conference on Trust, Privacy and Security in Digital Business, Springer, 2010, pp. 58–68.

[99] J. Parra-Arnau, D. Rebollo-Monedero, J. Forné, J. L. Muñoz, O. Esparza, Optimal tag suppression for privacy protection in the semantic web, Data & Knowledge Engineering 81 (2012) 46–66.

[100] J. Parra-Arnau, D. Rebollo-Monedero, J. Forné, Optimal forgery and suppression of ratings for privacy enhancement in recommendation systems, Entropy 16 (3) (2014) 1586–1631.

[101] N. Polatidis, C. K. Georgiadis, E. Pimenidis, H. Mouratidis, Privacy-preserving collaborative recommendations based on random perturbations, Expert Systems with Applications 71 (2017) 18–25.

[102] S. Halevi, Y. Ishai, A. Jain, E. Kushilevitz, T. Rabin, Secure multiparty computation with general interaction patterns, in: Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, ACM, 2016, pp. 157–168.

[103] Y. Lindell, B. Pinkas, Secure multiparty computation for privacy-preserving data mining, Journal of Privacy and Confidentiality 1 (1) (2009) 5.

[104] M. K. Reiter, A. D. Rubin, Crowds: Anonymity for web transactions, ACM transactions on information and system security (TISSEC) 1 (1) (1998) 66–92.

[105] R. Pathak, S. Joshi, Smc protocol for privacy preserving in banking computations along with security analysis, in: Internet, 2009. AH-ICI 2009. First Asian Himalayas International Conference on, IEEE, 2009, pp. 1–5.

[106] M. A. Rahman, M. H. Manshaei, E. Al-Shaer, M. Shehab, Secure and private data aggregation for energy consumption scheduling in smart grids, IEEE Transactions on Dependable and Secure Computing 14 (2) (2017) 221–234.

[107] M. E. Locasto, J. J. Parekh, A. D. Keromytis, S. J. Stolfo, Towards collaborative security and p2p intrusion detection, in: Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC, IEEE, 2005, pp. 333–339.

[108] A. C. Yao, Protocols for secure computations, in: Proceedings of the 23rd Annual Symposium on Foundations of Computer Science, SFCS '82, IEEE Computer Society, Washington, DC, USA, 1982, pp. 160–164.

[109] W. Du, M. J. Atallah, Privacy-preserving cooperative statistical analysis, in: Computer Security Applications Conference, 2001. ACSAC 2001. Proceedings 17th Annual, IEEE, 2001, pp. 102–110.

[110] M. J. Fischer, S. Micali, C. Rackoff, A secure protocol for oblivious transfer (extended abstract, in: In Journal of Cryptology, Springer-Verlag, 1996, pp. 191–195.

[111] C. Gentry, A fully homomorphic encryption scheme, Ph.D. thesis, Stanford, CA, USA (2009).

[112] R. Cramer, I. B. Damgård, et al., Secure multiparty computation, Cambridge University Press, 2015.

[113] R. Cramer, I. Damgård, U. Maurer, General secure multi-party computation from any linear secret-sharing scheme, in: Advances in Cryptology—EUROCRYPT 2000, Springer, 2000, pp. 316–334.

[114] Z. Yan, P. Zhang, A. V. Vasilakos, A survey on trust management for internet of things, Journal of network and computer applications 42 (2014) 120–134.

[115] S. Tonyali, K. Akkaya, N. Saputro, A. S. Uluagac, M. Nojoumian, Privacy-preserving protocols for secure and reliable data aggregation in iot-enabled smart metering systems, Future Generation Computer Systems.

[116] C. T. Barba, L. U. Aguiar, M. A. Igartua, J. Parra-Arnau, D. Rebollo-Monedero, J. Forné, E. Pallarès, A collaborative protocol for anonymous reporting in vehicular ad hoc networks, Computer Standards & Interfaces 36 (1) (2013) 188–197.

[117] J. Sen, Privacy preservation technologies in internet of things, in: Proceedings of the International Conference on Emerging Trends in Mathematics, Technology and Management, 2010, pp. 496–504.

[118] A. Erola, J. Castellà-Roca, A. Viejo, J. M. Mateo-Sanz, Exploiting social networks to provide privacy in personalized web search, Journal of Systems and Software 84 (10) (2011) 1734–1745.

