Privacy-enhancing technologies (PETs)

June 2023

ico.
Information Commissioner's Office

About this guidance 3
How can PETs help with data protection compliance? 4
What PETs are there? 17
Introduction 18
Differential privacy 19
Synthetic data 27
Homomorphic encryption (HE) 30
Zero-knowledge proofs 34
Trusted execution environments 36
Secure multiparty computation (SMPC) 40
Private set intersection (PSI) 45
Federated learning 49
Reference table 54
Case studies 59

About this guidance

■ Latest updates

> **19 June 2023** - we have created new PETs guidance, which is aimed at data protection officers and others who are using large personal data sets in finance, healthcare, research, and central and local government

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.

The first part of the guidance is aimed at DPOs (data protection officers) and those with specific data protection responsibilities in larger organisations. It focuses on how PETs can help you achieve compliance with data protection law.

The second part is intended for a more technical audience, and for DPOs who want to understand more detail about the types of PETs that are currently available. It gives a brief introduction to eight types of PETs and explains their risks and benefits.

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 comply with the data minimisation principle. You can do this by only processing the information you need for your purposes.
- PETs can also provide an appropriate level of security for your processing.
- You can use PETs to give people access to datasets that would otherwise be too sensitive to share, while ensuring people’s information is protected.
- Most PETs involve processing personal information. Your processing still needs to be lawful, fair and transparent.
- You **should** identify the risks to people by performing a case-by-case assessment of the processing (eg through a data protection impact assessment (DPIA)). This will determine if PETs are appropriate to mitigate those risks.
- You **should** not regard all PETs as a way to anonymise personal information. Not all PETs result in effective anonymisation, and you can achieve anonymisation without using them.

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 information use (this covers the legal definition of personal data in the UK GDPR);
- maximising information security; or
- empowering people.

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:

> ‘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 information 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 making the personal information unintelligible to anyone not authorised to access it.

Further Reading

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

External link

What are the benefits of PETs?

PETs can help reduce the risk to people, while enabling you to further analyse the personal information. The ability to share, link and analyse personal information in this way can give you 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 people whose data is in the dataset. Appropriate PETs can make it possible to give access to datasets that would otherwise be too sensitive to share.

What are the risks of using PETs?

You **should not** regard PETs 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 your processing;
- be clear about your purpose;

- understand and document how PETs can help you to comply with the data protection principles; and
- understand and address the issues PETs may pose to complying with the data protection principles (eg issues with accuracy and accountability).

Lack of maturity

Some PETs may not be sufficiently mature in terms of their scalability, availability of standards and their robustness to attacks. We list some factors you **should** consider to assess the maturity of PETs in the section How do we determine the maturity of a PET?.

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 that provides an appropriate level of support.

Mistakes in implementation

With insufficient expertise comes the risk of inappropriate implementation (eg poor key management when using technologies underpinned by encryption). This might mean that the PETs are not actually protecting people in the way you intended. Therefore, there are unaddressed risks to those people's rights and freedoms. You **should** also monitor attacks and vulnerabilities regularly, to ensure that you can put appropriate mitigation measures in place.

Lack of appropriate organisational measures

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 that you are using a trusted third party (ie an organisation 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 introduces some PETs that you can use to help you comply with your data protection by design obligations. They help you minimise the personal information you collect and integrate safeguards into the processing. Many aspects of PETs are also relevant for the public. However, this guidance focuses on PETs that organisations can use.

PETs that provide **input privacy** can significantly reduce the number of parties with access to personal information you are processing. Input privacy means that the party carrying out the processing cannot:

- access the personal information you are processing;
- access intermediate values or statistical results during processing (unless the value has been specifically selected for sharing); or
- derive inputs by using techniques such as side-channel attacks that use observable changes during processing (eg query timings or power usage) to obtain the input.

These types of PETs can also help you comply with the security, purpose limitation, storage limitation and data minimisation principles of the UK GDPR.

PETs that provide **output privacy** reduce the risk that people can obtain or infer personal information from the result of a processing activity. Output privacy measures and controls reduce the risk that personal information can be obtained or inferred from the result of a processing activity. This is regardless of whether the computation itself provides input privacy. Using a PET that provides output privacy is useful if you plan to:

- make anonymous statistics publicly available; or
- share the results of an analysis with a large group of recipients.

These types of PETs also help you comply with the storage limitation and data minimisation principles of the UK GDPR.

The table below gives further information on how the PETs covered in this guidance provide input or output privacy. If you are able to fulfil your purposes in this way, you **should** consider combining PETs to ensure that the processing satisfies both input and output privacy.

PET	Does the PET provide input privacy?	Does the PET provide output privacy?
Homomorphic encryption (HE)	In some cases If processing depends on the encrypted input of two or more parties, HE is not guaranteed to protect these inputs from the owner of the secret key.	No The output may contain personal information. It can be combined with output privacy approaches such as differential privacy.
Secure multiparty computation (SMPC)	Yes	No The output may contain personal information. It can be combined with output privacy approaches such as differential privacy.
Zero-knowledge proofs (ZKPs)	Yes	No It may be possible to learn something about a person, depending on the nature of the query.

Federated learning (FL)	No FL can be combined with other input privacy techniques, such as SMPC and HE.	No FL does not prevent personal information from being shared through the output. It can be combined with output privacy approaches such as differential privacy.
Synthetic data (SD)	No Using synthetic data does not address security risks when managing input data processed as part of the synthetic data generation process.	No Synthetic data does not inherently provide output privacy, but it can be combined with output privacy approaches such as differential privacy.
Trusted Execution Environments (TEEs)	Yes	No TEEs can also deploy output privacy techniques, providing the code executed within them includes specific computations that provide those features.
Differential Privacy (DP)	No DP does not address the security risks when processing input personal information which is highly identifiable (ie before noise is applied that makes it less identifiable). Global DP can be combined with input privacy PETs, such as SMPC, to protect the input personal information between the input sources and the party adding the noise.	Yes

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 people that the information you are processing is about. These can help you to fulfil the principle of data minimisation;
- focus on hiding and shielding information. These can help you achieve the requirements of the security principle; and
- split datasets. 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 information that reduces or removes people's identifiability

These aim to weaken or break the connection between someone in the original personal information and the derived information. Examples include:

- differential privacy; and
- synthetic data.

These PETs can effectively reduce risk to people. However, the resulting information may be less useful compared with the original information. This is because using these techniques can reduce how close the randomised answers to queries are compared to the real ones (ie those without "noise" applied). Noise randomly alters information in a dataset so that values such as people's direct or indirect identifiers are harder to reveal. These results may not be suitable if you need the actual personal information about people or datasets with higher utility (ie which contain more useful information that you can extract).

Example

A hospital needs to ensure that patients with underlying health conditions receive appropriate treatment. To achieve this purpose, it must process their health information.

This means the hospital cannot use a PET that reduces the identifiability of the patients (eg synthetic data), as it cannot then make the right decisions about their treatment.

Separately, the hospital also shares information with researchers studying regional trends of COVID-19 cases. In this case, the hospital generates synthetic data for the researchers, possibly in combination with differential privacy to achieve effective anonymisation.

PETs that focus on hiding, or shielding, data

These aim to protect people's privacy while not affecting the utility and accuracy of the information. For example:

- homomorphic encryption - this allows computation to be performed on encrypted data without revealing the plaintext;
- zero-knowledge proofs - these 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); and
- trusted execution environments (TEEs) - these protect the information from external operating systems and applications.

PETs that split datasets

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

This group of technologies define how you collect, distribute, store, query, and secure personal information, and how each component of the system communicates with each other. They may split information for computation or storage or provide dedicated hardware to prevent the operating system or other application from accessing the personal information. This reduces the risk of information from different datasets being linked.

Examples include:

- 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 **could** achieve anonymisation without using them.

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

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

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

Further reading

See the sections of our draft anonymisation guidance on identifiability and pseudonymisation for more information.

When should we consider using PETs?

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 your project, particularly for data-intensive projects that involve potentially risky uses of personal information. You **must** consider how you will comply with each of the data protection principles if you choose to use a PET.

If you are doing a data protection impact assessment (DPIA), and you have identified risks to people, then you **should** consider at this point whether PETs can mitigate those risks.

Which types of processing can benefit from using PETs?

PETs can help you reduce the risks to rights and freedoms that your processing may pose. For example, they can be suitable technical and organisational measures for the types of processing likely to result in a high risk to people. In particular, for processing that involves large-scale collection and analysis of personal information (eg artificial intelligence applications, Internet of Things and cloud computing services).

The table below is a non-exhaustive list of processing activities that may pose risks to people's rights and freedoms, and how PETs can aid your compliance by mitigating these risks. If the processing is likely to result in a high risk to people, you **must** complete a DPIA. However, you do not need to consult us, if your DPIA identified a high risk to people but you are able to apply PETs to reduce the risk, so it is no longer high (residual risk).

