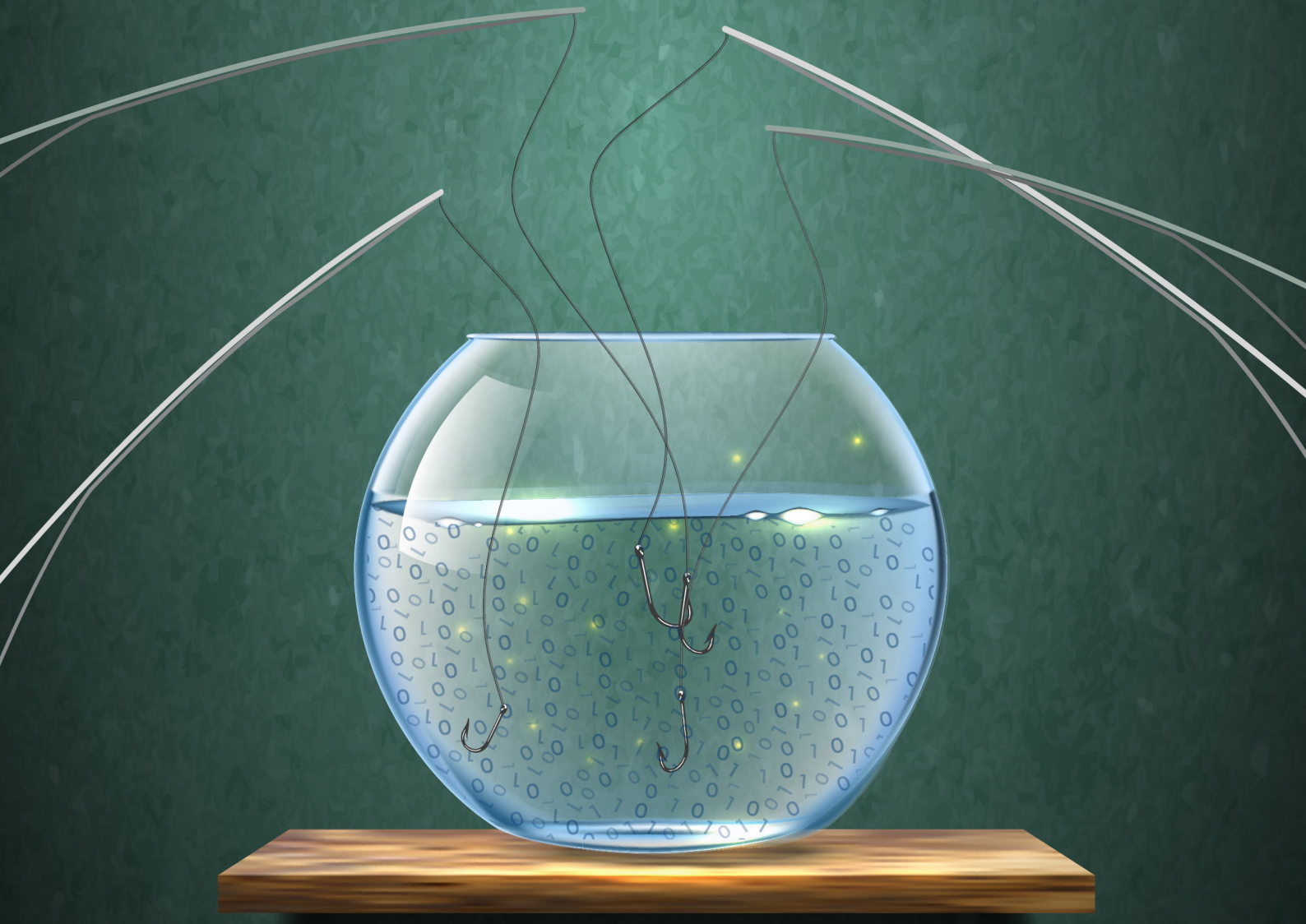
ACM India Minigraph

Privacy of Personal Information: Going Incog in a Goldfish Bowl

Sutapa Mondal
Mangesh S Gharote
Sachin P Lodha



ACMI-Minigraph-Oct2021

Contact us: acmi.minigraphs@gmail.com

ACM India Minigraph

PRIVACY OF PERSONAL INFORMATION: GOING INCOG IN A GOLDFISH BOWL

Sutapa Mondal, Mangesh S Gharote, Sachin P Lodha

TCS Research, Tata Consultancy Services Ltd, Pune.

sutapa.mondal@tcs.com, mangesh.g@tcs.com, sachin.lodha@tcs.com

ACM India Council and Association for Computing Machinery

Copy-editing
Yateendra Joshi

Design and layout
Dr. Rajnish Sharma and Neeraj Pandey, Chitkara University

Abstract

Each online interaction with an external service creates data about the user that is digitally recorded and stored. These external services may be credit card transactions, medical consultations, census data collection, voter registration, etc. Although the data is ostensibly collected to provide citizens with better services, privacy of the individual is inevitably put at risk. With the growing reach of the internet and the volume of data being generated, data protection and, specifically, preserving the privacy of individuals, have become particularly important.

This Minigraph gives an overview of privacy. We discuss how privacy has evolved over time. The privacy concepts are discussed using two fictitious characters, Swara and Betaal, and their interactions with a fictitious entity, namely Asha Hospital. An introduction to privacy and associated concerns (Sections 1 and 2) is followed by descriptions of two important privacy preserving techniques, k-anonymity (Section 3) and differential privacy (Section 4). Lastly, some new approaches to ensuring privacy are discussed (Section 5).

CCS CONCEPTS • Security and privacy • Database and storage security • Data anonymization and sanitization

Additional keywords: k-anonymity, Differential Privacy, Private Computation

FOREWORD

How does a non-specialist learn about a topical subject? This conundrum often arises for young computing professionals choosing a field of work, and for students at a career-planning stage. Subjects such as persistent memory, graph databases, artificial intelligence, machine learning, security, privacy and competitive algorithms are just a few examples of topics that are currently in vogue. Yet, except for those working in the area, few get a chance to really understand the motivation behind the work, the techniques used and the theoretical underpinnings. So, how does someone learn enough about a new area to make an informed decision about whether this is indeed their karmabhoomi?

Conventional wisdom would suggest taking a full course in the subject but that requires both learning opportunity and extended time. Further, published material, such as survey articles and monographs, are usually written for specialists and are rarely approachable by newcomers. On the other hand, popular articles may talk about a few interesting phenomena but usually do not cover enough theoretical background to initiate the reader into the field's structure.

ACM India Minigraphs represent a new initiative intended to bridge this gulf between "too shallow" and "too deep" resources. Specifically, each Minigraph is focused on a contemporary topic and expertly guides the reader from an introductory level to the stage where they can understand the problems in the field and the challenges that need further investigation. A carefully curated set of references point those interested in additional learning towards authoritative sources for further reading.

The idea of Minigraphs was first mooted by Mathai Joseph, a doyen of Indian computer science, and developed further in consultation with Hemant Pande (Executive Director, ACM India) and Rajeev Shorey (Chair of the ACM India Learning Initiatives Committee). Their deliberations led to the creation of an Editorial Board chaired by Mathai with Lipika Dey, Sudip Misra, Sanjiva Prasad and Yogesh Simmhan as members. A rigorous process of publishing the Minigraphs has been set up where each Minigraph manuscript is reviewed by an international committee of experts and then professionally copy-edited to conform to the standard ACM India Minigraph style. My profound thanks to all involved in this laudable exercise, which has collectively converted a fledgling idea into the reality of ACM India Minigraphs.

I am delighted to present the inaugural ACM India Minigraph on Privacy of Personal Information: Going Incog in a Goldfish Bowl, authored by Sutapa Mondal, Mangesh S. Gharote and Sachin P. Lodha of TCS Research, and look forward to perusing its forthcoming volumes as well. Our intention is for Minigraph readers to learn about new topics, and your feedback on the initial issues will guide our team in planning future volumes.

Looking to the future, I warmly invite you to become an active member of the Minigraph community across the spectrum of roles as reader, author, reviewer, and editor. In the years to come, this compendium of distilled information should prove to be of substantive pedagogical value to not just India, but the world at large!

JAYANT R HARITSA
President, ACM India

1. Introduction

Most of us go through life identified by our name and gender. These are our public attributes, and we are usually willing to reveal them. In certain contexts, e.g., in a doctor's clinic, we may disclose personal details like age, height and weight even though these attributes may not be generally known in public. A doctor will know details about a patient's body and mind that even the patient may not know (or understand), but one may not welcome an attempt by a friend, even a close friend, to find about one's medical condition. Similarly, a doctor's interest in a patient's political beliefs may be inappropriate.

Some personal information may need to be made available to certain groups of people and still be protected from unwanted view. The groups could include medical investigators, say, those studying effective forms of treatment or tracking the spread of communicable diseases, who need such data for their research. Privacy remains important, and this creates a tension between the obvious advantages of such work and the potential disadvantages of the details being available and falling into wrong hands.

One way to protect privacy is not to reveal anything at all, but that is hardly practical. Another way is to define the contexts in which certain attributes may be revealed. Our family may know a great deal more about us than most other people do, but we expect such personal information to remain confined to members of the family. In cricket, players tend to observe the rule that 'what happens on the pitch stays on the pitch'. More generally, we usually have no objection to being part of national statistics in terms of age, gender, city, state, first language, and so on, because we may believe that this kind of information cannot be used to reveal our identity: in other words, the information is sufficiently 'anonymized'. But can such anonymization be compromised?

