Chapter 5: Privacy-enhancing technologies (PETs)

Draft anonymisation, pseudonymisation and privacy enhancing technologies guidance

September 2022

Information Commissioner's Office

About this guidance

This guidance discusses privacy-enhancing technologies (PETs) in detail. Read it if you have questions not answered in the Guide, or if you need a deeper understanding to help you apply PETs in practice. It is aimed at DPOs and those with specific data protection responsibilities in larger organisations.

If you haven't yet read the 'In brief' page on PETs in the Guide to Data Protection, you should read that first. It introduces this topic and sets out the key points you need to know, along with practical checklists to help you comply.

Contents

How can PETs help with data protection compliance? 3

What are privacy-enhancing technologies (PETs)? 3

How do PETs relate to data protection law? 4

What are the benefits of PETs? 4

What are the risks of using PETs? 5

What are the different types of PETs? 5

Are PETs anonymisation techniques? 7

When should we consider using PETs? 7

How should we decide whether or not to use PETs? 8

How do we determine the maturity of a PET? 8

What PETs are there? 11

Introduction 12

Homomorphic encryption (HE) 12

Secure multiparty computation (SMPC) 15

Private set intersection (PSI) 20

Federated learning 23

Trusted execution environments 26

Zero-knowledge proofs 28

Differential privacy 30

Synthetic data 35

Reference table 36

How can PETs help with data protection compliance?

At a glance

- PETs can help you demonstrate a 'data protection by design and by default' approach to your processing.
- PETs can help you to comply with the data minimisation principle by ensuring you only process the data you need for your purposes, and provide an appropriate level of security for your processing.
- You can use PETs to give access to datasets which would otherwise be too sensitive to share, while ensuring individuals' data is protected.
- However, you should not regard PETs as a "silver bullet" for data protection compliance. Your processing still needs to be lawful, fair and transparent.
- You should perform a case-by-case assessment (eg, through a data protection impact assessment (DPIA)) to determine if PETs are appropriate for your aims.

In detail

- What are privacy-enhancing technologies (PETs)?
- How do PETs relate to data protection law?
- What are the benefits of PETs?
- What are the risks of using PETs?
- What are the different types of PETs?
- Are PETs anonymisation techniques?
- When should we consider using PETs?
- How should we decide whether or not to use PETs?
- How do we determine the maturity of a PET?

What are privacy-enhancing technologies (PETs)?

PETs are technologies that embody fundamental data protection principles by minimising personal data use, maximising data security, and/or empowering individuals. Data protection law does not define PETs. The concept covers many different technologies and techniques. The European Union Agency for Cybersecurity (ENISA) refers to PETs as:

Quote

‘software and hardware solutions, ie systems encompassing technical processes, methods or knowledge to achieve specific privacy or data protection functionality or to protect against risks of privacy of an individual or a group of natural persons.’

How do PETs relate to data protection law?

PETs are linked to the concept of ‘data protection by design’, and are therefore relevant to the technical and organisational measures you put in place. They can help you implement the data protection principles effectively and integrate necessary safeguards into your processing.

PETs can help you demonstrate a ‘data protection by design and by default’ approach by:

- complying with the data minimisation principle, by ensuring you only process the data you need for your purposes;
- providing an appropriate level of security;
- implementing robust anonymisation or pseudonymisation solutions; and
- minimising the risk that arises from personal data breaches, by rendering the personal data unintelligible to anyone not authorised to access it.

Relevant provisions in the legislation

See UK GDPR Article 25 and Recital 78 (data protection by design and by default) and Articles 5(1)(f), 32 and Recital 83 (security)

What are the benefits of PETs?

PETs can help reduce the risk to individuals, while enable further analysis of personal data without a controller necessarily sharing it, or a processor having access to it. The ability to share, link and analyse personal data in this way can provide valuable insights while ensuring you comply with the data protection principles.

By using PETs, you can obtain insights from datasets without compromising the privacy of the individuals whose data is in the dataset. Appropriate PETs can make it possible to give access to datasets which would otherwise be too sensitive to share.

What are the risks of using PETs?

PETs should not be regarded as a silver bullet to meet all of your data protection requirements. Your processing must still be lawful, fair and transparent. Before considering PETs, you should assess the impact of the decision-making process, purpose specification (ie specifying a legitimate purpose for processing) and how you can comply with accuracy and accountability requirements.

Lack of maturity

Some PETs may not be sufficiently mature in terms of their scalability, availability of standards and their robustness to attacks. We provide some factors you should consider to assess maturity of PETs later in this guidance.

Lack of expertise

PETs can require significant expertise to set up and use appropriately. Insufficient expertise can lead to mistakes in implementation, and a poor understanding of how to configure the PET to deliver the appropriate balance of privacy and utility. If you do not have required expertise then you should consider using an off-the-shelf product or service which provides an appropriate level of support.

Mistakes in implementation

There may be differences between the implementation of a PET in theory and its practical application. Risks to individuals' rights and freedoms may arise as a result. Attacks and vulnerabilities should also be monitored regularly, to ensure that appropriate mitigation measures can be put in place.

A lack of appropriate organisational measures can lower or even completely undermine the effectiveness of a PET. Depending on the threat model, some PETs can assume a trusted processor is used (ie a processor trusted not to act in a malicious or negligent manner). In this case, assurances are mainly derived from organisational controls, including legal obligations (such as contractual controls), monitoring and auditing processes.

What are the different types of PETs?

This guidance provides an introduction to some PETs that you can use to help you comply with your 'data protection by design' obligations. They help you minimise the personal data you collect, and integrate safeguards into the processing. Many aspects of PETs are also relevant for individuals. However, this guidance focuses on PETs that organisations can use.

Several categories of PETs can help achieve data protection compliance, including 'data protection by design and default'. These include PETs that:

- reduce the identifiability of the individuals to whom the data you are processing relates. These can help you to fulfil the principle of data minimisation;
- focus on hiding and shielding data. These can help you achieve the requirements of the security principle; and
- split or control access to personal data. These can help you to fulfil both the data minimisation and security principles, depending on the nature of the processing.

PETs that derive or generate data which reduces or removes the identifiability of individuals

These aim to weaken or break the connection between an individual in the original personal data and the derived data. Examples include:

- differential privacy; and
- synthetic data.

These PETs can effectively reduce risk to individuals. However, the resulting data may provide less utility compared with the original data. This may reduce how close the randomised answers to queries are to the real ones (ie those without noise applied). This may mean the results may not be suitable for some types of processing where the actual results are required to fulfil the purposes.

PETs that focus on hiding, or shielding, data

These aim to protect individuals' privacy while not affecting the utility and accuracy of the data. For example:

- homomorphic encryption, which allows computation to be performed on encrypted data without revealing the plaintext; and
- zero-knowledge proofs, which allow one party to prove to another party that something is true, without revealing what that something is or indeed anything else (such as the underlying data).

PETs that split datasets or control access to certain parts of the data

These PETs aim to minimise the amount of personal data shared and to ensure confidentiality and integrity, while not affecting the utility and accuracy of the data.

They take a systems and data architectures approach to processing, managing, and storing personal data. These approaches define how personal data is collected, distributed, stored, queried, and secured, and how each component of the system communicates with each other. They may split data for computation or storage, or provide dedicated hardware to prevent the operating system or other application from accessing the personal data.

The nature of the processing using these approaches means that the linkability risk between the split data is significantly reduced.

Examples include:

- trusted execution environments (TEEs);
- secure multi-party computation (SMPC), including private-set intersection (PSI); and
- federated learning.

Are PETs anonymisation techniques?

PETs and anonymisation are separate but related concepts. Not all PETs result in effective anonymisation, and you can achieve anonymisation without using them.

At the same time, PETs can play a role in anonymisation, depending on the circumstances. For example, you can configure differential privacy methods to prevent information about specific individuals being revealed or inferences about them being made.

However, the purpose of many PETs is to enhance privacy and protect the personal data you process, rather than to anonymise that data. This means that:

- many PET use-cases still involve personal data; and
- when you deploy such techniques, you still need to meet your data protection obligations.