Processing activity	Possible risks to people	PETs which may aid compliance
Processing involving artificial intelligence, machine learning, and deep learning applications	Possible risks to people involved in the training dataset include model inversion, model inference and attribute inference attacks. These can reveal people's identities, or may result in learning sensitive information about them.	PETs can help you assess and mitigate these risks. For example: homomorphic encryption ensures that only parties with the decryption key can access the information. This protects the information that is being processed (eg to train the AI model); SMPC can protect information sent to global model; differential privacy adds random noise during training to ensure the final model does not memorise information unique to a particular person's personal information; federated learning can minimise the amount of centrally held personal information and reduce the transfer of personal information between parties; and synthetic data can be used at the training stage to reduce the amount of personal information used to train artificial

		intelligence. Processing activities involving AI may require you to complete a DPIA. For more information, see our DPIA guidance.
Processing involving data matching that means combining, comparing or matching personal information obtained from multiple sources Eg sharing financial transactions to prevent fraud and money laundering	Possible risks to people include collecting more information than is required for the purposes and security threats during transfer of personal information.	PETs can help you assess and mitigate these risks. For example: SMPC and PSI can minimise the information shared and protect it during computation Processing activities involving matching information or combining datasets from different sources mean that you **must** complete a DPIA. For more information, see our DPIA guidance.
Processing involving IoT applications Eg smart technologies (including wearables)	Possible risks to people include: Collecting more information than required for the purposes Security threats due to data breaches Identifying people or learning about their activities through collection of sensitive attributes	PETs can help you assess and mitigate these risks. For example: Federated learning can be used to train machine learning models on a large number of decentralised IoT devices (eg wearable devices, autonomous vehicles). Depending on the circumstances of the processing, you can also use other PETs, such as SMPC, HE and DP, when you process personal information collected from IoT devices. Processing activities involving IoT may require you to complete a DPIA (eg

		large-scale processing of health information from wearables). For more information, see our DPIA guidance.
Processing involving data sharing between organisations, particularly data sharing likely to result in a high risk to people	Possible risks to people include sharing more information than the party you are sharing it with needs for their purposes and security threats (eg data breaches)	PETs can help you assess and mitigate these risks. For example: SMPC, PSI and FL (when used with other PETs) can minimise the information transferred between parties. HE can enhance security by preventing parties accessing the input information without affecting utility You **should** carry out a DPIA for data sharing operations. You **must** do this if sharing it is likely to result in a high risk to people. See our data sharing code for further guidance.
Processing involving cloud computing applications	Possible risks to individuals include increased risk of security threats from attackers due to performing computations in untrusted environments	HE, TEEs and SMPC can be used for cloud computing processing tasks to provide enhanced security.
Processing involving anonymisation of personal information	Re-identification of people in information that has not been effectively anonymised	DP can prevent people from being identified in published information or limit the amount of personal information released from queries. You **must** ensure that the risk of re-identification is sufficiently remote. Read our draft guidance on anonymisation for further information.

How should we decide whether or not to use PETs?

If you are considering using PETs to address privacy risks, you **should** do a DPIA to understand how your use of the PET will impact your data processing. Your assessment **must** consider:

- the nature, scope, context and purposes of your processing;
- the risks your processing poses to people's rights and freedoms;
- whether you are using a PET to address a recognised data protection risk, and how it does so; and
- the state-of-the-art and costs of implementation of any PETs.

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

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

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

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

You **must** consider the **state-of-the-art** to understand whether the PET 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.

You **must** consider the **cost** of a technique as a factor in deciding 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, read the section on data protection by design and security in our 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. TRLs are based on actual usage, integration, and tests with existing systems and use cases.

Some models (eg ENISA's PETs maturity assessment) combine TRLs with various quality measures including:

- scalability;
- quantified assumptions about how trustworthy the entities involved in the processing are;
- security measures in place; 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 information 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 does not 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 when designing and implementing data protection measures. You **must** 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).

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.

Further Reading

Relevant provisions in the legislation - See Article 25 and Recital 78 of the UK GDPR

External link

Further reading – ICO guidance

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

Further reading

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)

For more information on PETs:

- 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.
- The Royal Society’s 2023 report “From privacy to partnership” (external link, PDF) also provides further information about PETs use cases and relevant standards.
- The CDEI’s PETs adoption guide provides a question-based flowchart to aid decision-makers in thinking through which PETs may be useful in their projects.

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 information between different parties).
- Differential privacy generates anonymous statistics. This is usually done by randomising the computation process that adds noise to the output.
- Synthetic data provides realistic datasets in environments where access to large real datasets is not possible.
- Homomorphic encryption provides strong security and confidentiality by enabling computations on encrypted data without first decrypting it.
- Zero-knowledge proofs (ZKP) provide data minimisation by enabling someone to prove private information about themselves without revealing what it actually is.
- Trusted execution environments enhance security by enabling processing by a secure part of a computer processor that is isolated from the main operating system and other applications.
- Secure multiparty computation (SMPC) provides data minimisation and security by allowing different parties to jointly perform processing on their combined information, without any party needing to share all of its information with each of the other parties.
- Federated learning trains machine learning models in distributed settings while minimising the amount of personal information shared with each party. Using federated learning alone may not achieve appropriate protection of personal information. It may also require specific expertise to design mitigations (eg by combining with other PETs at different stages of your processing).

In detail

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

Introduction

There are many PETs that you **could** consider as part of your data protection compliance. This section outlines some of these, and summarises 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 **could** procure specialist expertise beyond this guidance.

We plan to update this guidance as technology develops to reflect changes in the state-of-the-art.

Differential privacy

- What is differential privacy and what does it do?
- How does differential privacy assist with data protection compliance?
- What do we need to know about implementing differential privacy?
- What are the risks associated with using differential privacy?

What is differential privacy and what does it do?

Differential privacy is a property of a dataset or database, providing a formal mathematical guarantee about people's indistinguishability. It is based on the randomised injection of noise.

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".

Epsilon represents the worst-case amount of information inferable from the result by any third party about someone, including whether or not they participated in the input.

Noise allows for "plausible deniability" of a particular person's personal information being in the dataset. This means that it is not possible to determine with confidence that information relating to a specific person is present in the data.

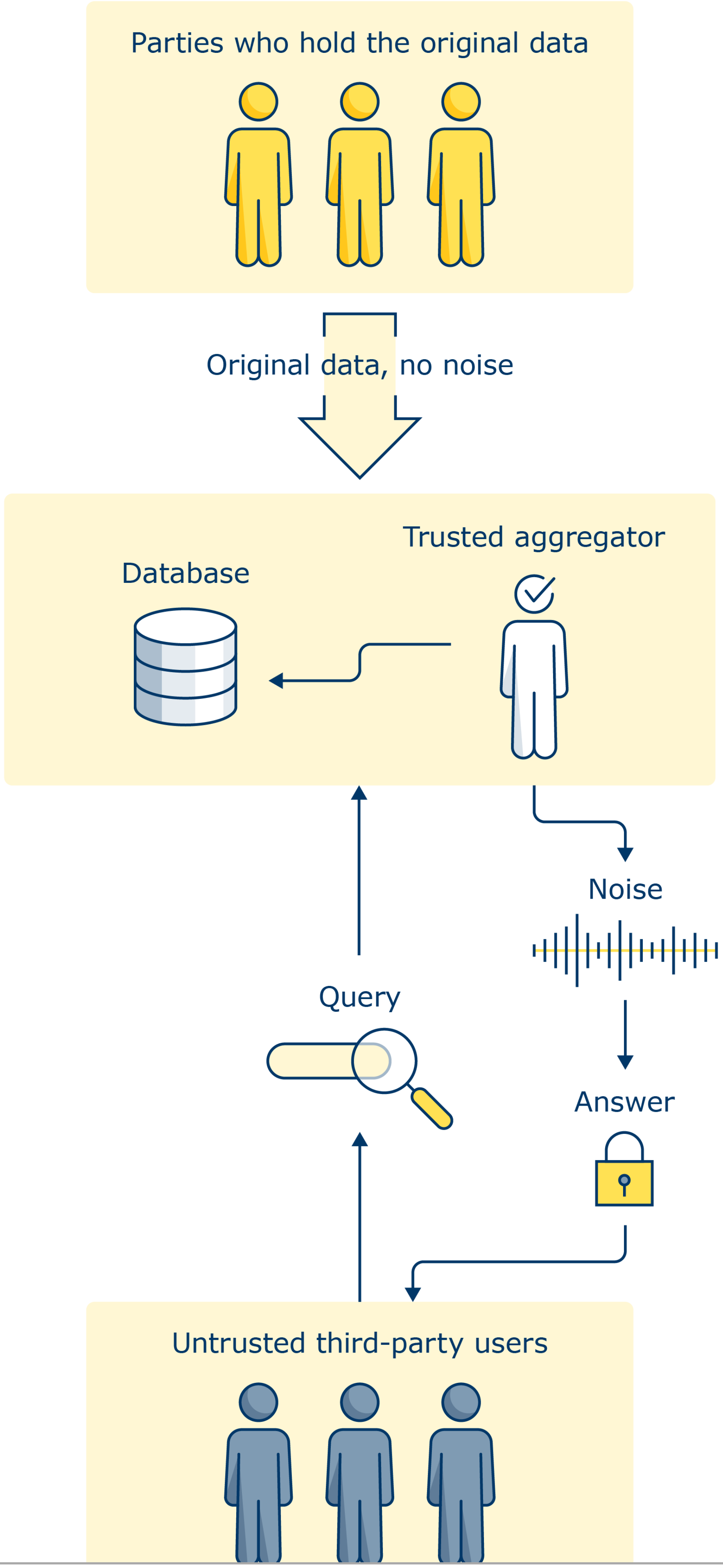
There are two ways for the privacy budget to be enforced:

- interactive, query-based DP – this is where noise is added to each query response and querying is terminated once the privacy budget is met (ie where the information obtained from queries reaches a level where personal information may be revealed); and
- non-interactive DP – this is where the level of identifiable information is a property of the information itself, which is set for a given privacy budget. This approach can be useful for publishing anonymous statistics to the world at large.

There are two types of differential privacy available:

- global differential privacy - this adds noise during aggregation; and
- local differential privacy:- this is where each user adds noise to individual records before aggregation.

Global (or centralised) differential privacy involves an "aggregator" having access to the real data. Each user of the system that differential privacy is being used in sends information 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 or the noise is embedded in the entire dataset). The noise is added during computation of the final result before it is shared with the third party. The main disadvantage of this approach is that the central aggregator has to access the real data. All the users have to trust the aggregator to act appropriately and protect people's privacy.