[119] D. Rebollo-Monedero, J. Forne, J. Domingo-Ferrer, Query profile obfuscation by means of optimal query exchange between users, IEEE Transactions on Dependable and Secure Computing 9 (5) (2012) 641–654.

[120] K. Rajendran, M. Jayabalan, M. E. Rana, A study on k-anonymity, l-diversity, and t-closeness techniques, IJCSNS 17 (12) (2017) 172.

[121] J. Domingo-Ferrer, V. Torra, A critique of k-anonymity and some of its enhancements, in: Availability, Reliability and Security, 2008. ARES 08. Third International Conference on, IEEE, 2008, pp. 990–993.

[122] P. Samarati, Protecting respondents identities in microdata release, IEEE transactions on Knowledge and Data Engineering 13 (6) (2001) 1010–1027.

[123] M. Gruteser, D. Grunwald, Anonymous usage of location-based services through spatial and temporal cloaking, in: Proceedings of the 1st international conference on Mobile systems, applications and services, ACM, 2003, pp. 31–42.

[124] B. Gedik, L. Liu, Protecting location privacy with personalized k-anonymity: Architecture and algorithms, IEEE Transactions on Mobile Computing 7 (1) (2008) 1–18.

[125] Y.-M. Ye, C.-C. Pan, G.-K. Yang, An improved location-based service authentication algorithm with personalized k-anonymity, in: China Satellite Navigation Conference (CSNC) 2016 Proceedings: Volume I, Springer, 2016, pp. 257–266.

[126] G. Ghinita, P. Kalnis, S. Skiadopoulos, Prive: anonymous location-based queries in distributed mobile systems, in: Proceedings of the 16th international conference on World Wide Web, ACM, 2007, pp. 371–380.

[127] H. Shin, V. Atluri, J. Vaidya, A profile anonymization model for privacy in a personalized location based service environment, in: Mobile Data Management, 2008. MDM'08. 9th International Conference on, IEEE, 2008, pp. 73–80.

[128] D. Riboni, L. Pareschi, C. Bettini, Shadow attacks on users' anonymity in pervasive computing environments, Pervasive and Mobile Computing 4 (6) (2008) 819–835.

[129] A. Machanavajjhala, J. Gehrke, D. Kifer, M. Venkitasubramaniam, l-diversity: Privacy beyond k-anonymity, in: Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on, IEEE, 2006, pp. 24–24.

[130] N. Li, T. Li, S. Venkatasubramanian, t-closeness: Privacy beyond k-anonymity and l-diversity, in: Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on, IEEE, 2007, pp. 106–115.

[131] F. Liu, K. A. Hua, Y. Cai, Query l-diversity in location-based services, in: 2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware, IEEE, 2009, pp. 436–442.

[132] B. Bamba, L. Liu, P. Pesti, T. Wang, Supporting anonymous location queries in mobile environments with privacygrid, in: Proceedings of the 17th international conference on World Wide Web, ACM, 2008, pp. 237–246.

[133] M. Xue, P. Kalnis, H. K. Pung, Location diversity: Enhanced privacy protection in location based services, in: International Symposium on Location-and Context-Awareness, Springer, 2009, pp. 70–87.

[134] Y. Wang, Y. Xia, J. Hou, S.-m. Gao, X. Nie, Q. Wang, A fast privacy-preserving framework for continuous location-based queries in road networks, Journal of Network and Computer Applications 53 (2015) 57–73.

[135] A. Ye, Y. Li, L. Xu, Q. Li, H. Lin, A trajectory privacy-preserving algorithm based on road networks in continuous location-based services, in: Trustcom/BigDataSE/ICESS, 2017 IEEE, IEEE, 2017, pp. 510–516.

[136] L.-P. Sondeck, M. Laurent, V. Frey, The semantic discrimination rate metric for privacy measurements which questions the benefit of t-closeness over l-diversity, in: SECRYPT 2017: 14th International Conference on Security and Cryptography, Vol. 6, Scitepress, 2017, pp. 285–294.

[137] D. Riboni, L. Pareschi, C. Bettini, S. Jajodia, Preserving anonymity of recurrent location-based queries, in: Temporal Representation and Reasoning, 2009. TIME 2009. 16th International Symposium on, IEEE, 2009, pp. 62–69.