Further reading

See the sections of this guidance on identifiability and pseudonymisation for more information.

When should we consider using PETs?

PETs can help you achieve compliance with the data protection principles, particularly data minimisation, purpose limitation and security. They can help you protect individuals' privacy and effectively implement 'data protection by design'.

Whether a specific PET, or combination of PETs, is appropriate for your processing depends on your particular circumstances. You should consider implementing PETs at the design phase of any project. A data protection impact assessment (DPIA) is a useful tool that can guide your considerations.

PETs are particularly suitable in contexts that involve large-scale collection and analysis of personal data (eg AI applications, Internet of Things and cloud computing services).

How should we decide whether or not to use PETs?

You should perform a DPIA to determine if PETs are appropriate to meet your aims. Your assessment should consider:

- the nature, scope, context and purposes of your processing;
- the state-of-the-art and costs of implementation of any PETs; and
- the risks your processing poses to individuals' rights and freedoms.

The **nature** of the processing is what you plan to do with the personal data.

The **scope** of the processing is what the processing covers.

The **context** of the processing is the wider picture, including internal and external factors which might affect expectations or impact of the processing.

The **purpose** of the processing is the reason why you want to process the personal data.

You should consider the PET's **state-of-the-art** to understand whether it is sufficiently mature for your purposes, and to check that you keep informed about the PETs available as the market changes. You are not required to implement the newest technologies available.

The **cost** of a technique can be a factor in considering which PET to implement, rather than a reason for not implementing any privacy-enhancing measure.

Further reading

See our DPIA guidance for more information on nature, scope, context and purpose of the processing.

For further guidance, you should read the section on data protection by design and security in the ICO draft guidance on pseudonymisation.

How do we determine the maturity of a PET?

There are different ways to determine a PET's maturity. Technology Readiness Levels (TRLs) are a common approach. These categorise PETs into discrete categories of maturity from conceptual to market-ready products.

Some models (eg ENISA's PETs maturity assessment) combine TRLs with various quality measures including scalability, trust assumptions and levels of protection, and versatility for different purposes. These are used to generate a rating based on market maturity and the PET's quality.

Other approaches to assessing PET suitability focus more on:

- the protections the PET provides;
- the risks of personal data leakage for a given threat model used; and
- scalability and complexity issues.

Some PETs may be theoretical, immature or unscalable. These can be challenging to implement. Just because something exists at the cutting edge doesn't mean you have to implement it to comply with data protection law – particularly if it is not yet practical to do so.

Some PETs are newer or more theoretical than others, and standardisation can therefore be at its early stages. Where standards do exist, you should take them into account in the design and implementation of data protection measures. You should ensure that appropriate technical and organisational measures are in place to mitigate against risks for a given threat model, as defined by relevant standards (eg ISO and IETF standards).

For example, standards can provide further detail and guidance about:

- specific attacks and how these can be mitigated;
- technical and organisational measures required for a given threat model (eg contractual controls and security measures such as access control); and
- technical and organisational measures required to ensure the security properties are maintained (eg management of cryptographic keys and tuning and security parameters).

We have produced a table on the availability of industry standards for PETs.

Relevant provisions in the legislation

See Article 25 and Recital 78 of the UK GDPR

Further reading – ICO guidance

Read our guidance on data protection by design and by default.

Further reading outside this guidance

For more information on methodologies for assessing the maturity of PETs, see guidance from the European Union Agency for Cybersecurity (ENISA), including:

- Readiness analysis for the adoption and evolution of PETs (2016)
- PETs controls matrix: a systematic approach for assessing online and mobile privacy tools (2016)
- PETs: evolution and state of the art (2017)
- A tool on PETs knowledge management and maturity assessment (2018)
- ENISA’s PETs maturity assessment repository (2019)
- The Royal Society’s 2019 report-“Protecting privacy in practice” (external link, PDF) also provides information about the current use, development and limits of PETs.

What PETs are there?

At a glance

- PETs are available for a variety of purposes (eg secure training of AI models, generating anonymous statistics and sharing data between different parties).
- Homomorphic encryption provides strong security and confidentiality by enabling computations on encrypted data without first decrypting it.
- Secure multiparty computation (SMPC) provides data minimisation and security by allowing different parties to jointly perform processing on their combined data, without any party needing to share its all of its data with each of the other parties.
- Federated learning trains machine learning models in distributed settings while minimising the amount of personal data shared with each party.
- Trusted execution environments provide enhanced security by enabling processing by a secure part of a computer processor, which is isolated from the main operating system and other applications.
- Zero-knowledge proofs (ZKP) provide data minimisation by enabling an individual to prove private information about themselves without revealing what it actually is.
- Differential privacy generates anonymous statistics by adding noise to individual records.
- Synthetic data provides realistic datasets in environments where access to large real datasets is not possible.

In detail

- Introduction
- Homomorphic encryption (HE)
- Secure multiparty computation (SMPC)
- Private set intersection (PSI)
- Federated learning
- Trusted execution environments
- Zero-knowledge proofs
- Differential privacy
- Synthetic data
- Reference table

Introduction

There are many PETs which you may consider as part of data protection compliance. The purpose of this section is to outline some of these, and summarise their benefits for compliance, residual risks and implementation considerations.

This section is not:

- a comprehensive list of PETs;
- an ICO endorsement of any particular PET; or
- a deep technical examination of each PET.

Depending on your circumstances you may need to procure specialist expertise beyond this guidance.

We plan to update this guidance in due course as technology develops to reflect changes in the state-of-the-art (eg as new techniques become available).

Homomorphic encryption (HE)

What is homomorphic encryption and what does it do?

Homomorphic encryption allows you to perform computations on encrypted data without first decrypting it. The computations themselves are also encrypted. Once you decrypt them, the result is an output identical to what would have been produced if you had performed the computation on the original plaintext data.

There are three types of homomorphic encryption:

- fully (FHE);
- somewhat (SHE); and
- partial (PHE).

The HE scheme you choose will depend on the nature, scale and the purpose of your processing and the level of utility you require to fulfil your purposes. You also need to consider the number of different types of mathematical operations the HE scheme supports, as well as any limit to how many operations the scheme can perform.

Type of HE	When would this type of HE be appropriate?
FHE	FHE allows you to compute any function, as there are no limitations in terms of the types of operations it supports or their complexity. This flexibility means it provides good

	protection and utility. However, the more complex the operation, the more resource and time may be required.
SHE	SHE permits fewer additions and multiplications on encrypted data. The amount is also fixed in advance. This in turn means that there is a limit on the types of functions it can support.
PHE	PHE provides good performance and protection, but limited utility. It supports only addition or multiplication operations, but not both. As with SHE, there is a limit on the types of (but not the number of) functions it can support.

HE uses a public key-generation algorithm to generate a pair of private and public keys, and an evaluation key. The evaluation key is needed to perform computations on the encrypted data when it is shared with the entity that will perform them. This entity does not need access to the private key to perform the analysis. The client, who retains the private key, can then decrypt the output obtain the result they require. Any entity that has only the public and the evaluation keys cannot learn anything about the encrypted data in isolation.

How does HE assist with data protection compliance?

In the context of processing, this activity likely counts as "consultation" and "use" of personal data, with the encryption itself being "adaptation or alteration" of that data. That means the data is still personal data. It has just been treated in a particular way that enhances the privacy of those to whom it relates.

HE can help you to ensure:

- **security and confidentiality**. It can minimise the risk from data breaches if they occur, as personal data remains encrypted at rest, in transit and during computation. For example, HE renders the data unintelligible to an attacker, the risks to individuals are reduced, and therefore no notification to individuals is required under Article 34 of the UK GDPR; and
- **accuracy.** It provides a level of assurance that the result of a computation is the same as if you performed it on unencrypted data – providing you ensure the inputs are correct prior to encryption taking place. This is because HE does not require you to alter the data in other ways (eg, adding "noise" like differential privacy) that mean the result may be different from performing the processing on unencrypted data.