Parties who hold the original data
Original data, no noise
Trusted aggregator
Database
Noise
Query
Answer
Untrusted third-party users

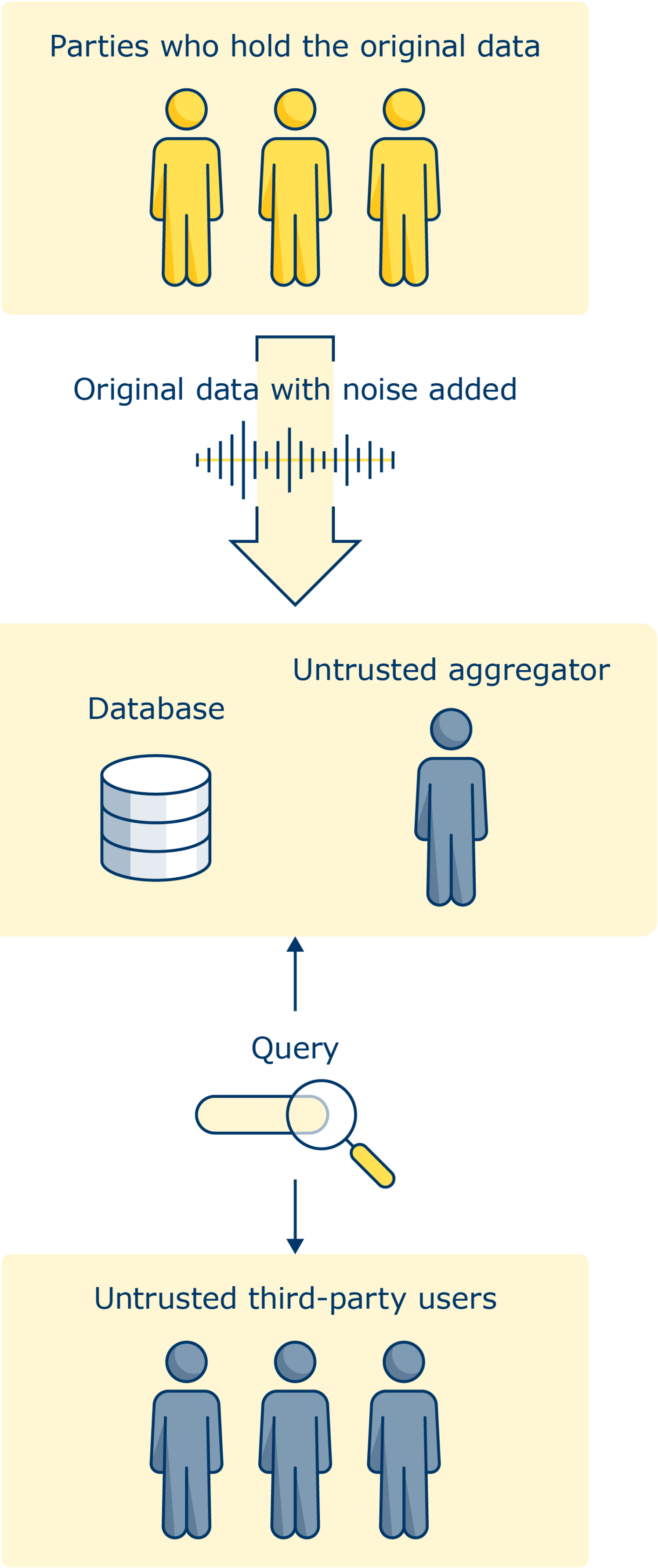
Example

Global differential privacy was used by the US Census Bureau when collecting personal information from people for the 2020 US Census. This was done to prevent matching between an person's identity, their information, and a specific data release. The US Census bureau was considered a trusted aggregator. In other words, they handled the information in line with the expectations of the participants and had robust controls in place.

Local differential privacy has the user of the system (or a trusted third party on a person's behalf) 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 is responsible for adding noise to their own information, the total noise is much larger than global differential privacy. Local differential privacy requires many more users to get useful results.

There are several key differences between the two models:

- the global model leads to more accurate results with the same level of privacy protection, as less noise is added;
- the global model provides deniability of people's non-participation (ie you cannot prove whether a person's information was in the dataset);
- the local model provides deniability of a person's record content, but not record association (the ability to link with other personal information which relates to that person);
- the local model is not necessarily suitable for producing anonymous information (eg statistics). However, you can use it to mitigate sensitive attribute inference (ie attributes that should not be linkable to a person, such as gender or race). For example, if something new could be learned about a known person.


Parties who hold the original data
Original data with noise added
Untrusted aggregator
Database
Query
Untrusted third-party users

Example

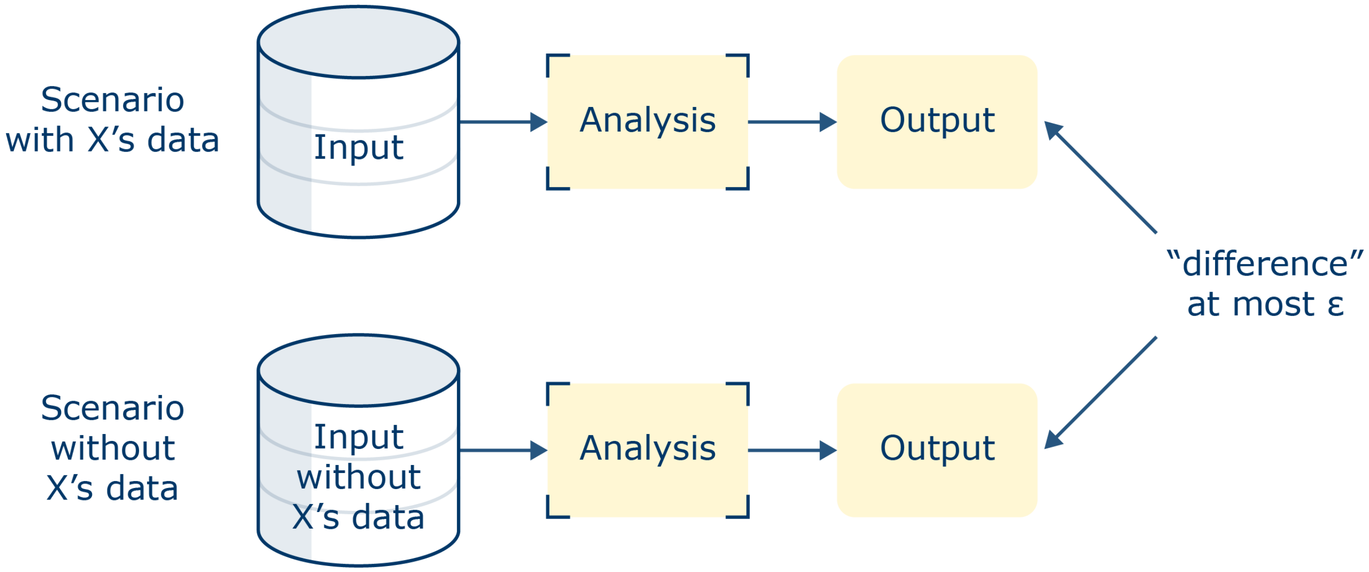
A smartphone OS (Operating System) developer wants to know the average number of minutes a person uses their device in a particular month, without revealing the exact amount of time.

If the smartphone OS developer wants to know this, then they should build local differential privacy into their product. This would work by default so that the person's sensitive attributes are protected and the person using the device does not have to do anything.

Instead of asking the exact amount of time, the person’s device adds any random value as noise. For example, a random number that with high probability lands in the range of -50 to +50 to the actual number of minutes they use their device and give the smartphone OS developer just the resultant sum of it. For example, if someone 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, local DP can be applied by a user so their attributes (eg device usage times) is noised and protected but they could still be identified from other identifiers (eg their IP address).

The diagram below shows the difference between a real-world computation (where a specific person’s information is included in the processing) and an optout scenario (where the person’s information 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 a person’s information 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 the local model often use higher values of epsilon than the global model, due to the higher amount of noise required. If you require anonymous information as output, you can set epsilon so that the relative difference in the result of the two scenarios is so small that it is unlikely anyone could identify a specific person in the input.

How does differential privacy assist with data protection compliance?

You can use differential privacy to:

- anonymise personal information, as long as you add an appropriate level of noise; and
- query a database to provide anonymous information (eg statistics).

Both models of differential privacy are able to provide anonymous information as output, as long as a sufficient level of noise is added to the data. The local model 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 and useful information than the global model.

Any original information retained by the aggregator in the global model or the individual parties in the local model is personal information in their hands. This also applies to any separately held additional information that may re-identify. For example, device IP address, unique device ID of people stored by the aggregator in the global model or the recipient of the information in the local model. However, in either model, the output may not be personal information in the hands of another party, depending on whether or not the risk of re-identification is sufficiently remote in their hands.

What do we need to know about implementing differential privacy?

Using differential privacy may not be beneficial due to noise addition. It is challenging to generate differentially private outputs that provide strong protection and good utility for different purposes. However, differential privacy can be useful for statistical analysis and broad trends, rather than for detecting anomalies or detailed patterns within data.

What are the risks associated with using differential privacy?

Differential privacy does not necessarily result in anonymous information. If you do not configure differential privacy properly, there is a risk of personal information leakage from a series of different queries. For example, if the privacy budget is poorly configured, an attacker can accumulate knowledge from multiple queries and re-identify someone.

Each subsequent release of a dataset constructed using the same underlying people further accumulates epsilon values. You should deduct the accumulated epsilon values from your overall privacy budget. This helps you ensure that no further queries can be answered once your privacy budget is exhausted. You **should** take this into account when setting your privacy budget and 'per-query' epsilon. For example, excessive queries early in the analysis can lead either to noisier outputs later, or no outputs at all, in order to remain within the privacy budget.

For example, a release mechanism requires 10 queries to produce the dataset to be released, using an ε of

0.1 in each query. It is generated once a year to provide yearly updates. Each year a privacy cost of one is being incurred, so after five years the privacy cost is five.

You **should** tune differential privacy on a case-by-case basis. You **could** consider obtaining expert knowledge for best results. Your privacy budget assessment **should** consider:

- the overall sensitivity of the information, which you can determine 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 personal information in any query you perform on the information. You **should** also consider contractual controls to mitigate malicious parties increasing the total amount of information they hold by making similar queries and then sharing them between each other.