Privacy as a philosophical, psychological, and legal concept has been studied in almost all social sciences. The first era of privacy can be traced back to the Privacy Act of 1974 of the United States federal law, which emphasized the code of fair information practice that governs the collection, maintenance, and use of PII (personally identifiable information). Since then, the notion of data privacy has evolved over time. In 1977, Dalenius (Dalenius, 1977) articulated a desideratum which stated that nothing about an individual should be learnable from a database that cannot be learned without access to the database. He raised concerns around the disclosure of data from statistical datasets. Similarly, Ruth Gavison (Gavison, 1980) argued that an individual's interest in privacy is related to their concern about the data being accessible to others, i.e., the extent to which others can view or access information about an individual. Gavison then emphasized that the privacy of an individual can be viewed as 'hiding in the crowd'. Later, Helen Nissenbaum (Nissenbaum, 2004) put forward an interesting construct of Contextual Integrity as a benchmark for privacy. This construct states that information gathering and dissemination be appropriate to the context and comply with the norms governing the distribution of that information. These viewpoints became the foundation of the privacy preserving techniques that are discussed later in this minigraph.

Many organizations have played an important role in promoting and defining guidelines for privacy:

- The Organisation for Economic Co-operation and Development (OECD) demonstrated international consensus on general guidance concerning the collection and management of personal information. These guidelines enable the mapping of privacy from theory to practice.
- The International Association of Privacy Professionals (IAPP) defines information privacy as the right to have some control over how your personal information is collected and used.
- The National Institute of Standards and Technology (NIST), a non-regulatory agency of the US Department of Commerce, came up with a framework for improving privacy through enterprise risk management (Lefkovitz & Boeckl, 2020).

2. Privacy: The Grand Challenge

Swara regularly visited Asha Hospital for medical treatment. Then one day she was told about a mobile healthcare app (application). The app enabled her to see the doctors' visiting schedules, book online appointments, and make online payments. After entering her personal details, she could view her medical history and laboratory reports of past examinations. For Swara, the app made it very convenient to interact with the hospital and manage her health records—and it helped the hospital to administer, manage, and serve its current and prospective patients.

2.1 A closer look at the data stored by Asha Hospital

Asha Hospital uses a database to maintain the records of its patients. The database consists of multiple tables that capture patients' personal information, medical history, and other details required by the hospital. Figure 1 shows a view of the patient's details stored in a database table, such as the national ID number (NID), name, ethnicity, date of birth, gender, postal code, marital status, and the diagnosed disease. Attributes such as NID and name can uniquely identify any patient and hence are referred to as personally identifiable information (PII). Disease is a sensitive attribute, because people typically do not like to make their ailments public.

The data stored in the database can serve several purposes, such as designing new medical drugs and monitoring and limiting the spread of diseases. For example, hospital data can be used to study and improve the efficacy of medical drugs for treating diseases such as cancer or neurological disorders. The focus of these studies also involves building an understanding of which medicines work best for a given age group, gender, or ethnic group. Thus, this kind of medical data is of tremendous importance.

Leakage or unauthorized disclosure of medical data can violate patient privacy, and the violation can have financial, mental, or social impacts. For example, some diseases like HIV or COVID may carry a social stigma, and disclosure can severely affect the ability of the patient to lead a normal life. Personally identifiable information and other personal data can be used for identity theft and this may take a great deal of time to remedy. Leakage of financial information, such as credit card details, can lead to fraudulent online payments. Thus, both security and privacy of data are critical.

Swara wonders

Is it safe to share personal details on this web app? Apart from information on the disease, why is other information being collected as well? Who has access to this data? What if the hospital were to share her data with a third party?

2.2 Betaal – the Malevolent

Betaal is a person with malicious intent who is on the prowl looking for sensitive information. Let us see how Betaal might be able to intrude on Swara's privacy.

- If Betaal learns that Swara's record is in the dataset, the leak is known as membership disclosure (MSD). This can have serious consequences, especially if she is suffering from an illness that she does not want to be made public.

NID	Name	Ethnicity	Date of birth	Gender	Postal code	Marital status	Disease
1121-4572-4532	Amit	Gujarati	09/27/64	M	400013	Divorced	HIV
1001-5720-2134	Swara	Marathi	09/13/64	F	400015	Married	HIV
2656-9089-2141	Elena	Bengali	09/21/72	F	410013	Married	Cancer
7690-5678-4365	Arjun	Gujarati	09/02/72	M	410016	Married	Heart disease
2154-4329-0090	Tara	Marathi	09/08/64	F	400014	Widowed	Cancer

Figure 1: Asha Hospital patient table

Similarly, MSD in a credit defaulters' list could have a severe effect on the concerned individuals' ability to get loans, or execute financial transactions. MSD in a list of Indians holding accounts in Swiss banks, for example, may have negative consequences even if an individual's particular account is legal.

- If Betaal learns about the disease that Swara suffers from, the leak is known as sensitive attribute disclosure (SAD). With SAD, any ambiguity left with MSD is removed. Since Betaal knows the actual value(s) of the sensitive attribute(s), he may use this information to harm Swara. Thus, the consequences of SAD can be more severe than those of MSD.
- If Betaal has access to Swara's entire record in the dataset, the leak is known as identity disclosure (IDD), the highest degree of disclosure, and it can have even worse consequences than those of MSD and SAD. For example, IDD can result in identity theft and allow Betaal to use Swara's personal information to carry out fraudulent or malicious activities on the web, knowing that any attempt to trace the perpetrator of such activities would lead the investigators to Swara. Medical identity theft would also allow Betaal to make use of healthcare services intended for Swara.

2.3 Disclosure risk

The amount of information Betaal can access can lead to different types of disclosure. In any dataset, these disclosures are closely related and can be ordered by severity as follows:

MSD << SAD << IDD

However, their likelihoods follow the reverse order, i.e., the likelihood of MSD is greater than that of SAD and this in turn is greater than the likelihood of IDD because each is subsumed in the other, in the given order. Since risk is defined as the 'likelihood of disclosure' times the 'impact of disclosure', if disclosure does happen, any one of these disclosures could potentially carry a bigger risk than the other two, depending on the overall context.

Swara wonders

How could Betaal access her data? Aren't there any privacy guidelines for sharing data? What about asking for her consent before processing her data for any particular purpose?

Each disclosure allows Betaal to intrude on Swara's privacy, either directly or indirectly. Betaal could be an unauthorized or an authorized user of the system. As an unauthorized user, Betaal would be an outsider; for example, he could be a health insurance agent or a pharmaceutical company representative with malicious intent to access the data for private gain. Because of the possible impact of a leak, information security has been recognized as a critical area that must be addressed by every organization, be it a hospital or a corporate office. Therefore, we can expect that there will be sufficient security controls to prevent an unauthorized user from accessing data, thereby protecting personal data and preserving the privacy of an individual.

As an authorized user, Betaal would be an insider. He could be a member of the hospital staff, or in administration or the IT department, and so on. Even if he were an authorized user, Swara's entire personal data ought not to be shared with Betaal.

This raises a major question: What data can be made available to an authorized user, such that

a) this user can carry out necessary work, including accessing some of Swara's personal data, while also
b) ensuring that Swara's privacy requirements are met?

Note

- In the context of security, it is assumed that the user and the attacker are different; however, in the context of privacy, a legitimate user may also be an attacker.
- For security, the malicious intent of an unauthorized user can be mitigated by security controls, whereas for privacy, a different set of controls is needed.
- A good security framework is not enough to ensure privacy: security takes care of the 'bad guys' (i.e., unauthorized users), whereas privacy is also concerned about the 'good guys' (i.e., authorized users) who have legitimate access to the data.

2.4 The data juggle

Is there some way to make only part of the data available to Betaal and mitigate the risks of MSD, SAD, and IDD?

What if we remove the identifying data (PII)?

After removal of the PII attributes, such as the NID and name (Figure 2), can the record representing Swara still be precisely identified by Betaal? In practice, the 'removal' may be implemented as 'replacement by fictitious values' to satisfy database or application constraints and requirements.

This seems like a promising solution as the possible membership in the dataset as well as the identity are not revealed. Will this preserve Swara's privacy? Sadly, that is not the case.

In the late 1990s, the Group Insurance Commission (GIC) providing health insurance to Massachusetts State employees used exactly this strategy for privacy protection. Latanya Sweeney (Sweeney, 1998), then a PhD student at the Massachusetts Institute of Technology, was able to identify most of the data subjects simply by taking the 'join' of the publicly available voter registration data with the healthcare data provided by GIC. The attack resulted in Identity (and Membership as well as Sensitive Attribute) Disclosure. Therefore, removing PII alone does not guarantee privacy.

What if we further shuffle the sensitive data?

Swara may not mind Betaal knowing that she is being treated in Asha Hospital: that is, she may be fine with MSD. More important for her could be to safeguard against SAD: to make sure that Betaal does not discover her disease. To overcome this, column-wise shuffling of the sensitive values (Figure 3) seems to be a possible solution. But this does not work, because even though the values are shuffled, their distribution remains the same, and that may be enough for Betaal to infer Swara's disease with high confidence. This is particularly true when the data is unevenly distributed, for example, if there are many patients but only a small number of distinct diseases in the dataset.

What if we remove everything?

