EUROPEAN UNION AGENCY
FOR CYBERSECURITY

DATA PSEUDONYMISATION: ADVANCED TECHNIQUES & USE CASES

Technical analysis of cybersecurity measures in data protection and privacy

JANUARY 2021

ABOUT ENISA

The European Union Agency for Cybersecurity, ENISA, is the Union's agency dedicated to achieving a high common level of cybersecurity across Europe. Established in 2004 and strengthened by the EU Cybersecurity Act, the European Union Agency for Cybersecurity contributes to EU cyber policy, enhances the trustworthiness of ICT products, services and processes with cybersecurity certification schemes, cooperates with Member States and EU bodies, and helps Europe prepare for the cyber challenges of tomorrow. Through knowledge sharing, capacity building and awareness raising, the Agency works together with its key stakeholders to strengthen trust in the connected economy, to boost resilience of the Union's infrastructure, and, ultimately, to keep Europe's society and citizens digitally secure. For more information, visit www.enisa.europa.eu.

CONTACT

For contacting the authors please use isdp@enisa.europa.eu.
For media enquiries about this paper, please use press@enisa.europa.eu.

CONTRIBUTORS

Cedric Lauradoux (INRIA), Konstantinos Limniotis (HDPA), Marit Hansen (ULD), Meiko Jensen (Kiel University), Petros Eftasthopoulos (NortonLifeLock)

EDITORS

Athena Bourka (ENISA), Prokopios Drogkaris (ENISA)

ACKNOWLEDGEMENTS

We would like to thank the European Data Protection Supervisor (EDPS), Technology and Privacy Unit, for reviewing this report and providing valuable comments.

We would also like to thank Giuseppe D'Acquisto (Italian DPA), Nils Gruschka (University of Oslo) and Simone Fischer-Hübner (Karlstad University) for their review and valuable comments.

LEGAL NOTICE

Notice must be taken that this publication represents the views and interpretations of ENISA, unless stated otherwise. This publication should not be construed to be a legal action of ENISA or the ENISA bodies unless adopted pursuant to the Regulation (EU) No 2019/881.
This publication does not necessarily represent state-of the-art and ENISA may update it from time to time.

Third-party sources are quoted as appropriate. ENISA is not responsible for the content of the external sources including external websites referenced in this publication.

This publication is intended for information purposes only. It must be accessible free of charge. Neither ENISA nor any person acting on its behalf is responsible for the use that might be made of the information contained in this publication.

COPYRIGHT NOTICE

ISBN 978-92-9204-465-7 - DOI 10.2824/860099.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Pseudonymisation is an established and accepted data protection measure that has gained additional attention following the adoption of the General Data Protection Regulation (GDPR) [1] where it is both specifically defined and many times referenced as a safeguard.

ENISA, in its prior work on this field, has explored the notion and scope of data pseudonymisation, while presenting some basic technical methods and examples to achieve pseudonymisation in practice. In this new report, ENISA complements its past work by discussing advanced pseudonymisation techniques, as well as specific use cases from the specific sectors of healthcare and cybersecurity. In particular, the report, building on the basic pseudonymisation techniques, examines advanced solutions for more complex scenarios that can be based on asymmetric encryption, ring signatures and group pseudonyms, chaining mode, pseudonyms based on multiple identifiers, pseudonyms with proof of knowledge and secure multi-party computation. It then applies some of these techniques in the area of healthcare to discuss possible pseudonymisation options in different example cases, while also exploring the possible application of the data custodianship model. Lastly, it examines the application of basic pseudonymisation techniques in common cybersecurity use cases, such as the use of telemetry and reputation systems.

Based on the analysis provided in the report, the following basic conclusions and recommendations for all relevant stakeholders are provided.

Defining the best possible technique

As it has been stressed also in past ENISA's reports, there is no fit-for-all pseudonymisation technique and a detailed analysis of the case in question is necessary in order to define the best possible option. To do so, it is essential to take a critical to look into the semantics (the "full picture") before conducting data pseudonymisation. In addition, pseudonymisation is only one possible technique and must be combined with a thorough security risk assessment for the protection of personal data.

Data controllers and processors should engage in data pseudonymisation, based on a security and data protection risk assessment and taking due account of the overall context and characteristics of personal data processing. This may also comprise methods for data subjects to pseudonymise personal data on their side (e.g. before delivering data to the controller/processor) to increase control of their own personal data.

Regulators (e.g. Data Protection Authorities and the European Data Protection Board) should promote risk-based data pseudonymisation through the provision of relevant guidance and examples.

[1] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679&from=EN

Advanced techniques for advanced scenarios

While the technical solution is a critical element for achieving proper pseudonymisation, one must not forget that the organisational model and its underlying structural architecture are also very important parameters of success. Advanced techniques go together with advanced scenarios, such as the case of the data custodianship model.

Data controllers and processors should consider possible scenarios that can support advanced pseudonymisation techniques, based – among other – on the principle of data minimisation.

The research community should support data controllers and processors in identifying the necessary trust elements and guarantees for the advanced scenarios (e.g. data custodianship) to be functional in practice.

Regulators (e.g. Data Protection Authorities and the European Data Protection Board) should ensure that regulatory approaches, e.g. as regards new technologies and application sectors, take into account all possible entities and roles from the standpoint of data protection, while remaining technologically neutral.

Establishing the state-of-the-art

Although a lot of work is already in place, there is certainly more to be done in defining the state-of-the-art in data pseudonymisation. To this end, research and application scenarios must go hand-in-hand, involving all relevant parties (researchers, industry, and regulators) to discuss joined approaches.

The European Commission, the relevant EU institutions, as well as Regulators (e.g. Data Protection Authorities and the European Data Protection Board) should support the establishment and maintenance of the state-of-the-art in pseudonymisation, bringing together all relevant stakeholders in the field (regulators, research community, and industry).

The research community should continue its efforts on advancing the existing work on data pseudonymisation, addressing special challenges appearing from emerging technologies, such as Artificial Intelligence. The European Commission and the relevant EU institutions should support and disseminate these efforts.

Towards the broader adoption of data pseudonymisation

Recent developments, e.g. in international personal data transfers, show clearly the need to further advance appropriate safeguards for personal data protection. This will only be intensified in the future by the use of emerging technologies and the need for open data access. It is, thus, important to start today the discussion on the broader adoption of pseudonymisation in different application scenarios.

Regulators (e.g. Data Protection Authorities and the European Data Protection Board), the European Commission and the relevant EU institutions should disseminate the benefits of data pseudonymisation and provide for best practices in the field.

1. INTRODUCTION

Pseudonymisation is an established and accepted data protection measure that has gained additional attention following the adoption of the General Data Protection Regulation (GDPR), where it is both specifically defined (Article 4(5) GDPR)[2] and many times referenced as a safeguard. Technical and organisational measures, in particular for security and data protection by design, comprise pseudonymisation. The application of pseudonymisation to personal data can reduce the risks to the data subjects concerned and help controllers and processors meet their data protection obligations. Nevertheless, not every so-called pseudonymisation mechanism fulfils the definition of the GDPR, and pseudonymisation techniques that may work in one specific case to achieve data protection, may not be sufficient in other cases[3]. Still, the basic concept of substituting identifying data with pseudonyms can contribute to reducing data protection risks.

1.1 BACKGROUND

Given the growing importance of pseudonymisation for both data controllers and data subjects, ENISA has been working over the past years on this topic, in co-operation with experts and national regulatory authorities. Indeed, ENISA issued its first relevant report in January 2019 (ENISA, 2019 - 1) presenting an overview of the notion and main techniques of pseudonymisation in correlation with its new role under the GDPR. A second ENISA report followed in November 2019 (ENISA, 2019 - 2) with a more detailed analysis of the technical methods and specific examples and best practices for particular data sets, i.e. email addresses, IP addresses and more complex data sets. In addition, a dedicated workshop on pseudonymisation[4] was co-organised by ENISA and the Data Protection Authority of the German Federal State of Schleswig-Holstein (ULD) in November 2019 in order to exchange information and experience among key stakeholders[5].

While work and regulatory guidance in the field is growing[6], it is apparent that further effort is needed, especially addressing specific application scenarios and different types of datasets. Both ENISA's reports and the conclusions of the ULD-ENISA workshop lead towards this direction, which could eventually support the development of "a catalogue of techniques" or a "cookbook" towards applying pseudonymisation in practice in different application scenarios.

[2] It has to be noted that personal data that has been pseudonymised is still regarded as "personal data" pursuant to Article 4(1) GDPR and must not be confused with "anonymised data" where it is no longer possible for anyone to refer back to individual data subjects, see Recital 28 GDPR.
[3] In order to fully understand the role of pseudonymisation for the processing of personal data, a full analysis of the legal situation in the specific case would also be required.
For the assessment of concrete processing operations, controllers and processors must take account of all factors playing a role for the risk to the fundamental rights of individuals induced by the processing as such and by potential breaches of security, also going beyond technical and organisational measures considered in this study.
[4]https://www.enisa.europa.eu/events/uld-enisa-workshop/uld-enisa-workshop-pseudonymization-and-relevant-security-technologies
[5] https://www.enisa.europa.eu/events/uld-enisa-workshop/uld-enisa-workshop-notes/view
[6] See also EDPS and Spanish DPA joint paper on the introduction of hash as pseudonymisation technique, https://edps.europa.eu/data-protection/our-work/publications/papers/introduction-hash-function-personal-data_en

Should this be achieved, it would be a significant step towards the definition of the state-of-the-art for pseudonymisation techniques.

Against this background and following previous relevant ENISA's work[7], the Agency decided under its 2020 work-programme to elaborate further on the practical application of data pseudonymisation techniques.

1.2 OBJECTIVES

The overall scope of this report is to continue past ENISA's work by providing (on the basis of the previous analysis) specific use cases for pseudonymisation, along with more advanced techniques and scenarios that can support its practical implementation by data controllers or processors.

More specifically, the objectives of the report are as follows:

- Explore further advanced pseudonymisation techniques which were not covered in prior ENISA's work, based on cryptographic algorithms and privacy enhancing technologies.

- Discuss specific application use cases where pseudonymisation can be applied, analysing the particular scenarios, roles and techniques that could be of interest in each case. In particular, for the scope of the report, use cases are presented in two different sectors: (a) healthcare information exchange; (b) cybersecurity information exchange with the use of innovative technologies (e.g. machine learning technologies).

It should be noted that the selection of the use cases was based on the fact that the specific sectors (healthcare, cybersecurity) represent quite common cases for the application of pseudonymisation in several real-life situations. At the same time, the selected use cases also reflect diverse requirements with regard to pseudonymisation, e.g. in terms of the scenarios/roles involved, as well as in terms of the techniques that could be applied in practice.

The target audience of the report consists of data controllers, data processors and manufacturers/producers of products, services and applications, Data Protection Authorities (DPAs), as well as any other interested party in data pseudonymisation.

The document assumes a basic level of understanding of personal data protection principles and the role/process of pseudonymisation. For an overview of data pseudonymisation under GDPR, please also refer to relevant ENISA's work in the field (ENISA, 2019 - 1) & (ENISA, 2019 - 2).

The discussion and examples presented in the report are only focused on technical solutions that could promote privacy and data protection; they should by no means be interpreted as a legal opinion on the relevant cases.

[7] https://www.enisa.europa.eu/topics/data-protection/privacy-by-design

1.3 OUTLINE

The outline of the remaining part of the report is as follows:

- Chapter 2 provides an overview of the basic scenarios, pseudonymisation techniques and policies discussed under (ENISA, 2019 - 2).
- Chapter 3 presents a number of advanced pseudonymisation techniques, including asymmetric encryption, ring signatures, chaining mode, Merkle trees, pseudonyms with proof or ownership, secure multiparty computation and secret sharing schemes.
- Chapter 4 analyses pseudonymisation techniques and application scenarios in the area of healthcare. It particularly focuses on the use of the tree-based pseudonyms approach and the data custodianship model.
- Chapter 5 discusses the application of pseudonymisation in the broader area of cybersecurity technologies.
- Chapter 6 summarises the previous discussions and provides the main conclusions and recommendations for all related stakeholders.

This report is part of the work of ENISA in the area of privacy and data protection[8], which focuses on analysing technical solutions for the implementation of GDPR, privacy by design and security of personal data processing.

[8] https://www.enisa.europa.eu/topics/data-protection

2. PSEUDONYMISATION BASICS

As mentioned in (ENISA, 2019 - 2), the most obvious benefit of pseudonymisation is to hide the identity of the data subjects from any third party (other than the Pseudonymisation Entity, i.e. the entity responsible for pseudonymisation). Still, pseudonymisation can go beyond hiding real identities and data minimisation into supporting the data protection goal of unlinkability and contributing towards data accuracy.

When implementing pseudonymisation, it is important to clarify as a first step the application scenario and the different roles involved, in particular the role of the Pseudonymisation Entity (PE), which can be attributed to different entities (e.g. a data controller, a data processor, a Trusted Third Party or the data subject), depending on the case. Under a specific scenario, it is then required to consider the best possible pseudonymisation technique and policy that can be applied, given the benefits and pitfalls that each one of those techniques or policies entails. Obviously, there is not a one-size-fits-all approach and risk analysis should in all cases be involved, considering privacy protection, utility, scalability, etc.

In that regard, this Chapter provides a brief overview of the basic pseudonymisation scenarios and techniques, as these are outlined in (ENISA, 2019 - 2), which will be then further complemented and analysed in the next Chapters of the report.

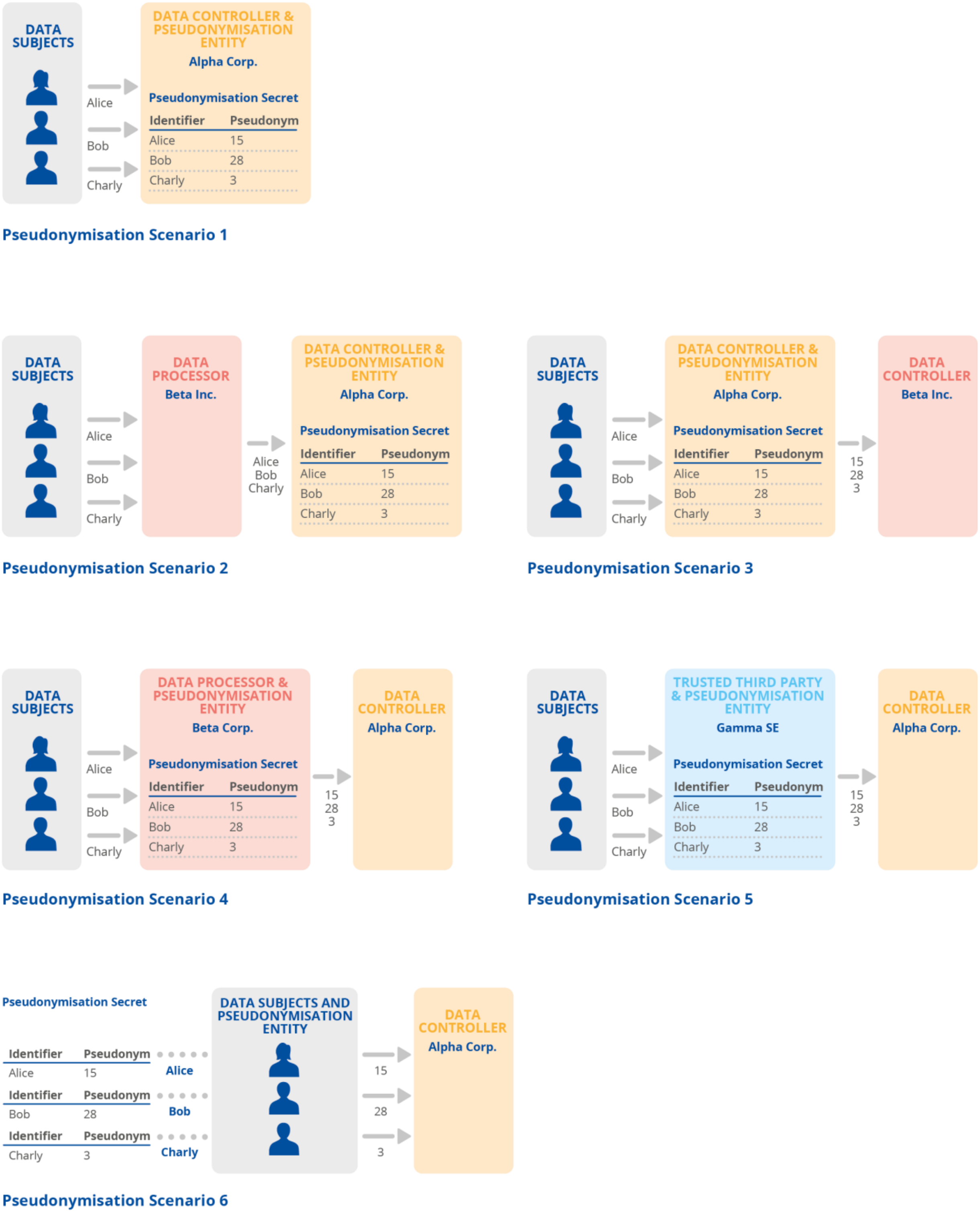
2.1 PSEUDONYMISATION SCENARIOS

Six different pseudonynimisation scenarios are discussed in (ENISA, 2019 - 2) and are presented in Figure 1 below. The defining difference between the scenarios is firstly the actor who takes the role of the Pseudonymisation Entity (PE) and secondly the other potential actors that may be involved (and their roles).