You **should** consider whether it is likely that:

- an attacker could accumulate knowledge on a person from the inputs or intermediate results;
- an attacker could accumulate knowledge from multiple queries; and
- malicious parties could collude to pool 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 they are reduced to a remote level.

Further reading – ICO guidance

For more information on assessing identifiability, see our draft anonymisation guidance “How do we ensure anonymisation is effective?”

Further reading

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.

For more information on using differential privacy for machine learning applications, see Microsoft’s blog on privacy-preserving machine learning and the Google blog Federated Learning with Formal Differential Privacy Guarantees.

Synthetic data

- What is synthetic data and what does it do?
- How does synthetic data assist with data protection compliance?
- What are the risks associated with using synthetic data?
- Is synthetic data anonymous?

What is synthetic data and what does it do?

Synthetic data is ‘artificial’ data generated by data synthesis algorithms. It replicates patterns and the statistical properties of real data (which may be personal information). It is generated from real data using a model trained to reproduce its characteristics and structure. This means that your analysis of the synthetic data 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 you are not able to access large datasets.

There are two main types of synthetic data:

- “partially” synthetic data - this synthesises only some variables of the original data. For example, replacing location and admission time with synthetic data in an A&E admission dataset, but maintaining the real causes for admission); and
- “fully” synthetic data - this synthesises all variables.

How does synthetic data assist with data protection compliance?

Synthetic data requires real data to generate it, which may involve processing personal information. You can use data synthesis to generate large datasets from small datasets. In cases where you can create synthetic data instead of collecting more personal information, this can help you comply with the data minimisation principle as it reduces your processing of personal information.

You **should** consider synthetic data for generating non-personal data in situations where you do not need to, or cannot, share personal information.

What are the risks associated with using synthetic data?

You **should** assess whether the synthetic data you use is an accurate proxy for the original data. The more that the synthetic data mimics real data, the greater the utility it has, but it is also more likely to reveal someone’s personal information.

If you are generating synthetic data from personal information, any inherent biases in the information will be carried through. For example, if the underlying dataset is not representative of your population of interest (eg your customers), neither will the synthetic data which you generate from it. 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 for people (ie legal or health consequences). If so, you **must** assess and mitigate any bias in the information.

To mitigate these risks, you **should** use diverse and representative training data when generating synthetic data, and continuously monitor and address any biases that may arise.

Is synthetic data anonymous?

This depends on whether the personal information on which you model the synthetic data can be inferred from the synthetic data itself. Assessing re-identification risks involved with synthetic data is an ongoing area of development. You **should** focus on the extent to which people 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, membership inference attacks and attribute disclosure risk. These can increase the risk of inferring a person's identity. You **could** protect any records containing outliers from these types of linkage attacks with other information through:

- suppression of outliers (data points with some uniquely identifying features); or
- differential privacy with synthetic data.

However, it may reduce the utility of the information and introduce a degree of unpredictability in the characteristics of the information.

You **should** consider the purposes and context of the processing when you decide what risk mitigation measures are appropriate balanced against what you need to fulfil your purposes. For example, if you intend to use synthetic data in a secure setting (eg a trusted research environment), then the risk of attack may be reduced. You **could** consider less stringent measures (eg adding less noise than if you were releasing the information to the world at large when using differential privacy). In some cases, you will not be able to achieve an acceptable balance of utility and protection. For example, if you need to capture outliers as part of your purposes for fraud detection. If you do not have the required expertise, you **should** consult an external expert in setting an appropriate privacy budget when using differential privacy. This will ensure the best trade-off of protection and utility for your purposes.

Further reading

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).

For more information on how synthetic data can be combined with differential privacy, see "Synthetic

Text Generation with Differential Privacy: A Simple and Practical Recipe” (external link, PDF)

“Generating and Evaluating Synthetic UK Primary Care Data: Preserving Data Utility & Patient Privacy” discusses the key requirements of synthetic data to validate and benchmark machine learning algorithms as well as reveal any biases in real-world data used for algorithm development (external link, PDF)

Homomorphic encryption (HE)

- What is homomorphic encryption and what does it do?
- How does HE assist with data protection compliance?
- What do we need to know about implementing HE?
- What are the risks associated with using homomorphic encryption?

What is homomorphic encryption and what does it do?

Homomorphic encryption allows you to perform computations on encrypted information 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).

You **should** choose the HE scheme based on the nature, scale and the purpose of your processing and the type of computation you require to fulfil your purposes. You **should** also 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 th operation, the more resource and time may be required.
SHE	SHE permits fewer additions and multiplications on encrypted information. The ar functions it can support.
PHE	PHE provides good performance and protection, but it only supports addition or m 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 information 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 to 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?

HE can help you to ensure:

- **security and confidentiality** - it can minimise the risk from data breaches, if they occur. This is because personal information remains encrypted at rest, in transit and during computation. For example, because HE renders the information unintelligible to an attacker, the risks people are reduced; 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, as long as you ensure the inputs are correct prior to encryption taking place. This is because HE does not require you to alter the information in other ways (eg adding "noise" like differential privacy). This means 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 an organisation 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 increase communications cost. It may therefore not be appropriate if you process large volumes of personal information.

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 costs are likely to become less significant over time, and FHE may therefore become more viable for large-scale processing operations. At present, FHE is suitable for some types of computation (eg addition operations), but it is still not feasible for many types of processing due to the computational cost.

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

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:

- 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.

What are the risks associated with using homomorphic encryption?

HE has similar risks to encryption more generally. You **should** 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 private key can be used to decrypt the outputs. You **must** therefore use appropriate technical and organisational measures to keep it secure. You **must** also have processes in place to generate a new key immediately if the original is compromised.

The security of most HE schemes is based on hard mathematical problems that 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 some additional risks that HE may introduce. For example:

- FHE does not protect against:
  - differencing attacks (attacks using background knowledge about someone to learn sensitive information about them by analysing multiple statistics which includes their information); or
  - dataset recovery through multiple queries.

  Therefore, you **should** consider additional measures, such as rate limiting (ie, controlling the rate at which queries may be submitted) and therefore the number of queries permitted.
- While HE protects inputs and data during computation, it does not protect the output once it has been decrypted. So, if the output is personal information, you should put in place other encryption measures to mitigate the risks of this information being compromised.

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

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.

HEBench creates a benchmarking framework for homomorphic encryption to better understand performance (this can be useful tool as an indicator in considering the maturity of HE for your processing).

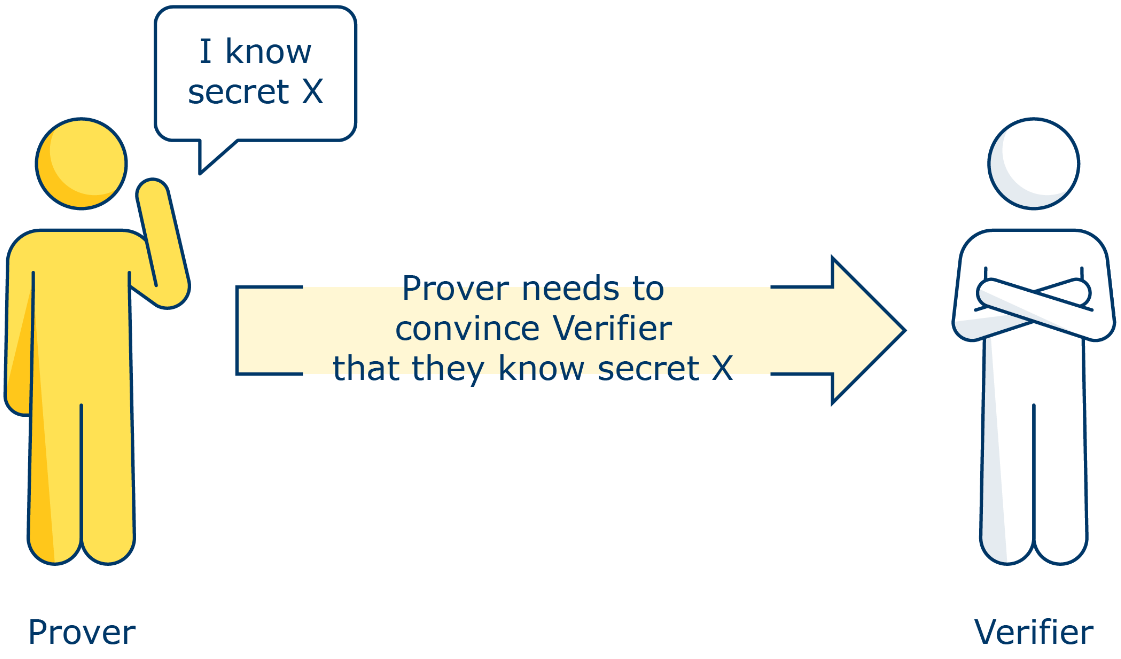
Zero-knowledge proofs

- What is a zero-knowledge proof and what does it do?
- How do ZKPs assist with data protection compliance?
- How does using ZKPs impact our ability to achieve the purpose of the processing?
- What are the risks associated with using ZKPs?

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

A zero-knowledge proof (ZKP) refers to any protocol where a prover (usually a person) 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.



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
- supporting 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 someone is over a particular age), is likely to still relate to a person depending on the nature of the query. Therefore, it is still personal information. You **could** use ZKPs to help you achieve data protection compliance with:

- the **data minimisation principle** as they limit the amount of personal information to what is required; and
- the **security principle** as confidential information, such as actual age, does not have to be shared with other parties.

How does using ZKPs impact our 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. 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 using ZKPs?

Poor implementation of the protocol can cause weaknesses, for example:

- 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 **must** ensure that the technical and organisational measures you use are consistent with the underlying protocol specification, and you have taken appropriate measures to address any security risks.