Replace all the records with synthetic data that approximates the original data (Figure 4). This approach looks very promising for safeguarding privacy because such data does not point to any individual in the real world. Synthetic data preserves both structural and characteristic properties of the original data. Initial techniques for generating synthetic data were based on estimating the data distributions. However, with emerging techniques like generative adversarial networks (GANs), it is now feasible to generate synthetic data that

NID	Name	Ethnicity	Date of birth	Gender	Postal code	Marital status	Disease
		Gujarati	09/27/64	M	400013	Divorced	HIV
		Marathi	09/13/64	F	400015	Married	HIV
		Bengali	09/21/72	F	410013	Married	Cancer
		Gujarati	09/02/72	M	410016	Married	Heart disease
		Marathi	09/08/64	F	400014	Widowed	Cancer

Figure 2: Dataset without personally identifiable information

NID	Name	Ethnicity	Date of birth	Gender	Postal code	Marital status	Disease
		Gujarati	09/27/64	M	400013	Divorced	Cancer
		Marathi	09/13/64	F	400015	Married	HIV
		Bengali	09/21/72	F	410013	Married	Heart disease
		Gujarati	09/02/72	M	410016	Married	Cancer
		Marathi	09/08/64	F	400014	Widowed	HIV

Figure 3: Data after shuffling a sensitive attribute, namely <Disease>

Original Data

NID	Name	Ethnicity	Date of birth	Gender	Postal code	Marital status	Disease
1121-4572-4532	Amit	Gujarati	09/27/64	M	400013	Divorced	HIV
1001-5720-2134	Swara	Marathi	09/13/64	F	400015	Married	HIV
2656-9089-2141	Elena	Bengali	09/21/72	F	410013	Married	Cancer
7690-5678-4365	Arjun	Gujarati	09/02/72	M	410016	Married	Heart disease
2154-4329-0090	Tara	Marathi	09/08/64	F	400014	Widowed	Cancer

Synthetic Data

NID	Name	Ethnicity	Date of birth	Gender	Postal code	Marital status	Disease
0001-0020-2341	TUV	Bengali	09/08/60	F	410013	Separated	Cancer
1020-7520-1234	DET	Gujarati	08/01/64	M	400012	Married	HIV
2120-0984-1421	FRE	Marathi	02/02/72	M	400013	Widowed	HIV
6709-8765-5634	QWE	Marathi	09/11/71	F	410012	Divorced	HIV
4512-9234-0109	XYZ	Gujarati	10/29/64	M	400011	Not married	Heart disease

Figure 4: Original data transformed into synthetic data

appears very similar to real-world data. This synthetic data can be used for carrying out various data analytics while preserving privacy at the same time. But the limitation is the loss in accuracy of the results obtained and hence the approach would not be always suitable for some applications. For example, machine learning models for cancer prediction would require very high accuracy and very high precision. This may not be possible using only synthetically generated data.

2.5 Tug of war: privacy vs utility

These *what-ifs* suggest some data-sharing possibilities that could preserve privacy. That said, the other side of the privacy coin is 'data utility'. The purpose behind providing data access to an authorized user is for some productive purpose. Therefore, attention should not be limited to the question "How anonymous is the

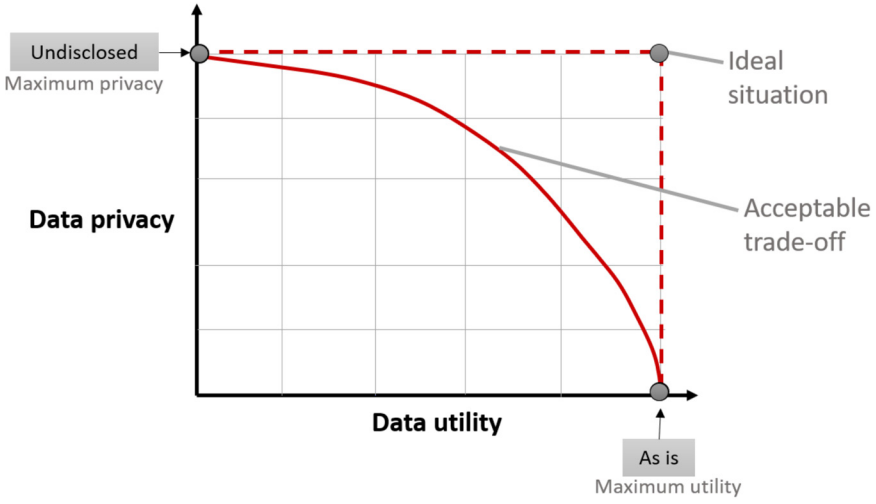


Figure 5: Data privacy vs Data utility

anonymized data?" but should also address the other question, "How meaningful is the anonymized data?"

The grand challenge of privacy is finding a balance between fully disclosed data and completely withheld data. Figure 5 shows that when data is shared "as is", its utility is maximum but privacy is minimum and when nothing is disclosed, privacy is maximized but utility is limited.

Swara wonders

What are these privacy-preserving techniques? How do they differ from one another?

The ideal solution would be when both privacy and utility are at their maximum possible levels. To be at one end or the other, i.e., complete privacy or complete utility, is not acceptable in most situations. Owing to the conflicting requirements of privacy and utility, achieving the ideal solution is challenging. Therefore, there is a strong need for privacy-preserving techniques that can provide a balance between privacy and utility. Use of such techniques would allow Swara to methodically share her personal data that is useful in a specific context and prevent Betaal from breaching her privacy.

3. K-Anonymity

Traditional methods such as randomization, shuffling and, data swapping have been able to preserve privacy to an extent, but the risk of data disclosure still exists. To overcome these limitations, a search started for better techniques that can offer conclusive utility and privacy guarantees. In this section, we introduce and discuss k-anonymity, one such privacy preserving technique.

3.1 Introduction

Swara wonders

If Asha Hospital shares her data, will her privacy be preserved? Will her data be anonymized before sharing?

We saw in the earlier section that merely removing the PII doesn't help in ensuring anonymity. What if Swara's record were made to look the same as the other 99 records in the dataset? Swara's record will no longer be identifiable among those 100 records. Let us generalize this 100 to k. This will make it difficult for Betaal to re-identify Swara among k similar records; it is like hiding Swara in the crowd. This notion of privacy highlighted by Ruth Gavison in the 1980s later emerged as an idea behind k-anonymization. k-anonymization makes k records look similar in a dataset, i.e., data about every individual is hidden among k similar records.

Swara wonders

What led to k-anonymity? How can one achieve k-anonymity?

We discussed Latanya Sweeney's attack in Section 2. Sweeney had observed that many patients had a unique combination of <date of birth, gender, and zip code (postal code)> attributes in the de-identified dataset released by the hospital. Further, she observed that these three attributes were listed along with names and addresses in publicly available voter registration data. Sweeney then linked the data released by the hospital with the voter registrations data and was able to re-identify sensitive health information of even the Governor of Massachusetts! This kind of attack is known as a linkage attack, in which one or more records in a de-identified dataset can be re-identified by uniquely linking them with identified records in a publicly available dataset. Sweeney effectively showed that merely removing the PII is not sufficient to preserve privacy.

Sweeney also observed that with just <date of birth, gender, zip code> attributes, 87% of the US population is uniquely identifiable. Such attributes are referred to as quasi-identifiers. These attributes, when combined, can act as PII. Figure 6 illustrates certain categories of attributes that are relevant for our discussion: identifiers (NID, name), quasi-identifiers (date of birth, gender, postal code), and the sensitive attribute (disease). Sweeney's linkage attack showed that publishing a record with quasi-identifier(s) is as bad as publishing it with explicit identity (PII) attributes. An attack aimed at IDD, assuming the attacker already knows that the targeted individual is part of the data, requires only the corresponding record to be identified. This assumption of prior MSD in the attack model makes it even easier for the attacker to uncover the identity of a given individual who is part of the dataset.

Owing to the possibility of such attacks and limitations in the practice of personal data protection and privacy, researchers started looking for better privacy-preserving techniques. Samarati and Sweeney introduced *k-anonymity* as a privacy-preserving technique (Samarati & Sweeney, 1998), especially

NID	Name	Date of birth	Gender	Postal code	Disease
2321-4572-2221	Aabha	09/27/74	F	400013	Obesity
4151-1391-7002	Nita	09/21/74	F	400014	Chest pain

Figure 6: Categorization of attributes (identifiers, quasi-identifiers, sensitive attribute)

to mitigate such linkage attacks. Initially, k-anonymity focused on publishing data with person-specific structured data in a privacy-aware manner. For example, release of patient records should come with scientific guarantees such that the data remains of practical use and no individual can be re-identified.

Definition: Published data is said to have the k-anonymity property if the information for an individual cannot be distinguished from k–1 other individuals whose information also appears in the data (Samarati & Sweeney, 1998).

3.2 A closer look at k-anonymity

Figure 7 illustrates an example of 3-anonymization of relational data (where k = 3) such that the transformed quasi-identifiers must appear in at least three records. The transformation of quasi-identifiers can be carried out using techniques such as *generalization* and *suppression*. For suppression, some or all the values of an attribute may be replaced by '^'. For example, in Figure 7 all the values of the name attribute are suppressed. In practice, instead of '^', attribute values are replaced by random values. For generalization, on the other hand, individual values of attributes are replaced by a value representing a much broader range or category. For example, Figure 7 shows that the attribute postal code has been generalized using a single strategy of replacing the last three digits represented by '*', whereas for the attribute 'Date of birth', multiple strategies have been used wherein in some cases the month has been generalized and in others, the month, the date, and the year have been generalized. In practice, instead of '*', the values are replaced by random values from the domain of the attribute values.

Generalization can be done using taxonomy trees, as illustrated in Figure 8. The taxonomy tree (Figure 8a) defined for the postal code shows a generalization hierarchy. The lower level of the taxonomy tree represents district information with the last three digits suppressed, whereas the higher levels depict the region and the zone. Similarly, Figure 8b shows a taxonomy tree for generalization of the 'age' attribute such that a value of age (say, 22) in a dataset can be generalized to any random value in the range [21-25].