Clearly, in all three first scenarios in Figure 1, the data controller is the PE, either acting alone (scenario 1) or involving a processor before pseudonymisation (scenario 2) or after pseudonymisation (scenario 3). In scenario 4, the PE is the processor that performs pseudonymisation on behalf of the controller (thus, controller maintaining still control over the original data). Scenario 5 sets a Trusted Third Party entity, outside the control of the data controller, as PE, therefore involving an intermediary to safeguard the pseudonymisation process. Lastly, scenario 6 provides for data subjects to be the PE and, thus, control an important part of the pseudonymisation process.

Later in this report we will explore the practical application of these scenarios in specific cases, especially scenarios 1 and 3 under cybersecurity use cases (Chapter 5) and scenarios 5 and 6 under healthcare use cases (Chapter 4). For the scenario 5 particularly we will further detail the notion of the Trusted Third Party (data custodian) and the forms that it could take in the healthcare sector.

Figure 1: Basic pseudonymisation scenarios



Identifier	Pseudonym
Alice	15
Bob	28
Charly	3

Pseudonymisation Scenario 1

Identifier	Pseudonym
Alice	15
Bob	28
Charly	3

Pseudonymisation Scenario 2

Identifier	Pseudonym
Alice	15
Bob	28
Charly	3

Pseudonymisation Scenario 3

Identifier	Pseudonym
Alice	15
Bob	28
Charly	3

Pseudonymisation Scenario 4

Identifier	Pseudonym
Alice	15
Bob	28
Charly	3

Pseudonymisation Scenario 5

Identifier	Pseudonym
Alice	15

Identifier	Pseudonym
Bob	28

Identifier	Pseudonym
Charly	3

Pseudonymisation Scenario 6

2.2 PSEUDONYMISATION TECHNIQUES AND POLICIES

The basic pseudonymisation techniques that can be applied in practice, as also discussed in (ENISA, 2019 - 2) are as follows:

- **Counter:** the simplest pseudonymisation function, where the identifiers are substituted by a number chosen by a monotonic counter. Its advantages rest with its simplicity, which make it a good candidate for small and not complex datasets. It provides for pseudonyms with no connection to the initial identifiers (although the sequential character of the counter can still provide information on the order of the data within a dataset). However, the solution may have implementation and scalability issues in cases of large and more sophisticated datasets.

- **Random Number Generator (RNG):** a similar approach to the counter with the difference that a random number is assigned to the identifier. It provides strong data protection (as, contrary to the counter, a random number is used to create each pseudonym, thus it is difficult to extract information regarding the initial identifier, unless the mapping table is compromised). Collisions, however, may be an issue[9], as well as scalability, depending on the implementation scenario.

- **Cryptographic hash function:** directly applied to an identifier to obtain the corresponding pseudonym with the properties of being a) one-way and b) collision free[10]. While a hash function can significantly contribute towards data integrity, it is generally considered weak as a pseudonymisation technique, as it is prone to brute force and dictionary attacks (ENISA, 2019 - 2).

- **Message authentication code (MAC)**: similar to a cryptographic hash function except that a secret key is introduced to generate the pseudonym. Without the knowledge of this key, it is not possible to map the identifiers and the pseudonyms. MAC is generally considered as a robust pseudonymisation technique from a data protection point of view. Recovery might be an issue in some cases (i.e. if the original identifiers are not being stored). Different variations of the method may apply with different utility and scalability requirements. HMAC (Bellare, Canetti, & Krawczyk, 1996) is by far the most popular design of message authentication code used in Internet protocols.

- **Symmetric encryption:** the block cipher is used to encrypt an identifier using a secret key, which is both the pseudonymisation secret and the recovery secret. Using block ciphers for pseudonymisation requires to deal with the block size. Symmetric encryption is a robust pseudonymisation technique, with several properties being similar to MAC (i.e. the aforementioned properties of the secret key). One possible issue in terms of data minimisation is that the PE can always reverse the pseudonyms, even if there is no need to store the initial individuals' identifiers.

[9] Still, it should be noted that cryptography-based constructions of pseudo-random number generators are available, which can avoid collisions if they are properly configured and could be possibly similarly used to provide pseudonyms (e.g. discrete logarithm based constructions (Blum, Feldman, & Micali, 1984).

[10] This holds under the assumption that a cryptographically strong hash function is used. Moreover, it is essential that hashing should be applied to appropriate individual's identifiers (e.g. hashing the first name and last name may not avoid collisions, if this combination does not constitute an identifier in a specific context – i.e. there may be two individuals with the same fist name and last name). More details are given in (ENISA, 2019 - 1) (ENISA, 2019 - 2).

Independently of the choice of the technique, the pseudonymisation policy (i.e. the practical implementation of the technique) is also critical to the implementation in practice. Three different pseudonymisation policies have been considered to that end:

- **Deterministic pseudonymisation**: in all the databases and each time it appears, Id is always replaced by the same pseudonym $pseudo$.
- **Document randomised pseudonymisation:** each time Id appears in a database, it is substituted with a different pseudonym $(pseudo_1, pseudo_2,...)$; however, Id is always mapped to the same collection of ($pseudo_1, pseudo_2$) in the dataset A and B.
- **Fully randomised pseudonymisation:** for any occurrences of Id within a database A or B, Id is replaced by a different pseudonym ($pseudo_1, pseudo_2$).

As summarised in (ENISA, 2019 - 2), the choice of a pseudonymisation technique and policy depends on different parameters, primarily the identified level or risk and the expected/identified utilisation of the pseudonymised dataset In terms of protection, random number generator, message authentication codes and encryption are stronger techniques as they prevent by design exhaustive search, dictionary search and random search. Still, utility requirements might lead the Pseudonymisation Entity (PE) towards a combination of different approaches or variations of a selected approach. Similarly, with regard to pseudonymisation policies, fully-randomised pseudonymisation offers the best protection level but prevents any comparison between databases. Document-randomised and deterministic functions provide utility but allow linkability between records.

Using the aforementioned scenarios, techniques and policies as a basis for any practical implementation of pseudonymisation, Chapter 3 explores more advanced techniques that often rely on the basic existing techniques, while offering advanced protection, along with other properties. Chapters 4 and 5 discuss how both basic and advanced techniques can be employed in practice with specific examples and use cases.

3. ADVANCED PSEUDONYMISATION TECHNIQUES

In Chapter 2 we presented a number of pseudonymisation techniques (alongside with relevant policies and scenarios) that can improve the level of protection of personal data, provided that the pseudonymisation secrets used to create the pseudonyms are not exposed. However, in order to address some specific personal data protection challenges, typical pseudonymisation techniques, such as pseudonymisation tables or conventional cryptographic primitives (ENISA, 2019 - 2) may not always suffice. It is possible though to create pseudonyms addressing more complex situations, whilst the risks of a personal data breach are minimised.

This Chapter reviews some of these solutions, based on cryptographic techniques, and discusses what problems they could be used to solve in practice. In particular, the following techniques are presented:

- Asymmetric encryption.
- Ring signatures and group pseudonyms.
- Chaining mode.
- Pseudonyms based on multiple identifiers or attributes.
- Pseudonyms with proof of ownership.
- Secure multiparty computations.
- Secret sharing schemes.

For each technique we analyse its application to support pseudonymisation, pointing out possible examples, as well as shortcomings in this context.

3.1 ASYMMETRIC ENCRYPTION

Although symmetric encryption is most commonly used (compared to asymmetric encryption) in the area of pseudonymisation, asymmetric encryption has some interesting properties that could also support data minimisation and the need-to-know principle, while providing robust protection.

Asymmetric encryption enables the possibility to have two different entities involved during the pseudonymisation process: (i) a first entity can create the pseudonyms from the identifiers using the Public pseudonymisation Key (PK), and (ii) another entity is able to resolve the pseudonyms to the identifiers using the Secret (private) pseudonymisation Key (SK)[11]. The entity who applies the pseudonymisation function and the entity who can resolve the pseudonyms into the original identifiers *do not have to share the same knowledge*.

For example, a data controller can make available its public key PK to its data processors. The data processors can collect and pseudonymise the personal data using the PK. The data controller is the only entity which can later compute the initial data from the pseudonyms. Such a scenario is strongly related to the generic scenario of a data processor being the

[11] Actually other combinations are also possible, as they are being discussed later on; for example, utilising the private key may allow for proof of ownership of a pseudonym (see Section 3.5, Chapter 3).

Pseudonymisation Entity (see Scenario 4 in Section 2.1 Chapter 2), with the additional advantage, in terms of protecting individuals' identities, that the processors do not have the pseudonymisation secret[12]. It is not possible to achieve such pseudonymisation scheme using symmetric encryption because the data controller and the data processor need to share the same pseudonymisation secret.

Similarly, a Trusted Third Party (TTP) may publish its public key PK to one or more data controllers. In such a scenario, the TTP can resolve any pseudonym created by a data controller using its private key SK (e.g. at the request of a data subject); such scenario may also be relevant to cases of joint controllership, where a controller is performing the pseudonymisation and another controller only receives the pseudonymised data for further processing (see Scenario 5 in (ENISA, 2019 - 2)).

Therefore, asymmetric encryption facilitates the *delegation* of pseudonymisation. However, pseudonymisation using asymmetric encryption needs to be carefully implemented (see also (ENISA, 2019 - 1)). For example, textbook application of RSA (Rivest, Shamir, & Adleman, 1978) or Rabin scheme (Rabin, 1979) both fail to achieve strong pseudonymisation. Indeed, since the encryption key PK is publicly available, an adversary knowing both PK and the set of original identifiers can perform an exhaustive search attack for those schemes. Therefore It is important to use a randomised encryption scheme – i.e. at each encryption, a random value (nonce) is being introduced to ensure that for given input (user's identifier) and PK, the output (pseudonym) cannot be predicted (ENISA, 2019 - 1). Several asymmetric encryption algorithms are by default randomised, like Paillier (Paillier, Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, 1999) or Elgamal (Elgamal, 1985). It should be noted that, by these means, a fully-randomised pseudonymisation policy is achieved – i.e. a different pseudonym is derived each time for the same identifier, without changing the pseudonymisation process or the pseudonymisation secret (see Section 5.2.3 (ENISA, 2019 - 2)).

Although in cryptographic applications the usage of asymmetric encryption algorithms implies that the relevant PKs are available to everyone (including adversaries), in the context of pseudonymisation we may deviate from this assumption (thus allowing for more flexibility in designing pseudonymisation schemes)[13]; indeed, the PK in such cases is needed to be known only by the pseudonymisation entities (regardless of their role – i.e. data controllers, data processors, data subjects), since these are the only entities, which will need to utilise this PK to perform pseudonymisation – and, thus, this public key should be distributed to the Pseudonymisation Entities through a secure channel. However, even if the PK is indeed available to everyone, the inherent security properties of asymmetric encryption ensure that an adversary will not be able to reverse pseudonymisation, under the assumption that a cryptographically strong asymmetric algorithm is being used[14].

It is worth mentioning that certain asymmetric encryption schemes support homomorphic operations (Armknecht, et al., 2015). Homomorphic encryption is a specific type of encryption, allowing a third party (e.g. a cloud service provider) to perform certain computations on the ciphertexts without having knowledge of the relevant decryption key[15]. For instance, the product of two pseudonyms created using Paillier's scheme (which is homomorphic) is the pseudonym of sum of the two identifiers. This advantage, in terms of cryptographic operation, can be also a drawback in terms of pseudonymisation. An adversary can substitute a pseudonym by the

[12] Assuming that the data processors do not store the mapping between original identifiers and the derived pseudonyms.

[13] This may be necessary in some cases, e.g. if it is not desirable that any other party, apart from the Pseudonymisation Entity, has adequate information to allow that party generating valid pseudonyms for a specific application.

[14] Indeed, an adversary, even if the PK is unknown, can resolve any pseudonyms if he/she knows either (i) the original identifiers and all the nonces or (ii) the secret key SK. For the first case (i), the adversary must know all the nonces used during the pseudonymisation which represents a large quantity of data. The second case (ii) is rather unlikely to occur if the secret key SK is protected with appropriate measures (e.g. encrypted on an access-controlled device).

[15] For example, an additively homomorphic encryption scheme with *E* and *D* as the encryption and decryption functions respectively, satisfies the following: For any input arithmetic messages $m_1, m_2$and the corresponding ciphertexts $c_1 = E(m_1), c_2 = E(m_2)$ obtained by the same encryption key, performing the decryption $D(c_1 + c_2)$ with the corresponding private key yields the sum $m_1 + m_2$ (i.e. adding ciphertexts results in the encryption of the sum of the original messages).

product of other pseudonyms P1 and P2 without knowing the public key PK or even the original identifiers associated to P1 and P2; therefore, if the sum of two identifiers is also a meaningful identifier (for example, in case of numerical identifiers with no prescribed format), a valid pseudonym can be generated by an adversary without having access to the pseudonymisation secret. This issue can also occur with certain symmetric encryption schemes. Consequently, if the homomorphic property is present, appropriate safeguards should also be in place (for example, appropriate integrity measures to ensure that it is not possible to tamper with the pseudonyms).The generation speed and the size of the pseudonym obtained using asymmetric encryption can also be an issue. These parameters are strongly correlated to the size of the keys[16]. For certain setups, the key size can be up to 2018 or 3096 bits. However, it is possible to use elliptic curves cryptography to reduce this cost to 256 bits (Paillier, Trapdooring Discrete Logarithms on Elliptic Curves over Rings, 2000), (Joye, 2013). There are efficient implementations of elliptic curves cryptography that reduce the performance gap with symmetric encryption.

Several pseudonymisation schemes based on asymmetric encryption have already been proposed. A typical application is to make available healthcare data to research groups; more precisely, by using fully randomised pseudonymisation schemes based on asymmetric cryptography (ENISA, 2019 - 1), we may ensure that the identifiers (e.g. social security number or medical registration number or any other identifier) of a given patient are not linkable. For instance, a participant may have different local pseudonyms at doctors X, Y, Z, and at medical research groups U, V, W – thus providing domain-specific pseudonyms to ensure unlinkability between these different domains; by these means, doctors will store both the real name/identity of their patients and their local pseudonyms, but researchers will only have (their own) local pseudonyms. . As characteristic examples, ElGamal cryptosystem has been used in (Verheul, Jacobs, Meijer, Hildebrandt, & de Ruiter, 2016) and Paillier in (He, Ganzinger, & Hurdle, 2013), (Kasem-Madani, Meier, & Wehner).

Another application of asymmetric encryption for pseudonymisation is outsourcing. In (Lehmann, 2019), a distributed pseudonymisation scheme based on ElGamal is proposed. An entity can pseudonymise a dataset without learning neiither any sensitive data nor the created pseudonyms. It is also used as a building block to create a more advanced form of pseudonymisation like in (Camenisch & Lehmann, (Un)linkable Pseudonyms for Governmental Databases, 2015), (Camenisch & Lehmann, Privacy-Preserving User-Auditable Pseudonym Systems, 2017).

As another characteristic example, in which asymmetric cryptographic primitives have an essential role, is the case of the so-called linkable transaction pseudonyms, introduced in (Weber, 2012). By the approach described therein, users may generate their own transaction pseudonyms – i.e. short-term pseudonyms – providing unlinkability (that is different pseudonyms each time for the same user), but with the additional property that some linkability can be present in a step-wise re-identification fashion (for example, authorised parties may link pseudonyms without being able though to reveal the actual identity or may check if a pseudonym corresponds to a user with specific attributes). However, in the work presented in (Weber, 2012), not simply asymmetric encryption but more complex cryptographic primimitives such as zero-knowledge proofs and threshold encryption are being used; such primitives are being individually discussed next in this Chapter.

3.2 RING SIGNATURES AND GROUP PSEUDONYMS

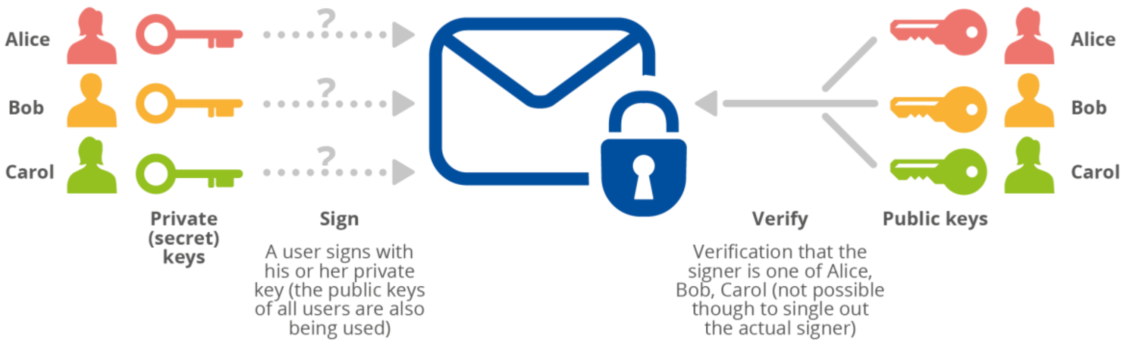
The notion of digital signatures is widely used in many applications, constituting a main cryptographic primitive towards ensuring both the integrity of the data as well as the authentication of the originating user, i.e. the so-called signer of the message. The underlying idea of a conventional digital signature is that anybody can verify the validity of the signature,

[16] https://www.keylength.com/

which - in the typical scenario - is associated with a known signer. Typically, asymmetric encryption provides the means for implementing digital signatures, since they are both based on the safe concept of Public and Private key, as well as on a Trusted Third Party (TTP) issuing the keys. In many pseudonymisation schemes, like (Camenisch & Lehmann, (Un)linkable Pseudonyms for Governmental Databases, 2015), (Camenisch & Lehmann, Privacy-Preserving User-Auditable Pseudonym Systems, 2017), (Lehmann, 2019), signature schemes are also combined with other primitives (e.g. asymmetric encryption) to achieve advanced properties like auditability.