Further reading

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

Trusted execution environments

- What is a trusted execution environment and what does it do?
- How do TEEs assist with data protection compliance?
- What are the risks associated with using TEEs?

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 runs code and accesses information in a way that is isolated from the rest of the system.

As defined by the Confidential Computing Consortium, a TEE is an environment that provides a level of assurance that unauthorised entities cannot:

- view information while it is in use within the TEE ("data confidentiality;
- add, remove or alter information while it is in use in the TEE ("data integrity"); and
- add, remove, or alter code executing in the TEE ("code integrity").

TEEs are made up of software and hardware components. TEEs are isolated from the rest of the system. This means that the operating system or hypervisor (a process that separates a computer’s operating system 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 access information outside the TEE, but applications outside the TEE cannot access information in the TEE.

Using a TEE gives you a higher level of trust in validity, isolation and access control in the information and code stored in this space, when compared to the main operating system. Therefore, this makes the applications running inside that space 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 information, and you do not need to add any noise to it.

You **could** use TEEs 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;

- providing the ability for relying parties (eg clients) to attest attributes of a TEE remotely and cryptographically (including initial code deployed in the TEE) before sharing secrets, such as encryption keys with the TEE. This ensures that TEE providers are meeting desired security and privacy properties;
- in a cloud context, to ensure that the computation is securely outsourced. This means that the provider cannot learn anything about this information, while being able to prove that certain processing occurred and that the systems were not compromised;
- 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 do 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 which provides input privacy. This ensures that the information is protected from disclosure and provides a level of assurance of data integrity, data confidentiality, and code integrity. Therefore, 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 using TEEs?

Scalability can be an issue for large-scale processing due to a lack of available memory. However, this is becoming less of an issue with the latest TEE implementations. The combined use of TEEs with other PETs (eg machine learning using SMPC), is still an open research topic.

Processing in shared environments may pose higher risks. You **should** be aware of published security flaws on TEEs. Vulnerabilities are often unique to specific TEE types, so you **should** conduct your own research of threats and mitigations in each case. Possible attacks include, but are not limited to:

- '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 computer's memory and use this to infer whether or not a user has accessed information in related memory locations.

TEE implementation can introduce several risks, for example:

- Hardware TEEs only provide certain security properties. You **must not** assume that TEEs provide a complete security solution for the processing;
- other risks arise from poor architectural, design and implementation choices. For example, isolated Virtual Machines (VMs) can increase the size of the attack surface; and
- code within the TEE can also pose a risk, both in terms of the programs and their configuration. Insecure or buggy code cannot mitigate security risks.

In order to mitigate these risks, you **must:**

- ensure that you put in place the security properties that are available and configure them appropriately to mitigate potential security risks;
- use well-reviewed and analysed code implementations, such as low-level firmware and secure operating systems; and
- avoid writing your own software stacks for TEEs (but if you have to write your own code, make sure that the programs are written carefully and audited to ensure nothing observable about their execution could leak information, eg memory access patterns to personal information).

TEEs protect inputs and data during the computation but do not provide output privacy unless the code running within the TEE includes specific computations that provide that feature.

TEEs rely on the public key infrastructure (PKI) of the manufacturer for verification (eg making attestations that code is running on genuine compatible hardware before input is decrypted). This means that the manufacturer's cryptographic PKI must also be trusted (in addition to their implementation of TEE hardware).

Further reading

Commercial TEE solutions are widely available, for example:

- Microsoft Azure confidential computing; and
- Amazon AWS Nitro Enclaveshttps://aws.amazon.com/ec2/nitro/nitro-enclaves/.

These are isolated and constrained virtual machines that run alongside an Amazon EC2 instance that is responsible for starting the enclave and communicating with it. By design, the enclave’s network traffic must go through the parent EC2 instance.

For more information on TEEs and confidential computing, read:

- 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) and CYBERNETICA - An Overview of Vulnerabilities and Mitigations of Intel SGX Applications (external link, PDF) for further information on particular vulnerabilities in some types of CPUs.

Secure multiparty computation (SMPC)

- What is secure-multiparty computation (SMPC) and what does it do?
- How does SMPC assist with data protection compliance?
- What do we need to know about implementing SMPC?
- What are the risks associated with using SMPC?

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

SMPC is a protocol (a set of rules for transmitting information between computers) that allows at least two different parties to jointly process their combined information, without any party needing to share all of its information 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”. This refers to the division of a secret and its distribution among each of the parties. This means that each participating party’s information is split into fragments to be shared with other parties. Secret sharing is not the only way to perform SMPC, but it the most common approach used in practice.

Each party’s information cannot be revealed to the others unless some proportion of fragments of it from each of the parties are combined. As this would involve compromising the information 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 information 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 (to be 3 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 £45,000 partials)

Average expenditure (total **£15,000** divided by number of parties)

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

You should note that this is a simplified example. In reality, additional calculations on the secret shares are required to ensure the value of the shares cannot be leaked.

How does SMPC assist with data protection compliance?

SMPC is a way to ensure that the amount of information you share is limited to what is necessary for your purposes, without affecting the utility or accuracy of the data. 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. This is because the shared information is not stored together, and also when it is being processed by separate parts of the same organisation.

If your purposes require you to provide personal information to the SMPC computation, you **must** assess whether the information you receive from the output is personal information. 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 SMPC operations that can be challenging in practice, including:

- using SMPC to replace missing information with substituted values;
- eliminating duplicate copies of repeating information; 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 -this outsources the computations to a trusted provider. It 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 - this 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.

Above a certain "threshold” (number of secret shares), it may be possible for the input data to be reconstructed (eg by one or more of the parties, or an attacker), if the secret shares are combined together. Therefore, you **should** determine what the appropriate threshold your use of SMPC involves.

The threshold for reconstruction influences the risk of collusion and reidentification. The required threshold depends on the threat model used. A threat model that requires a greater proportion of the parties to be honest poses a higher risk than one that requires a lower proportion. For example, if all but one parties must be honest, then compromise of two parties would undermine the security of the protocol. Furthermore, some attack models may allow more than one party to act 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 **could** need to obtain further expertise in secret sharing when assessing the context and purpose for your use of SMPC. For most use cases, an organisation would typically not develop SMPC directly, but rather use a protocol designed by an expert cryptographer.

What are the risks associated with using 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'.

You **should** distinguish between the security and trust assumptions associated with secret sharing and the trust assumptions associated with the analysis. For example, a dishonest party can faithfully follow the protocol, and act as an honest participant in the secret sharing protocol. But also can use knowledge of its own data, or use false data, to learn something about the other party's (or parties') information through similar techniques as those used in differencing attacks.

The security model appropriate for your circumstances depends on the level of inherent risk of a malicious party learning something about a person. Or corrupting their inputs so that it may have a detrimental effect on someone.

Generally, if you are using stronger threat models you will use more computational resources as further checks are required to ensure that the parties are not acting in bad faith. You **should** 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 **must** carry out accuracy checks to ensure the inputs have not been manipulated by a corrupted party. You **could** 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 information during the computation but does not protect the output. Where the output is personal information, you **should** implement appropriate encryption measures for information at rest and in transit to mitigate the risk of personal information 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 information.

Further reading

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?
- How does PSI assist with data protection compliance?
- What are the risks associated with using PSI?

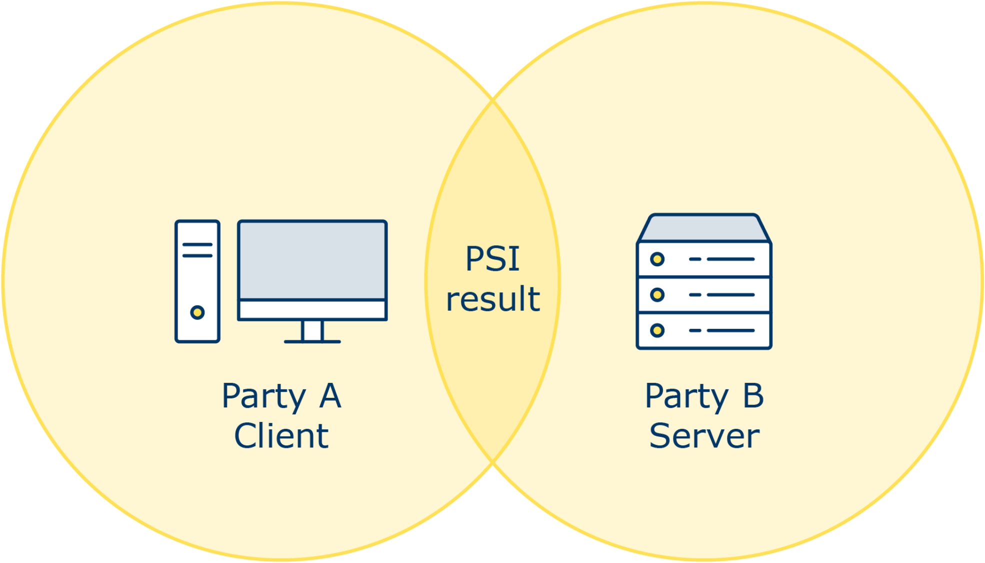
What is private set intersection (PSI) and what does it do?

PSI is a specific type of SMPC that 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. You **could** also use it 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 information that the client can query to determine if it holds any matching information with the server.

PSI can work in two ways:

- the parties 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 information 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
- with cryptographic expertise, PSI protocols can be 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 information about people’s health.

Organisation A processes information about people’s vaccination status, while Organisation B processes information about people’s specific health conditions.

Organisation B needs to determine the percentage of people 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 information they process, while still achieving their purposes.

A third party provides the PSI protocol. The computation involves processing the personal information that both organisations hold. However, the output of that computation is the number of people 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 information needed to achieve the purpose. Therefore, it enhances people’s privacy.

What are the risks associated with using PSI?

PSI introduces some risks that you **must** mitigate. These include:

- risks of re-identification from inappropriate intersection size or over-analysis;
- the introduction of a third party to the processing when using PSI may increase data protection risks to people if it is compromised, and
- the potential for one or more of the parties to use fictional data in an attempt to reveal information

about people.