[138] C. Dwork, Differential privacy: A survey of results, in: International Conference on Theory and Applications of Models of Computation, Springer, 2008, pp. 1–19.

[139] L. Wang, D. Yang, X. Han, T. Wang, D. Zhang, X. Ma, Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation, in: Proceedings of the 26th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2017, pp. 627–636.

[140] H. Shin, S. Kim, J. Shin, X. Xiao, Privacy enhanced matrix factorization for recommendation with local differential privacy, IEEE Transactions on Knowledge and Data Engineering.

[141] Z. Zhang, Z. Qin, L. Zhu, J. Weng, K. Ren, Cost-friendly differential privacy for smart meters: Exploiting the dual roles of the noise, IEEE Transactions on Smart Grid 8 (2) (2017) 619–626.

[142] G. Cormode, S. Jha, T. Kulkarni, N. Li, D. Srivastava, T. Wang, Privacy at scale: Local differential privacy in practice, in: Proceedings of the 2018 International Conference on Management of Data, ACM, 2018, pp. 1655–1658.

[143] Ú. Erlingsson, V. Pihur, A. Korolova, Rappor: Randomized aggregatable privacy-preserving ordinal response, in: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security, ACM, 2014, pp. 1054–1067.

[144] A. G. Thakurta, A. H. Vyrros, U. S. Vaishampayan, G. Kapoor, J. Freudiger, V. R. Sridhar, D. Davidson, Learning new words, uS Patent 9,594,741 (Mar. 14 2017).

[145] B. Ding, J. Kulkarni, S. Yekhanin, Collecting telemetry data privately, in: Advances in Neural Information Processing Systems, 2017, pp. 3571–3580.

[146] D. Riboni, C. Bettini, Differentially-private release of check-in data for venue recommendation, in: Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on, IEEE, 2014, pp. 190–198.

[147] R. Chen, B. C. Fung, S. Y. Philip, B. C. Desai, Correlated network data publication via differential privacy, The VLDB Journal 23 (4) (2014) 653–676.

[148] H. To, G. Ghinita, L. Fan, C. Shahabi, Differentially private location protection for worker datasets in spatial crowdsourcing, IEEE Transactions on Mobile Computing 16 (4) (2017) 934–949.

[149] H. J. Asghar, P. Tyler, M. A. Kaafar, Differentially private release of public transport data: The opal use case, arXiv preprint arXiv:1705.05957.

[150] R. Geambasu, T. Kohno, A. A. Levy, H. M. Levy, Vanish: Increasing data privacy with self-destructing data., in: USENIX Security Symposium, Vol. 316, 2009.

[151] S. Wolchok, O. S. Hofmann, N. Heninger, E. W. Felten, J. A. Halderman, C. J. Rossbach, B. Waters, E. Witchel, Defeating vanish with low-cost sybil attacks against large dhts., in: NDSS, 2010.

[152] L. Zeng, Z. Shi, S. Xu, D. Feng, Safevanish: An improved data self-destruction for protecting data privacy, in: 2nd IEEE International Conference on Cloud Computing Technology and Science, IEEE, 2010, pp. 521–528.

[153] R. Perlman, The ephemerizer: Making data disappear.

[154] A. Zarras, K. Kohls, M. Dürmuth, C. Pöpper, Neuralyzer: flexible expiration times for the revocation of online data, in: Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy, ACM, 2016, pp. 14–25.

[155] B. Chor, E. Kushilevitz, O. Goldreich, M. Sudan, Private information retrieval, Vol. 45, 1998.

[156] W. Gasarch, A survey on private information retrieval, The Bulletin of the EATCS 82 (72-107) (2004) 1.

[157] H. Sun, S. A. Jafar, The capacity of private information retrieval, IEEE Transactions on Information Theory 63 (7) (2017) 4075–4088.

[158] C. Aguilar-Melchor, J. Barrier, L. Fousse, M.-O. Killijian, Xpir: Private information retrieval for everyone, Proceedings on Privacy Enhancing Technologies 2016 (2) (2016) 155–174.

[159] R. Ostrovsky, W. E. Skeith, III., A survey of single-database private information retrieval: Techniques and applications, in: Proceedings of the 10th International Conference on Practice and Theory in Public-key Cryptography, PKC'07, Springer-Verlag, Berlin, Heidelberg, 2007.