Several advanced digital signature techniques are known, with diverge properties, each aiming to a different challenge based on the requirements of a specific application. One such scheme is the so-called ring signature. A ring signature is a digital signature that is created by a member of a group of users, so as to ensure the following property: the verifier can check that the signature has indeed been created by a member from this group, whilst he/she cannot determine exactly the person in the group who has created the signature. In other words, the identity of the signer is indistinguishable from any other user of this group. The first approach for ring signatures scheme has been proposed by (Rivest, Shamir, & Tauman, 2001). Ring signatures do not necessitate a TTP. This concept is based on asymmetric cryptography, as it is assumed that each possible signer (i.e. the k^{th} amongst n users, $1 \leq k \leq n$) is associated with a public key P_k and a relevant secret (private) key S_k. In this scheme, any user from the group can generate, for any given message *m*, a signature *s* by appropriately using his/her secret key and the public keys of all the other members of the group. A verifier with access to the public keys of all members of the group is able to confirm that a given signed message m has been signed by a member of the group, but he/she cannot identify explicitly which user is the actual signer. In their original paper, (Rivest, Shamir, & Tauman, 2001) described ring signatures as a way to leak a secret; for instance, a ring signature could be used to provide a verifiable signature from “a high-ranking official” (i.e. a member of a well-determined group of officials), without revealing though who exactly is the official that signed the message.

Figure 2: The ring signature operation


Alice
Bob
Carol
?
?
?
Private (secret) keys
Sign
A user signs with his or her private key (the public keys of all users are also being used)
Verify
Verification that the signer is one of Alice, Bob, Carol (not possible though to single out the actual signer)
Public keys
Alice
Bob
Carol

A variant of traditional ring signatures, being called linkable ring signatures, has been proposed in (Liu & Wong, 2005), which allows any of *n* group members to generate a ring signature on some message, with the additional property that all signatures from the same member can be linked together.

Although ring signatures are often being mentioned as anonymous signatures in the literature, they actually constitute pseudonymous data. Indeed, such signatures are in fact uniquely associated to a person (under the assumption that the group of possible signers consists of individuals), despite the fact that no other entity can explicitly re-identify the signer. However, the secret key of the signer suffices to prove, if it is revealed, that the signature has been generated by him/her. Therefore, we actually have a pseudonymous scheme, allowing for a specific utilisation (i.e. verifying that the data stem from a well-determined group of users), in which the pseudonymisation secret (i.e. the secret key[17]) is under the sole control of the data subject.

Ring signatures are being recently used, as a privacy enhancing technology, for the creation of the so-called anonymous cryptocurrencies (see, for, example, the open-source technology Cryptonote[18]); in this framework, ring signatures may provide the means for implementing untraceable payments – i.e. for each incoming transaction, all possible senders are equiprobable. In other words, a verifier can only verify that a signer of a transaction belongs to a specific group of users, without being able to explicitly pinpoint the user that signed the transaction. Despite the use of the term "anonymous cryptocurrency", these data are actually pseudonymous – and not anonymous – data, where the user (signer) owns his/her pseudonymisation secret.

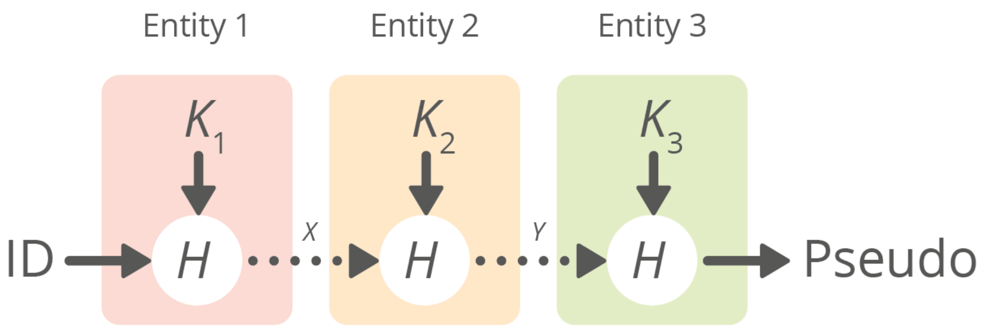
Group pseudonyms have been used in many contact tracing protocols (like Pronto-C2 (Avitabile, Botta, Iovino, & Visconti, 2020)) proposed during the COVID-19 pandemic. The idea is that each time two data subjects meet, a pseudonym is created with a contribution from each data subject. After the encounter, they both have computed the same pseudonym. Each data subject has a list of group or encountered pseudonyms. If one of them is exposed, all his/her group pseudonyms are published on a public board and all the contacts can check if they have been exposed. This pseudonymisation scheme is randomised in such that when two data subjects meet again, they always obtain a new group pseudonym to avoid any malicious traceability.

3.3 CHAINING MODE

As discussed in (ENISA, 2019 - 2), a secure cryptographic hash function is rarely expected to be an appropriate pseudonymisation technique. Authentication codes and keyed-hash functions must be preferred – which include the use of a secret key. However, more advanced techniques can be obtained by appropriately chaining hash functions, as discussed next.

Chaining the outputs of multiple cryptographic hash functions was first proposed by Lamport (Lamport, 1981) to store passwords. This idea has been generalised to create key derivation functions (Krawczyk, 2010) and password hashing functions (Biryukov, Dinu, & Khovratovich, 2016), which can be used to pseudonymise personal data.

Figure 3: A typical hash chain



Previous approaches of chaining (Lamport, 1981), (Krawczyk, 2010), (Biryukov, Dinu, & Khovratovich, 2016) involved only one entity, however the approach of chaining keyed hash functions discussed in this report is distributed (Figure 3). It is a layered approach: i.e. several somehow intermediate pseudonyms are (temporarily) generated, in order to finally obtain the

[17] This actually constitutes the additional information needed to allow re-identification, according to the Article 4(5) of the GDPR.
[18] https://cryptonote.org/

pseudonym, which is the output of the last hash function. Each layer is computed by a different entity[19] and each entity holds a secret used to obtain an intermediate pseudonym.

As depicted in Figure 3, K_1 is used to obtain the temporary value $X = H_{K_1}(ID)$. Value X is then transmitted to the second entity which computes $Y = H_{K_2}(X)$. Finally, the last entity computes the $Pseudo = H_{K_3}(Y)$. Such a chain mitigates the risk of a data breach. An adversary needs to compromise the three entities in order to reverse the pseudonymisation, i.e. he/she must know K_1, K_2, K_3.

The only drawback of chaining is that pseudonym resolution requires to have the three entities to cooperate. However, on the other side, this ensures an additional property that cannot be achieved by a single keyed hash function; any entity receiving an intermediate pseudonym cannot reverse it, whereas the first entity (which obviously knows the original identifiers) is not able to match the final pseudonyms with the identifiers (of course, these properties hold under the assumption that the secret keys are not exchanged between the pseudonymisation entities). For example, the recipient of the final (or even any intermediate) pseudonym may perform statistical/scientific analysis on the pseudonymous data without being able to map the pseudonyms to the original users' identifiers. A hash chain can be further generalised into more complex structures.

Apparently, the notion of chaining pseudonymisation mechanisms could also be applied more generally – i.e. not only for cryptographic hash functions, but also for other techniques (e.g for typical symmetric cryptographic algorithms). Actually, depending on the application scenario, each entity may apply a different pseudonymisation technique in such a chaining approach, thus allowing for more flexibility which in turn may give rise to more sophisticated pseudonymisation schemes.

3.4 PSEUDONYMS BASED ON MULTIPLE IDENTIFIERS OR ATTRIBUTES

Pseudonymisation is usually considered as the processing of an identifier into a pseudonym (one-to-one mapping). It is possible to slightly modify this definition to add new properties. The pseudonym can be the processing of several identifiers (many-to-one mapping). The identifiers can be homogeneous, i.e. they have the same type (only phone number for instance) and they are related to different individuals. Otherwise, they are heterogeneous and they match different attributes of a single individual (social security number, phone number, first name and last name). Any case in between is possible. Any known pseudonymisation technique can be easily applied to more than one identifiers – e.g. a keyed hash function, as pseudonymisation primitive, may have, as input data, a combination of more than one identifiers of an individual in order to derive a pseudonym for him/her (see also (ENISA, 2019 - 1)). However, to ensure some additional properties of such pseudonyms which correspond to many-to-one-mappings, more sophisticated approaches are needed; this is discussed next.

Cryptographic accumulators (Benaloh & de Mare, 1993), (Fazio & Nicolosi, 2002)] are best fitted to implement a many-to-one pseudonymisation scheme. A cryptographic accumulator can accumulate a set *L* of values into a unique, small value *z* in such a way that it is possible only for elements $y \in L$ to provide a proof that a given y actually has been accumulated within *z*. Such a proof is called a witness *w*.

To illustrate this short definition, we provide an example based on *Merkle Tree* (Merkle, 1987). This cryptographic data structure is a binary tree constructed through hash functions (which in turn could be seen as a generalisation of hash chains). This tree structure could be

[19] These entities may have specific roles in terms of personal data protection. For example, these entities could be joint controllers (Article 26 of the GDPR), each of them with a well described role.

appropriately used for pseudonymisation purposes, as follows: a) the root of the tree is the pseudonym; b) the leaves of the tree correspond to the authentication codes of the identifiers computed using a message authentication code *G* and different keys[20]. In such case, the inner nodes of the tree are computed using a cryptographic hash function *H*. The role of the authentication codes is to ensure that no dictionary attack is possible. The root and the inner nodes of the tree are computed using *H* to let anybody verify that a leaf is associated to a given root *z* (i.e. being the witness w_i for the corresponding ID_i).

For example, let us consider the Merkle tree in Figure 4. The pseudonym has been derived by four identifiers (ID_1, ID_2, ID_3 and ID_4) and, thus, it depends on all of them. To prove that a known identifier ID_1 has contributed in deriving the root pseudonym z, the contributor of ID_1 reveals the corresponding key k1 (which was used for constructing the leaf of the tree corresponding to ID_1), as well as the following information:

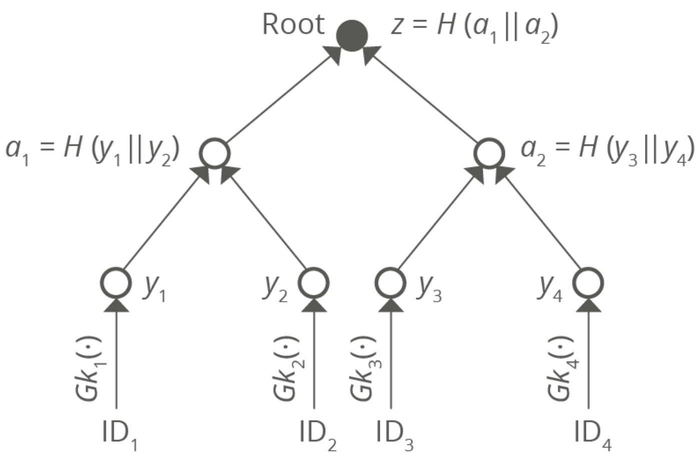
$y_1 = G_{k1}(ID_1)$ (actually y_1 is computed by the verifier who knows ID_1 and k_1)

$a_1 = H(y_1||y_2)$ (y_2 is provided as part of the witness w_1 of ID_1, to compute a_1)

$z' = H(a_1||a_2)$ (both a_1 and a_2 are also parts of the witness w_1 of ID_1).

If z' ≠ z then ID_1 does not belong to the set L accumulated into z. Otherwise, it belongs to z.

Figure 4: A Merkle tree with $2^2 = 4$ leaves



In general, each contributor knows ID_i and the corresponding witness w_i (including the corresponding key k_i) A contributor can later reveal ID*i* and w_i to prove he/she has contributed to z. Actually, this property of Merkle trees is widely used in constructing one-time signature schemes that achieve post-quantum security.

It is important to notice that it is impossible to revert the tree, i.e. recover any values ID_1, ID_2, ID_3 or ID_4 while knowing only its root (i.e. the accumulated pseudonym). If a subset of identifiers, ID_1 and ID_3 for instance, has been revealed, it is still not possible to recover the other identifiers ID_2 and ID_4. It is only possible to know that ID_2 and ID_4 have accumulated into z if and only if their corresponding witnesses w2 and w4 have been revealed.

[20] In a typical Merkle tree, the leaves are simple (i.e. unkeyed) hash values of some initial data. In the context of pseudonymisation, since a simple hash function is generally considered as a weak technique, it is preferable to employ a secret key to derive the leaves of the tree. Although in this report we refer to authentication codes, other approaches could also be considered – e.g. the leaves could be derived by encryption of the original identifiers.

Many designs of cryptographic accumulators have been proposed through the years. There are many designs now based on hash functions only (Nyberg, 2005), elliptic curves (Tartary, 2008) or bilinear mapping (Camenisch, Kohlweiss, & Soriente, An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials, 2009). They support different operations like dynamic modifications (addition or revocation) (Barić & Pfitzmann, 1997), (Badimtsi, Canetti, & Yakoubov, 2020).

An interesting observation is that the above properties of Merkle trees as pseudonymisation primitives could be preferable in cases that a user-generated pseudonym is needed - i.e. in cases that the Pseudonymisation Entity coincides with the individual. Indeed, an individual may produce a pseudonym based on a list of more than one identifiers of his/her so as: i) no identifier can be computed by any party having access to this pseudonym, ii) the individual is able to prove, at any time, that this pseudonym is bound to a specific identifier from this list (i.e. allowing individual's identification or authentication, depending on the context), without revealing the secret information or any other identifier from the list. This is also strongly related to the so-called pseudonyms with proof of ownership, as discussed in Section 3.5, Chapter 3.

Structures as the Merkle trees (which are binary trees) can be appropriately generalised. Indeed, any tree-structure starting with several types of personal data as its leaves and appropriately moving upwards via employing hashing operations preserves somehow the same properties as described above. Actually, the value at each internal node in this tree structure - which is the hash of a set of values - can be seen as an intermediate pseudonym, depending on one or more individual's attributes (i.e. being an accumulator of these values). The value z' of each intermediate pseudonym does not allow computation of the original personal data (i.e. pseudonymisation reversal), but allows for verification whether, for a given initial set of values, these values have accumulated into the pseudonym z' or not. Each intermediate pseudonym may be handled by a different entity[21]. A concrete practical example in this direction is presented in Chapter 5.

3.5 PSEUDONYMS WITH PROOF OF OWNERSHIP

As already discussed, pseudonymisation is a data protection technique which aims at protecting the identity of individuals by substituting their identifiers by pseudonyms. However, pseudonymisation may in certain cases interfere with the exercise of the rights that a data subject has on his/her data as defined in the GDPR (Articles 15 to 20) [22]. For example, in cases where the data controller does not have access to original identifiers but only to pseudonyms[23], then any request from a data subject to the data controller can be satisfied only if the data subject is able to prove that the pseudonym is related to his/her own identity; indeed, although the pseudonym is a type of an identifier in such a context, if its association with a specific data subject cannot be appropriately established, then the data controller cannot satisfy relevant data subject requests.