NID	Name	Date of birth	Gender	Postal code	Disease
^	^	09/**/72	F	410***	Hypertension
^	^	09/**/72	F	410***	Cancer
^	^	**/**/7*	M	520***	Heart disease
^	^	**/**/7*	M	520***	Heart disease
^	^	**/**/7*	M	520***	Heart disease
^	^	09/**/72	F	410***	Hypertension

Figure 7: Example of 3-anonymous relational data

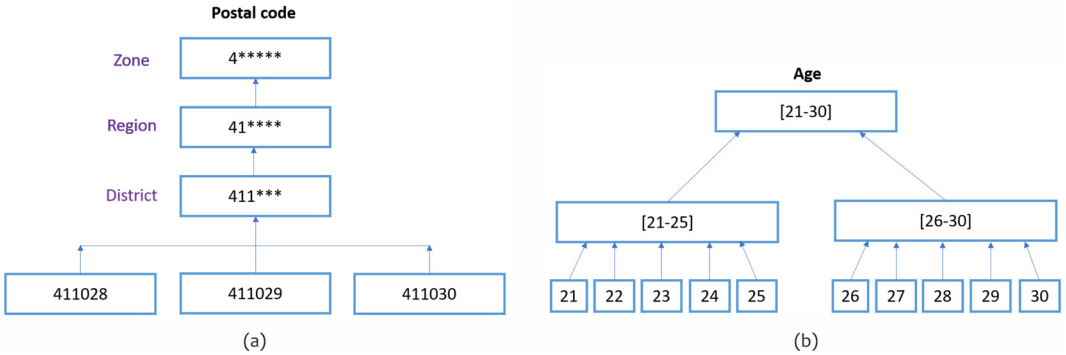


Figure 8: Taxonomy trees with generalized hierarchy for attributes

Higher generalization allows more records to be mapped and hence a higher level of privacy to be achieved, though this might significantly impact data utility. Further, generalizing all the records of a relational database using a single strategy for an attribute may not be the best strategy. For example, if in a dataset there are many records with similar demographics, then grouping some records by generalizing to the region level (Postal code: 41****) and others to the district level (Postal code: 411***) would prove to be a better strategy to preserve privacy and enhance utility of the data. Thus, one can also make local changes rather than only global changes to the attributes.

This privacy-preserving transformation of data is referred to as recoding. In global recoding, a particular detailed value must be mapped to the same generalized value in all records. Local recoding allows the same detailed value to be mapped to different generalized values in each anonymized group.

Can we achieve optimal k-anonymity by transforming a minimum amount of data? It has been proved that achieving optimal k-anonymity is an NP-hard problem for multidimensional data (Meyerson & Williams, 2004). Various researchers have proposed approximation algorithms to achieve near optimal k-anonymity (Aggarwal et al., 2005).

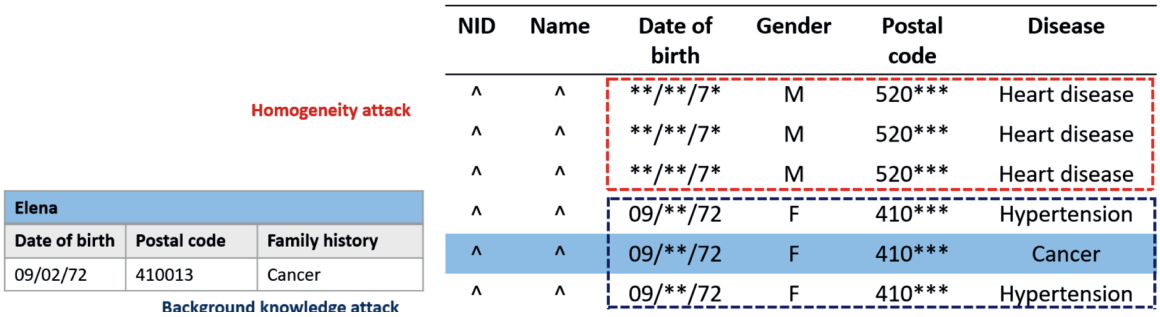
3.3 Extensions to k-anonymity

Now let us take a deeper look at the k-anonymized table (Figure 9), which shows that all k (= 3) individuals with the postal code generalized to the value (520***) have the same sensitive value (in this case, they all suffer from heart diseases). Although k-anonymization of data prevents linkage attacks and an attacker cannot link to other databases with a high degree of certainty, it may still reveal sensitive information. This is known as a homogeneity attack, where all k individuals have the same sensitive value. Similarly, if an attacker has additional information about an individual, it may be possible to re-identify the record with high probability, leading to a background knowledge attack. For example, Figure 9 shows that if an attacker knows the date of birth, postal code and family history of a person (say, Elena), then the attacker can guess Elena's record with high probability.

Thus, k-anonymity does not give any scientific guarantees against such attacks. Moreover, the choice of 'k' for an acceptable level of k-anonymity poses another challenge. Further, information is lost during generalization or suppression of records for achieving k-anonymity: the higher the generalization, the lower the utility. These mechanisms do skew or bias the statistics of the dataset, leading to a trade-off between utility and privacy.

To overcome some of these drawbacks, different variants of k-anonymity were proposed. To meet the requirement that the values of sensitive attributes be well represented in each group, l-diversity was introduced, in which any sensitive attribute should have l distinct values in every group. However, it also involves suppressing or adding rows that would alter the distribution of data. Such suppression or addition raises concerns over the statistical validity of conclusions drawn from the dataset.

These shortcomings led to t-closeness, an



NID	Name	Date of birth	Gender	Postal code	Disease
^	^	**/**/7*	M	520***	Heart disease
^	^	**/**/7*	M	520***	Heart disease
^	^	**/**/7*	M	520***	Heart disease
^	^	09/**/72	F	410***	Hypertension
^	^	09/**/72	F	410***	Cancer
^	^	09/**/72	F	410***	Hypertension

Figure 9: Possible attacks on k-anonymized data

extension of k-anonymity and l-diversity: t-closeness captures the notion that the distribution of a sensitive attribute in any k-subset is not only l-diverse but also close to the distribution of the attribute in the overall dataset. Further, the distance between the two distributions is measured by the threshold t. These extensions of k-anonymity cater to some of the limitations, but not all. For example, the dimensionality of data continues to be a challenge: for data with high dimensionality, such as a time series data, it becomes quite hard to provide the same privacy guarantees as for low dimensional data.

Organizations today are entrusted with personal data of their customers and their employees to provide various services. Although they may have obtained the data lawfully, it may be infeasible for them to get the consent of data subjects each time to perform any data processing. Also, it is highly likely that for some kinds of analyses of personal data, the data subjects may not be comfortable in giving their consent at all. Further, data subjects may view any processing of their personal data with different degrees of concern depending on their perceptions. In such scenarios, it becomes important how personal data should be managed such that it does not violate the privacy expectations of an individual. Here, suitably anonymized data can act as a proxy for real data. This has led many organizations to come up with privacy solutions for data anonymization and protection. There are also open-source tools such as µArgus, and Datafly (Sweeney, 2002), ARX data anonymization, Anonimatron, sdcMicro, etc., that provide k-anonymization as one of their functionalities. Many IT firms like Google, IBM, and TCS have privacy solutions that use k-anonymity as one of the privacy-preserving techniques.

k-anonymity has gained much of its visibility from privacy-aware data publishing scenarios. However, there is also research on performing classification and data mining tasks using k-anonymity (Ciriani et al., 2008). Further, the application scope of k-anonymity has been extended beyond relational databases to anonymizing combinatorial structures like graphs (Stokes & Torra, 2012).

3.4 Discussion

In this section, we discuss the choice of k in k-anonymity, some practical aspects of publishing anonymized data, quasi-identifiers, the ideal amount of generalization to achieve the desired anonymization, and how to k-anonymize efficiently.

What is the right choice of k for k-anonymization?

In the United States, the Health Insurance Portability and Accountability Act (HIPAA) sets the standard for sensitive patient data protection. The HIPAA 'safe harbor' model requires removal of geographic subdivisions smaller than a state, except for the initial three digits of the US zip code if the geographic unit formed by combining all zip codes with the same initial three digits contains more than 20,000 people. Therefore, groups formed by generalizing US zip codes with the first three digits will have at least 20,000 people. Consequently, the HIPAA defined 20,000 as the standard value of k for k-anonymity. Another US act, namely the Family Education Rights and Privacy Act (FERPA), sets the standards for protecting personal information of students and their families. However, in the FERPA, many statisticians considered a cell size of 3 to be the absolute minimum needed to mitigate disclosure using k-anonymity, though a large value of k (5 or 10) can be used to prevent disclosure. This shows how two standards for health (HIPAA) and education (FERPA) in the same country differ in their choice of k. Here, the choice of k is pre-defined for applications governed under these regulatory mandates. However, for applications where there is no regulatory requirement, choosing k to provide the right level of privacy versus utility trade-off becomes a challenge. An approach to choosing k could be to vary the value of k over a range of values and determine the change in generalized information loss (a measure of utility) of the dataset. The value of k corresponding to the acceptable generalized information loss can then be chosen as the value of k.

That said, finding the appropriate value of k is still an open research problem, and researchers have proposed several approaches, like probabilistic models and, multi-objective optimization models.

How does one identify quasi-identifiers?

Identification of quasi-identifiers (Lodha & Thomas, 2007) is a primary problem because it has a direct impact on the effectiveness of the k-anonymity method. Figure 10 illustrates the number of records identifiable for varying sets of attributes that could be potential quasi-identifiers. For example, if the table comprises only the Ethnicity column (A), then only one record (Punjabi) is unique. Further, when column B (Gender) is added, three records become unique (identifiable). Similarly, adding column C (Postal code) makes two more records unique. Hence, with the increase in information, a larger number of records becomes identifiable. The choice of quasi-identifiers can become more complex with an increase in the dimensionality of data. Also, the problem becomes challenging with the uncertainty in the additional data being published by others. In that case, it may turn out that some of our published attributes must be considered as quasi-identifiers.

What should be the ideal amount of generalization to achieve desired anonymization?

The ideal degree of generalization depends on publicly available information. Public- and private-sector organizations in many countries publish information in the public domain to achieve greater information transparency and to make it easier for citizens to access their data. For example, from a public-sector point of view, these publicly available databases or services can be part of e-Government initiatives; examples are publishing transportation information like vehicle registration, legal court proceedings data, or information related to various schemes like the Mahatma Gandhi National Rural Employment Guarantee Act (MNREGA). Similarly, private-sector organizations may inadvertently publish information that should not be published. This gives an opportunity for private aggregators to misuse such information. They may make use of open-source intelligence (OSINT) tools that search multiple publicly available sources. This gives them a comprehensive view of the data which can be used to leak highly specialized information. Thus, these open data sources may make it easier to carry out a linkage attack. OCEAN at IIIT Delhi is a project that aggregates data from open government data sources and demonstrates its linking power by building family trees of citizens. This shows that any organization wanting to publish data about citizens must apply extreme generalization to prevent re-identification through linkage attacks.