[160] P. Mittal, F. G. Olumofin, C. Troncoso, N. Borisov, I. Goldberg, Pir-tor: Scalable anonymous communication using private information retrieval., in: USENIX Security Symposium, 2011, p. 31.

[161] N. Cao, C. Wang, M. Li, K. Ren, W. Lou, Privacy-preserving multi-keyword ranked search over encrypted cloud data, IEEE Transactions on parallel and distributed systems 25 (1) (2014) 222–233.

[162] I. Lazrig, T. Moataz, I. Ray, I. Ray, T. Ong, M. Kahn, F. Cuppens, N. Cuppens, Privacy preserving record matching using automated semi-trusted broker, in: IFIP Annual Conference on Data and Applications Security and Privacy, Springer, 2015, pp. 103–118.

[163] F. Baldimtsi, O. Ohrimenko, Sorting and searching behind the curtain, in: International Conference on Financial Cryptography and Data Security, Springer, 2015, pp. 127–146.

[164] C. Chen, X. Zhu, P. Shen, J. Hu, S. Guo, Z. Tari, A. Y. Zomaya, An efficient privacy-preserving ranked keyword search method, IEEE Transactions on Parallel and Distributed Systems 27 (4) (2016) 951–963.

[165] P. Chaudhari, M. L. Das, Privacy-preserving attribute based searchable encryption., IACR Cryptology ePrint Archive 2015 (2015) 899.

[166] M. J. Atallah, K. B. Frikken, Privacy-preserving location-dependent query processing, in: Pervasive Services, 2004. ICPS 2004. Proceedings. The IEEE/ACS International Conference on, IEEE, 2004, pp. 9–17.

[167] R. Paulet, M. G. Kaosar, X. Yi, E. Bertino, Privacy-preserving and content-protecting location based queries, IEEE transactions on knowledge and data engineering 26 (5) (2014) 1200–1210.

[168] M. Ullah, R. Khan, M. A. Islam, Poshida, a protocol for private information retrieval, in: Innovative Computing Technology (INTECH), 2016 Sixth International Conference on, IEEE, 2016, pp. 464–470.

[169] D. X. Song, D. Wagner, A. Perrig, Practical techniques for searches on encrypted data, in: Proceedings of the 2000 IEEE Symposium on Security and Privacy, SP '00, 2000, pp. 44–.

[170] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky, Searchable symmetric encryption: Improved definitions and efficient constructions, in: Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS '06, 2006, pp. 79–88.

[171] N. Cao, C. Wang, M. Li, K. Ren, W. Lou, Privacy-preserving multi-keyword ranked search over encrypted cloud data, IEEE Transactions on Parallel and Distributed Systems 25 (1) (2014) 222–233.

[172] N. H. Sultan, F. A. Barbhuiya, A secure re-encryption scheme for data sharing in unreliable cloud environment, in: Proceedings of the 2016 IEEE World Congress on Services (SERVICES), 2016, pp. 75–80.

[173] C. Guo, X. Chen, Y. Jie, F. Zhangjie, M. Li, B. Feng, Dynamic multi-phrase ranked search over encrypted data with symmetric searchable encryption, IEEE Transactions on Services Computing (2017) 1–1.

[174] W. Sun, S. Yu, W. Lou, Y. T. Hou, H. Li, Protecting your right: Verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud, IEEE Transactions on Parallel and Distributed Systems 27 (4) (2016) 1187–1198.

[175] W. Sun, S. Yu, W. Lou, Y. T. Hou, H. Li, Protecting your right: Attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud, in: Proceedings of the 2014 IEEE Conference on Computer Communications, IEEE INFOCOM, 2014, pp. 226–234.

[176] B. Hu, Q. Liu, X. Liu, T. Peng, G. Wang, J. Wu, DABKS: Dynamic attribute-based keyword search in cloud computing, in: Proceedings of the 2017 IEEE International Conference on Communications (ICC), 2017, pp. 1–6.

[177] I. H. Witten, A. Moffat, T. C. Bell, Managing gigabytes: compressing and indexing documents and images, Morgan Kaufmann, 1999.