Therefore, in the cases as described above, it may be useful to create pseudonyms with proof of ownership. Such pseudonymisation techniques do exist (Montenegro & Castelluccia, 2004). The solution described verifies that the pseudonyms are hiding and binding. A pseudonym P is created by a data subject from a given identifier ID and later transferred to a data controller (Figure 5). The data controller must not be able to recover any information from the pseudonym P (hiding property). This property is important to avoid exposing the personal data of the data subject. At the same time, it must not be possible to find another identifier ID'≠ ID that is

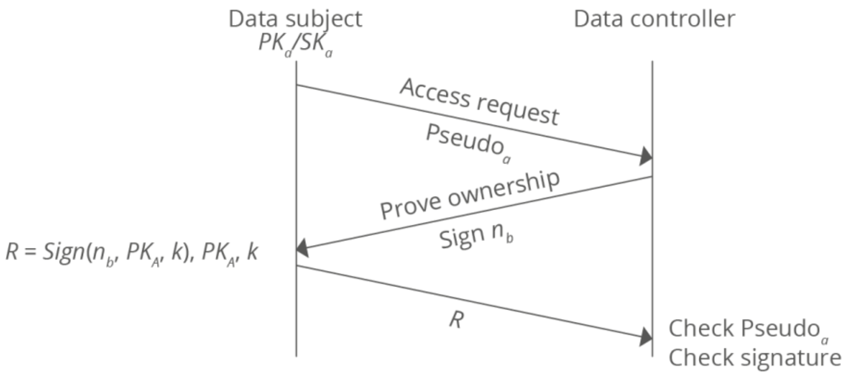
[21] Again, as in the case of chaining, an appropriate joint controllership could be possibly established, assigning relevant responsibilities (vis-a-vis to the data processors).
[22] Recalling the Article 11 of the GDPR, if the purposes for which a controller processes personal data do not require the identification of a data subject by the controller, the controller shall not be obliged to maintain, acquire or process additional information in order to identify the data subject. In such cases Articles 15 to 20 shall not apply, except where the data subject, for the purpose of exercising his or her rights under those Articles, provides additional information enabling his or her identification.
[23] This may be a requirement from a data protection perspective – i.e. if the data controller does not need to process direct identification information for the purposes of processing.

associated to P. This is the binding property. This property is needed to avoid any ambiguity on the identity of the data subject associated with a pseudonym, since otherwise it would be impossible to differ between two data subjects. It prevents impersonation attack when a right is exercised on the data. These two properties, hiding and binding, can be achieved by cryptographic commitment scheme. In (Montenegro & Castelluccia, 2004), the authors have chosen a commitment scheme based on message authentication codes. They have also considered that the identifier is a public key from an asymmetric encryption scheme. When the data subject needs to exercise his/her rights (an access request for instance), he/she needs to succeed a challenge/response protocol and to open the commitment. The data controller asks the subject to sign challenge n_b using its private key SK_a. The data subject signs the challenge and provides all the values needed to let the data controller verify the pseudonym: it includes, apart from the signature R, the public key PK_a and the value k. To verify that the request is valid, the data controller must ensure that PK_a matches the pseudonym and that the signature is correct using PKa.

The data subject can later prove to the data controller that he/she owns the pseudonym. It allows the data controller to check if any request made by the data subject related to a specific pseudonym is legitimate or not.

Figure 5: Proof of ownership

(first, the subject creates Pseudo_a=MAC(ID,k) with ID the subject long-term identifier and k a secret key used once; then, the subject can prove ownership of the value)



In the scheme proposed in (Montenegro & Castelluccia, 2004), the data controller learns at the end the identifier of the data subject. It is possible to avoid this situation by using zero-knowledge proof during the challenge/response phase.

3.5.1 Zero-Knowledge Proof

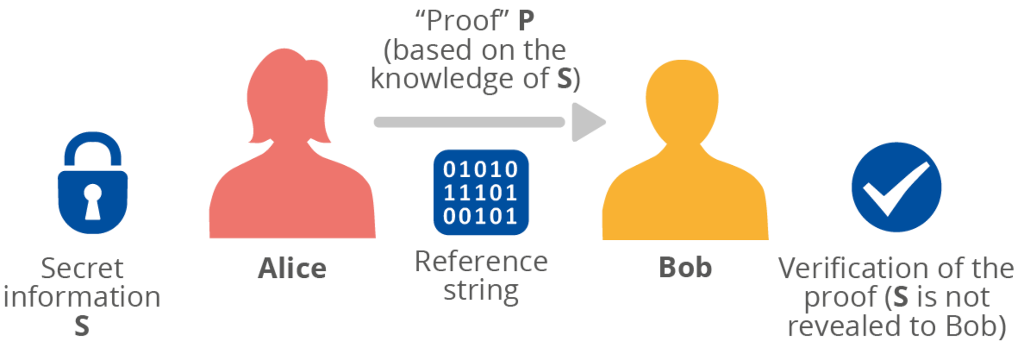
A known cryptographic primitive is the so-called Zero-Knowledge Proof (ZKP), which is actually, in the typical scenario, a term describing any protocol achieving the following: a party (prover) is able to prove to another party (verifier) that he/she is in the possession of a secret without revealing any information about the secret itself. ZKPs were first introduced for identity verification (Feige, Fiat, & Shamir, 1988), by providing the means to prove identity without revealing authentication information (but proving only that the correct authentication information is in the possesion of the prover). More generally, zero knowledge proofs involve proving that a statement is true, without revealing the details of the statement. Initially, this was achieved in an interactive way – i.e. a series of messages are needed to be exchanged between a prover and a verifier, a ZKP should satisfy the following properties (Goldwasser, Micali, & Rackoff, 1985).

- Completeness: In the case where the statement is correct, the honest prover will persuade the honest verifier that the fact corresponding to the statement is correct.

- Soundness: In the case where the statement is false, the adversarial prover cannot persuade the honest verifier that statement is correct, except with negligible probability.
- Zero-knowledge: In the case where the statement is correct, the verifier figures out nothing more than the fact that the statement is correct.

Non-interactive zero knowledge proofs were first studied in (Blum, Feldman, & Micali, 1984). Such a system utilises only a message sent to a verifier by the prover – which suits better with several applications than the classical interactive proofs. To achieve this, a common reference string model is introduced, meaning that a reference string shared between the prover and the verifier should be securely established, since only the prover and the verifier should have access to it (Figure 6, where Alice and Bob are the prover and the verifier respectively).

Figure 6: Zero-knowledge proof for pseudonymisation


"Proof" **P** (based on the knowledge of **S**)
01010
11101
00101
Secret information **S**
Alice
Reference string
Bob
Verification of the proof (**S** is not revealed to Bob)

In the context of pseudonymisation, if an individual associated with a pseudonym needs to prove that he/she is the owner of that pseudonym, without revealing his or her exact identity, a ZKP may provide the solution. As a concrete example of such a scenario, we refer to the usage of ZKP for (being called) anonymous transactions in cryptocurrencies. In these cases, zero-knowledge proofs are used to allow verification of the transactions without the verifiers (miners) knowing anything about the transactions' contents (and, by these means, the senders and the receivers of the transactions are concealed). This is the case, e.g. in the Zcash system[24], in which the sender of a transaction (being shielded) constructs a proof to show that: i) the input values sum to the output values for each shielded transfer, ii) he/she has the corresponding private keys, giving him/her the authority to spend, iii) the private spending keys are cryptographically linked to a signature over the whole transaction, in such a way that the transaction cannot be modified by a party who did not know these private keys.

3.6 SECURE MULTIPARTY COMPUTATION

In cryptography, a secure Multiparty Computation (MPC) protocol allows a set of parties to jointly compute a function of their secret inputs without revealing anything but only the output of the function. The first such protocol was introduced in the 1986 by Yao for the two-party case (Chi-Chih Yao, 1986), whereas one year later the multiparty case was first studied by (Goldreich, Micali, & Wigderson, 1987). Several applications of secure MPC protocols are known (not all of them associated with data pseudonymisation), including privacy-preserving auctions and private comparisons of lists.

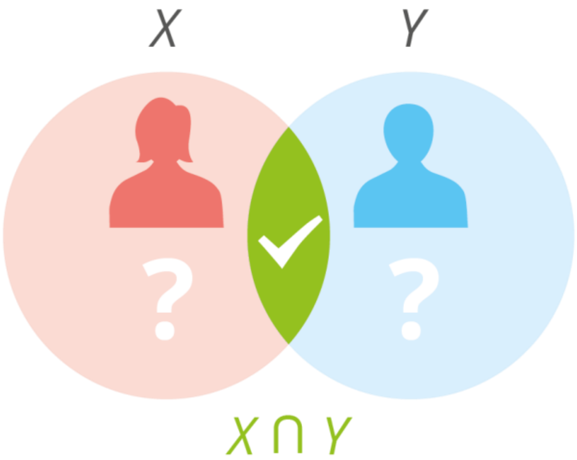
A specific case of secure MPC is the private set intersection protocol, in which two parties with private lists of values wish to find the intersection of the lists, without revealing anything apart from the elements in the intersection (Figure 7). Several MPC protocols exist for this problem. An idea that is described in (Kolesnikov, Kumaresan, Rosulek, & Trieu, 2016), which - as we will discuss next - is related to pseydonymisation, rests with the use of a so-called oblivious

[24] See https://z.cash/technology/zksnarks/

Pseudorandom Function (PRF) *F*; namely, this is a two-party protocol between a sender S and a receiver R so as, for a secret key *k* provided by S (and being hidden from R) and for any input *v* from R, R computes the value $F_k(v)$ (without learning the key *k*) and S does not learn the input *v*. In the aforementioned private set intersection protocol, the steps are as follows (based on the simplified description in (Lindell, 2020)), assuming that the first party has a private set ($x_1, x_2, . . . , x_n$) and the second party has a private set ($y_1, y_2, . . . , y_n$):

1. The first party chooses a key *k* for a PRF *F*.
2. The two parties execute *n* oblivious pseudorandom function evaluations: in the i-th execution, $1 \leq i \leq n$, the first party inputs *k* and the second party inputs y_i.
3. As a result, the second party learns $F_k(y_1), . . . , F_k(y_n)$, while the first party does not get any information on $y_1,\ldots,y_n$.
4. The first party, since he/she knows *k*, computes $F_k(x_1), \ldots, F_k(x_n)$ and sends the list to the second party.
5. The second party computes the intersection between the lists $(F_k(x_1), . . . , F_k(x_n))$ and $(F_k(y_1), . . . , F_k(y_n))$ and outputs all values y_j for which $F_k(y_j))$ is in the intersection; note that the party knows these values since he/she knows the association between y_j and $F_k(y_j)$, $1 \leq j \leq n$, whereas he/ she cannot find out any x_i, $1 \leq i \leq n$.

Figure 7: Private set intersection



In the above scenario, if the private lists consist of personal data, we actually have a pseudonymisation scheme (despite the fact that this is not explicitly stated in the literature). Therefore, despite the fact that, in general, a secure MPC protocol is not a pseudonymization primitive per se, in some cases – as in this scenario – it actually provides the means for sophisticated pseudonymisation schemes. For example, in the above case, let us assume that x_i, y_j are e-mail addresses of users, $1 \leq i, j \leq n$. Therefore, the outputs of the PRF function $F_k(x_i)$ and $F_k(y_j)$ are actually pseudonyms, where the key *k* is the pseudonymisation secret. It should be stressed though that, it is not a typical cryptographic scheme for pseudonymisation, since the second entity is able to compute the pseudonyms corresponding to his/her list without having access to the pseudonymisation secret.

Such techniques for private set intersection may be the proper solutions in terms of personal data protection requirements in several cases which necessitate comparison of two different lists from two different data controllers without revealing anything else than their common entries. For example, it could be applied in case that two health insurance companies wish to ensure that no one has taken out the same insurance with both of them (Hazay & Lindell, 2008). It could also be applied for advertising purposes (i.e. measuring ad conversion rates by comparing the list of people who have seen an ad with those who have completed a transaction, where these lists are held by the advertiser and by merchants, respectively (Pinkas, Schneider, & Zohner, 2019)).

There are several different oblivious PRFs, with different design characteristics (e.g. based on either asymmetric or symmetric cryptographic operations, hash function, bitwise operations etc). The main issue in such protocols is their performance, which is evaluated by means of the computation and (protocol) communication cost. Such protocols are in general computationally slower than naïve approaches which provide solutions to the same problem but with weaker pseudonymisation (i.e. through hashing the datasets and comparison of the hashed lists). However, implementations and executions of secure MPC protocols are practical; the approach in (Kolesnikov, Kumaresan, Rosulek, & Trieu, 2016) performs private computation of the intersection of two million-size sets in about 4 seconds. A new approach has been recently presented in (Chase & Miao, 2020), achieving balance between communication and computation costs.

3.7 SECRET SHARING SCHEMES

Secret sharing schemes can be seen as specific instances of secure Multiparty Computation (MPC) protocols. More precisely, secret sharing schemes are well known cryptographic techniques, aiming to appropriately split a secret information D into n parts D_1, D_2, . . ., D_n so as to ensure the following:

- Knowledge of k (or more) of D1, D2, . . ., Dn allow to compute D (where k is a design parameter)
- Knowledge of k − 1 (or fewer) of D1, D2, . . ., Dn is not sufficient for the computation of D.

Such schemes are also known as (k, n) threshold schemes. The most famous secret-sharing scheme has been proposed by Shamir and, as it is stated in his work (Shamir, 1979), this is an approach to securely manage a secret cryptographic key. Indeed, storing the key in a single, well-guarded location is unreliable in terms of single misfortune or corruption, whereas storing multiple copies of the key at different locations increases the risk of security breaches. Instead, by using a (k, n) threshold scheme with n = 2k - 1, a robust key management scheme is derived: we can recover the original key even if almost the half (k -1) of the n pieces are destroyed, whilst at the same time an adversary cannot reconstruct the key even if any k -1 such segments are compromised. As Shamir states in (Shamir, 1979), "*threshold schemes are ideally suited to applications in which a group of mutually suspicious individuals with conflicting interests must cooperate (...) By properly choosing the k and n parameters we can give any sufficiently large majority the authority to take some action while giving any sufficiently large minority the power to block it"*.

A secret sharing scheme can be also used to split an identifier into distinct segments (i.e. pseudonyms in our context), one for each different recipient, so as to ensure that pseudonymisation reversal is feasible only under specific prerequisites. More precisely, let us assume that the Pseudonymisation Entity substitutes - through a mapping procedure - the user's identifier[25] by carefully chosen pseudonyms. Each of these pseudonyms is irreversible (i.e. its recipient cannot compute the original identifier), under the assumption that the pseudonymisation mapping remains secret. Moreover, the unlinkability property is ensured since all these pseudonyms are different. However, exploiting the properties of the secret sharing scheme, these pseudonyms can be used for reidentification later on, only if a well-determined number of the recipients (each carrying a different pseudonym for the same entity) agree to exchange their pseudonyms. Such a property may be desirable in several cases. For example, this approach is proposed in (Biskup & Flegel, 2000) with the aim to pseudonymise

[25] It should be pointed out that, actually, the notion of identifier in this case could be quite general. In the extreme case, even all the personal data in a dataset can be used as an input to a secret sharing scheme.

auditing log files of a system so as to ensure that pseudonymisation reversal will occur only if a suspicious activity - as it is defined according to a specific threshold - is present: only in such a scenario, the parties storing the pseudonyms (i.e. the log events analysers) are able to derive the original identifier by exchanging the corresponding values of the pseudonyms. Otherwise, no identification of users from their relevant log data is possible. More recently, another secret sharing scheme has been used in (Li, Pei, Liao, Sun, & Xu, 2019) to protect vehicular identity privacy in a Vehicular Ad Hoc Network (VANET); VANETs constitute a main application field for the Internet-of-Things (IoT), which generally poses several personal data protection challenges (a recent survey on the types of pseudonyms that are being proposed for IoT applications is given in (Akil, Islami, Fischer-Hübner, Martucci, & Zuccato, 2020).

Due to the inherent properties of the secret sharing schemes, we may conclude that the pseudonymisation secret is somehow shared between several entities[26]. Actually, in the above scenario, each pseudonym plays also, in a way, the role of a share of the pseudonymisation secret: indeed, in a (k, n) scheme, combination of any k such shares (but no less) suffices to extract the original identifier. However, such an idea of sharing the pseudonymisation secret can be also applied in other pseudonymisation techniques. For example, let us consider a pseudonymisation technique whose pseudonymisation secret is a secret key. The secure storage of this key may be similarly based on a secret share amongst several entities, as described above. Recalling that the pseudonymisation secret is strongly related to the additional information that is needed to attribute the pseudonymous data to a specific data subject, as well as that the GDPR explicitly states under Article 4(5) that such additional information should be kept separately and be subject to technical and organisational measures, it becomes evident that secret sharing schemes may provide the means to achieve this goal. Thus, the complex task of selecting the optimal secret sharing parameters and settings for securely storing the secret data shares, while meeting all of end user’s requirements and other restrictions, becomes of high importance (such challenges and restrictions in secret sharing schemes are being discussed in (Framner, Fischer-Hübner, Lorünser, Alaqra, & Pettersson, 2019).

3.8 CONCLUSION

In this Chapter we reviewed a number of advanced techniques that can provide more sophisticated pseudonymsation solutions in real world scenarios. It must be emphasised, however, that advanced techniques rely always on cryptographic primitives and are not recommended for simple pseudonymisation cases where basic techniques (as those described in Chapter 2) would normally suffice. A case-by-case approach should be followed to select the best possible technique for the scenario in question. In the next two Chapters we explore such options in the areas of healthcare and cybersecurity.

[26] Again, this could possibly be a case of joint controllership in certain scenarios.

4. PSEUDONYMISATION USE CASES IN HEALTHCARE

In the previous Chapters we explored basic and advanced pseudonymisation techniques that can be applied in different contexts. When it comes to real-world application of pseudonymisation, combination of different approaches can provide unique advantages, allowing for utility while preserving protection on a high level. At the same time, such solutions require careful implementation so as to maintain those beneficial effects throughout the lifecycle of the application.