How can k-anonymity be efficiently achieved?

Researchers have demonstrated that multidimensional k-anonymity is an NP-hard problem (Meyerson & Williams, 2004). However, there exist approximation algorithms (Aggarwal et al., 2005) that can achieve k-anonymity but are not scalable. On the other side, a probabilistic approach to k-anonymity (Lodha & Thomas, 2007) using dynamic programming provides a time-optimal algorithm for k-anonymity. Heuristic methods such as k-optimize (Bayardo & Agrawal, 2005) also yield effective results.

However, with the current emphasis on AI-driven analytics, there is a visible change in the definition of privacy and data protection, which demonstrates the need for privacy preserving techniques that provide much stronger guarantees and offer a wider scope for different applications.

(A)
Ethnicity
Gujarati
Marathi
Gujarati
Gujarati
Marathi
Punjabi
Marathi

(A)	(B)
Ethnicity	Gender
Gujarati	F
Marathi	F
Gujarati	M
Gujarati	M
Marathi	M
Punjabi	M
Marathi	F

(A)	(B)	(C)
Ethnicity	Gender	Postal code
Gujarati	F	400013
Marathi	F	400014
Gujarati	M	400014
Gujarati	M	400015
Marathi	M	400013
Punjabi	M	400014
Marathi	F	400014

Figure 10: Choice of quasi-identifiers

Takeaways

Linkage attacks showed that removing identifiers alone does not preserve privacy. Hence, k-anonymity emerged as a prominent privacy-preserving technique. Here, generalization is performed over true information, which makes it more acceptable than other strategies. Also, k-anonymity and its variants can limit linkage attacks, homogeneity attacks, and background attacks. From the industrial standpoint, k-anonymity has gained wider visibility because of its acceptance in regulatory compliance, especially in the United States.

However, k-anonymity has some drawbacks, such as MSD and information loss. Also, generalization requires a taxonomy tree for every quasi-identifier in the dataset on which k-anonymity must be performed, and this will require the intervention of domain experts even if the taxonomies are auto-generated. Further, depending upon the use-case for performing k-anonymity, the level of generalization for every attribute may vary. This motivated researchers to come up with purpose- driven strategies such as assigning weights to every attribute to measure their relative importance and accordingly generalize the attributes (Lodha & Thomas, 2007) or performing multidimensional suppression such that values are suppressed only on certain records depending upon other attribute values (Kisilevich et al., 2009). k-anonymity provides limited scientific guarantees that an individual cannot be re-identified. However, with the advancement in computing power and the availability of digital datasets, the risk of re-identification exists. These limitations motivate researchers to look for better privacy preserving techniques.

4. Differential Privacy

Suppose that before sharing the data, some noise is injected, or a synthetic dataset is created that has the same statistical properties as the original dataset. These are some of the ways by which privacy can be preserved. In this section we introduce Differential Privacy (DP) as a privacy preserving technique.

4.1 Introduction

Cynthia Dwork (Dwork et al., 2006) introduced Differential Privacy to protect an individual's privacy by injecting carefully calibrated random noise to make the data non-truthful. The ingenuity of DP lies in allowing meaningful analyses to be drawn over the dataset while preserving the privacy of individuals. However, the original motivation behind DP can be traced back to Dalenius's notion of privacy concerns over statistical datasets: "Learning anything about me should be hard", i.e., it should be difficult to learn anything about an individual without direct access to the database.

Swara wonders

How does differential privacy work? How does DP make it safe for third parties to perform data analysis?

Differential Privacy is intended to preserve privacy while analysing data. In a typical DP set-up, the data curator is assumed to be trustworthy and acts as the central body. The data curator holds the data of individuals that make up the database. With a trusted curator, DP can be operated in two different modes: an online or interactive mode or an offline or non-interactive mode.

In the interactive mode (Figure 11), the data analyst queries the database adaptively, i.e., the analyst modifies successive queries depending on the responses to earlier queries. A query is a function applied to the database; for example, "How many people in the database have tuberculosis?" is a query. Figure 11 shows how a DP mechanism is interposed between the database and the data analyst. To each query of an analyst, a noisy response is generated, thereby preserving privacy.

Asha Hospital maintains a database of patients who took the COVID-19 test. Suppose Swara is part of this database and Betaal wants to find out whether Swara was COVID-19 positive. He poses two queries to the database: "How many people in the database have COVID-19?" and "How many people not named Swara are there in the database who have COVID-19?" The difference in the two results will tell Betaal whether Swara tested positive or negative. Such an attack is known as a differencing attack and it shows that queries over large databases can reveal sensitive information of an individual or a group of individuals. However, if the responses to the queries were perturbed, then the actual information does not get revealed. This is how an interactive mode of DP would help to safeguard the actual information.

Another safety strategy is to operate in offline or non-interactive mode (Figure 12). The curator produces a 'synthetic database' using DP mechanism that has the same statistical properties as the original dataset. After releasing the data, the curator plays no further role, and the original data may even be destroyed. Thus, with a synthetic database, it becomes difficult to re-identify an individual. Further, such synthetic data can be shared for performing quality analyses.

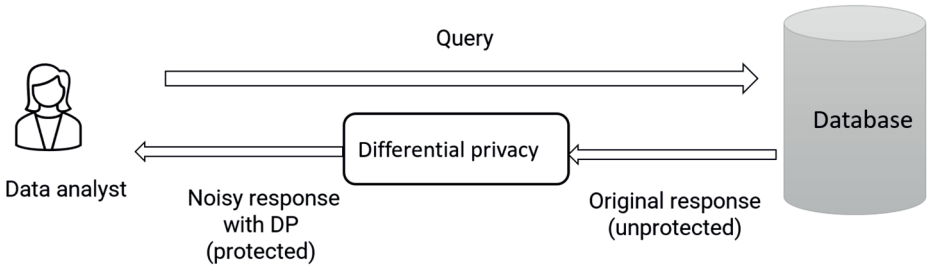


Figure 11: Differential Privacy in the interactive (online) mode

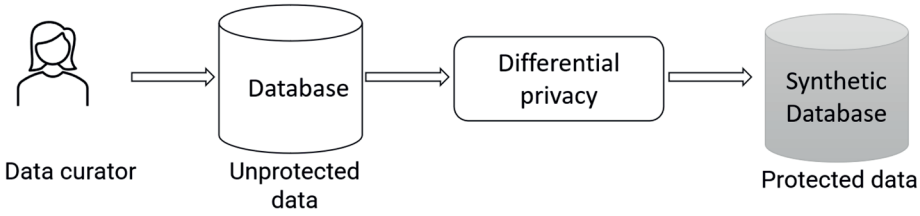


Figure 12: Differential Privacy in the non-interactive (offline) mode

Swara wonders

What is a typical differential privacy mechanism? Does DP provide any formal guarantees for privacy?

4.2 A closer look at differential privacy

Consider an algorithm that analyses a dataset and computes statistical properties such as the mean, variance, median, mode, etc. Such an algorithm is said to be differentially private if by looking at the output one cannot tell whether any individual's data is included in the original dataset or not. In other words, the guarantee of a differentially private algorithm is that its behaviour hardly changes with the absence or presence of an individual in the dataset. Figure 13 shows that a DP mechanism F produces similar output distributions when applied on the database with candidate D and without candidate D. Most notably, this guarantee holds for any individual and any dataset.

Thus, regardless of how unique any single individual's details are, and regardless of the details of anyone else in the database, the guarantee of DP still holds.

Mathematically, DP can be defined as follows:

A randomized algorithm M with domain $N^{|x|}$ is ϵ-differentially private if for all $S \subseteq Range(M)$ and for all $x, y \in N^{|x|}$ such that $\|x - y\| \leq 1$:

$$Pr[M(x) \in S] \leq exp(\epsilon) \times Pr[M(y) \in S]$$

The distribution of the curator's output M(x) on the database x is nearly the same as M(y) on database y, where the databases x and y differ by only one individual's record and M is a randomized algorithm guaranteeing ϵ-differential privacy: ϵ determines the indistinguishability of two databases x and y, i.e., the deviation in the response to a query over both the databases is governed by ϵ. This gives a formal guarantee that individual-level information about participants in the database is not leaked. This depicts a crucial detail about DP wherein MSD is avoided as well as making it difficult for other disclosure risks (SAD and IDD) to take place.

The crucial feature of DP is that it defines privacy as a quantifiable measure using the parameter ϵ and not as a binary condition such as whether the data of an individual was leaked or not. Essentially, ϵ determines how much noise is added to the computation, so it can be treated as a tuning knob for balancing privacy and utility. Each differentially private analysis can be tuned to provide more, or less, privacy.

Swara wonders

How is this guarantee achieved in software systems? How is differential privacy applied?

Differentially private algorithms, or DP mechanisms, are randomized algorithms that add noise at key points. The Laplace mechanism can be used

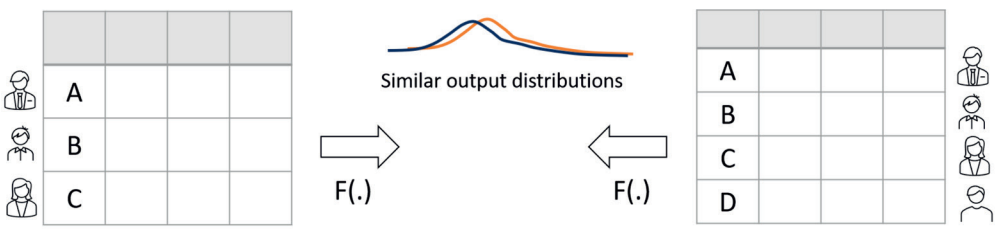


Figure 13: Differentially private mechanism

to make aggregate queries (e.g., count, sum, means, etc.) differentially private. This uses the Laplace probability distribution centred at 0 and scaled by $1/\epsilon$ to sample the random noise. Perturbing the real value by adding the obtained noise results in a masked response.

How is a simple differential privacy mechanism differentially private?