HE can also be a building block for other PETs such as private-set intersection and federated learning.

HE can provide a level of guarantee to a controller when outsourcing a computation in an untrusted setting, without the other party ever learning about the "original" unencrypted data, the computation, or result of the computation.

What do we need to know about implementing HE?

FHE can add significant computational overhead (several thousand times slower than processing plaintext) and increased communications cost. It may therefore not be appropriate if your processing involves large volumes of personal data.

FHE's performance deficit is reducing due to technological progress – for example, increasing computational power and efficiency improvements of the FHE algorithms. This means challenges relating to computational overhead and cost are likely to become less significant over time, and FHE may in turn become more viable in the context of large-scale processing operations. However, at present FHE performs better for some types of computation like addition operations, but it is still not feasible for many types of processing.

Other schemes such as PHE and SHE are less affected by overhead but are more limited in terms of mathematical operations they support.

What are the risks associated with the use of homomorphic encryption?

HE has similar risks to encryption more generally. You need to ensure that you:

- choose the right algorithm;
- choose the right key size;
- choose the right software; and
- keep the key secure.

This is particularly important with HE because the secret key can be used to decrypt the outputs. You must therefore use appropriate technical and organisational measures to keep it secure. You must also ensure you have processes in place to generate a new key immediately in case the original is compromised.

The security of most HE schemes is based on hard mathematical problems which are currently considered to be secure even against quantum computers. You should monitor the effectiveness of your HE scheme as decryption technologies continue to develop.

There are also off-the-shelf HE products and services, including open-source solutions. These can help you to implement HE if you do not have the sufficient technical expertise. For example, these products and services can provide things like:

- the underlying cryptographic operations;
- application programming interfaces (APIs);
- key generation;
- encryption and decryption; and
- particular addition or multiplication functions.

Additionally, industry efforts to standardise HE schemes are ongoing. You should monitor the effectiveness of the solution you choose as technologies continue to develop.

Further reading – ICO guidance

For more information on protecting encryption keys, read our guidance on encryption.

For more information about assessing identifiability, see the identifiability section of our anonymisation guidance.

Further reading outside this guidance

The current version of the community standard for homomorphic encryption includes further guidance on best practices.

OpenMined's blog on "What is homomorphic encryption?" provides further information on the mathematical operations that underpin HE.

This link provides a curated list of Homomorphic Encryption libraries, software and resources

Secure multiparty computation (SMPC)

What is secure-multiparty computation (SMPC) and what does it do?

SMPC is a protocol (a set of rules for transmitting data between computers) that allows at least two different parties to jointly perform processing on their combined data, without any party needing to share its all of its data with each of the other parties. All parties (or a subset of the parties) may learn the result, depending on the nature of the processing and how the protocol is configured.

SMPC uses a cryptographic technique called "secret sharing", which refers to the division of a secret and its distribution among each of the parties. This means that each participating party's data is split into fragments to be shared with other parties.

Each party's data cannot be revealed to the others unless some proportion of fragments of the data of each of the parties are combined. As this would involve compromising the data security of a number of different parties, in practice it is unlikely to occur. This limits the risks of exposure through accidental error or malicious compromise and helps to mitigate the risk of insider attacks.

Example

Three organisations (Party A, Party B and Party C) want to use SMPC to calculate their average expenditure. Each party provides data about their own expenditure – this is the "input" that will be used for the calculation.

SMPC splits each party's information into three randomly-generated "secret shares". For example, Party A's input – its own total expenditure – is £10,000. This is split into secret shares of £5,000, £2,000 and £3,000. Party A keeps one of these shares, distributes the second to Party B and the third to Party C. Parties B and C do the same with their input data.

Party	**Input data**	**Secret share 1 (to be kept)**	**Secret share 2 (to be distributed)**	**Secret share 3 (to be distributed)**
A	£10,000	**£5,000**	**£2,000**	**£3,000**
B	£15,000	**£2,000**	**£8,000**	**£5,000**
C	£20,000	**£7,000**	**£4,000**	**£9,000**

When this process is complete, each party has three secret shares. For example, Party A has the secret share it retained from its own input, along with a secret share from Party B and another from Party C. The secret shares cannot reveal what each party's input was – ie Party A does not learn the total expenditure of Parties B or C, and so on.

Each party then adds together their secret shares. This calculates a partial result both for each party and the total expenditure of all three. The SMPC protocol then divides the total by the number of parties – three, in this case – giving the average expenditure of each: £15,000.

Party	**Input data**	**Secret share kept**	**Secret share received**	**Secret share received**	**Partial sum**
A	£10,000	**£5,000**	**£4,000**	**£5,000**	£14,000
B	£15,000	**£2,000**	**£2,000**	**£9,000**	£13,000
C	£20,000	**£7,000**	**£8,000**	**£3,000**	£18,000

Total expenditure (sum of partials)	£45,000
Average expenditure (total divided by number of parties)	**£15,000**

No single party is able to learn what the other's actual expenditure is.

How does SMPC assist with data protection compliance?

SMPC is a way to ensure that the amount of data you share is limited to what is necessary for your purposes. It can help you to demonstrate:

- the **security** principle, as the inputs of other parties are not revealed and internal or external attackers cannot easily change the protocol output; and
- the **data minimisation** principle, as no one should learn beyond what is absolutely necessary. Parties should learn their output and nothing else.

SMPC can also help to minimise the risk from personal data breaches when performing processing with other parties, as the shared data is not stored together, and also when data is being processed by separate parts of the same organisation.

If your purposes require you to provide personal data to the SMPC computation, you need to assess whether the data you receive from the output is personal data. You should consider applying differential privacy to the output to further reduce risks of identifiability.

What do we need to know about implementing SMPC?

SMPC is an evolving and maturing concept. It may not be suitable for large-scale processing activities in real-time, as it can be computationally expensive. There are some other open research problems, including the use of SMPC for replacing missing data with substituted values, eliminating duplicate copies of repeating data and record linkage where matches in data sets to be joined are inexact.

Currently, effective use of SMPC requires technological expertise and resources. This may mean that you cannot implement SMPC yourself. However, SMPC has different deployment models, meaning that it may be possible for you to use it. These include:

- the delegated model, which outsources the computations to a trusted provider. This can also be a good approach if you are reluctant to participate in the protocol due to security and confidentiality concerns – for example, the risk of collusion between other parties or mismatched levels of security between parties; and

- the hybrid model, which involves an external provider running one of the servers, while you run the other in-house using the same technical and organisational measures. This approach still requires a solid understanding of the technology.

Beyond a certain point, it may be possible for the input data to be reconstructed (eg by one or more of the parties, or an attacker). It is therefore important to determine what the appropriate "threshold" is for the number of secret shares your use of SMPC involves.

The threshold for reconstruction influences the risk of collusion and re-identification. The required threshold depends on the threat model used. A threat model which requires a greater proportion of the parties to be honest poses a higher risk than one which requires a lower proportion. For example, if all but one parties must be honest, then compromise of a single party would undermine the security of the protocol. If only half of the parties are required to be honest, then this would require further parties to be compromised or behave maliciously.

There are several parameters that you should consider when you determine the appropriate number of shares. These include:

- the number of parties involved;
- the underlying infrastructure you intend to use;
- the availability of that infrastructure; and
- the calculations you intend to make and the input data required.

To avoid collusion between parties, you should ensure appropriate trust mechanisms are in place, particularly if multiple parties involved in the process use the same underlying infrastructure. These may include robust access controls, logging and auditing mechanisms and a strong contractual framework.

You may need to obtain further expertise in secret sharing when assessing the context and purpose for your use of SMPC.

What are the risks associated with the use of SMPC?

SMPC protocols are designed for a variety of threat models that make assumptions about an attacker's capabilities and goals. The models are based on allowed actions that dishonest parties are allowed to take without affecting its privacy properties. This is an important underlying concept behind the design of SMPC.