This Chapter focuses on the healthcare domain, more precisely, the collection of medical data from patients, and the processing of such data at hospitals and subsequent medical research institutions. Starting from an example scenario, we first demonstrate how pseudonymisation can be best employed in different use cases. We then further analyse how the scenario could evolve with the use of the data custodian model, in cases where a Trusted Third Party is needed to safeguard the pseudonymisation process.

4.1 EXAMPLE SCENARIO

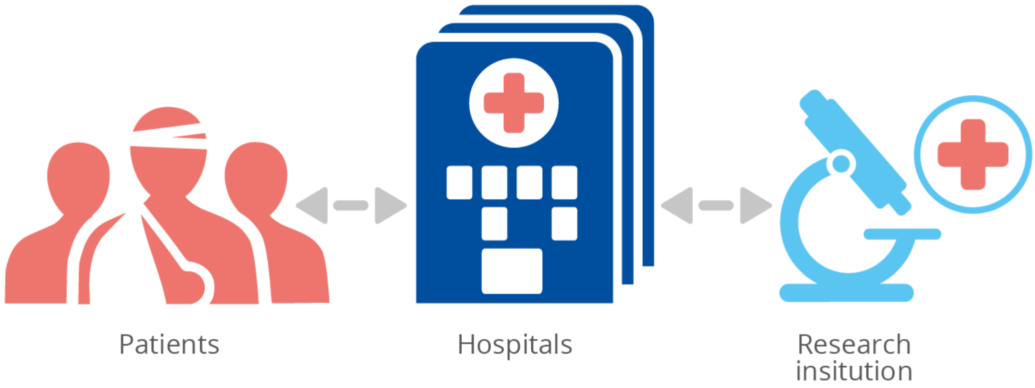
Medical records of patients serve multiple purposes. They are used to inform doctors about the relevant medical history of a person, they are used for healthcare insurance organisations to calculate financial aspects of disease treatments, or they are used by other medical practitioners (e.g. in other countries) if those need to rapidly learn about the most relevant medical conditions of emergency patients. Research organisations may e.g. have an interest in statistical data on diagnoses and medications. Beyond these, there obviously are many other interest groups for medical data of patients.

The main issue with all of these different purposes is that each purpose only needs access to certain parts of a medical data record, but not necessarily to the full record at once. A doctor needs access mostly to the relevant medical data, but not necessarily to the insurance-related financial aspects. A healthcare insurance company should best-possibly not have access to many details about the exact diagnosis nor medical history, as long as it is not relevant to payments. Medical research organisations may only get access to the binary information on whether a patient is treated with a certain medication or not, potentially in combination with the diagnosis, but certainly neither the person’s identifiers (like real name) nor exact medical history nor financial data.

In this field, pseudonymisation can provide protection of sensitive information of patients against – accidental or intentional – access by any of those parties. The act of pseudonymisation helps in separating the medical facts from the identity of the patient, potentially allowing medical research to be performed on pseudonymised data.

In light of this domain of application of pseudonymisation, we focus on a hypothetical data exchange environment for such medical data, in order to illustrate a way where pseudonymisation may protect the privacy of patients while enabling processing of medical data for specific valid purposes.

Figure 8: Example scenario for medical research



In this setting, we will initially differentiate between three different actors: the *patients*, whose medical data is stored and processed, the *hospitals* that store the patient's data, and the data processing organisation, which in our case will be a single medical *research institution* (Figure 8). Obviously, the research institution should not learn about the exact medical conditions of an individual patient, however, there might be an interest to perform statistical analyses on the correlations of medical treatments and disease conditions (like symptoms, durations, emergency states, etc.). This causes a dilemma to the hospitals. When handing out the patient's medical record in plain text to the research institution, the research institution can perform their statistical analysis, but at the same time can easily identify individual patients and their medical history. On the other hand, if a hospital does not hand over the data to the research institution, statistical analyses on correlations between medications and symptoms becomes impossible, limiting the ability to do valid, large-scale research on medical conditions.

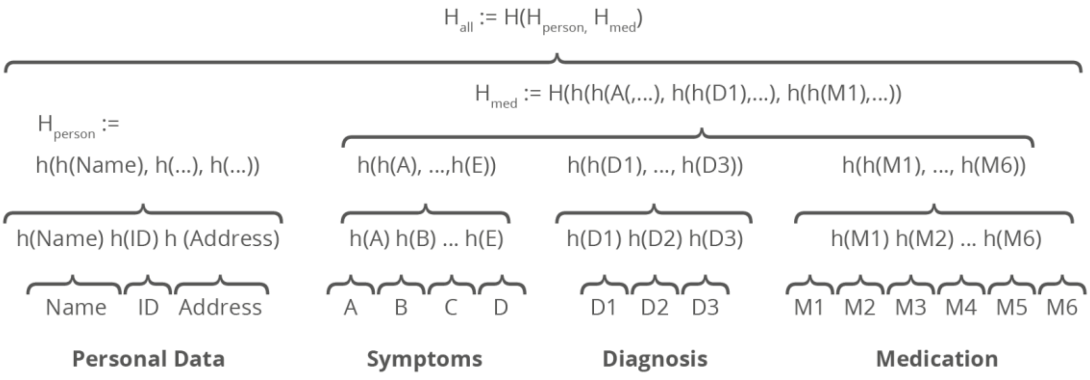
Hence, it becomes necessary to find a way to reveal some part of the medical data to the research institution in such a way that statistical analysis of the data still is feasible, while at the same time, linkability of individual medical data records to the correlated patient is prevented to the best extent possible. In other words, it becomes necessary to find a way to process medical data without revealing that very same medical data. Examples for this scenario are described next.

4.2 PSEUDONYMISATION USE CASES

4.2.1 Patient record comparison use-case

Assume that two hospitals, considered merely as data storage locations in this example, need to decide whether they both share the same, up-to-date version of a certain patient's medical record. Due to delays in data digitalisation or transmission from the involved hospitals, a situation may arise in which it is unclear whether the patient's data record in the different data storage locations is complete and consistent among all storage entities (i.e. hospitals and their potential IT subcontractors). Hence, it becomes necessary to run a protocol that compares patient-specific data records, consisting of personal data (like name, health insurance identifier, home address) and medical data of that patient (like symptoms, diagnosis, treatments and medications). Without the use of pseudonymisation techniques, this would require one hospital to send all that information to the second hospital, which then does the comparison of all data fields of the patient's medical record. Obviously, this approach reveals all medical and personal data of the patient concerned to the second hospital – irrespective of whether that patient record exists at that hospital's data storage or not.

Figure 9: Example of a tree-based pseudonymisation approach for medical patient records



This disclosure of personal data could easily be avoided by utilising a hash tree scheme for pseudonymisation of the patient's medical record prior to comparison (see Section 3.4, Chapter 3). As illustrated in Figure 9, this approach requires the sending hospital to pseudonymise each single entry of the medical data record by means of an appropriate pseudonymisation function. For the sake of simplicity, we will refer to this pseudonymisation function as *hashing*, however, applying just a plain cryptographic hash function to the data values in consideration is typically not sufficient for achieving a reasonable level of protection, as also explained in Section 3.4, Chapter 3 (see also further details at (ENISA, 2019 - 2)).

In this approach, each patient-specific data entry (like name, health insurance ID, address, etc.), as well as each associated medical data entry (like symptoms, diagnoses, medications) is hashed individually, resulting in a large set of hash values – the pseudonyms. Here, it must be noted that these pseudonyms do not necessarily all independently represent personal data, as e.g. the hash value of a disease name by itself is not linked to any human individual. However, in the context of a personal health record, the existence or absence of such a hash value obviously becomes personal information of the patient in consideration. Also, for this scheme to work, it is of critical importance to make sure that identical symptoms, diagnoses, and medications are represented in a textually identical format, as even a slight change in syntax (like different capitalization of words) would result in different hash values, hence reflecting different semantics.

Without the ability to uncover the plaintext version of these pseudonyms, it is hardly possible to re-identify the individual data entries themselves in this set (given that a secure pseudonymisation function was utilised for hashing). However, if two patients share the same set of symptoms, they would still share some identical (or at least linkable) pseudonyms: those of the specific symptoms shared by both patients (but only if a deterministic or document-randomised pseudonymisation function is chosen (ENISA, 2019 - 2)). This would allow the data-receiving hospital to easily uncover these pseudonyms and learn the true symptoms of a patient. The same weakness holds true for diagnosis and medication data. Hence, in order to keep the receiving hospital from performing such easy discrimination attacks, we add a second level of pseudonymisation to the set of pseudonyms created. Therein, we create new pseudonyms (named level-2 pseudonyms) resulting from the initial pseudonyms (level-1 pseudonyms). Those level-2 pseudonyms are created by taking all level-1 pseudonyms of a certain class of data items (like, all symptom pseudonyms or all medication pseudonyms of the patient record concerned) as plaintext. In order to guarantee comparability of level-2 pseudonyms later-on, it is necessary to sort all level-1 pseudonyms, e.g. by alphabet, concatenate them, and utilise the resulting string of characters as plaintext input to the level-2 pseudonymisation function. If the level-2 pseudonyms of two different patient records are identical, this implies that the whole correlated set of level-1 pseudonyms was identical, implying that all the symptoms/medications/diagnoses/patient data records were identical.

To utilise this concept of tree-based pseudonymisation further, we can now create level-3 pseudonyms over all level-2 pseudonyms of a patient record, e.g. creating a pseudonym representing all medical data in such record, i.e. spanning over all symptoms, diagnoses, and medications. Depending on the level of granularity given in the specific patient data record format, other levels of pseudonymisation can be applied in a similar manner, until the single, top-most pseudonym over all lower-level pseudonyms is created (labelled H_{all} in Figure 9).

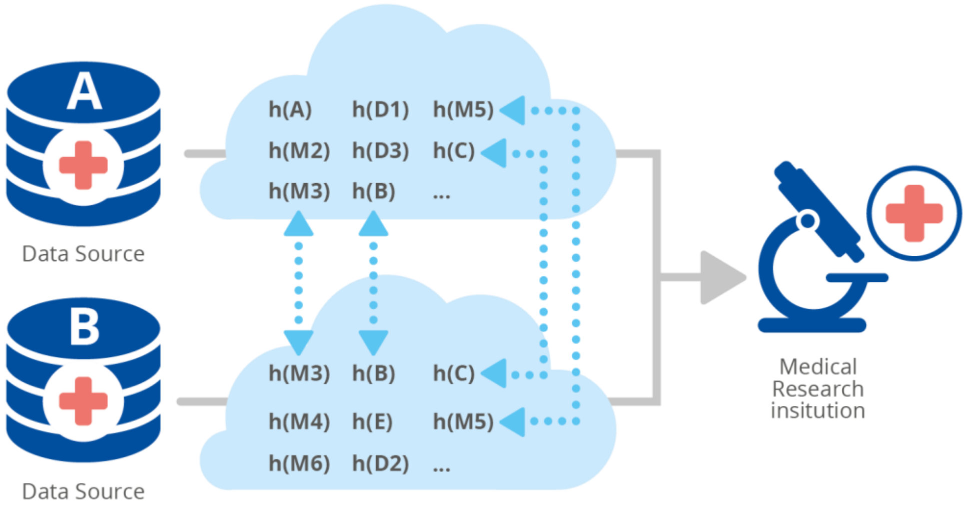
Once this pseudonymisation is completed, the task of comparison of two patient data records is reduced to the comparison of their top-level pseudonyms only. If those are identical, all data in the two patient records are identical as well. If they differ, only the set of pseudonyms from the next-lower level are sent to the second hospital, revealing as little as possible additional details on the patient's exact medical conditions. If a pseudonym is found to be identical, all lower-level pseudonyms of that subset of data items must be identical as well. This way, the comparison of two medical data records can easily be reduced to comparison of the particular levels of pseudonyms, allowing to easily identify the exact differences and their data items by running an appropriate pseudonym exchange protocol.

4.2.2 Medical research institution use-case

Beyond sheer comparison of patient data records, another common utilisation of such medical data is the detection of correlations and patterns among symptoms and medications, attempting to identify new ways of diagnosis or treatment for certain diseases. This task is typically performed not at the hospitals themselves, but is outsourced to dedicated medical research institutions that analyse the data from many different sources. Hence, at these research institutions, the data of multiple patients must be analysed for common patterns of medical relevance.

For this type of analysis, the identity of the patients is not directly relevant, and there normally is no need for the research institution to learn about the true identity of any of the patients whose data gets analysed at the research institution. An exception to this assumption occurs when a patient's data itself might reveal a new diagnosis, e.g. due to having the same patterns of symptoms and medications as all others that share the new diagnosis. In such cases, it becomes necessary to re-identify the particular patient in order to notify him/her (and his/her doctors) of the new diagnosis.

Figure 10: Medical research institution use case



In this setting, the pseudonymisation scheme described in Section 4.2.1 (Chapter 4) can unfold its ideal potential. The task of detecting correlations and statistical patterns in symptoms and medications can easily be performed via comparison of level-1 pseudonyms (as illustrated in Figure 10), without even ever revealing the true value of the underlying symptom or medication. Thus, the research institution can easily work on level-1 pseudonyms only, never learning any real symptom or medication itself (assuming a strong pseudonymisation scheme robust to the common dictionary and brute force attacks is utilised, (ENISA, 2019 - 2) – see also Chapter 2 for an overview). This way, the identity and personal medical record of the patients is largely protected, yet the intended utility of data analytics over symptoms and medications remains feasible. A drawback of this approach consists in the limitation of the utility scope: this pseudonymization scheme does not automatically support other queries than the pattern correlation presented here.

Let us next consider the scenario where a patient needs to be re-identified to notify him/her of a newly discovered diagnosis (or other relevant medical assumption) uncovered by the research institution. Obviously, the research institution cannot and should not be able to contact the patient directly, so as to protect his/her identity. Hence, it becomes necessary for the research institution to contact the data-storing hospital for that patient, and trigger a patient notification performed by that hospital.

For handling such a case, the research institution only needs to store an identifier for the hospital it received the data from, the patient-related personal pseudonym (H_{person} from Figure 9), as well as the set of level-1 pseudonyms received for that patient from that hospital. In case of detection of a relevant medical condition that requires notification, this H_{person} pseudonym is sent to the hospital whose patient the data came from. Then, that hospital can uncover the H_{person} pseudonym locally, re-identify the patient concerned, and perform the notification and other relevant tasks as necessary. This way, the research institution itself never learns the identity of the patient, yet delivers a new diagnosis to that patient.

4.2.3 Distributed storage use-case

Assuming medical data to be stored not just in one single database, but being copied over a set of different databases operated by different organisations for reasons of security and availability, the pseudonymisation scheme described above can even be improved in terms of protection by applying appropriate secret-sharing techniques (see Section 3.7, Chapter 3).

Assume a medical patient record to be stored in the databases of several different hospitals. If done in plain text, this would enable each single hospital to investigate the full medical data record of a patient, along with personal data like name and address – an obvious risk to privacy and data protection of those patient's data. The common countermeasure to this is to apply a level of encryption on specific parts/identifiers of the patient record (or the whole record in certain cases) prior to sharing it with other hospitals[27]. This way, the data record itself stays available, but access to identifiers is only given to those organisations that also have access to the secret key utilised in the encryption task. Assuming that one secret key is utilised per patient, that key acts as the pseudonymisation secret utilisable to uncover the pseudonym, which would be the encrypted patient identifiers from the patient record itself. In such a setting, the risk of re-identification is mostly reduced to the risk of malicious utilisation of that very pseudonymisation secret.

In this setting, even the secret key utilised as pseudonymisation secret can be protected better than plain storage in a secret space, by utilising the secret sharing scheme described in Chapter 3.7 on the encryption key as its identifier. One approach could be to split the encryption key into a set of secret shares, which then are copied to the very same hospitals storing also the (potentially also encrypted) patient record itself. Then, in case a re-identification of the

[27] Note the difference between pseudonymisation and encryption, as highlighted in (ENISA, 2019 - 2).

patient becomes necessary, a sufficiently large set of participating hospitals must provide their secret shares accordingly. If this happens, the secret key can be restored, the patient identifiers from the patient record can be decrypted, and thus the patient can be re-identified.

An interesting aspect of this scheme consists in the encryption approach taken. If a standard symmetric encryption scheme is utilised, where encryption and decryption both use the very same secret key, that secret key becomes the secret to share. In this case, neither creating nor uncovering a pseudonym works without access to that pseudonymisation secret.

If, however, asymmetric encryption is utilised, there are two different keys: private key and public key. In that case, the private key is necessary for uncovering the pseudonyms created, hence that one must be shared by means of a secret sharing scheme. The public key, however, can easily be utilised for creating new pseudonyms (i.e. encrypting new patient record data) without the need for resolving any shared secret. Each hospital itself may decide to add (personal) data to a patient's medical record, simply by encrypting an addendum to the original record, and copying that addendum to all hospitals that share the particular patient record. Still, in order to uncover the identity behind the encrypted addendum, a collaboration of a set of hospitals along the constraints of the secret sharing scheme is necessary (in order to uncover the private key and decrypt both original patient record identifiers and addendum).

4.3 ADVANCED PSEUDONYMISATION SCENARIO: THE DATA CUSTODIANSHIP

While the selected techniques described in Section 4.2 of this Chapter are of high relevance, the roles of the parties involved (and especially the party that assumes the role of the Pseudonymisation Entity) are central as to the overall effectiveness of the approach. To this end, in this Section we explore an advanced pseudonymisation scenario that could be of great use in healthcare and beyond, that of the *data custodianship*.