You **should** consider whether you can incorporate differential privacy into the PSI protocol to prevent the risk of singling out of the output – providing the output is sufficiently useful to fulfil your purposes. This approach is generally less error prone than trying to manually constrain protocol parameters to rule out specific attacks.

You **should** choose an appropriate intersection size. This is because a low **intersection size** may allow the party computing the intersection to single out people within that intersection in cases where a person'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. You **should** therefore decide on an appropriate "threshold" for intersection size and remove any outliers to mitigate this risk.

Once you agree an intersection size, you **could** 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, **could** provide additional mitigations.

Re-identification can also happen due to **over-analysis**. This involves performing multiple intersection operations that may either reveal or remove particular people 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. You **should** define this type of technical measure 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 information). 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 information in the intersection they do not actually have mutually in common with the other party. If it is personal information, there is a risk that the malicious party could access sensitive information that may have detrimental effects to people.

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

If you and other organisations use PSI to match people from your separate databases, you **must** also maintain **referential integrity** to ensure each record is matched accurately. Linking across datasets becomes more difficult if the information is held in a variety of formats. There may be a risk that some people 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 information is an exact match for both parties.

Your choice of using a third party will depend on whether it is likely to reduce the risk in comparison to using a two-party protocol. When performing a DPIA, you **should** document the risks and choose the most suitable option for mitigating the risks for your circumstances. If you are using a third party for the computation or storage of sets, you **must** ensure appropriate technical and organisational measures are in place to mitigate the risk of any personal information being compromised.

Federated learning

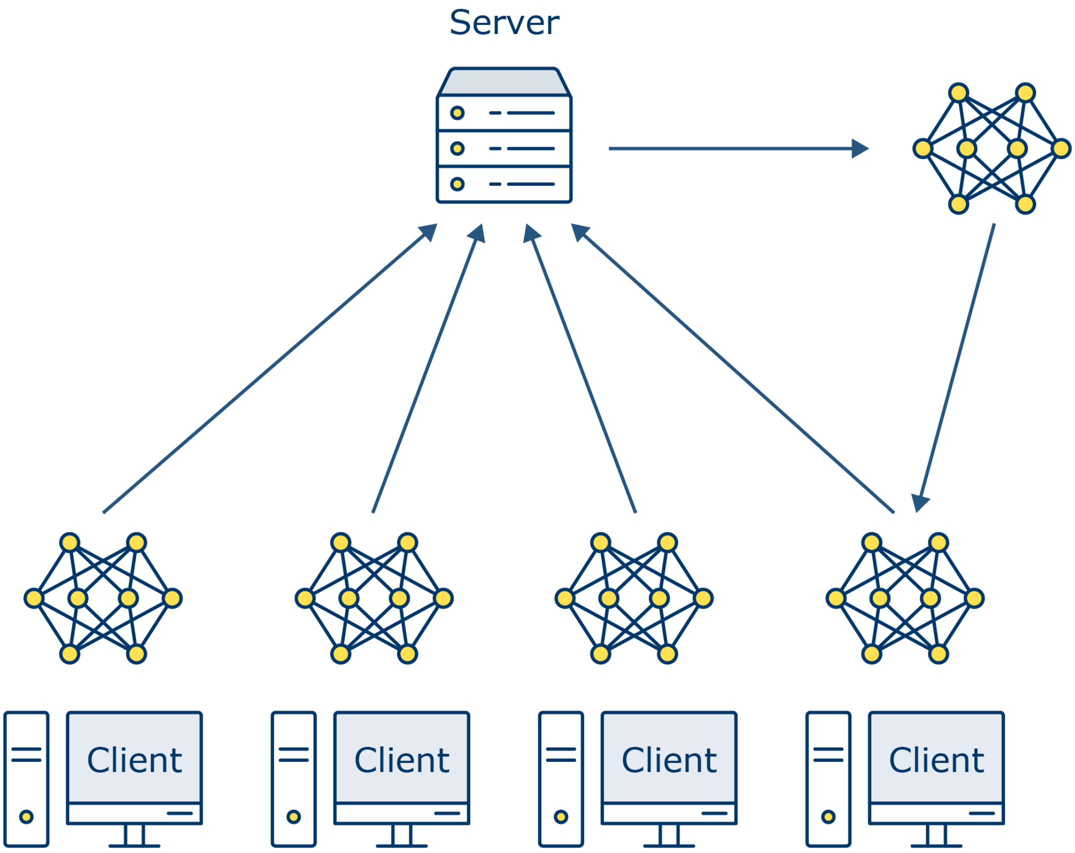
- What is federated learning and what does it do?
- How does FL assist with data protection compliance?
- What do we need to know about implementing federated learning?
- What are the risks associated with using FL?

What is federated learning and what does it do?

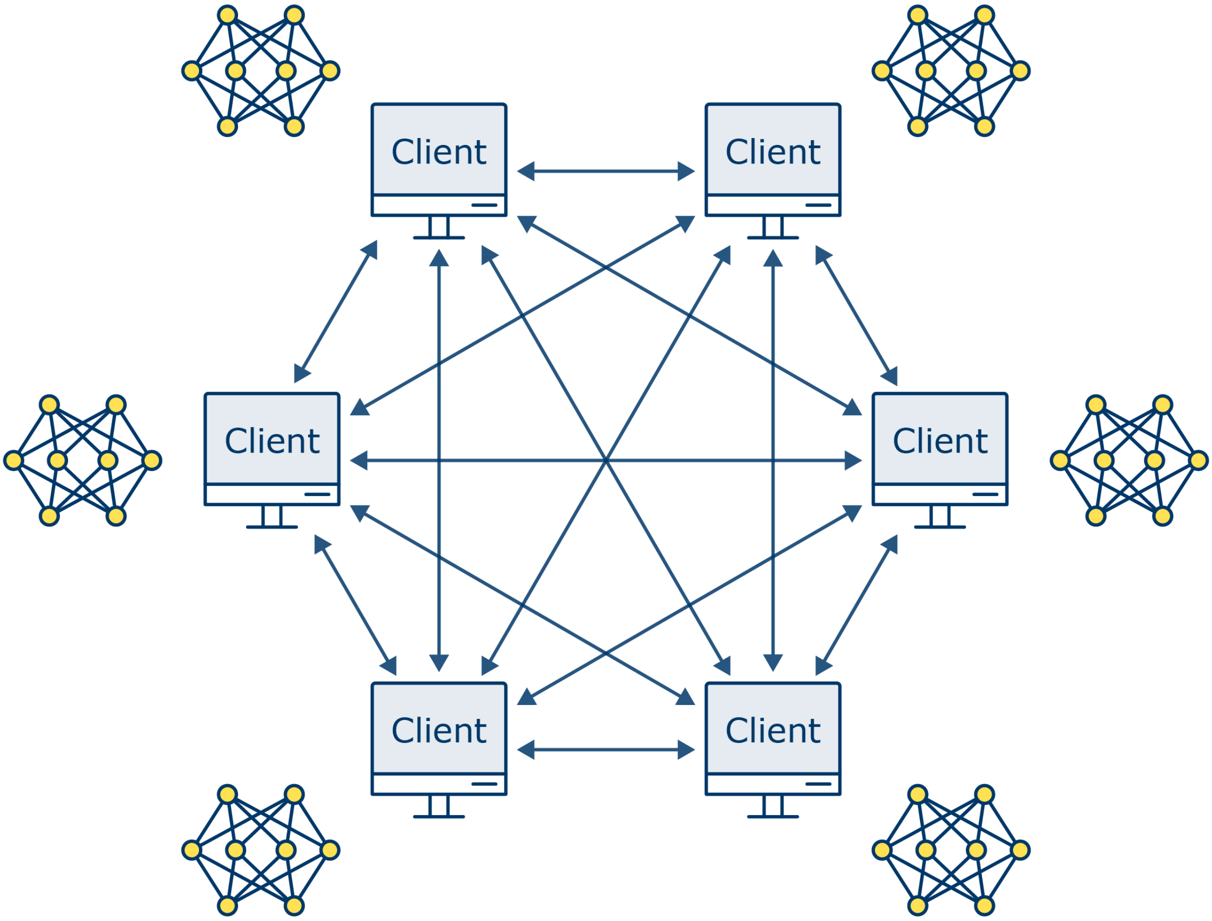
Federated learning (FL) is a technique that allows multiple different parties to train AI models on their own information (‘local’ models). They then combine some of the patterns that those models have identified (known as “gradients”) into a single, more accurate ‘global’ model, without having to share any training information 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 coordination server. This process repeats itself to constantly refine and improve the model. A trusted third-party, is a requirement when using centralised federated learning, which is not the case when using SMPC.



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 because 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 information 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 that 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 information.

What do we need to know about implementing federated learning?

While decentralised FL can be cheaper than training a centralised model, it still incurs 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 and will increase proportionally as the size of the dataset increases.

You **should** also consider:

- the choice of encryption algorithm to encrypt local model parameters;
- the local model 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 using FL?

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

The information shared as part of FL may indirectly expose private information 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 by 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 that are exchanged with the co-ordination server, you **should** combine FL with other PETs. For example, you **could** 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 that 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 information of any particular person used to train it, this increases the risk of singling them out. You **could** use differential privacy to add noise and hide the fact that you used a particular person's information in the training task. This makes it less certain which data points actually relate to a particular person. This is more effective if the number of people 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.