Suppose Asha Hospital provides counselling to patients with drug addiction. The hospital maintains the data of such patients collected through the healthcare app. Now, suppose Betaal – who can query this database – wants to know whether Amit is receiving such counselling. Betaal can craft multiple queries as part of the differencing attack, such that they reveal Amit's counselling status as 'Yes' or 'No'. For example, suppose Betaal uses the COUNT query and the result is 30, i.e., thirty people are receiving the counselling service. If the response to the second COUNT query that excludes Amit's name is 29, then Betaal can conclude that Amit is receiving counselling and therefore addicted to drugs. However, if the second COUNT query result is 30, Betaal would conclude the exact opposite. If we use the interactive DP set-up having a mechanism such as the Laplace mechanism (refer to Figure 11), Betaal will always receive noisy results. The result may be centred around 29 or 30, but it may return values like 28 or 31, or even 27 or 32 with an even smaller probability. Therefore, it is difficult for Betaal to be sure whether the true answer is 29 or 30. This goes back to the notion of Dalenius, wherein, despite having access to the database, Betaal's knowledge of whether Amit is receiving counselling or not will not change because the noisy response does not add anything to his prior knowledge.

There are many mechanisms with related algorithms that can be used instead of the Laplace mechanism: for example, an exponential mechanism, a private multiplicative weights algorithm and, a multiplicative weights exponential algorithm. With such mechanisms, a realization of DP-based software systems is possible. However, there are practical challenges: for instance, if the same query must always receive the same noisy response, then it requires a look-up into the log of historic responses. There is no leakage of information, since the answer remains the same, but a log look-up can be expensive in terms of space and time.

What if Betaal poses a query structurally different from but equivalent to the query he has posed earlier? It is difficult for the system to establish the equivalence of two queries as this is known to be computationally hard. Therefore, while DP has several advantages over the traditional privacy-preserving approaches, there are certain limitations.

- Identifying the ideal privacy loss parameter ϵ to have high utility while preserving privacy remains a challenge.
- The privacy guarantee in DP exists only for a bounded number of queries, which is a function of the number of distinct data subjects represented in the dataset (Ullman, 2016). Thus, the challenge of designing privacy-preserving mechanisms that can handle an arbitrary number of queries remains open.
- Differential Privacy is open to side channel attacks: an adversary can learn facts about the data by monitoring side channels. A classic example is the timing-channel attack. Suppose a query computes the frequency of various medical diagnoses, and Betaal (the adversary) knows that the computation will take 51 µs if a person has cancer and 49 µs otherwise. Then, by merely observing the amount of time spent, Betaal can learn whether a given individual has cancer.
- Sensitive attribute disclosure is possible, for example, an adversary can build a classifier over differentially private data that can predict sensitive information (Cormode, 2011).

So far, we have discussed the standard DP model where the data curator is trustworthy. However, the data curator may be compromised and hence becomes an untrusted source. This necessitates a shift in the DP model from standard differential privacy (SDP) to local differential privacy (LDP).

Local differential privacy treats the aggregator as untrusted. Figures 14 and 15 outline the two settings. In the SDP model (Figure 14), noise is injected into the original database belonging to a trusted curator, and the end-user can then perform an analysis. In the LDP model (Figure 15), the noise is injected locally, i.e., at the individual level for each data subject, and such perturbed data is aggregated by the untrusted curator. This way, privacy control stays with the data subject.

Further, with privacy regulations such as GDPR (General Data Protection Regulation, a law that governs the processing of personal data belonging to European Union citizens and residents) and CCPA (California Consumer Privacy Act, a law that regulates how businesses worldwide are allowed to handle personal information related to California residents), large organizations use the LDP model to avoid the liability arising from misuse of storing sensitive user data. So, with trust assumptions, LDP is more attractive for deployment in DP-based systems. However, the utility of the statistics released with LDP is poorer than of those with SDP because in the LDP model, perturbation happens locally at each individual's end, resulting in larger noise addition. One fallout of this model is that there is no single source of unperturbed data anymore. Consequently, the gap between SDP and LDP can be interpreted as 'high trust assumptions with high utility' in SDP and 'lower trust assumptions with lower utility' in LDP. Some recent research work exploits cryptographic primitives to bridge the trust–utility gap between SDP and LDP to have the best of both worlds. Figure 16 shows that SDP can achieve high utility (lower error) while LDP does not rely on a trusted curator and achieves lower utility (high error). The goal is to achieve the utility of the SDP under the more practical assumptions of LDP. Use of cryptographic primitives opens a new direction of research to evolve DP as a promising privacy preserving technique (Wagh et al., 2021)

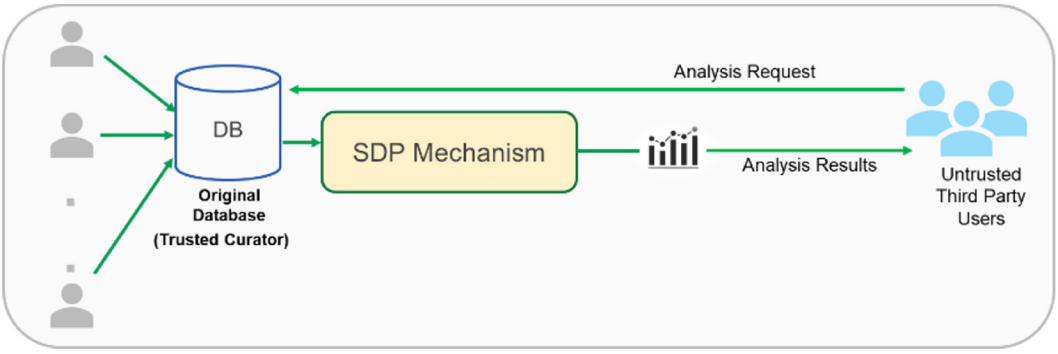


Figure 14: Standard Differential Privacy

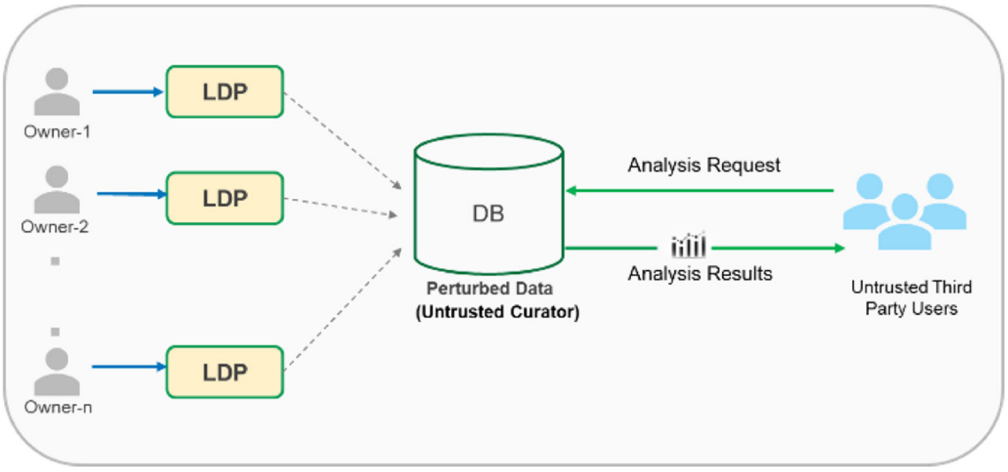


Figure 15: Local Differential Privacy

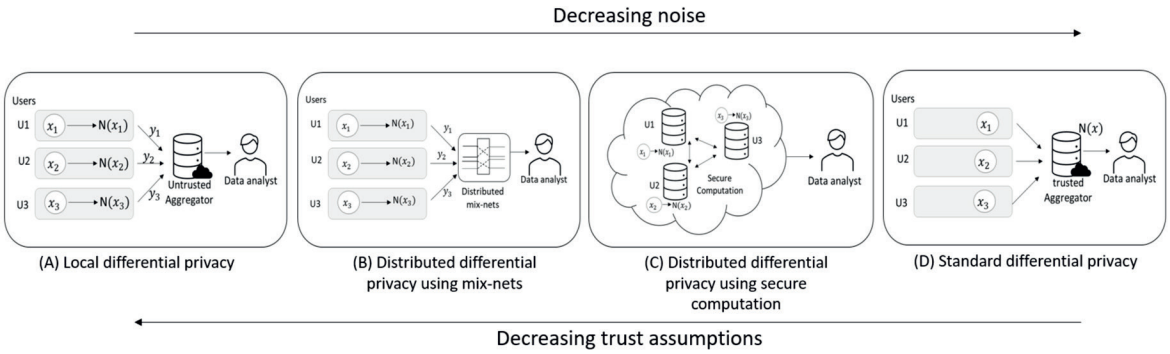


Figure 16: Deployment scenarios for Differential Privacy (adopted from Figure 2 in (Wagh et al., 2021))

4.3 An industrial outlook

Differential Privacy as a technique has a much wider role to play in many application areas including cyber physical systems (Figure 17) such as smart grid transportation, healthcare systems, Internet of Things (IoT), autonomous automobile systems and other application areas like social networks and, sensitive data mining. For example, smart grid transportation relies on smart meters that record household energy consumption and communicate this back to power providers. This can save money and aid energy conservation but will also create massive amounts of new data that can reveal intimate details about households and the people who live in them. The risk of misuse of such data raises a new set of privacy concerns for consumers, making it necessary to incorporate privacy preserving techniques into such systems. Similarly, for healthcare and medical systems, data collected by IoT devices, such as blood pressure, blood sugar levels, and sometimes even the location details, also need to be obtained in a privacy-aware manner.