[178] W. K. Wong, D. W.-l. Cheung, B. Kao, N. Mamoulis, Secure knn computation on encrypted databases, in: Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, ACM, 2009, pp. 139–152.

[179] R. Rivest, L. Adleman, M. Dertouzos, On data banks and privacy homomorphisms, in: Foundations on Secure Computation, Academia Press, 1978.

[180] P. Cohn, Introduction to Ring Theory, Springer Undergraduate Mathematics Series, Springer, 2000.

[181] S. Micali, C. Rackoff, B. Sloan, The notion of security for probabilistic cryptosystems, Vol. 17, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1988.

[182] T. Elgamal, A public key cryptosystem and a signature scheme based on discrete logarithms, Vol. 31, 1985, p. 469–472.

[183] R. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Vol. 21, 1978, pp. 120–126.

[184] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in: Proceedings of the 17th international conference on Theory and application of cryptographic techniques, EUROCRYPT'99, Springer-Verlag, Berlin, Heidelberg, 1999.

[185] D. Boneh, E. Goh, K. Nissim, Evaluating 2-dnf formulas on ciphertexts, in: Proceedings of the Second international conference on Theory of Cryptography, TCC'05, Springer-Verlag, Berlin, Heidelberg, 2005.

[186] M. Van Dijk, C. Gentry, S. Halevi, V. Vaikuntanathan, Fully homomorphic encryption over the integers, in: Proceedings of the 29th Annual International Conference on Theory and Applications of Cryptographic Techniques, EUROCRYPT'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 24–43.

[187] N. P. Smart, F. Vercauteren, Fully homomorphic encryption with relatively small key and ciphertext sizes, in: Proceedings of the 13th International Conference on Practice and Theory in Public Key Cryptography, PKC'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 420–443.

[188] J. H. Cheon, J.-S. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi, A. Yun, Batch fully homomorphic encryption over the integers, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2013, pp. 315–335.

[189] Z. Brakerski, V. Vaikuntanathan, Efficient fully homomorphic encryption from (standard) lwe, SIAM Journal on Computing 43 (2) (2014) 831–871.

[190] T. Plantard, W. Susilo, Z. Zhang, Lll for ideal lattices: re-evaluation of the security of gentry–halevi's fhe scheme, Designs, Codes and Cryptography 76 (2) (2015) 325–344.

[191] Z. Erkin, T. Veugen, T. Toft, R. L. Lagendijk, Generating private recommendations efficiently using homomorphic encryption and data packing, IEEE transactions on information forensics and security 7 (3) (2012) 1053–1066.

[192] S. Badsha, X. Yi, I. Khalil, A practical privacy-preserving recommender system, Data Science and Engineering 1 (3) (2016) 161–177.

[193] S. Badsha, X. Yi, I. Khalil, E. Bertino, Privacy preserving user-based recommender system, in: Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on, IEEE, 2017, pp. 1074–1083.

[194] N. H. Sultan, F. A. Barbhuiya, M. Laurent, Icauth: A secure and scalable owner delegated inter-cloud authorization, Future Generation Computer Systems 88 (2018) 319–332.

[195] M. Naehrig, K. Lauter, V. Vaikuntanathan, Can homomorphic encryption be practical?, in: Proceedings of the 3rd ACM workshop on Cloud computing security workshop, ACM, 2011, pp. 113–124.

[196] T. Dierks, E. Rescorla, Rfc 5246 - the transport layer security (tls) protocol version 1.2, Tech. rep. (aug).

[197] T. Ylonen, C. Lonvick, The secure shell (ssh) protocol architecture.

[198] O. Levillain, A study of the tls ecosystem, Ph.D. thesis, Institut National des Télécommunications (2016).

[199] N. Doraswamy, D. Harkins, IPSec: the new security standard for the Internet, intranets, and virtual private networks, Prentice Hall Professional, 2003.

[200] P. M. K. Ahonen, Virtual private networks, uS Patent 6,976,177 (Dec. 13 2005).

[201] A. Lakbabi, G. Orhanou, S. El Hajji, Vpn ipsec & ssl technology security and management point of view, in: Next Generation Networks and Services (NGNS), 2012, IEEE, 2012, pp. 202–208.