4.3.1 Notion of data custodianship

As discussed earlier, the Pseudonymisation Entity (PE) is the entity responsible of processing identifiers into pseudonyms using the pseudonymisation function. It can be a data controller, a data processor (performing pseudonymisation on behalf of a controller), a Trusted Third Party or a data subject, depending on the pseudonymisation scenario (see also Chapter 2 for an overview)[28].

This definition clarifies that there may be multiple parties involved in a pseudonymisation scenario, specifically for using the pseudonymisation function. Dedicating the processing to a specific party allows for some advantages, in particular in cases where only the pseudonymised data, but not the identifiers, should be accessible for other parties.

The concept of establishing trusted intermediaries for supporting confidentiality and protection of identifying data has a long tradition. For instance, a health professional will have to maintain professional secrecy regarding the sensitive data of his/her patients, but it may be possible to pseudonymise the data and make them available for researchers. For this purpose, *data custodians* can be employed in a scenario of pseudonymising data and providing access under predefined conditions.

While there is no precise definition of the term data custodianship, it may comprise various functionalities in a pseudonymisation scenario, depending on its involvement in the pseudonymisation process, the data handling and the provision of access to other parties.

[28] See also Article 4 (5) GDPR – the objective is "that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organisational measures to ensure that the personal data are not attributed to an identified or identifiable natural person".

Currently, data custodianship (or similar concepts such as data trustees or intermediaries) is being discussed on the one hand for allowing data access under specific conditions to researchers or companies in an interconnected data ecosystem and on the other hand for shielding data against unwanted or unlawful access, e.g. from jurisdictions that cannot guarantee adequate levels of data protection.

In general, involving a data custodian entity can be regarded as an organisational safeguard for handling the data. In some scenarios, the data custodian has to be a legally, spatially and personally autonomous and independent party with legal accountability; in other scenarios, the data custodian may be implemented as a technical service in the architecture (Pommerening, et al., 2006).

Looking at the role of a data custodian in the pseudonymisation process, it may fulfil the role of assigning pseudonyms by applying the pseudonymisation function to the identifying data. In this case, the custodian does not necessarily have access to the comprehensive data records; it may only hold a pseudonymisation mapping table. The custodian would be involved in the process of recovery, i.e. inverting the pseudonymisation function. The data custodian would have to maintain the informational separation of powers (Bundesministerium des Innern, 2017), i.e. to provide a reliable service in applying the pseudonymisation function, keeping the necessary data on the mapping between identifying data and pseudonyms in a secure way, and possibly – under predefined conditions – conducting the recovery.

In other scenarios, the data custodian may focus on the storage of pseudonymised data, as being provided by the data controller, and facilitating access after the authorised parties have proven legitimacy and fulfilment of constraints. In this case, the focus is on a fair way of sharing data[29]. A data custodian may collect pseudonymised data from various data controllers and provide larger repositories to which access may be provided. The reliable service of the data custodian would encompass availability and integrity of the stored data as well as checking the authorisation before allowing access.

For enhanced control on how the data is processed, the data custodian may not directly provide access, but process the pseudonymised data according to the specification of the user (Information Commissioner's Office (ICO), 2012). Thus, the repository would not be directly accessible from other parties, but these could provide their processing operations (e.g. code) to the data custodian which would send back the results. This may also allow for checks for a potential identifiability of the data that otherwise would have been overlooked.

Added functionality of a data custodian could comprise the provision of synthetic data that is not directly related to the identifying data or the pseudonymised data, but still show sufficient structural equivalence with the original data set or share essential properties or patterns of those data. Synthetic data is being used instead of real data as training data for algorithms or for validating mathematical models.

In Section 4.1 of this Chapter, we assumed a certain hospital (e.g. the standard hospital of the city a patient lives in) to store the patient record and manage all access requests including those from researchers who should work with pseudonymised data only. This is a common scenario of today, but there are alternatives. Some patients might not trust a certain hospital to also perform well as an IT provider for managing data access to researchers or other parties. Some patients might prefer to store their own personal medical data on their own, personal storage devices, and reveal relevant pseudonymised information to research institutions only under their own control. Some patients might prefer to have their (pseudonymised) data stored by a different hospital than their own city's hospital (e.g. if relatives work at the latter). Some patients might prefer to choose an independent, trustworthy third party to store and manage

[29] See http://www.rfii.de/download/the-data-quality-challenge-february-2020/

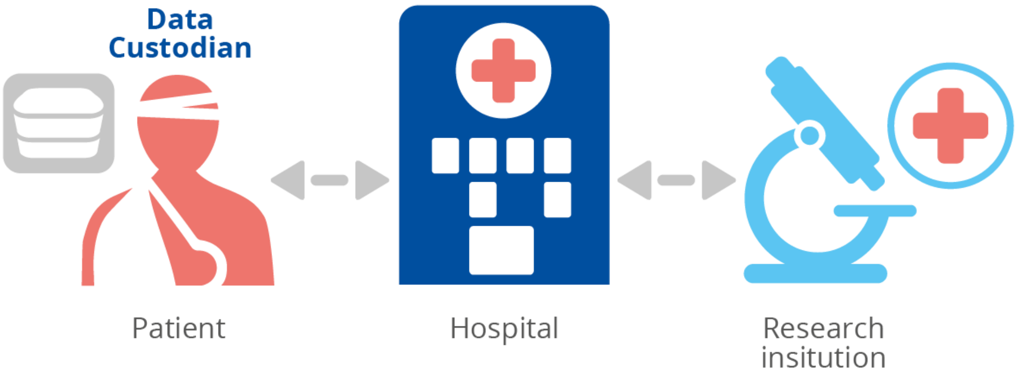
their pseudonymised medical data, be it a private company or a dedicated government agency. And finally, some patients might decide not to trust any single party, but to have their data being stored in a shared yet secure distributed data storage, spread among many different parties and organisations.

For each of the cases above, there exist technical solutions to enable such type of data storage and access management, but they all require different technical architectures to be implemented. In the following, different types of data custodians are illustrated in the hospital scenario, assuming that pseudonymisation is performed to allow access for researchers (EDPS, 2020).

4.3.2 Personal Information Management System (PIMS) as data custodian

The *personal information management system (PIMS)* concept is characterised as "new technologies and ecosystems which aim to empower individuals to control the collection and sharing of their personal data" (EDPS, 2016). The PIMS concept can be applied to our scenario: All medical data is stored on devices of and in the domain of control of the patient, i.e. of the human individual the medical data relates to (Figure 11). While storage of data alone may e.g. be realised in the shape of a smart card the patients carry with them when contacting a doctor, such as a healthcare insurance ID card, PIMS provide for more advanced functionality. In this case, all control on data access (including the pseudonymised data) is managed by the patients themselves: if they provide access via their PIMS, they provide access to the data. If the technical architecture allows it, access to that medical data can even be restricted to parts of the total data records, or to pseudonymised data sets only.

Figure 11: Scenario with PIMS as data custodian

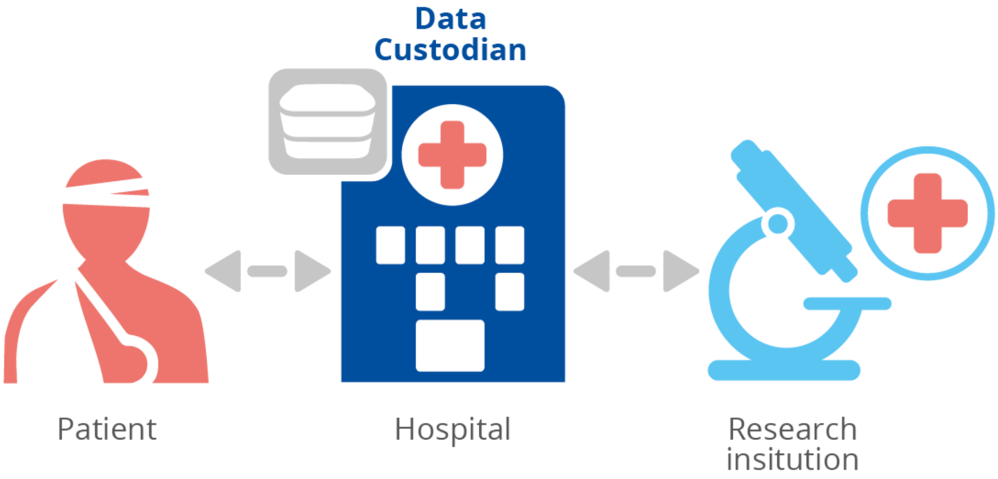


While for regular treatment of patients and its necessary documentation it often would not be sufficient for health professionals to fully rely on the data that the patient is willing to disclose in the specific situation, PIMS may be used for consent-based access to pseudonymised data for research. For example, the patient may be reached via a communication address (such as a specific e-mail address) and asked for consent to provide access to the (pseudonymised) data for a research study. The PIMS could also be implemented as an app on a smartphone for giving (or withdrawing) consent and facilitating the access to data stored at the patient's side or at some other location trusted by the patient. The PIMS (as well as the provider of the system where the patient data is stored) would function as data custodian.

4.3.3 Data custodian as a part of the hospital

The data custodian could also be acting as part of or directly on behalf of the hospital so that the data is stored at the organisation that creates the data, e.g. the hospital of the patient where the medical data is metered and entered into the medical record. Given that this organisation is to be trusted by the patient anyways, it may be a reasonable data storage option also for pseudonymised data.

Figure 12: Scenario with the hospital as data custodian

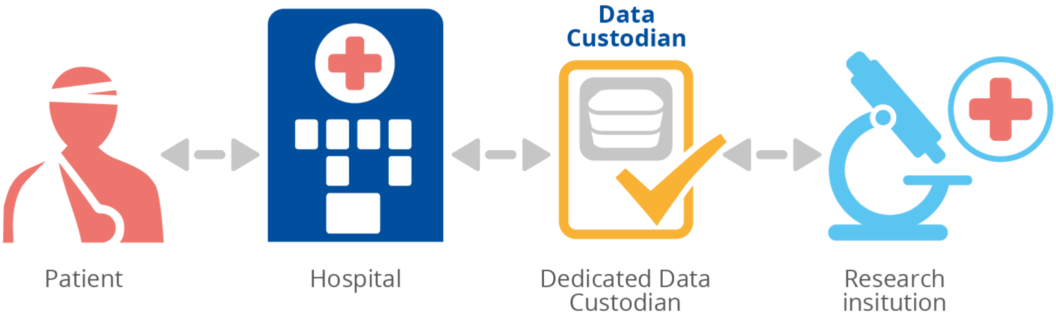


However, if a patient changes hospital (e.g. on holidays or when moving to a new city), the data source concept may have issues with portability of data – also for researchers who would need to use pseudonymised data sets over a longer time span. Determining the correct medical institutions to ask for patient data may be time-consuming and error-prone.

4.3.4 Data custodian as an independent organisation

The typical data custodian model implies data storage at an external, trustworthy organisation that stores and manages the (pseudonymised) patient data and potentially participates in the pseudonymisation process. The data custodian processes personal data and may keep, or be able to derive, identifying data. Whenever any other organisation demands access to the medical data of a patient from the data custodian, the data custodian validates the legitimacy of the request against the conditions and constraints demanded by the corresponding patient, and serves the data access (and potential pseudonymisation actions) in the name and interest of the patient. Similar to a notary, a data custodian therefore serves also as a representative of the patient and as a fiduciary in case of conflicts regarding access to a patient's medical data. In this approach, an essential requirement is that the data custodian is trusted by the patients. Therefore, it is essential for a patient to decide upon their data custodian themselves, based on their individual trust relationships, rather than being forced to trust a certain data custodian just because it also happens to be the local hospital.

Figure 13: Scenario with a dedicated, independent data custodian organisation



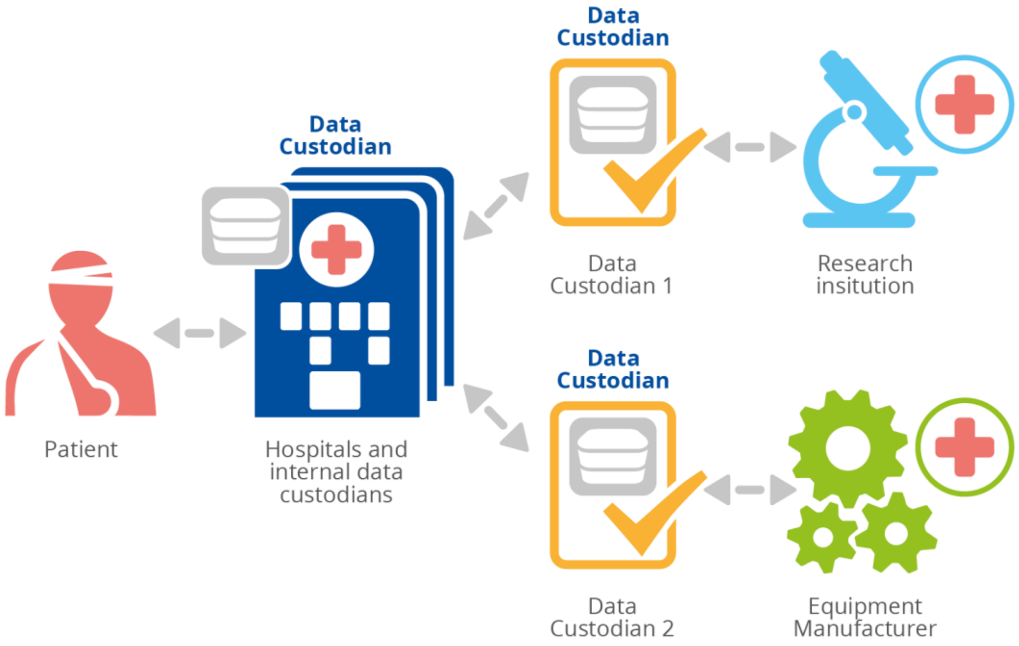
Here, a problem rises if a patient does not want to trust a single organisation, be it a hospital or a dedicated organisation, to honestly and securely manage their personal data. In that case, it becomes necessary to eliminate the threat of a single data custodian in favour of a set of semi-trustworthy data custodians that collaborate – an interconnected data custodian network.

4.3.5 Interconnected data custodian network

In general, an interconnected data custodian network approach reflects a paradigm in which there exists no single data custodian for a patient, but instead a network of entities, be it private persons, consortia of hospitals, or dedicated companies, that jointly and collaboratively store and manage the personal data of patients. Such approaches have emerged already in many other application domains, such as blockchain technology for shared management of financial data, peer-to-peer networks for shared storage of arbitrary data, or the Onion Routing services such as TOR[30], for collaborative – and thereby largely protected – access to Internet services. Each of these applications is realised as a distributed, de-centralised architecture whose trust is not maintained by a single central instance, but merely by the assumption that it is unlikely for a large fraction of participants to maliciously collaborate and violate the rules and conditions of the distributed architecture. Hence, in such de-centralised architectures, the trust demand is split among several entities, increasing the trust a patient might have in such a system.

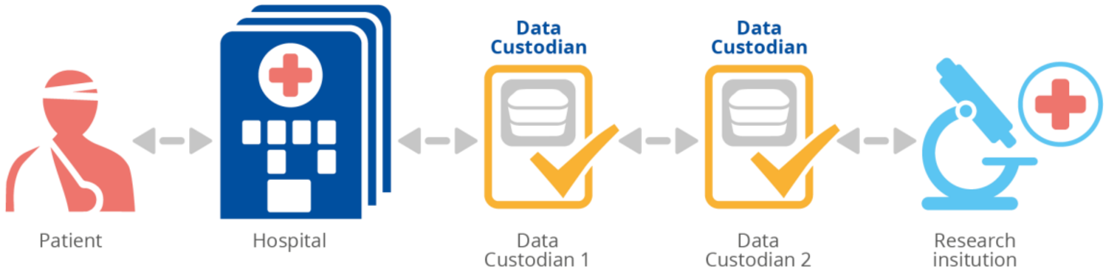
Depending on the scenario, different kinds of separation and splitting tasks among the interconnected data custodians are suitable. For instance, manufacturers of technical hospital equipment such as tomographs may need machine information for maintenance or optimisation that sometimes may contain personal information. This data may be handled by a different data custodian than pseudonymised health records for medical research.

Figure 14: Scenario with multiple in-house or dedicated data custodian organisations



[30] See: www.torproject.org/

Figure 15: Scenario with a chain of data custodian organisations



Also, the pseudonymisation process may be performed by integrating more than one data custodian. This is specifically proposed for biobanks that contain sensitive material and profit from a double pseudonymisation, so that not one entity alone can recover the identifying information.

5. PSEUDONYMISATION USE CASES IN CYBERSECURITY

While healthcare, as explored in Chapter 4, might seem quite an expected application area for data pseudonymisation, this need might not be as apparent in the cybersecurity sector, e.g. in malware or antivirus protection technologies. However, most modern cybersecurity technologies today no longer rely on static, signature-based protection, but rather depend on security telemetry analytics – such as correlating suspicious events that reveal the existence of an advanced threat, training Machine Learning systems to classify threats, establishing reputation-based protection, building behavioural threat models, etc. As such, cybersecurity technologies rely strongly on the processing of personal data. In this Chapter we discuss some of the cases where pseudonymisation could be utilised in this context, in order to provide for security analytics, while preserving privacy and data protection.