An SMPC protocol can be compromised, resulting in reconstruction of the input data or the results of the computation being incorrect. For example, an external entity or a participating party can act in bad faith. In the SMPC context these are known as 'corrupted parties'.

The security model appropriate for your circumstances depends on the level of inherent risk of a malicious party learning something about an individual, or corrupting their inputs such that it may have a detrimental effect on an individual.

Generally, stronger threat models require more computational resources. It is good practice to perform intruder testing on the SMPC protocol operation using the threat model assumptions for a given adversary, as provided in the design of the protocol. For example, you should test the impact of corrupted inputs on the computation and the security of the communications channels between the parties.

By design, using SMPC means that data inputs are not visible during the computation, so you need to carry out accuracy checks. You can do this in several ways, such as:

- ensuring the design has measures in place to protect against corruption of the input values (eg a process for checking the input values and contractual requirements on accuracy);
- ensuring that data validation and correction is part of the SMPC protocol you choose, and that both processes are executed on the inputs;
- checking the output after the computation is complete, so you can evaluate whether the result is true (this process is known as "sanity checks");
- bounds checking to ensure values are not corrupted; and
- ensuring technical and organisational measures are in place (eg robust access controls, logging and auditing mechanisms to mitigate the risk of collusion between parties).

SMPC protects data during the computation but does not protect the output. Where the output is personal data, you should implement appropriate encryption measures for data at rest and in transit to mitigate the risk of personal data being compromised.

Further reading – ICO guidance

Read the section of this guidance on identifiability for more information on the motivated intruder test and assessing the identifiability of personal data.

Further reading outside this guidance

The publications below provide additional information on implementation considerations, threat models and use cases for SMPC.

For an extensive overview of SMPC, including an assessment of various methods and a summary of what problems it can solve, see "A Pragmatic Introduction to Secure Multi-Party Computation" (external link, PDF).

ENISA's 2021 report "Data Pseudonymisation: Advanced Techniques and Use Cases" summarises SMPC.

Private set intersection (PSI)

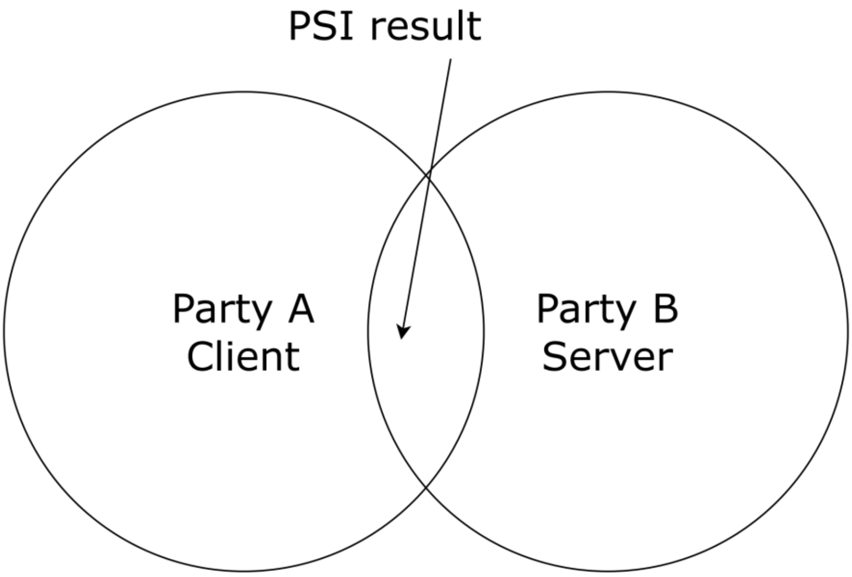
What is private set intersection (PSI) and what does it do?

PSI is a specific type of SMPC which allows two parties, each with their own dataset, to find the "intersection" between them (ie the elements the two datasets have in common), without revealing or sharing those datasets. It can also be used to compute the size of the intersection or aggregate statistics on it.

The most common type of PSI is the client-server subtype, where only the client learns the PSI result. The client can be the user of a PSI service or the party who will learn the intersection or intersection size (the number of matching data points between the two parties), depending on the purposes. The server hosts the PSI service and holds data which the client can query to determine if it holds any matching data with the server.

PSI can work in two ways:

- the data owners interact directly with each other and need to have a copy of their set at the time of the computation, known as traditional PSI; or
- the computation of PSI or the storage of sets can be delegated to a third-party server, known as delegated PSI.



The most efficient PSI protocols are highly scalable and use a variety of methods, including other privacy enhancing techniques such as hashing or homomorphic encryption.

How does PSI assist with data protection compliance?

PSI can help to achieve **data minimisation** as no data is shared beyond what each party has in common.

PSI offer the same benefits as other SMPC protocols, such as:

- no single party being able to have a 'global view' of all combined identifiable input data from both parties;
- the parties involved in each stage of the processing receiving the minimum amount of information tailored to their requirements, preventing purpose creep; and
- PSI protocols being modified to show only anonymous aggregate statistics from the intersection, depending on the requirements of the sharing.

Example – Using Private Set Intersection

Two health organisations process personal data about individuals' health. Organisation A processes data about individuals' vaccination status, while Organisation B processes data about individuals' specific health conditions.

Organisation B needs to determine the percentage of individuals with underlying health conditions who have not been vaccinated.

Ordinarily, this may require Organisation A to disclose its entire dataset to Organisation B so the latter can compare with its own. By using PSI, it does not need to do so. In fact, both organisations can minimise the amount of personal data processed while still achieving their purposes.

A third party provides the PSI protocol. While the computation involves processing of the personal data that both organisations hold, the output of that computation is the number of individuals that are not vaccinated who have underlying health conditions. Organisation B therefore only learns this, and does not otherwise process Organisation A's dataset directly.

This minimises the personal data needed to achieve the purpose. This in turn enhances individuals' privacy.

What are the risks associated with the use of PSI?

PSI introduces some risks that you need to mitigate. These include:

- risks of re-identification from inappropriate intersection size or over-analysis; and

- the potential for one or more of the parties to use fictional data in an attempt to reveal information about individuals.

You should choose an appropriate intersection size. This is because a low **intersection size** may allow the party computing the intersection to single out individuals within that intersection in cases where an individual's record has additional information associated with it (eg numerical values for hospital visits). These values can be added together and used for publishing aggregates (known as the intersection sum).

If an identifier has a unique associated value, then it may be easy to detect if that identifier was in the intersection by looking at the intersection sum and whether one of the identifiers has a very large associated value compared to all other identifiers. In that case, if the intersection sum is large, it is possible to infer that that identifier was in the intersection.

The intersection sum may also reveal which identifiers are in the intersection, if the intersection is too small. This could make it easier to guess which combination of identifiers could be in the intersection in order to obtain a particular intersection sum. Deciding on an appropriate "threshold" for intersection size and removing any outliers is therefore important to mitigate this risk.

Once you agree an intersection size, you can set the computation process to automatically terminate the PSI protocol if it is likely to result in a number below this. Additionally, halving the size of the intersection as well as the size of the inputs can provide additional mitigations.

Re-identification can also happen due to **over-analysis**. This involves performing multiple intersection operations which may either reveal or remove particular individuals from the intersection. In other words, this can lead to re-identification through singling out. Rate-limiting can be an effective way of mitigating this risk. This type of technical measure should be defined in any data sharing agreement.

Some PSI implementations may not **ensure input is checked** (ie that parties use real input data as opposed to non-genuine or fictional data). Others may not prevent parties from arbitrarily changing their input after the computation process begins.

This is an issue because it allows a malicious party to reveal data in the intersection they do not actually have mutually in common with the other party. If the data is personal data, there is a risk that the malicious party could access sensitive information, which may have detrimental effects to individuals.

You can mitigate this risk by ensuring that the inputs are checked and validated, and independently audited.