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	How does it support data protection compliance?
Secure multiparty computation	Cryptographic key protection within a single organisation: Secure multiparty computation allows an organisation to split its secret keys across multiple hosts. Pseudonymisation within a single organisation. Privacy-preserving analytics (eg training neural networks, evaluating decision trees). Secure collaborative computation (eg processing that requires multiple parties to share information between them for joint analysis of the combined data). Can be used to speed up the validation process for AI models.	IEEE 2842-2021 – IEEE Recommended practice for secure multi-party computation. ITU-T X.1770 Technical guidelines for secure multi-party computation. The IETF is currently developing a draft multi-party privacy-preserving measurement (PPM) protocol standard.	Requires significant computational resources. Communication costs can be high.	Data minimisation Security

Homomorphic encryption	Leverage cloud computing and storage services securely, as information held off-site is encrypted but can be processed. Secure machine learning as a service: information can be processed without giving processor access to encrypted information. Secure collaborative computation.	Community standard for homomorphic encryption. ISO/IEC 18033-6:2019 - IT Security techniques — Encryption algorithms — Part 6: Homomorphic encryption Also in development: ISO/IEC WD 18033-8 - Information security — Encryption algorithms — Part 8: Fully Homomorphic Encryption	Scalability and computation speed can be an issue. Fully homomorphic encryption is unsuitable for real-time information analysis.	Accuracy Security
Differential privacy	Performing statistical analysis with privacy guarantees (ie that presence or absence of an person in the information does not affect the final output of the algorithm significantly). Useful for allowing databases to be queried without releasing information about people in the database.	No standard available.	No consensus over the optimal trade-off of privacy and utility. The level of noise added will depend on the circumstances of the processing.	Reduce identifiability of personal information or render it as anonymous information Purpose limitation
Zero-knowledge proofs	Proving claims about personal information	ZKProof Community	Weaknesses in Zero-knowledge	Data minimisation

	(eg nationality, solvency, age, transactions).	Reference (2019) ISO/IEC 9798-5:2009 - Information technology — Security techniques — Entity authentication — Part 5: Mechanisms using zero-knowledge techniques	proof implementations can be caused by poor implementation of the protocol. Interactive protocols may be more vulnerable to side channel or timing attacks as they require the prover to send multiple messages.	Security
Generating synthetic data	Use cases that require access to large amounts of information (eg model training, research and development).	No standard available, although work on this topic is underway in IEEE SA - Synthetic data	Synthetic data may not represent outliers present in the original personal information. Requires assessment on whether the personal information on which the synthetic data was trained can be reconstructed. Further additional measures (eg differential privacy) may be required to protect against singling out.	Data minimisation Purpose limitation
Federated learning	Applications where the aggregation of	IEEE 3652.1-2020 – IEEE Guide for	The devices or entities	Data minimisation

	information into a centralised data server is not feasible or desirable (eg building models of user behaviour from device data, without it leaving the devices or carrying out research on information from multiple entities without it being transmitted between them).	architectural framework and application of federated machine learning.	contributing information need to have 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. This requires sufficient bandwidth. Requires other PETs to provide privacy to people's information. This may affect performance and scalability.	Security (when combined with other PETs)
Trusted execution environments	Protection against software attacks. Used for processing, particularly confidential information within an existing system or device.	IEEE SA - IEEE 2830-2021, IEEE SA - IEEE 2952-2023 and NISTIR 8320, Hardware-Enabled Security: Cloud and Edge Computing Use Cases | CSRC In development:	May be vulnerable to side channel attacks. These attacks monitor certain properties of the system, such as the time required to perform an operation, to learn sensitive	Accountability Security

	Other standardisation initiatives are being developed by Trusted Execution Environment Provisioning (teep) (ietf.org) GlobalPlatform, the Trusted Computing Group and Confidential Computing Consortium	information.

Case studies

The following case studies are examples of how you could use or combine the techniques discussed in our guidance about privacy enhancing technologies (PETs).

If you are considering implementing PETs, these case studies will show you how you can do this in practice and what objectives you can achieve with these techniques. We do not require or encourage you to follow these techniques, they are simply examples of good implementation.

We have developed these case studies with organisations who use anonymisation and pseudonymisation techniques in innovative ways. These organisations collaborated with us voluntarily, and we did not pay or otherwise compensate them for doing so.

We will add further case studies as we develop them with organisations. If you are using PETs and you think other organisations would benefit from learning about how you use them, please contact our Technology team to discuss developing a case study.

Case study 1: homomorphic encryption for data sharing

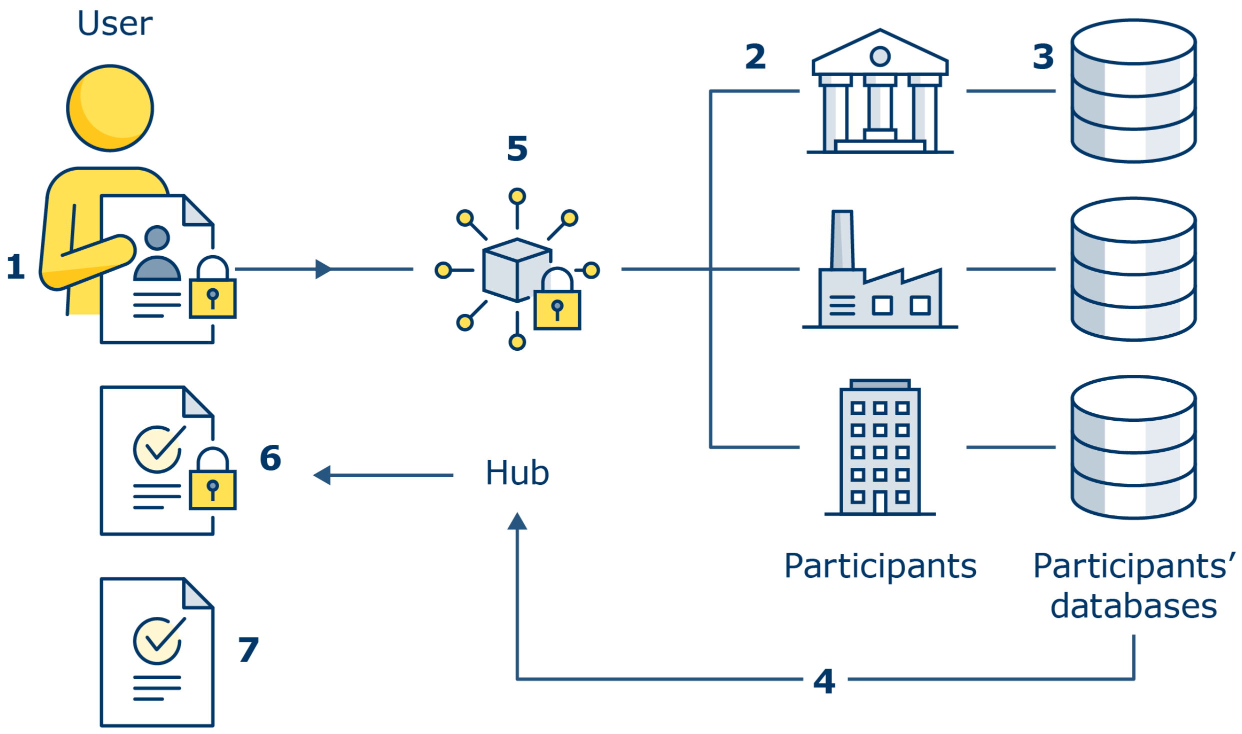
Developed in collaboration with Duality Technologies

Context

A group of law enforcement agencies and financial services organisations have formed a consortium to share personal information to detect and prevent financial crimes and related harms (eg fraud, money laundering, and cybercrimes). For these purposes, the members operate as independent data controllers. A “hub”, which acts independently to the other parties, acts as an intermediary. It receives and forwards queries to the other members, and then collects, aggregates, and forwards responses.

When a member of the consortium (controller A) conducts a financial crime investigation, it can submit a homomorphically encrypted query about a person to other members within the consortium (controllers B, C and D). The query will ask other members if they hold information for a particular person that is linked to financial crime activity.

The query is then sent to members via the hub to controllers B, C and D. These controllers send their homomorphically encrypted responses back to the hub. The hub aggregates them, before sharing the response with controller A, who is the only party to see the final response.



1. User creates and encrypts query, which is sent to the hub
2. Hub distributes query to other participants, masking the original inquirer and query from view
3. Encrypted query runs on each participants' database
4. Encrypted results sent back to the Hub
5. Hub aggregates the encrypted results
6. Aggregated and encrypted results sent back to user
7. Results are decrypted and actioned

Personal information that is encrypted in queries includes:

- customer identifiers and contact information (eg names, email addresses, postal addresses, ID numbers);
- financial transaction information (eg dates, amounts, counterparties);
- information about criminal convictions and offences;
- device information (eg IP address, device ID); and
- indicators of fraud or other financial crime.

Objective

Financial institutions who are members of the consortium process customer personal information to detect and prevent financial crime. Each member of the consortium wants to be able to share personal information about actual or suspected instances of fraud or other financial crime committed by its customers (or by known criminals, in the case of the law enforcement agencies). They all benefit from this reciprocal sharing of similar personal information by the other members.

Technical measures

Each member of the consortium uses fully homomorphic encryption (FHE) techniques to ensure an appropriate level of security when sharing personal information between members. A SQL-like query language is used to construct the queries.

The personal information in the query is homomorphically encrypted, rendering it pseudonymised to controller A, as it holds the private key required for decryption. Only controller A can convert the information back into personal information. For example, the identifiers for a customer are encrypted as below (where XXXXX represents an encrypted field in the query):

> “Do any accounts owned by [John Smith; NI Number: AB1234C; date of birth: 01/01/1980] have confirmed fraud flags?”
>
> “Do any accounts owned by [xxxxxxx; NI Number: xxxxxxx; date of birth: xxxxxxx] have confirmed fraud flags?”

The homomorphically encrypted query is then sent to members (controllers B, C and D) via the hub. Using homomorphic encryption techniques, controllers B, C and D can perform data matching on the encrypted query with their own customer information. However, they do not ‘see’ the original personal information in the query parameters and are prevented from learning which records in their data may have matched the query. This means controllers B, C and D can automatically respond to the query without needing to decrypt the personal information. The hub is used to route the encrypted query and results. It cannot see the query parameters nor the results that are encrypted (and the hub cannot decrypt them as it does not hold the decryption key).