5.1 THE ROLE AND SCOPE OF SECURITY TELEMETRY

Many cybersecurity products traditionally relied on a number of static threat signatures (e.g. malware signatures) able to detect a threat on an endpoint, which often represents/belongs to a user. Although this tactic served as well for many years (and is still being used within some static protection engines), modern threats have evolved in ways that require a more sophisticated and scalable approach to keep up with the threat actors. This evolution is characterized by many elements, such as the speed with which malware variants are created and distributed, the many new threat vectors for delivering attacks (e.g. files, email, websites, mobile apps, malicious documents, etc.), and the need for quickly identifying new threats – before even having a chance to analyse them in depth within a lab environment.

To that end, most cybersecurity vendors have shifted their efforts towards behavioural and analytics-based protection mechanisms, making use of the opportunities offered by the latest developments in data analytics and machine learning. For instance, a URL (Uniform Resource Locator) reputation system can greatly improve the cybersecurity of users by warning them of malicious URLs while browsing the Internet. An oversimplified version of such a system could be a simple “blocklist” of confirmed bad URLs. In practice, however, this is far from sufficient – given the scale and fluidity of the Web. By analysing correlations between the URLs a user is visiting and the threats or infections encountered by that user, we are able to train a system capable of effectively protecting users from bad URLs, while also keeping up with the volatility of the threat landscape. This type of correlation requires a large corpus of field-collected data – often referred to as *security telemetry* – in order to perform the necessary correlations and train the model that is eventually deployed.

The collection of real-world security telemetry is vital to the effectiveness of modern analytics-based cyber defences. Such telemetry can be collected in ways that do not involve users (e.g. honeypot infrastructure), but in most cases the most reliable types of security telemetry is crowd-sourced – by analysing and collecting telemetry from the user end-points. As such, collecting security telemetry is a sensitive task that requires certain necessary steps, including user consent (i.e. the user agrees to participate in the telemetry collection program in order to improve the overall ability of the community to detect and protect against threats), clear statements for the types of telemetry collected and its use, as well as employing appropriate technical measures for the protection of users’ identities. These steps are especially important in some cases, where sensitive data are involved in the telemetry collection process. For instance, when creating a URL reputation system, it is necessary to collect and utilise telemetry that relates to users’ Web browsing behaviour – which may contain personal information. While in most cases, telemetry analytics would not require the identification of users, however, as we explain in the next Sections of this Chapter, there are still cases that this might be necessary in order to provide for sufficient security protection. These are the types of situations where it is imperative to employ data pseudonymisation.

5.2 A USE CASE ON REPUTATION SYSTEM TRAINING AND USER-TAILORED PROTECTION

Some of the most successful machine learning (ML) systems deployed today are using the “wisdom of the crowd” in order to achieve adequate coverage of vast population space, like URLs and downloaded files. To that end, Reputation Systems (RS) attempt to assign a reputation score to an entity (e.g. a URL or a file download candidate) by collecting and correlating telemetry related to the entity in question. For instance, if downloading a particular file has been associated with a number of suspicious or malicious outcomes (e.g. computers getting infected), or has been associated with poor hygiene (e.g. file is overwhelmingly prevalent on infected computers), then a reputation system may capture this correlation outcome and use it to warn users accordingly. This process can only achieve high rates of effectiveness if large populations of both benign and malicious datasets are analysed and correlated, in order to train a model capable of making the distinction between the two. Even though the analysis of the bulk of the data (i.e. benign data) can be done without user identification, any discoveries of malicious behavior would eventually need to be analysed and delivered to the users with specific (de-identified) details that can help protect or clean up the system. In the following, we explore how pseudonymisation could be applied in this context, analysing possible scenarios, roles and techniques.

5.2.1 Entities and roles

A generalised pseudonymisation scenario for a reputation system typically consists of: 1) the data subjects (e.g. Alice), who voluntarily participate in the reputation system and the cybersecurity features it enables, 2) the Pseudonymisation Entity PE, 3) the Data Controller DC (e.g. security telemetry collection entity), 4) and the Data Processor DP (i.e.the reputation system). More specifically, the real-world deployments tend to be as follows:

- Similar to Scenario 1 Section 2.1 (Chapter 2), the data subjects share the data directly with the Data Controller (the cybersecurity company). The DC will receive and analyse the data internally, and will, therefore, act as the Pseudonymisation Entity as well as the reputation system (i.e. there is no Data Processor). This scenario (Figure 16) is very common, for instance in the consumer security market, where the entire collection and analysis of the data is performed by the consumer product provider.
- The second common case, reflected by Scenario 3 Section 2.1 (Chapter 2), involves “Sending Pseudonymised Data to a Processor” (Figure 17) . This scenario captures cases where an analytics system (e.g. reputation system) is provided as a third-party service by a data analytics provider. The cybersecurity company (e.g. an IoT home security product provider) may act as the Data Controller who takes the role of the PE and passes pseudonymised data to the processor (analytics provider).

Figure 16: Cybersecurity company collecting all data and queries and acts as DC & PE

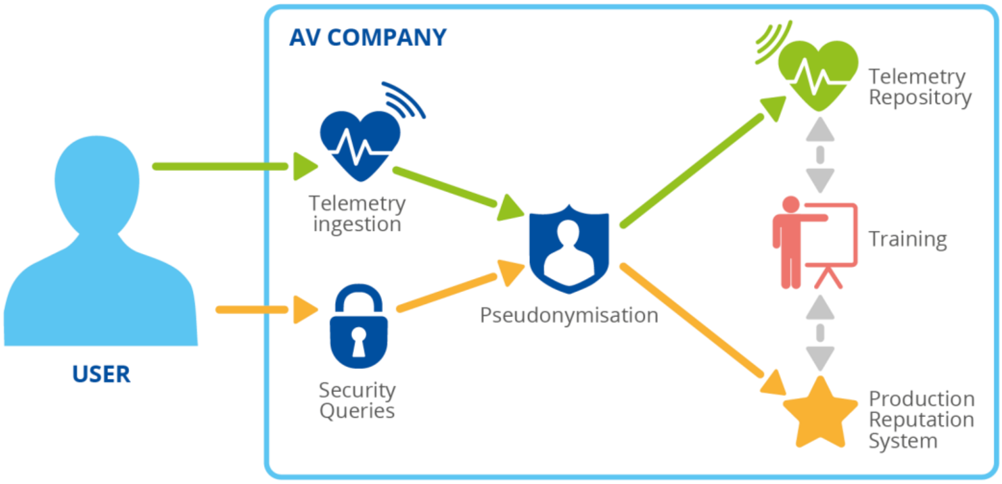
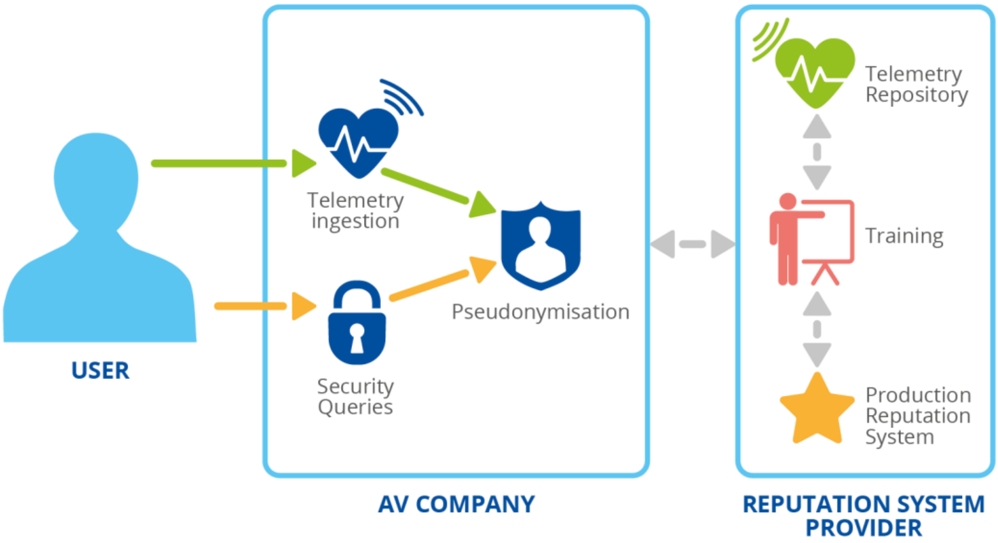


Figure 17: Cybersecurity company (DP & PE) performs pseudonymisation before queries are passed on to the DP (Reputation System Provider)



Alternatively, the data processor, may also perform the pseudonymisation on behalf of the controller (if trusted to do so), although this does not appear to be a common scenario in practice.

5.2.2 File Reputation

When analysing files, a security service aims at determining whether a file is malicious (e.g. a piece of malware), or benign. In many cases, however, the determination is not binary: many files fall in a "grey area", where it is unclear whether they are strictly malicious or benign. In that sense, one can think of file analysis results as a score (e.g. from 0 to 9), where the middle range is "undetermined". One could chose to not convict such files, risking the possibility of missing a lot of threats (high false negative rate - FNR). On the opposite extreme, one could follow a conservative approach and convict all grey files – leading to a lot of unfair convictions (high false positive rate - FPR) that may render the entire system unusable. In most cases, security vendors have to resort to other methods in order to classify grey files as accurately as possible. This type of problem lends itself as a great application for reputation technology.

A file reputation system aims at classifying a file based on the reputational characteristics of the file – as opposed to "traditional" file analysis. The reputational characteristics of a file F on Alice's computer include properties such as 1) what other files F is installed with, 2) what malicious files coexist along with F, 3) what is Alice's computer hygiene (based on observed numbers of incidents - such as infection rate, malicious downloads, malware activity on that computer, etc.), 4) how many computer's containing F are found to be infected/compromised/abused, etc. By combining these and many other similar factors a file reputation system can identify high-confidence correlations allowing the system to produce a score for the grey file in question.

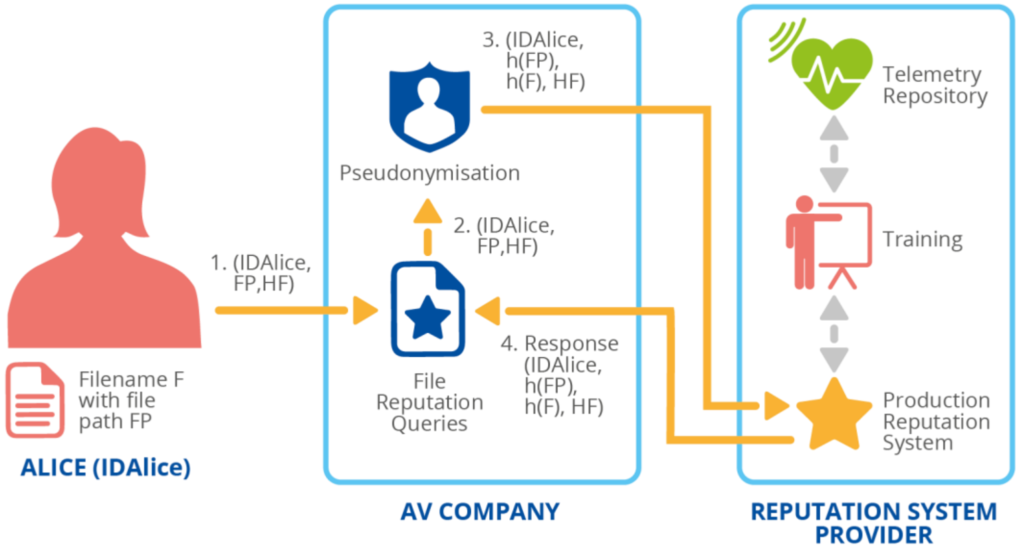
In the following we explore the possibility to employ pseudonymisation at two different phases of the reputation system: a) training (with a large corpus of data) and b) production. Without loss of generality, for the purposes of this example, let us assume that the data controller is also performing the pseudonymisation step, while a third-party DP is contracted for the file reputation analytics. The high-level flow of the system would be as follows:

- During the bootstrapping phase, the unique pseudonym is created for each data subject (e.g. IDAlice) by the PE.
- For each (new) file F on Alice's endpoint, a query is sent to the PE component of the DC, containing the following: IDAlice, F's full path, F's content hash HF.
- The PE component of the DC generates a cryptographic hash of a) the full/relative path of F, b) the filename of F. These hashes are stored in the DC, along with IDAlice and HF.

Let us now consider the two phases of the file reputation system:

- During the training phase the pseudonymised data gathered by the DC is used to train the algorithms that are used to perform the reputational scoring. This requires a "ground truth" data set used by the DP in order to generate the graphs, ML models and all other algorithmic tools necessary. At a high level, the training process consists of two iterative steps: a) calculate the model parameters, and b) check the accuracy of the system – if the accuracy is not satisfactory, return to step (a). Although step (a) could be performed on pseudonymised data, the accuracy check of step (b) would require the verification of the correctness of the results (as well as the calculation of FPR and FNR). In order to perform these checks, DP would need to communicate the results to the PE component of the DC, who will re-identify the data, calculate the accuracy/correctness metrics and respond to DP accordingly. This process of checking results against the de-identified ground truth training data set is essential for the efficacy of the system and will continue until the desired accuracy has been achieved. Furthermore, the system will need to be re-trained periodically, in order to keep up with new threats.
- After the system has been trained, it is deployed in production. When a reputation query for file F is submitted to the system it is routed to the DP. Upon checking, if the DP classifies the file as benign, no further action is needed. If, however, the DP classifies F as malicious, then the appropriate feedback needs to be passed back to the user. The DP responds back to the DC in order to reverse the pseudonymisation, identify the user in question and provide them with plain text information regarding the malicious file F. The request flow steps are depicted on Figure 18 – if the DP response is negative (low reputation detected), then the response of step 4 is re-identified and an alert is sent to Alice.

Figure 18: A file reputation query by Alice, step-by-step.


1. (IDAlice, FP,HF)
Filename F with file path FP
ALICE (IDAlice)
3. (IDAlice, h(FP), h(F), HF)
Pseudonymisation
2. (IDAlice, FP,HF)
File Reputation Queries
4. Response (IDAlice, h(FP), h(F), HF)
AV COMPANY
Telemetry Repository
Training
Production Reputation System
REPUTATION SYSTEM PROVIDER

Note that, in the above scheme, the file content hash (HF) is used as a (unique) file identifier, so as to identify the file in question, regardless of name and file path. In that sense, HF is not a pseudonymised identifier. A simple cryptographic hash suffices, in general, for computing HF as the input domain consists of any possible file of any form (large search space), making dictionary and brute-force attacks impractical. On the contrary, the user ID (IDAlice), the file name and the file path information are hashed for pseudonymisation purposes. A simple hash function can be used in this case as well, but as the search space is a lot smaller, it is important to consider the potential for dictionary attacks as discussed in (ENISA, 2019 - 1) and (ENISA, 2019 - 2) for cases where the input domain of the hash function is somehow predictable[31]. If the estimated risk of unauthorised reversing of such hashed values is deemed to be unacceptable, other approaches implementing deterministic pseudonymisation policy are also possible (e.g. a keyed hash function or symmetric encryption for facilitating the pseudonymisation reversal by the PE, with the key being the pseudonymisation secret – see Chapter 2 for an overview).

The selection of hashing strategy primarily depends on the threat model assumed for the situation at hand: in an “honest but curious” model (where adversaries are curious about the plain text values but will not launch an attack on the system), a simple hash may be sufficient. When the threat model involves more aggressive/malicious adversaries, a keyed hash function would be more appropriate, especially when hashing data from a narrower domain (e.g. file paths in a known operating system).

5.2.3 URL Reputation

Similar to file reputation, cybersecurity companies often rely on URL reputation in order to protect their users from malicious websites. One can say that URL reputation datasets can be considered even more sensitive, as they can contain personal information that can be explicitly or implicitly extracted by analysing the browsing history of users. For example, it has been shown that browsing histories can be linked to some social networks profiles (and, thus, to individuals) due to the fact that users are more likely to click on links posted by accounts they follow, which in turn can be publically available information (Su, Schukla, Goel, & Narayanan, 2017).

[31] See, e.g., the case of hashing IP addresses in (ENISA, 2019 - 2).

Therefore, the notion of an “identifier” of a user can indeed be very broad in some cases and, thus, protecting such information is considered very important. Training an effective URL reputation system, however, requires training models and identifying correlations similar to those needed for a file reputation system. In a similar workflow, when users are infected or otherwise negatively impacted by visiting a particular URL, the system needs to learn from this incident and train itself so as to protect other users. The role of pseudonymisation here is crucial, as the DP (URL reputation system responding to incoming queries) must de-identify the data and use it during the (re-)training phase in order to perform sanity checking and improve the classification models (Figure 16).