If you and other organisations use PSI to match individuals from your separate databases, you also need to ensure you maintain **referential**

integrity to ensure each record is matched accurately. Linking across datasets becomes more difficult when there may be variation formats in which the data items are held. There may be a risk that some individuals are not included or included by mistake. It is possible to reduce the risk of inaccurate matching by a number of techniques, including tokenisation and hashing. For example, if a common identifier is hashed by both parties, then the hashes will only match if the data is an exact match for both parties.

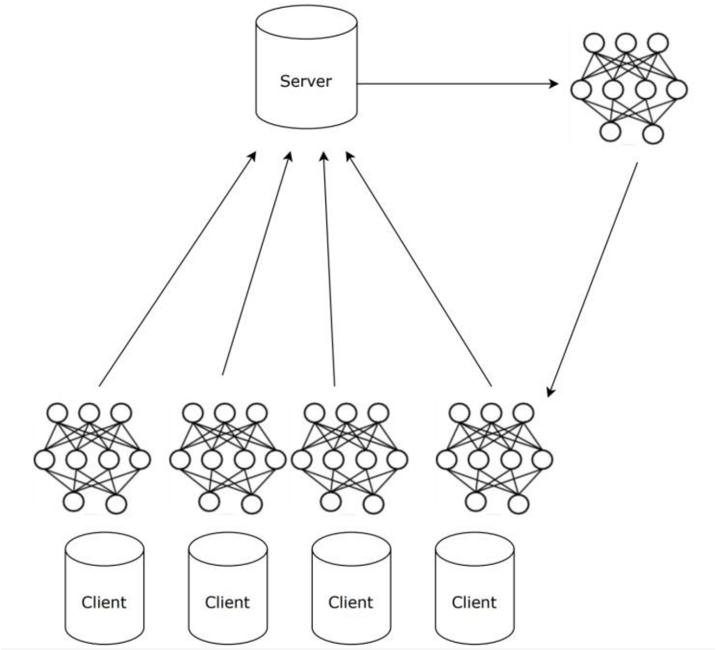
Federated learning

What is federated learning and what does it do?

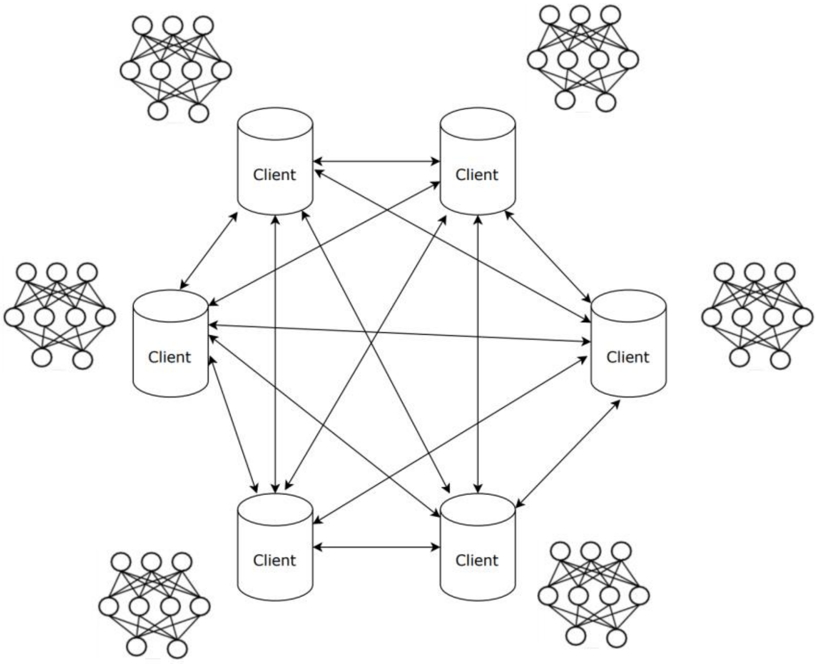
Federated learning (FL) is a technique which allows multiple different parties to train AI models on their own data ('local' models). They then combine some of the patterns that those models have identified (known as "gradients") into a single, more accurate 'global' model, without having to share any training data with each other. Federated learning has similarities with SMPC. For example, the processing involves multiple entities. However, FL is not necessarily a type of SMPC.

There are two approaches to federated learning: centralised design and decentralised design.

In **centralised FL**, a co-ordination server creates a model or algorithm, and duplicate versions of that model are sent out to each distributed data source. The duplicate model trains itself on each local data source and sends back the analysis it generates. That analysis is synthesised with the analysis from other data sources and integrated into the centralised model by the co-ordination server. This process repeats itself to constantly refine and improve the model.



In **decentralised FL**, there is no central co-ordination server involved. Each participating entity communicates with each other, and they can all update the global model directly. The decentralised design has some advantages since processing on one server may bring potential security risks or unfairness and there is no single point of failure.



How does FL assist with data protection compliance?

FL can help with data protection compliance in several ways, including:

- minimising the personal data processed during a model's training phase;
- providing an appropriate level of security (in combination with other PETs); and
- minimising the risk arising from data breaches, as no data is held together in a central location which may be more valuable to an attacker.

FL also can reduce risk in some use cases, but the addition of other PETs further mitigates the risk of attackers extracting or inferring any personal data.

What do we need to know about implementing federated learning?

As FL regularly transfers analysis into the global model, it can incur significant computational cost. This may make it unusable for large-scale processing operations. You should consider whether the training and testing time and memory usage is acceptable for your aims. This will depend on the scale of the processing as processing time, and will increase proportionally as the size of the dataset increases.

You should also consider:

- the choice of encryption algorithm;
- the parameter settings to be specified when reporting the training or testing time and required memory; and
- analysing the FL algorithm to determine its resource usage, so that you can estimate the resource requirements.

What are the risks associated with the use of FL?

When you use FL techniques, local machine learning (ML) models can still contain personal data. For example, the models may preserve features and correlations from the training data samples which could then be extracted or inferred by attackers.

The information shared as part of FL may indirectly expose private data used for local training of the ML model – for example, by:

- model inversion of the model updates;
- observing the patterns that those models have identified (known as 'gradients'); or
- other attacks such as membership inference.

The nature of FL means the training process is exposed to multiple parties. This can increase the risk of leakage via reverse engineering if an attacker can observe model changes over time, observe specific model updates (ie a single client update), or manipulate the model.

To protect the privacy of your training dataset and local model parameters which are exchanged with the co-ordination server, you should combine FL with other PETs. For example, you can use:

- SMPC to protect parameters sent from the clients to ensure that they do not reveal their inputs. For example, the Secure Aggregation protocol (a form of SMPC), has already been integrated into Google's TensorFlow Federated framework;
- homomorphic encryption to encrypt local model parameters from all participants. The coordination server receives an encrypted global model which can only be decrypted if a sufficient number of local models have been aggregated;
- differential privacy, to hide the participation of a user in a training task. If a model depends on the data of any particular individual used to train it, this increases the risk of singling out that individual. You can use differential privacy to add noise and hide the fact that a particular individual's data was used in the training task. This makes it less certain which data points actually relate to a particular individual. This

is more effective if the number of individuals in the dataset is large; and

- secure communications protocols (eg TLS) between clients (in the decentralised model) and between clients and the server (in the centralised model) to prevent man-in-the-middle attacks, eavesdropping and tampering on the connection between the clients and co-ordination server.

Further reading – ICO guidance

See the 'How should we assess security and data minimisation in AI' section of our Guidance on AI and data protection for further information on security risks posed by AI systems and available mitigation techniques.

Trusted execution environments

What is a trusted execution environment and what does it do?

A trusted execution environment (TEE) is a secure area inside a computing device's central processing unit (CPU). It allows code to be run, and data to be accessed, in a way that is isolated from the rest of the system.

TEEs are made up of software and hardware components. TEEs are isolated from the rest of the system, so that the operating system or hypervisor (a process that separates a computer's operating system (OS) and applications from the underlying physical hardware) cannot read the code in the TEE.

TEEs provide security services including:

- integrity of execution;
- secure communication with the applications running in the main operating system;
- trusted storage;
- key management; and
- cryptographic algorithms.