Tech Giants like Google, Microsoft, and Apple use DP in various application services, such as those that collect or release micro-data. For example, Microsoft used LDP to protect user privacy in applications like telemetry in Windows and, suggested replies in MS Office. Apple also uses LDP to help protect the privacy of user activity in a given time period while still gaining insights that improve the intelligence and usability of such features as Quick Type suggestions and Emoji suggestions. Google's RAPPOR system integrated in Chrome acquires data in a privacy-aware manner on how unwanted software hijacks user settings. Companies like IBM and Google provide libraries for carrying out machine learning tasks in a DP-aware manner. Although these companies have used DP in several applications, researchers have questioned whether such implementations of DP in practice provide adequate privacy guarantees. Applying DP to record-level data collection or release requires employing a large amount of noise to ensure a safe ϵ. If $\epsilon \leq 1$, the analytical utility of DP outputs is likely to

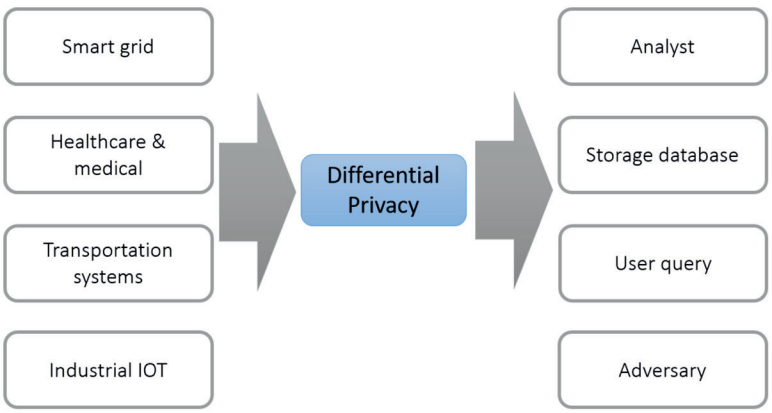


Figure 17: Use-case scenarios of Differential Privacy

be very poor (Bambauer et al., 2013). One way around this could be to use a very large value of ϵ to mitigate the utility problem. For example, Apple reportedly uses $\epsilon = 6$ in MacOS and even $\epsilon = 43$ for iOS 10 beta versions, whereas in RAPPOR, Google uses ϵ up to 9. This shows that the applicability of DP in practice is still a challenge as the privacy guarantee of DP is greatly diminished for such larger values of ϵ (Domingo-Ferrer et al., 2021).

4.4 Discussion

In the previous sections, we saw how the need for data privacy unfolded from the standard use-case of data publishing to privacy-driven analytics. Here DP has gained significant attention as it offers mathematical guarantees. However, there are challenges while mapping the theory of DP to practice.

What is the ideal differential privacy setting?

The ideal DP setting for a particular use-case is one that can mitigate the threats and risks of disclosure of sensitive data while keeping the data utility high. The requirement of privacy has always been part of the context. There is no universal DP setting that will work for all use-cases. In a situation where the data controller is a trusted entity, we can use the standard DP model, which ensures that the data is safe with the data controller; for the malicious data analyst, the model prevents the risk of disclosure by using different DP mechanisms. If the data controller is untrusted, the local DP model is used. The data is first safeguarded locally using DP mechanisms and then shared with the controller. This way the data controller has access to noisy data and not the original data, thus preventing its misuse. Further, it also prevents a malicious data analyst from leaking sensitive information. To conclude, based on the scenario, one can choose a suitable DP setting.

Which is the right Differential Privacy mechanism?

There is no universal DP mechanism that is effective for all use-cases. For example, the Laplace mechanism can be used only for numeric queries whereas the exponential mechanism can address both numeric and categorical data in the queries. Thus, the applicability of a mechanism varies based on the use-case and the type of data.

That said, there are many DP algorithms catering to specific use-cases. New algorithms are being proposed that don't meet the precise mathematical definition and are therefore referred to as 'almost differentially private'.

How to select the value of ϵ?

The value of ϵ can be used to determine the level of privacy. The smaller the value of ϵ, the better the privacy but the accuracy of the results may be affected. DP researchers suggest that ϵ greater than 6 may not be good from a privacy point of view (Greenberg, 2017). Although that is certainly a good goalpost, it often may not be achievable given the nuances of the use-case. Further, the choice of ϵ may vary from application to application depending upon the need for privacy in that context. In general, questions like "What value of ϵ is appropriate?" and "How much privacy is enough?" remain unanswered. There is no easy guide for this, and the best practices have not evolved yet.

When should the use of differential privacy be discontinued?

Privacy losses accumulate (Ullman, 2016), i.e., with each new query, the privacy guarantee decreases as additional information about the sensitive data is released. This implies that after a certain number of queries, the application of DP provides no privacy guarantees. Ideally, we would want the privacy loss to be small for strong privacy guarantees. Thus, to mitigate the risks with growing privacy loss, one can enforce a maximum privacy loss denoted by the privacy budget. Each query can be treated as a privacy expense that incurs an incremental privacy loss. Thus, if the number of queries exceeds this privacy budget, then one can stop answering the queries, thereby discontinuing DP. To not hit the privacy budget limit, one could attempt to make the privacy expense almost zero for most queries. However, the choice of parameters would typically mean very noisy responses and, therefore, negligible data utility. Thus, owing to either privacy or utility considerations, DP may not be suitable for long-running systems.

5. Emerging Techniques

OECD promotes privacy as a fundamental requirement and provides guidelines on the protection of privacy and regulating transborder flows of personal data. The OECD principles cover the essential dimensions of privacy that need attention when building any system. These principles are collection limitation, data quality, purpose limitation, use limitation, security safeguards, openness, individual participation, and accountability. These principles help to govern the privacy requirements of a system. Data privacy considerations exist during the entire lifecycle of data in any system. Therefore, privacy concerns of the data at rest, in transit, and in execution should be addressed. Techniques such as k-anonymity and DP provide a strong foundation for data privacy. However, in a scenario with a rise in the complexity of the systems, where the storage unit and computing unit may not be centralized, mitigating privacy disclosure risks is challenging. Such systems, based on, for example, IoT sensors, wearable computing devices, mobile computing and, smart meters require us to design stronger privacy techniques and protocols. These are privacy techniques and protocols that take into consideration the deployment architecture, compute availability at various nodes in the system, sensitive data flows and, various threat models.

Swara wonders

How can we go beyond k-anonymity and differential privacy to preserve privacy for complex systems? Can these existing techniques be extended?

Interestingly, research in this direction has been evolving rapidly, and different frameworks and methodologies are being proposed. Suppose the healthcare app used by Swara needs to offer a disease prediction functionality. Such a feature may help Swara; however, using it would require reporting symptoms from time to time. Based on the symptoms, the app would train a predictive machine learning model using inputs from all its subscribed users. Since all this sensitive information is being stored and processed, Swara and other patients are bound to have serious privacy concerns. Let us now see how techniques such as federated learning could be applied here to get the best of both privacy and utility.

To build a global model for disease prediction while preserving privacy, a local model is trained over the data residing locally on each user's mobile device. The learned model parameters are sent by each user device to the cloud server where aggregation is performed to build a global model. This learned global model is then pushed to each user's mobile device for predictions. This process of learning and improving a model locally, then pushing those updates to build the global model centrally and pushing the global model back for local usage could continue. Note that the data storage is local, giving the user control over her data while computation happens in a remote server without operating on the raw data. Thus, learning in such a set-up enables privacy by sending parameters and not sending a users' data to a

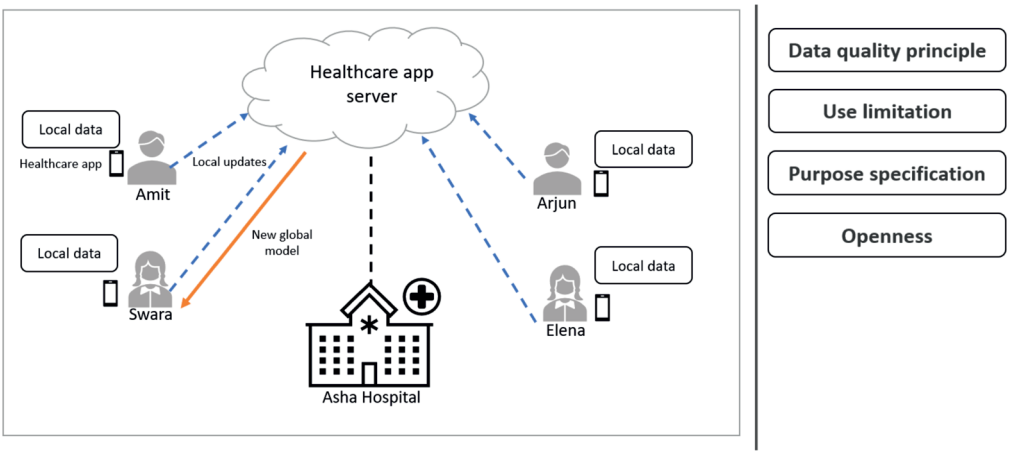


Figure 18: Federated learning architecture for a hospital setting

centralized compute server. This simple federated learning architecture (Figure 18) also helps to realize some of the OECD's privacy principles mentioned earlier.

These distributed architectures have further expanded with IoT data analytics. For example, as part of edge computing, heavy computation tasks get offloaded to the edge nodes while client devices such as IoT sensors and, smart gadgets are assigned a lightweight task whose output is then used to perform heavyweight tasks at the edge nodes. Here, the local differential privacy obfuscation (LDPO) Framework is being proposed to ensure data privacy and guarantee data utility for edge computing. The basic approach of the LDPO framework is to add noise to prevent leakage of private information. However, adding noise might degrade data utility, which is why the LDP-based data distillation model is proposed, which limits the collection of personal data while still maximizing data utility. The LDPO framework is based on components that involve learning the most compact and useful features of data using data minimization and perturbing these identified features with LDP for privacy guarantees. Further, these features are anonymized to a k-bit string using different hash functions so that the transformation yields a unique string. Lastly, the data is transmitted to edge servers, where feature rebuilding and distribution estimation are performed using hash functions for data reconstruction, thereby preventing sensitive attributes from getting exposed.

Suppose Swara has participated in a research study where her health parameters are being collected using a wearable health gadget. The LDPO framework, as illustrated in Figure 19, safeguards Swara's personal data and helps in realizing some OECD principles.