[202] S. Jahan, M. S. Rahman, S. Saha, Application specific tunneling protocol selection for virtual private networks, in: Networking, Systems and Security (NSysS), 2017 International Conference on, IEEE, 2017, pp. 39–44.

[203] A. Rao, J. Sherry, A. Legout, A. Krishnamurthy, W. Dabbous, D. Choffnes, Meddle: middleboxes for increased transparency and control of mobile traffic, in: Proceedings of the 2012 ACM conference on CoNEXT student workshop, ACM, 2012, pp. 65–66.

[204] B. Ramsdell, Secure/multipurpose internet mail extensions (s/mime) version 3.1 message specification.

[205] R. A. Popa, N. Zeldovich, S. Verma, R. S. Battat, A. D. Burrow, Secure sharing, uS Patent 9,954,684 (Apr. 24 2018).

[206] K. Ermoshina, F. Musiani, H. Halpin, End-to-end encrypted messaging protocols: An overview, in: International Conference on Internet Science, Springer, 2016, pp. 244–254.

[207] J. Camenisch, A. Lehmann, (un) linkable pseudonyms for governmental databases, in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, ACM, 2015, pp. 1467–1479.

[208] K. Wouters, K. Simoens, D. Lathouwers, B. Preneel, Secure and privacy-friendly logging for egovernment services, in: Availability, Reliability and Security, 2008. ARES 08. Third International Conference on, IEEE, 2008, pp. 1091–1096.

[209] T. Pulls, R. Peeters, K. Wouters, Distributed privacy-preserving transparency logging, in: Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society, ACM, 2013, pp. 83–94.

[210] J. Camenisch, A. Lehmann, Privacy-preserving user-auditable pseudonym systems, in: IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, 2017, pp. 269–284.

[211] N. Kaaniche, M. Laurent, Blockchain-based data usage auditing, in: IEEE International Conference on Cloud Computing 2018, IEEE, 2018.

[212] J. Camenisch, S. Krenn, A. Lehmann, G. L. Mikkelsen, G. Neven, M. Ø. Pedersen, Formal treatment of privacy-enhancing credential systems, in: International Conference on Selected Areas in Cryptography, Springer, 2015, pp. 3–24.

[213] E. R. Verheul, B. Jacobs, C. Meijer, M. Hildebrandt, J. de Ruiter, Polymorphic encryption and pseudonymisation for personalised healthcare., IACR Cryptology ePrint Archive 2016 (2016) 411.

[214] S. Cui, S. Belguith, M. Zhang, M. R. Asghar, G. Russello, Preserving access pattern privacy in sgx-assisted encrypted search, in: 2018 27th International Conference on Computer Communication and Networks (ICCCN), 2018, pp. 1–9. doi:10.1109/ICCCN.2018.8487338.

[215] S. Cui, M. R. Asghar, G. Russello, Multi-cdn: Towards privacy in content delivery networks, IEEE Transactions on Dependable and Secure Computing (2018) 1–1.

[216] N. Kaaniche, M. Laurent, A blockchain-based data usage auditing architecture with enhanced privacy and availability, in: Network Computing and Applications (NCA), 2017 IEEE 16th International Symposium on, IEEE, 2017, pp. 1–5.

[217] N. Kaaniche, M. Mohamed, M. Laurent, H. Ludwig, Security sla based monitoring in clouds, in: 2017 IEEE International Conference on Edge Computing (EDGE), IEEE, 2017, pp. 90–97.

[218] M. Swan, Blockchain: Blueprint for a new economy, O'Reilly Media, Inc., 2015.

[219] M. Crosby, P. Pattanayak, S. Verma, V. Kalyanaraman, Blockchain technology: Beyond bitcoin, Applied Innovation 2 (2016) 6–10.

[220] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, S. Savage, A fistful of bitcoins: characterizing payments among men with no names, in: Proceedings of the 2013 conference on Internet measurement conference, ACM, 2013, pp. 127–140.

[221] D. Ron, A. Shamir, Quantitative analysis of the full bitcoin transaction graph, in: International Conference on Financial Cryptography and Data Security, Springer, 2013, pp. 6–24.

[222] A. Kosba, A. Miller, E. Shi, Z. Wen, C. Papamanthou, Hawk: The blockchain model of cryptography and privacy-preserving smart contracts, in: 2016 IEEE symposium on security and privacy (SP), IEEE, 2016, pp. 839–858.