Applications running in the TEE can only directly access their own data.

Using a TEE provides you with a higher level of trust in validity, isolation and access control in the data and code stored in this space, when compared to the main operating system. In turn, this asserts that the applications running inside that space are more trustworthy.

TEEs do not suffer from a loss of utility or additional overhead due to encryption. This is because the actual computation is performed on unencrypted data, and no noise needs to be added to it.

TEEs can be used for many applications, including:

- supporting biometric authentication methods (facial recognition, fingerprint sensor and voice authorisation). A TEE is used to run the matching engine and the associated processing required to authenticate the user. The TEE protects the biometric data, essentially forming a "buffer" against any non-secure apps located in mobile OSes.
- in a cloud context, to ensure that the computation is "securely" outsourced. This means that the provider cannot learn anything about the data involved and assure certain processing occurred, the integrity of systems, that configuration and management policies are applied and that billing accurately reflects the resources consumed;
- enabling secure multi-party computation on untrusted platforms;
- privacy in large scale data analytics and in enabling more privacy-aware machine learning 'as a service'; and
- Internet of Things (IoT) devices.

How does TEEs assist with data protection compliance?

TEEs ensure processing is limited to a specific part of a CPU with no access available to external code. This ensures that the data is protected from disclosure and provides a level of assurance of data integrity, data confidentiality, and code integrity. In turn, this can help you to comply with both the security principle and the requirements of 'data protection by design', depending on your context.

In addition, TEEs can assist with your data governance. For example, they can provide evidence of the steps you take to mitigate risks and enable you to demonstrate that these were appropriate. This can help you to comply with the accountability principle.

TEEs also have wider benefits. For example, they can provide strong manufacturing and supply chain security. This is because TEE implementations embed devices with unique identities via roots of trust (ie a source that can always be trusted within a cryptographic system). These enable key stakeholders in the supply chain to identify whether the device they are interacting with is authentic.

What are the risks associated with the use of TEEs?

Scalability can be an issue for large-scale processing due to a lack of available memory, as only limited data can be processed at any one time. The combined use of TEEs with other PETs (eg machine learning using SMPC), is an still an open research topic.

Processing in shared environments may pose higher risks. These are discussed in more depth in the following section.

You should be aware of published security flaws on TEEs. For example:

- 'side-channel' attacks – an attack based on extra information that can be gathered from way the TEE communicates with other parts of a computer. The most common in the context of TEEs are timing attacks. Attackers can learn information about processes sharing the same CPU, such as memory access patterns of the program that are revealed whenever data is transferred from the TEE to the main memory; and
- timing attacks can leak cryptographic keys or infer information about the underlying operation of the TEE. These attacks measure the access times to a series of specific locations in the computers memory, and use the information to infer whether or not a user has accessed data in related memory locations.

Additionally, TEE hardware only provides particular security properties. The software or system design must properly use these to benefit from the security properties. Insecure or buggy code within a TEE will often not mitigate security risks. If you are writing your own code to use within a TEE, you should ensure that programs are carefully written and audited to ensure that nothing observable about their execution could leak security-impacting sensitive information, for example information regarding memory access patterns to personal data.

Further reading outside this guidance

Commercial TEE solutions are widely available, for example:

- Microsoft Azure confidential computing; and
- Amazon AWS Nitro Enclaves;

For more information on TEEs and confidential computing see:

- Microsoft's "What is confidential computing?", which provides additional information on the benefits and use cases for TEEs;
- The Confidential Computing Consortium's publications "A Technical Analysis of Confidential Computing" (external link, PDF); and: "Confidential Computing: Hardware-Based Trusted Execution for Applications and Data" (external link, PDF)

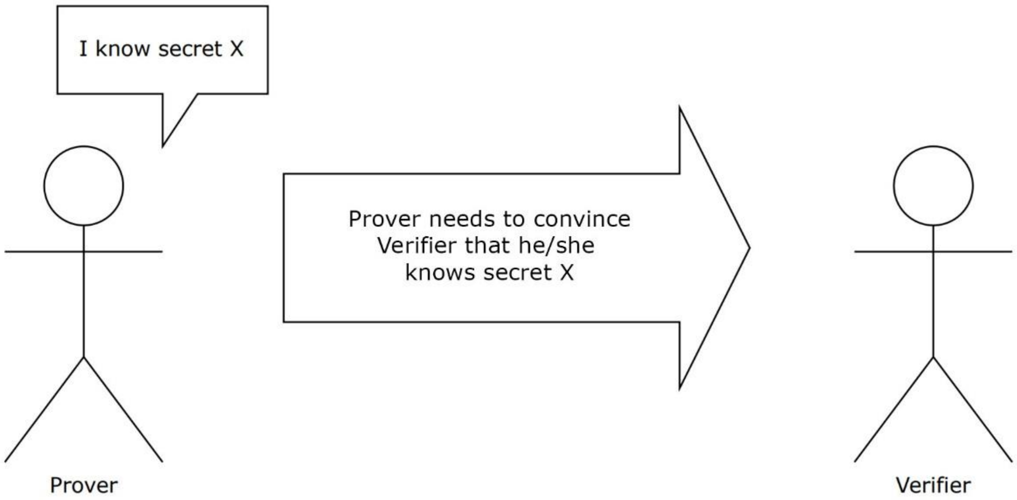
See the IEEE publication "On the Spectre and Meltdown Processor Security Vulnerabilities" (external link, PDF) for further information on particular vulnerabilities in some types of CPUs.

Zero-knowledge proofs

What is a zero-knowledge proof and what does it do?

A zero-knowledge proof (ZKP) refers to any protocol where a prover (usually an individual) is able to prove to another party (verifier) that they are in the possession of a secret (information they know but is unknown to the verifier).

For example, a prover can prove their age without revealing what it actually is. The prover can use a ZKP to prove to the verifier that they know a value X (eg proof they are over 18), without conveying any information to the verifier apart from the fact that the statement is true. The verifier challenges the prover such that the responses from the prover will convince the verifier if the X is true (ie that the prover is over 18).


I know secret X
Prover needs to convince Verifier that he/she knows secret X
Prover
Verifier

Existing applications of ZKPs include:

- confirmation a person is of a certain age (eg legally able to drive), without revealing their birth date;
- proving someone is financially solvent, without revealing any further information regarding their financial status; or
- demonstrating ownership of an asset, without revealing or linking to past transactions; and
- to support biometric authentication methods such as facial recognition, fingerprint sensor and voice authorisation on mobile devices.

ZKPs can be interactive, (ie require the service or verifier to interact with the prover), or non-interactive.

How do ZKPs assist with data protection compliance?

If you use a ZKP service, the information you receive (eg proof that an individual is over a particular age), is likely to still relate to an individual depending on the nature of the query. Therefore, it will still be personal data.

ZKPs can be a used to help you achieve data protection compliance with:

- the **data minimisation principle** as they limit the amount of personal data to what is required; and

- the **security principle** as confidential data such as actual age does not have to be shared with other parties.

How does the use of ZKPs impact the ability to achieve the purpose of the processing?

The algorithms and functions underpinning ZKPs provide a probable certainty as to whether the information is correct or not. This means the secret can be proved with a very high degree of certainty. When applying a ZKP to the design of a processing operation, you should assess whether this uncertainty reaches sufficiently low value for the risk to be accepted in the framework of that specific processing.

What are the risks associated with the use of ZKPs?

Poor implementation of the protocol can cause weaknesses such as code bugs, compromise during deployment, attacks based on extra information that can be gathered from the way the ZKP protocol is implemented and tampering attacks. You should ensure that the technical and organisational measures you use are consistent with the underlying protocol specification, and appropriate measures have been taken to address any security risks.

Further reading outside this guidance

See the current ZKP community reference document (external link, PDF) for more information regarding advanced ZKP techniques, including their advantages, disadvantages and applications.

Differential privacy

What is differential privacy and what does it do?