Besides the federated learning kind of distributed architectures, fully homomorphic encryption (FHE) and secure multi-party computation (SMPC) are cryptographic techniques that can be used for private computation on data (Figure 20).

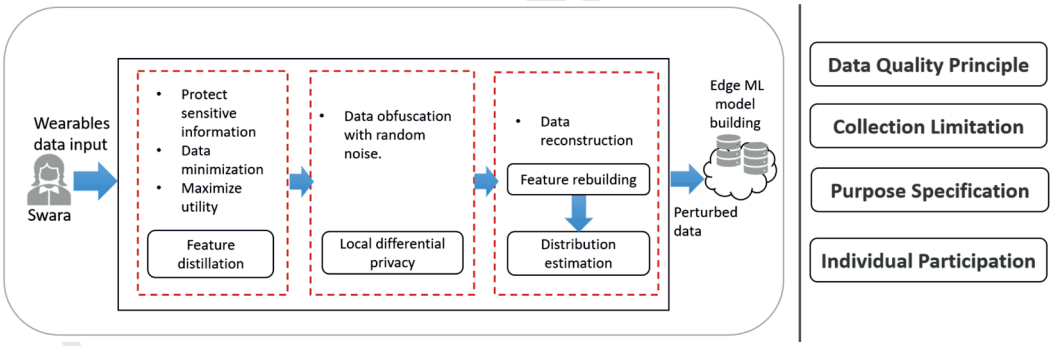


Figure 19: Local Differential Privacy Obfuscation framework

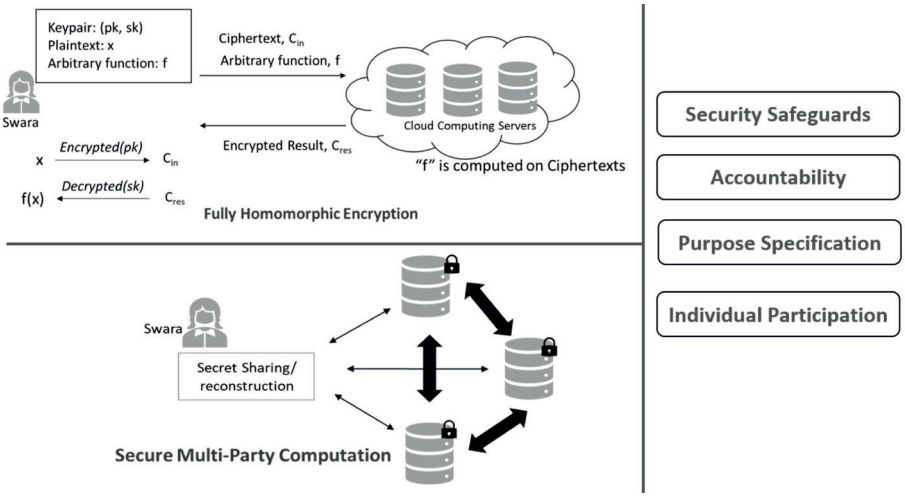


Figure 20: Privacy-preserving data transmission

Fully homomorphic encryption is an encryption scheme that enables analytical functions to be run directly on encrypted data while yielding the same encrypted results as if the functions were executed on plaintext. If Asha Hospital were to utilize FHE for disease prediction, Swara's records would be homomorphically encrypted on her device locally and sent to the cloud server for processing. The outcome of the prediction model running in the cloud would also be in the encrypted form that would be sent to Swara's device. Swara would be able to decrypt the response while no one else, including the cloud administrators, would be able learn anything about Swara's condition. Although this is exciting from a security and privacy standpoint, at the current state of the art, FHE computations run about a million times slower compared to the equivalent plaintext computations. That said, this is already a big improvement over the original trillion times slow down at the inception of FHE. Several open-source implementations of FHE schemes exist today, and efforts are on to come up with even more efficient schemes and implementations as well as to standardize FHE given its potential benefit to cloud computing.

Alternatively, SMPC allows multiple parties to perform computations on their private data to evaluate a function of common interest: SMPC is highly applicable to machine learning areas as it allows companies to offer their models to perform inferences on private data of clients while ensuring utmost privacy. For example, Asha Hospital's healthcare application central server could be hosted on a cloud with every registered patient of the hospital, including Swara, having the healthcare app on their device. Using SMPC, the cloud service provider could execute a trained classification model by securely sharing patient data and sending the securely computed result (such as prediction of a disease) back to the patient. Several SMPC techniques have been known for a while; however, many of them involve significant message-passing overheads. Research is in progress to develop inexpensive, efficient, and effective SMPC techniques. Also, there are attempts to judiciously combine SMPC and FHE techniques to come up with hybrid schemes having acceptable time and communication complexity.

6. Concluding Remarks

With the world's information now being reshaped in digital form, privacy of individual information has become a crucial concern for individuals and organizations. It is vital for organizations to understand and address the privacy concerns attached to any activity over data. In this minigraph, we discussed how Asha Hospital can apply various privacy-preserving techniques. Each of the techniques has different strengths and weaknesses, depending upon the context (use-case). There is no silver bullet to privacy yet, that is, there does not exist a universal approach to guarantee privacy, but by using state-of-the-art privacy preserving techniques, the potential damage caused due to a privacy breach can be largely averted.

Swara, being privacy aware, understands the value of the data she holds and the need to safeguard it. On the other side, organizations following privacy principles and using privacy-preserving techniques in their operations can prevent people like Betaal with malicious intent. This minigraph summarizes the journey, starting with seeing privacy as a challenge and responding with techniques such as k-anonymity, and then moving to formal assurance with DP. The promise of privacy goes beyond these two techniques and extends to new approaches, such as federated learning, LDPO, FHE, etc. We have also seen how building complex systems becomes easier if they incorporate well-accepted privacy principles.

Acknowledgments

Writing the minigraph was harder than we thought but more rewarding than we ever imagined. None of this would have been possible without the motivation and guidance provided by Mathai Joseph. He helped us shape this minigraph from ideation to realization. We also thank our referees, namely John Mitchell (Professor, Stanford University), Yuval Elovici (Professor, Ben Gurion University), and Sitaram Chamarty (Principal Scientist, TCS Research) for their valuable and constructive suggestions, which helped us to revise and refine the minigraph. We also thank Freya Barua (Specialist, Talent Development, TCS) for helping us with her expert language editing, content orientation, and critical revisions of the draft while keeping the content intact and yet ensuring that readers can relate to it easily.

We thank TCS Research for providing us a conducive environment for our work. Without the experience and support of our peers at TCS Research, this minigraph would not have been completed. Last but not the least, we thank ACM India for giving us this opportunity to produce a minigraph.

References

Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D., & Zhu, A. (2005). Approximation Algorithms for k-Anonymity. Journal of Privacy Technology.

Bambauer, J., Muralidhar, K., & Sarathy, R. (2013). Fool's Gold: An Illustrated Critique of Differential Privacy. Vanderbilt Journal of Entertainment & Technology Law, 16, 701.

Bayardo, R. J., & Agrawal, R. (2005, April). Data Privacy Through Optimal k-Anonymization. In 21st IEEE International Conference on Data Engineering, pp. 217-228.

Ciriani, V., Di Vimercati, S. D. C., Foresti, S., & Samarati, P. (2008). k-Anonymous Data Mining: A Survey. In Privacy-Preserving Data Mining, pp. 105-136. Springer, Boston, MA.

Cormode, G. (2011, August). Personal privacy vs population privacy: learning to attack anonymization. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1253-1261).

Dalenius, T. (1977). Towards a methodology for statistical disclosure control. statistik Tidskrift, 15(429-444), 2-1.

Domingo-Ferrer, J., Sánchez, D., & Blanco-Justicia, A. (2021). The Limits of Differential Privacy (and Its Misuse in Data Release and Machine Learning). Communications of the ACM, 64(7), 33–35.

Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2006). Calibrating Noise to Sensitivity in Private Data Analysis. In Theory of Cryptography Conference, pp. 265-284.

Gavison, R. (1980). Privacy and the Limits of Law. The Yale law journal, 89(3), 421-471.

Greenberg, A. (2017). How One of Apple's Key Privacy Safeguards Falls Short. Wired. https://www.wired.com/story/apple-differential-privacy-shortcomings/.

Kisilevich, S., Rokach, L., Elovici, Y., & Shapira, B. (2009). Efficient multidimensional suppression for k-anonymity. IEEE Transactions on Knowledge and Data Engineering, 22(3), 334-347.

Lefkovitz, N., & Boeckl, K. (2020). NIST Privacy Framework: An Overview.

Lodha, S., & Thomas, D. (2007, August). Probabilistic Anonymity. In International Workshop on Privacy, Security, and Trust in KDD, pp. 56-79.

Meyerson, A., & Williams, R. (2004, June). On the Complexity of Optimal k-Anonymity. In Proceedings of the Twenty-Third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 223-228.

Nissenbaum H (2004). Privacy as Contextual Integrity. Washington Law Review, 101-139.

Samarati, P., & Sweeney, L. (1998). Protecting Privacy When Disclosing Information: k-Anonymity and Its Enforcement Through Generalization and Suppression. Technical Report. SRI International.

Stokes, K., & Torra, V. (2012). Reidentification and k-Anonymity: A Model for Disclosure Risk in Graphs. Soft Computing, 16(10), 1657-1670.

Sweeney, L. (2002). k-Anonymity: A Model for Protecting Privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(05), 557-570.

Ullman, J. (2016). Answering n2 + o(1) Counting Queries with Differential Privacy Is Hard. SIAM Journal on Computing, 45(2), 473-496.

Wagh, S., He, X., Machanavajjhala, A., & Mittal, P. (2021). DP-Cryptography: Marrying Differential Privacy and Cryptography in Emerging Applications. Communications of the ACM, 64(2), 84-93.

Association for Computing Machinery
C/O Persistent Systems Ltd
"Pingala - Aryabhata"
Plot No 9A/12, CTS No.12A/12,
Erandawana, Pune 411004