[223] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, G. Maxwell, Bulletproofs: Short proofs for confidential transactions and more, in: Bulletproofs: Short Proofs for Confidential Transactions and More, IEEE, 2018, p. 0.

[224] G. Malavolta, P. Moreno-Sanchez, A. Kate, M. Maffei, S. Ravi, Concurrency and privacy with payment-channel networks, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2017, pp. 455–471.

[225] M. Al-Bassam, A. Sonnino, S. Bano, D. Hrycyszyn, G. Danezis, Chainspace: A sharded smart contracts platform, arXiv preprint arXiv:1708.03778.

[226] P. McCorry, S. F. Shahandashti, F. Hao, A smart contract for boardroom voting with maximum voter privacy, in: International Conference on Financial Cryptography and Data Security, Springer, 2017, pp. 357–375.

[227] Z. Ghahramani, Probabilistic machine learning and artificial intelligence, Nature 521 (7553) (2015) 452.

[228] G. Uchyigit, M. Y. Ma, Personalization techniques and recommender systems, Vol. 70, World Scientific, 2008.

[229] P. Hamet, J. Tremblay, Artificial intelligence in medicine, Metabolism 69 (2017) S36–S40.

[230] P. Brusilovski, A. Kobsa, W. Nejdl, The adaptive web: methods and strategies of web personalization, Vol. 4321, Springer Science & Business Media, 2007.

[231] V. Nikolaenko, S. Ioannidis, U. Weinsberg, M. Joye, N. Taft, D. Boneh, Privacy-preserving matrix factorization, in: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, ACM, 2013, pp. 801–812.

[232] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, N. Taft, Privacy-preserving ridge regression on hundreds of millions of records, in: Security and Privacy (SP), 2013 IEEE Symposium on, IEEE, 2013, pp. 334–348.

[233] S. Gao, J. Ma, C. Sun, X. Li, Balancing trajectory privacy and data utility using a personalized anonymization model, Journal of Network and Computer Applications 38 (2014) 125–134.

[234] D. Massaguer, B. Hore, M. H. Diallo, S. Mehrotra, N. Venkatasubramanian, Middleware for pervasive spaces: Balancing privacy and utility, in: ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing, Springer, 2009, pp. 247–267.

[235] Y. Wang, Y. O. Basciftci, P. Ishwar, Privacy-utility tradeoffs under constrained data release mechanisms, arXiv preprint arXiv:1710.09295.

[236] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, R. Attia, Phoabe: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted iot, Computer Networks 133 (2018) 141–156.

[237] S. Belguith, N. Kaaniche, G. Russello, Pu-abe: lightweight attribute-based encryption supporting access policy update for cloud assisted iot, in: 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), IEEE, 2018, pp. 924–927.

[238] S. Belguith, N. Kaaniche, M. Hammoudeh, T. Dargahi, Proud: Verifiable privacy-preserving outsourced attribute based signcryption supporting access policy update for cloud assisted iot applications, Future Generation Computer Systems.

[239] R. Canetti, Universally composable security: A new paradigm for cryptographic protocols, in: Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on, IEEE, 2001, pp. 136–145.

[240] J. Blömer, F. Eidens, J. Juhnke, Practical, anonymous, and publicly linkable universally-composable reputation systems, in: Cryptographers' Track at the RSA Conference, Springer, 2018, pp. 470–490.

[241] P. Waelbroeck, A. Khatchatourov, C. Levallois-Barth, Personal data and trust: What are the strategies of french citizen-consumers in 2017? (2017).

[242] R. Robert, L. Smit, The proposal for a directive on digital content: a complex relationship with data protection law, in: ERA Forum, Springer, 2018, pp. 1–19.

[243] Y. Balgobin, D. Bounie, M. Quinn, P. Waelbroeck, Payment instruments, financial privacy and online purchases, Review of Network Economics 15 (3) (2016) 147–168.

[244] D. Wright, P. De Hert, Introduction to privacy impact assessment, in: Privacy Impact Assessment, Springer, 2012, pp. 3–32.

[245] G. Wilkinson, General data protection regulation: No silver bullet for small and medium-sized enterprises, Journal of Payments Strategy & Systems 12 (2) (2018) 139–149.