Differential privacy is a method for measuring how much information the output of a computation reveals about an individual. It is based on the randomised injection of "noise". Noise is a random alteration of data in a dataset so that values such as direct or indirect identifiers of individuals are harder to reveal. An important aspect of differential privacy is the concept of "epsilon" or ε, which determines the level of added noise. Epsilon is also known as the "privacy budget" or "privacy parameter".

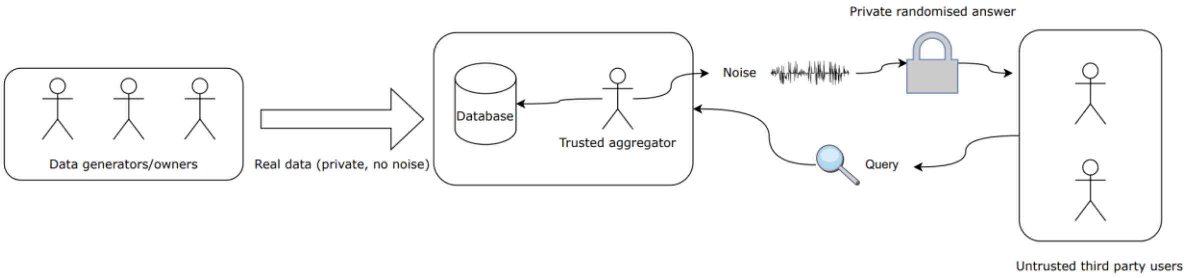
Noise allows for 'plausible deniability' of a particular individual's personal data being in the dataset (ie it is not possible to determine with confidence that information relating to a specific individual is a present in the data).

There are two types of differential privacy available:

- global differential privacy, which adds noise during aggregation; and

- local differential privacy, where each user adds noise to individual records before aggregation.

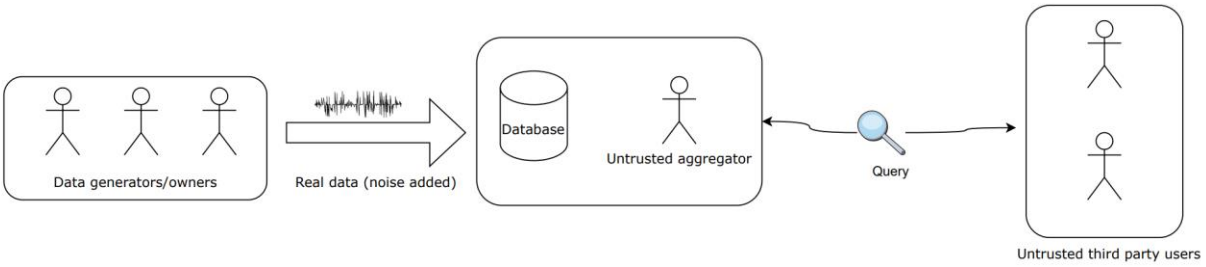
Global (or centralised) differential privacy involves the "aggregator" having access to the real data. Each user sends data to the aggregator without noise. The aggregator then applies a differentially private mechanism by adding noise to the output (eg a response to a database query). The noise is added only once, at the end of the process before sharing it with the third party. The main disadvantage of this approach is the requirement for the central aggregator to access the real data. All the users have to trust the aggregator to act appropriately and protect the privacy of individuals.


Data generators/owners
Real data (private, no noise)
Database
Trusted aggregator
Noise
Private randomised answer
Query
Untrusted third party users

Example

Global differential privacy was used by the US Census Bureau when collecting personal data from individuals for the 2020 US Census to prevent matching between an individual's identity, their data, and a specific data release. The US Census bureau was considered a trusted aggregator, in other words – the data was handled in line with the expectations of the participating individuals with robust controls in place.

Local differential privacy has the individual users applying the mechanism before they send anything to the aggregator. Noise is added to the individual (input) data points. The aggregator receives "noisy" data – this addresses the trust risk of global differential privacy as the real data is not shared with the aggregator. Since each user must add noise to their own data, the total noise is much larger than global differential privacy. Local differential privacy requires many more users to get useful results. The key difference between the two models is that the global model leads to more accurate results with the same level of privacy protection, as less noise is added.



Example

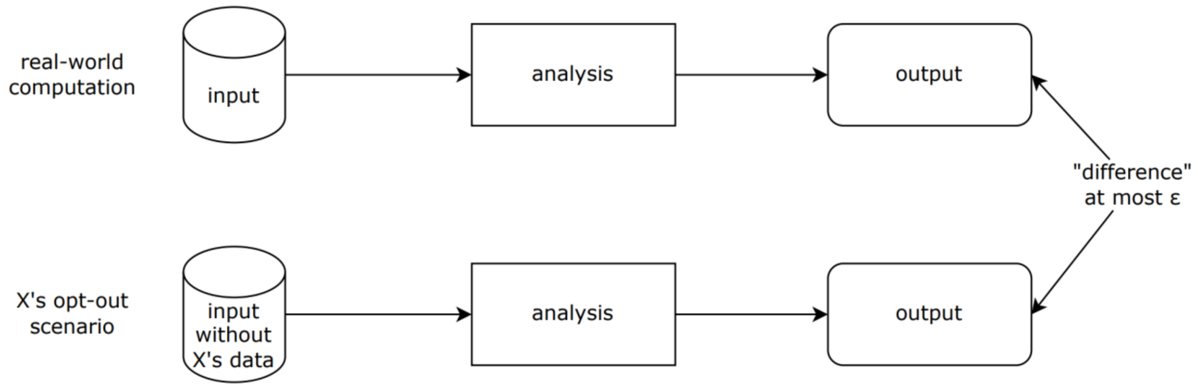
A smartphone provider wants to know the average number of minutes a individual uses their device within a given month without revealing the exact amount of time.

Instead of asking the exact amount of time, the individuals device adds any random value as noise (eg in the range of -50 to +50 to the actual number of minutes they use their device and give the smartphone provider just the resultant sum of it. That is if an individual had a monthly usage of 300 minutes by adding a random number of -50 to it, (300 + (-50)), they provide just the noised result, which is 250 minutes in this case.

The diagram below shows the difference between a real-world computation (where a specific individual’s data is included in the processing) and an opt-out scenario (where the individual’s data is not included). Epsilon (ε) is the maximum distance between a query on a database (real-world computation) and the same query on a database with a single entry added or removed.

Small values of ε provide very similar outputs when given similar inputs, and therefore provide higher levels of privacy as more noise is added. Therefore, it is more difficult to distinguish whether an individual’s data is present in the database. Large values of ε allow less similarity in the outputs, as less noise is added and therefore it is easier to distinguish between different records in the database.

Practical applications using local DP often use higher values of epsilon than global DP due to the higher amount of noise required. If anonymous information is required as output, epsilon can be set such that that the relative difference in the result of two the scenarios is so small that it is unlikely anyone could single out or infer, with confidence, anything about a specific individual in the input.



How does differential privacy assist with data protection compliance?

Differential privacy can be used to anonymise data for other purposes, providing an appropriate level of noise is added. Anonymous aggregates can be generated from personal data or it can be used to query a database to provide anonymised statistics.

Both models of differential privacy are able to provide anonymous information as output, providing a sufficient level of noise is added to the data. Local differential privacy adds noise to the individual (input) data points to provide strong privacy protection of sensitive attributes. As the noise is added to each individual contribution, this will result in less accurate, lower utility data than global differential privacy.

Any original data retained by the aggregator in the global model or the individual parties in the local model would be personal data in their hands. This also applies to any additional information that may reidentify. However, in either model, the output may not be personal data in the hands of another party.

What do we need to know about implementing differential privacy?

Using differential privacy can result in poor utility due to noise addition. It is challenging to generate differentially private outputs that provide strong protection and good utility for different purposes. Differential privacy can however be useful in the context of statistical analysis and broad trends, rather than for detecting anomalies or detailed patterns within data.

What are the risks associated with the use of differential privacy?