The hub aggregates the individual homomorphically encrypted query responses, so that controller A does not see the specific responses provided by the other members. Therefore, it does not know whether the person is a customer of controllers B, C or D. The hub is also unable to infer these insights. The data flows are depicted in the diagram above.

For example, if a bank wants to better understand the risk profile of one of its customers, it may want to know whether accounts owned by the customer are receiving transfers from high-risk jurisdictions.

1. The bank sends an encrypted query to the network asking, “Have accounts owned by [this person] received transfers from high-risk jurisdictions in the last 30 days? If so, how many transactions from how many jurisdictions?”
2. Each member provides an encrypted response to the request. The underlying response from each member may look like “Yes; 20 transactions from 5 high risk jurisdictions,” or “No”.
3. The hub receives the encrypted responses (which it cannot decrypt). It then calculates an encrypted risk score based on the inputs from each respondent. The hub cannot see the underlying information nor the

risk score it has generated.

4. The encrypted risk score is sent back to the bank, who can decrypt it. It is then able to understand the risk of a given customer, without knowing where else the customer might be banking, and without obtaining any information about specific transactions.

Organisational measures

The members underpin this processing by a contractual arrangement and information governance controls which include:

- data protection impact assessments (DPIAs);
- pre-defined types of queries and information they will share;
- processes for raising and correcting issues with inaccurate information. This includes a complete audit trail to address people's information rights requests;
- multi-factor authentication to ensure only authorised end users can access the system;
- enforcing permissions for individual users of the system to determine who has the right to deploy queries; and
- training for end users on the system, including how to submit queries and how to configure rules about which queries the member will participate in.

How do the technical and organisational measures achieve the objective?

The system supports UK GDPR compliance in three main ways:

- Helps to fulfil the requirements of the security principle by providing appropriate technical and organisational measures.
- Supports accuracy principles as using this technology produces results that are equivalent to those in the clear. Therefore, there is no negative impact on data utility or the accuracy of the results.
- The use of homomorphic encryption helps to comply with data protection by design obligations.

The technical and organisational measures significantly reduce the risk to people as:

- parties can only decrypt queries and results with permission. This ensures information is protected even when computations occur. Additionally, due to the aggregation performed by the hub, no party in the consortium knows which party made the enquiry or provided a response. This is a benefit for people as no unnecessary suspicion is raised by the enquiry, if the queried person is innocent of any financial crime; and
- the system provides higher levels of security compared to methods that do not employ homomorphic encryption. If they used 'traditional' methods of encryption, the controllers receiving the query would need to decrypt the personal information in order to provide a response. This creates additional risk as the information could be exposed to an attacker. By using homomorphic encryption, the information is never decrypted and therefore it reduces the risk from attacks.

Risk and mitigation

- Risk of HE key compromise:
    - a new key pair is generated for each computation session, and removed immediately afterwards. This

  - means that they cannot use old keys or old ciphertexts to find patterns and reverse engineer newly generated ciphertexts.

- Risk of hub compromise:
  - The hub never has access to unencrypted personal information. This means that if the hub is compromised, a hacker will not be able to access any personal information.
  - They can also protect connectivity to the hub by other methods such as a VPN or firewall.
- Handling results:
  - Decrypted results are kept in the consortium members’ dedicated platform. Within each, access to the results is available only to authorised users.
- Risk of collusion between members:
  - There are both contractual and technical controls against collusion. For instance, from a technical perspective, multiple parties have to cryptographically “agree” to decrypt a response. This helps reach agreement about sharing a result. From a contractual perspective, the parties have agreements with one another to prevent collusion.
- Risk of system attacks:
  - The rate at which they can make queries, the type of queries and the people with permissions to make them are all monitored and restricted. This mitigates the risk of attackers making repeated queries on the information to extract as much as possible and reconstruct the dataset.

Case study 2: differentially private mixed noise in financial services

Developed in collaboration with Privacy Analytics

Context: insights from credit information

BankCo, a financial services company, collects and processes personal information from a variety of sources (eg debt collection agencies, credit card companies, and public records). The company uses this information to help them evaluate risk factors and repayment options when providing loans to people.

Credit scores are based on a variety of personal financial information, including debts and repayment history. This information is particularly sensitive, so BankCo wants to anonymise the information it uses for modelling purposes. However, if excessive deviation is introduced by anonymising it, this could have significant repercussions when they use the information for modelling credit risk and risk management.

In order to reuse or share the collected information or insights for secondary purposes, they need to use appropriate technical and organisational measures. This will ensure that the information is effectively anonymised while preserving its accuracy and consistency. BankCo could share this information with other financial institutions to provide valuable information and intelligence on geographic trends and overall risk profiles that inform business development and outreach. The information can also augment BankCo’s predictive models to create a more complete picture of different markets.

Objective: increase collaboration with statistically useful information

BankCo implements randomisation using a differentially private noise addition method to anonymise the information for secondary purposes. BankCo wants to keep record-level information rather than aggregating it, as this preserves its statistical patterns.

BankCo's anonymisation process allows its internal researchers and other financial institutions to use the anonymised information and derive new insights.

Technical measures: mixed noise and risk metrics

In this scenario, differential privacy protects the information elements by introducing a level of uncertainty through randomisation (noise injection). This approach limits the amount of personal information that can be extracted by integrating the privacy budget (an information limit on how much can be inferred or learned about someone) into the differentially private dataset itself. This way, BankCo or its collaborators can carry out analytical processing without BankCo having to put any restrictions on the nature of the information queries. This is because BankCo has limited what can be inferred or learned before making the information available.

BankCo uses the risk of singling out, or uniqueness, as a risk threshold to determine the privacy budget for the dataset. This alleviates concerns that exist with differential privacy of variable protection across datasets for the same privacy budget. This approach also ensures that people are not identifiable by providing plausible deniability for anyone's contribution. The anonymisation process uses several differentially private noise addition schemes (eg they use a Laplace distribution because it meets the mathematical properties that are used to define differential privacy).

Table 1 shows a sample of the personal financial information of five people from a larger dataset to demonstrate the techniques used. BankCo's challenge was to set the privacy budget to ensure they add sufficient noise to anonymise the information while maintaining sufficient utility for the analysis.

Table 1: Example of personal financial information

Person	Age	Income (1,000's)	Assets (1,000's)	Debts (1,000's)	Credit Utilisation
1	24.3	65.0	245	45.0	0.460
2	25.5	63.0	270	48.0	0.450
3	27.6	75.0	324	60.0	0.490
4	29.8	85.0	375	74.0	0.520
5	30.1	90.0	395	71.0	0.520

To preserve statistical properties of computed outputs while introducing a measurable level of uncertainty, they used a mixed noise mechanism. This combines a normal distribution (bell curve) with a Laplace distribution. Laplace noise is used to manage particularly sensitive information points that would otherwise be at risk of singling someone out. This approach allows models to be calibrated to account for the (predominantly normal) noise while using a consistent set of analytical methods and tools.

Table 2 shows how BankCo randomised the information by adding noise. The ranges shown indicate the precision around each value once noise is added to the entries in the table. Overlap between the individual records can be seen due to the inclusion of confidence intervals. For example, person 1 and person 2 share similar profiles, and separately person 3 and person 4 share similar profiles. Person 3 (outlier), with the addition of normally distributed noise, still contains outlier values with no overlap with other people's information that could allow singling out. To prevent person 3 being at risk of singling out, BankCo introduces additional Laplace noise to this record, as shown in Table 2.

Table 2: Example with confidence intervals for randomisation

Person	Age	Income (1,000's)	Assets (1,000's)
1	(23.2-25.3)	(60-70)	(220-270)
2	(24.5-26.5)	(58-68)	(245-295)
3 (outlier)	(26.6-28.6)	(70-80)	(299-349)
3 (tuned)	(25.6-29.6)	(65-85)	(274-374)
4	(28.8:30.8)	(80-90)	(350-400)
5	(29.1:31.1)	(85-95)	(390-620)

For person 3 (tuned), they used Laplace noise to ensure:

- the person's record cannot be singled out (as there is no overlap between each person's information); and
- the resulting dataset is differentially private.

In practice, BankCo found that few records require this treatment when using this mixed noise mechanism.

Table 3 shows the anonymised differentially private dataset. In practice, a larger dataset would demonstrate greater variation than shown in the table, as such a small dataset would require significant randomisation to be differently private.

Table 3: Example with differentially private data

Person	Age	Income (1,000's)	Assets (1,000's)
1	24.1	66.3	245
2	26.4	64.7	270
3	27.7	72.3	344
4	29.9	85.5	376

5	30.1	91.2	414

How do the technical and organisational measures achieve the objective?

BankCo uses a secure data environment to access the information, and gives access to this environment to its collaborators when it shares the anonymised information. This approach reduces the risk of re-identification by limiting the sharing (only approved collaborators have access). As the information is shared only with known financial institutions, the risk of attackers is lower, so BankCo decides to apply less noise. This increases the utility of the information they share. BankCo uses best practice security measures (ISO 27001) for the data sharing environment. They take the following actions:

- access logging and control;
- monitoring and alerting;
- ensures that the financial institutions accessing the information understand and follow the terms of use;
- penetration testing and auditing;
- output screening; and
- ensuring that all institutions that access the data sharing environment meet staff training requirements and are regularly reminded of the terms of use of the information.

The infrequent inclusion of Laplace noise to deal with outliers in the mixed noise mechanism reduces the degree of randomisation they need. This improves the statistical usefulness of the information and ensures correct statistical inference and statistics that can be calculated with true confidence intervals.

The technical and organisational measures they use reduce the risk of re-identification to a sufficiently remote level by:

- mitigating risks of singling out;
- linkability;
- inference with other available information;
- strictly controlling information access and use; and
- performing periodic identifiability assessments to determine if the technical or organisational measures provide effective anonymisation.