Data treated by a differentially private algorithm does not necessarily result in anonymous information. If you do not configure differential privacy properly, there is a risk of personal data leakage from a series of different queries. For example, if the privacy budget is poorly configured, an attacker can accumulate knowledge from multiple queries, to re-identify an individual. Tuning differential privacy must be done on a case-by-case basis. You should consider obtaining expert knowledge for best results. Your privacy budget assessment should consider:

- the overall sensitivity of the data, which can be obtained by measuring the specific weight of a record on the result of the query;
- the nature of the attributes;
- the type of query made;
- the size of the population in the database;
- the number of queries that are likely to be made over the data lifecycle; and
- whether you set the privacy budget per user or globally (or both).

When setting an appropriate privacy budget to enforce limits on the number of queries made, you should consider the risk of unintended disclosure of sensitive information in any query to be performed on the data. You should also consider contractual controls to mitigate malicious parties making similar queries and then sharing them between each other, to increase the total amount of information each one holds.

You should consider whether it is likely that:

- an attacker could accumulate knowledge on an individual from the inputs or intermediate results,
- an attacker could accumulate knowledge from multiple queries; and
- malicious parties could collude to pool the results the results of their queries, and increase their collective knowledge of the dataset.

If there is still some risk, you should adjust the privacy budget and re-assess the risk until the risks are reduced to a remote level.

Further reading outside this guidance

For more information on the concept of differential privacy, see Harvard University's publication "Differential Privacy: A Primer for a Non-technical Audience" (external link, PDF). Harvard University also has a number of open source toolkits and resources available, such as its OpenDP Project.

For more information on differential privacy and the epsilon value, see Purdue University's publication "How Much Is Enough? Choosing ε for Differential Privacy" (external link, PDF).

The Government Statistical Service has an introduction on differential privacy for statistical agencies, accessible on request from the GSS website.

For an analysis of differential privacy in the context of singling out, linkability and inferences see section 3.1.3 of the Article 29 Working Party's Opinion 05/2014 on anonymisation techniques (external link, PDF).

OpenMined's blog on "Local vs global differential privacy" provides a useful description of the two types along with some code examples.

Synthetic data

What is synthetic data and what does it do?

Synthetic data is ‘artificial’ data generated by data synthesis algorithms, which replicate patterns and the statistical properties of real data (which may be personal data). It is generated from real data using a model trained to reproduce the characteristics and structure of that data. This means that when you analyse the synthetic data, the analysis should produce very similar results to analysis carried out on the original real data.

It can be a useful tool for training AI models in environments where access to large datasets is not possible.

There are two main types of synthetic data:

- “partially” synthetic data, which synthesises only some variables of the original data; and
- “fully” synthetic data, which synthesises all variables.

How does synthetic data assist with data protection compliance?

Synthetic data requires real data to generate it, which may involve the processing of personal data. However, data synthesis may allow large datasets to be generated from small datasets. This can help you comply with the data minimisation principle as it reduces or eliminates the processing of personal data.

You should consider synthetic data for generating non-personal data in situations where you do not need to, or cannot, share personal data. If you are generating synthetic derived from personal data, any inherent biases in the data will be carried through. You should:

- ensure that you can detect and correct bias in the generation of synthetic data, and ensure that the synthetic data is representative; and
- consider whether you are using synthetic data to make decisions that have consequences (ie legal or health consequences) for individuals.

What do we need to know about implementing synthetic data?

Generating synthetic data is an active research area and, at present, it may not be a viable solution for many data processing scenarios. Synthetic data is being considered as a type of statistical disclosure control method for open data release.

What are the risks associated with the use of synthetic data?

The degree to which synthetic data is an accurate proxy for the original data depends on the utility of the method and model. The more that the synthetic data mimics real data, the greater the utility it has. At the same time, it may be more likely to reveal individuals' personal data.

Assessing re-identification risk involved with synthetic data is an ongoing area of development. You should consider whether the synthetic data you generate is personal data. You should focus on the extent to which individuals are identified or identifiable in the synthetic data, and what information about them would be revealed if identification is successful.

Some synthetic data generation methods have been shown to be vulnerable to model inversion attacks. Using differential privacy with synthetic data can protect any outlier records from linkage attacks with other data. However, it may reduce the utility of the data and introduce a degree of unpredictability regarding the preservation of data characteristics.

Further reading outside this guidance

The links below provide useful reading on synthetic data techniques and their associated benefits and risks.

The ONS has proposed a high-level scale to evaluate the synthetic data based on how closely they resemble the original data, their purpose and disclosure risk.

For an evaluation of how synthetic data delivers utility, see Manchester University's publication "A Study of the Impact of Synthetic Data Generation Techniques on Data Utility using the 1991 UK Samples of Anonymised Records" (external link, PDF).

Reference table

The table below provides some example use-case applications for PETs discussed in this guidance, together with information about whether standards are available and known limitations. Your purposes may require a combination of techniques to provide the required protection at all the various stages of the data processing lifecycle. This is not an exhaustive list.

PET	Applications	Standards	Known weaknesses
Secure multiparty computation	Cryptographic key protection within a single organisation: Secure multiparty computation allows an organisation to split its secret	IEEE 2842-2021 – IEEE Recommended Practice for Secure Multi-Party Computation. ITU-T X.1770 Technical	Requires significant computational resources. Communication costs can be high.

	keys across multiple hosts. Pseudonymisation within a single organisation. Privacy-preserving analytics (eg training neural networks, evaluating decision trees).	guidelines for secure multi-party computation. The IETF is currently developing a draft multi-party privacy preserving measurement (PPM) protocol standard.	
Homomorphic encryption	Leverage cloud computing and storage services securely, as data held off-site is encrypted but can be processed. Secure machine learning as a service: data can be processed without giving processor access to encrypted data. Secure collaborative computation.	Community standard for homomorphic encryption.	Scalability and computation speed can be an issue. Fully homomorphic encryption is unsuitable for real-time data analysis.
Differential privacy	Performing statistical analysis with privacy guarantees (ie that presence or absence of an individual in the data will not affect the final output of the algorithm significantly). Useful for allowing databases to be queried without	No standard available.	No consensus over the optimal trade-off of privacy and utility. You will need to tailor level of noise added will depend on the circumstances of the processing. Requires expertise to add the right amount of noise.

	releasing information about individuals in the database.		
Zero-knowledge proofs	Proving claims about personal data (eg nationality, solvency, age, transactions).	ZKProof Community Reference (2019) ISO/IEC 9798-5 Information technology — Security techniques — Entity authentication — Part 5: Mechanisms using zero-knowledge techniques.	Weaknesses in Zero-knowledge proof implementations can be caused by poor implementation of the protocol. Interactive protocols may be more vulnerability to side channel or timing attacks as they require the prover to send multiple messages.
Generating synthetic data	Use cases which require access to large amounts of data (eg model training, research and development).	No standard available.	Synthetic data may not represent outliers present in the original personal data. You will need to assess whether the personal data on which the synthetic data was trained can be reconstructed. Further additional measures (eg differential privacy) may be required to protect against singling out.
Federated learning	Applications where the aggregation of data into a centralised data	IEEE 3652.1-2020 – IEEE Guide for Architectural Framework and	The devices or entities contributing data will need to have

	server is not feasible or desirable (eg building models of user behaviour from device data, without the user data leaving the devices or carrying out research on data from multiple entities without the data being transmitted between them).	Application of Federated Machine Learning.	compatible formats and standards to allow the analysis to be carried out locally. This also requires sufficient local computing power. Federated learning requires frequent communication between the participating entities, which requires sufficient bandwidth. Requires other PETs to provide privacy to individuals data, this may affect performance and scalability.
Trusted execution environments	Protection against software attacks. Used for processing particularly confidential data within an existing system or device.	IETF Trusted Execution Environment Provisioning (TEEP) Architecture (draft) 2021. Other standardisation initiatives are being developed by, GlobalPlatform, the Trusted Computing Group and Confidential Computing Consortium	Side channel attacks possible with some earlier implementation. These attacks monitors certain properties of the system, such as the time required to perform an operation, to learn sensitive information.