One example in the space of URL reputation, where user protection would be significantly improved by selective re-identification of pseudonymised data, is the case of “typo-squatting” detection. Typo-squatting refers to the case where attackers attempt to duplicate a website under a different domain that is very close in spelling to the original service (i.e. likely for user to visit the malicious URL due to a typo error). In order for the reputation system to defend against typo-squatting, it must be able to calculate the edit distance between the two URLs. Enabling this type of protection would require a reversal of the pseudonymised (e.g. hashed) URL. A somewhat better solution involves separately pseudonymising the main domain of the URL and the exact path under that domain – so as to provide a balance between utility and protection (ENISA, 2019 - 2). This allows the DP to train protection models by having access to the main domains in plain text, without seeing the details of the exact path – and thus fully exposing the specifics of the user’s activity on the website in question.

Similarly to the case of file reputation, a URL reputation system may be also based on a conventional cryptographic hash function as a pseudonymisation mechanism. However, in these cases the probability of unauthorised pseudonymisation reversal is not necessarily negligible since dictionary attacks may be viable (indeed, large sets of known URLs can be collected and hashed by an adversary, in order to derive the relevant hashed values and subsequently compare them with the pseudonymised ones). Therefore, although the last two cases (file reputation and URL reputation) share some common properties and requirements (i.e. they both need deterministic pseudonymisation policies, as well as capability of easy pseudonymisation reversal by the PE), different pseudonymisation techniques may be preferable, depending again on the threat model assumptions for the operating environment. In reality, although the DP and PE entities may be hosted within the same organisation, the PE function may be isolated, so as to ensure additional protection from DP agents (see also the discussion on data custodianship in Chapter 4).

5.3 USE CASES ON SECURITY OPERATIONS AND CUSTOMER SUPPORT CENTRES

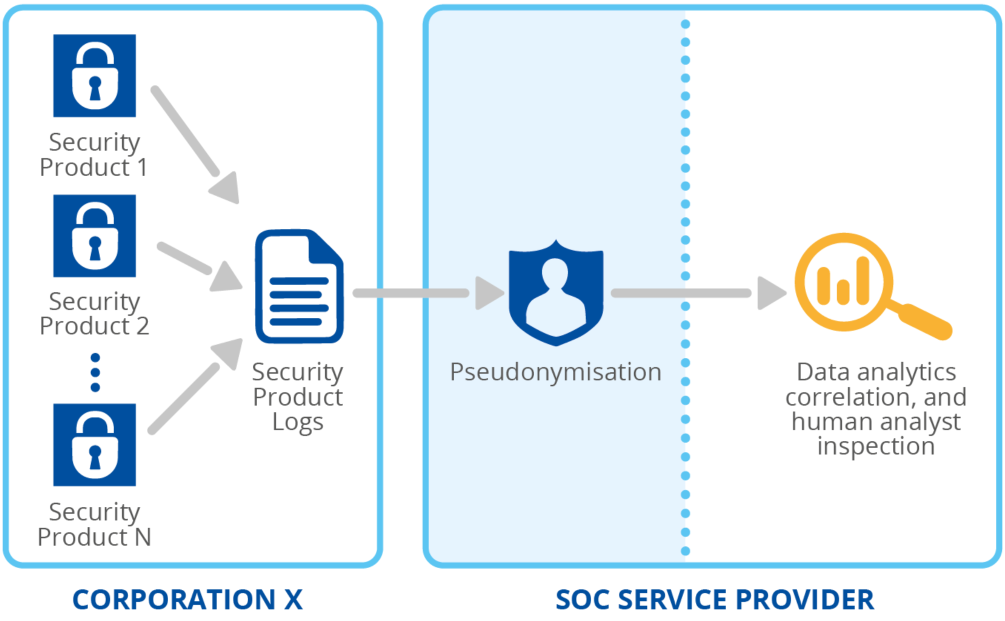
In many cases individuals or organisations are faced with unexpected circumstances, complex cyber security problems, or new threats. Those are typically the cases where customer support is contacted in hope of resolving the situation at hand. For an incident response or customer support function to be effective in dealing with a crisis (or even simple problems) it is often necessary to be able to quickly access (with the data owner’s explicit permission) all the necessary information relating to the circumstances of the issue. Pseudonymisation can protect users during their day-to-day regular interaction, while leaving open the possibility of an effective and satisfactory customer support and crisis management experience – as it is in the best interest of everyone to do so.

5.3.1 Security Operations Centers

In the context of enterprise cybersecurity, a Security Operation Center (or SOC) is a centralized unit within an organisation dealing with all aspects of cybersecurity operations. Since this is a highly technical and critical function, a lot of businesses chose to outsource their SOC to a third-party service. In practice, this involves a company X sharing all of its security-related data (e.g.

service, access, firewall, server logs, etc.) with a SOC service Y, which is responsible for analysing all the events included in the data and attempt to discover security incidents, if any. This task requires Y to employ very advanced correlation and analytics algorithms in order to discover meaningful security insights from a vast amount of sensitive information. In order to reduce the exposure of internal information from X to Y, and also reduce Y's liability during the analysis phase, it is common for the data to be pseudonymised before getting analysed by Y. By doing so, when Y is able to detect an incident, it can report it back to X for investigation (after the pseudonymisation has been reversed).

Figure 19: Example of pseudonymisation in the case of SOC



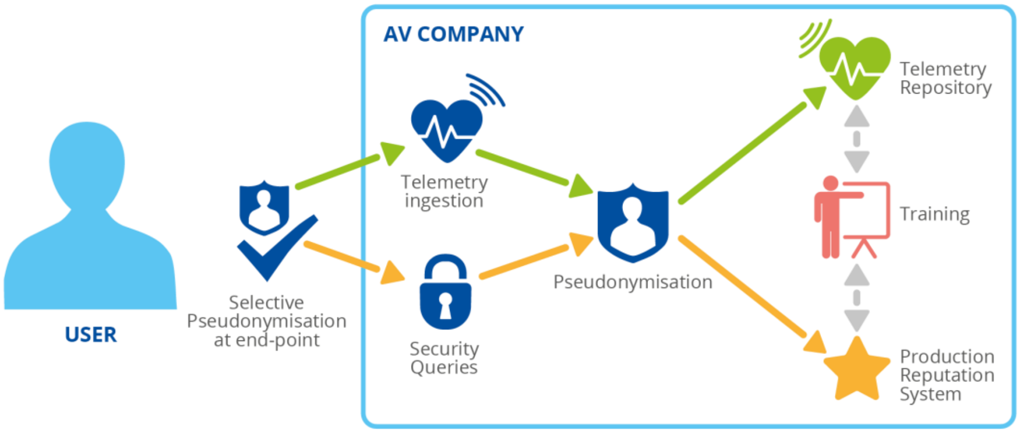
While obviously the best possible option would be for the DC (X) to perform the pseudonymisation, before data being send to the DP (Y), in practice most DCs do not have the capability to do so. Therefore, the most common approach, as depicted in Figure 19, is that the data is pseudonymised by the DP (Y) as soon as it is being received, but before being passed on to the analysts of Y. In such case, Y acts as the Pseudonymisation Entity (see scenario 4 in Section 2.1, Chapter 2). In order for the pseudonymisation process to be safeguarded, however, it is necessary in this example that the PE part within Y is clearly separated from the analytics part of Y. A number of different pseudonymisation techniques can be employed for this (see also in Chapter 3 for advanced techniques).

5.3.2 Consumer customer support

During the normal operation of a cyber security product (e.g. antivirus software, or an intrusion prevention/detection system) the data subject information (ID, queries, files, URLs, etc.) is pseudonymised to protect the identity of individuals. Similar to the file reputation example of 5.2.2, the data subject (Alice) is assigned a unique identifier (IDAlice). All information and queries are pseudonymised by PE (e.g. using a crypto hash for file hashes, URL domain hashes, URL path hashes, etc.) and then passed on to the DC and any necessary DPs (e.g. for file or URL reputation queries). In most consumer products (e.g. antivirus software on consumer PC), the PE role is implemented by the cybersecurity product provider, who is also the Data Controller. This operation model follows Scenario 1 of (ENISA, 2019 - 2) .

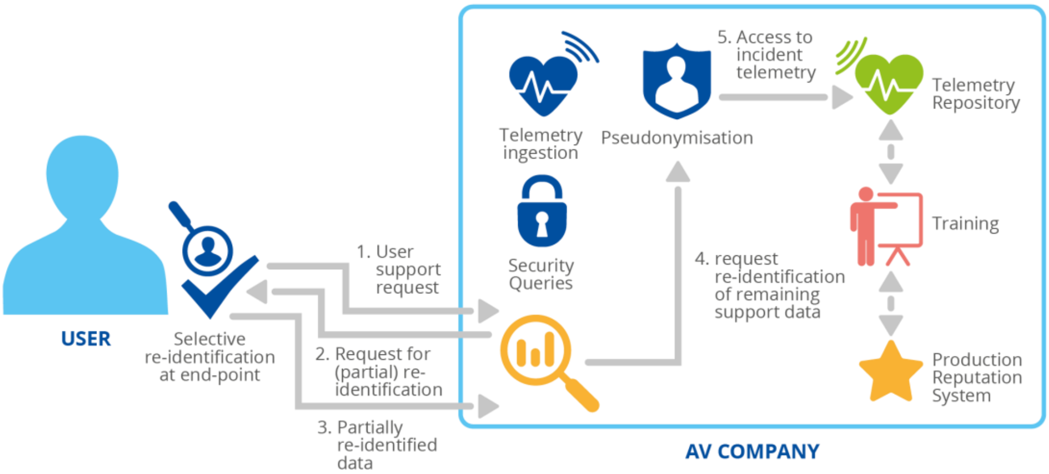
In some cases, or for certain types of data, the product in question may include an end-point component (e.g. antivirus agent running on the user's computer), also performing some basic pseudonymisation functions (e.g. potentially prompting for a user "secret" such as a pin or password) before any data is sent back to the DC. For example, such a secret may be used to pseudonymise the users' location at the end-point. The DC and DP can operate with pseudonymised location data (e.g. "home" can correspond to a pseudonymised location identifier). But if a phone is stolen or lost, the data can be re-identified in order to perform recovery. Figure 20 depicts this possibility of a potential "two-stage" pseudonymisation (as a slightly altered flow of Figure 16). In either case, the data residing at the DC are pseudonymised during normal operation.

Figure 20: Pseudonymised at the end-point



Let as now assume that at a given point in time, Alice visits URL W and this leads to the (explicit or implicit) download of file F. When F is accessed, a file reputation query will be performed as well as other antivirus checks. If the file is found to be malicious, the execution is blocked, but the URL needs to be marked as malicious in the URL reputation DP (update the URL reputation models). Let as assume, however, that the file is not convicted and suddenly Alice's computer begins to exhibit strange behaviours. Alice contacts customer support and reports the situation. In that case, Alice expects to be assisted and have the issue resolved. It is therefore in her best interest to allow for the customer support agent to access the details of her most recent activity, in order to perform an investigation, determine the cause of the problem and remediate it.

Figure 21: Flow of events for a user-initiated support call


Telemetry ingestion
Pseudonymisation
5. Access to incident telemetry
Telemetry Repository
Security Queries
Training
4. request re-identification of remaining support data
Production Reputation System
USER
Selective re-identification at end-point
1. User support request
2. Request for (partial) re-identification
3. Partially re-identified data
AV COMPANY

To that end, as depicted in Figure 21, the customer support agent will determine Alice's pseudonymous identifier IDAlice (possibly with the help of Alice – especially if Alice acts as partial/local PE, such as in the case of client-side pseudonymisation of personal data mentioned earlier). After acquiring IDAlice, the support agent will request all relevant (e.g. most recent) records belonging to Alice from the DC telemetry repository and de-identify them. Once the incident data have been provided to the support agent, the forensic study will begin and the result of the investigation will enable the agent to remediate and provide Alice with a solution. Note that in this example the process is triggered by Alice's explicitly request (and consent) for remediation of a crisis – as opposed to other examples (in which the system automatically performs the necessary remediation actions).

5.3.3 Protection gap and real-time protection

Cyber threats and cyber security are two areas of fast-paced change. Ensuring continuous and uninterrupted protection for users requires that threats are monitored continuously, protection gaps are addressed immediately, and performance improvements are made to make the whole experience seamless for users. Similar to other cases of detailed threat analysis, protection gaps require the ability to dissect the security telemetry data as needed and analyse the details of anomalous incidents – which may indicate new threats. When identifying a protection gap, the typical roadmap for providing protection for an emerging threat involves the analysis of the threat itself (e.g. malware payload), the sources of the threat (e.g. IP addresses, URLs, social media accounts, etc.), and the implications for the affected users (e.g. files infected locally, spyware installed on end-points, etc.). Similar information is necessary for addressing performance concerns for security products. In Section 5.2 of this Chapter we described the process necessary for addressing the protection gap related to the reputation engine updates (model retraining, etc.). In the case of customer support, described in the previous Section, it is often the case that customers become witnesses to novel, emerging threats. When Alice contacts customer support to seek help against a threat that was not detected/prevented by her security software, it is important for the customer support agent to, not only remediate the issue for Alice, but also escalate the concern in order to improve protection from the threat in question for everyone's benefit. In the context of the previous example, if the malware file that Alice downloaded infected her computer, the customer support agent (with Alice's permission) would need to forward all relevant information (URL, domain, file path, file hash, etc.) to the security response team that would analyse the data in question and bridge the protection gap accordingly.

Another very clear example of this type of protection gap remediation workflow is the one related to fraud detection systems, including credit card transaction monitoring, dark web monitoring, etc. As these services access financial and other personal information, it is important to employ pseudonymisation methods to protect these fields. Like in previously discussed cases, in most real-world applications, a PE sanitises the data using crypto hashes, before handing them over to a DP. When fraud is detected (e.g. a suspicious or unauthorised credit card transaction) it is important for the DC (as well as the authorities sometimes) to access the details of the fraudulent activity in order 1) protect other customers (e.g. who was Alice scammed by and how, what was the context, the methods and the delivery vehicle for the scam, etc.), and 2) assist in the investigation of the fraudulent activity. These types of incidents have increased significantly in recent years and it is very common for DCs to request the re-identification of data for investigation purposes.

5.4 ADDITIONAL CYBERSECURITY USE CASES

While in this Chapter we explored only a few cases of application of pseudonymisation in the area of cybersecurity, there are many other promising areas, such as the one of risk analytics, as well as that of fair and accountable Machine Learning systems (where protection of personal data must be ensured, while also allowing for system transparency and accountability). Forensics analysis and evidence discovery are other important areas that can benefit from data pseudonymisation.

6. CONCLUSIONS AND RECOMMENDATIONS

Pseudonymisation is increasingly becoming a key security technique and a way to implement data minimisation in various contexts, providing a means that can facilitate personal data processing, while offering strong safeguards for personal data protection. Complementing previous relevant ENISA's work, in this report we analyse advanced pseudonymisation techniques and specific use cases that can help towards the definition of the state-of-the-art in this field.

Obviously, as we demonstrate throughout the report, there is not a single solution on how and when to apply pseudonymisation; in fact different solutions might provide equally good results in specific scenarios, depending on the requirements in terms of protection, utility, scalability, etc. By the same token, pseudonymisation can be a "simple" option to adopt, but it can also comprise of a very complex process, both at technical, as well as organisational levels.

Based on the analysis provided in the report, in the following we draw some basic conclusions and recommendations for all relevant stakeholders, as regards the wider practical adoption of data pseudonymisation.

Defining the best possible technique

As it has been stressed also in past ENISA's reports, a risk based approach[32] to pseudonymisation is fundamental to truly unfold the potentials of this set of technologies. There is no fit-for-all pseudonymisation technique and a detailed analysis of the case in question is necessary in order to define the best possible option. For instance, although simple hash would not provide adequate data protection in most cases, we show in this report that appropriate elaboration of this technique (as in the case of chaining mode or Merkle trees) can significantly increase the protection level. At the same time, different techniques can solve different types of problems (as for example the case of asymmetric encryption that can provide for the delegation of pseudonymisation).

Another very important element in this discussion is the very wide notion of pseudonymous data. As we showed in the healthcare examples, while a certain piece of data might not constitute pseudonymous (personal) data in itself, it automatically assumes this characteristic, when it becomes part of the broader dataset (e.g. part of the pseudonym of a medical data record). Independently of the technique used and the level or risk, it is, thus, critical to look into the semantics or otherwise the "full picture" before conducting data pseudonymisation.

This being said, it is also important to note that pseudonymisation, while being a prominent security and data protection measure, it is still not the only possible solution; in fact, pseudonymisation must be combined with a thorough security risk assessment for the protection of personal data.

Data controllers and processors should engage in data pseudonymisation, based on a security and data protection risk assessment and taking due account of the overall context and characteristics of personal data processing. This may also comprise methods for data subjects to

[32] Taking into account specifically the risks for rights and freedoms of natural persons (as required by the GDPR).

pseudonymse personal data on their side (e.g. before delivering data to the controller/processor) to increase control of their own personal data[33].

Regulators (e.g. Data Protection Authorities and the European Data Protection Board) should promote risk-based data pseudonymisation through the provision of relevant guidance and examples.

Advanced techniques for advanced scenarios

While the technical solution is a critical element for achieving proper pseudonymisation, one must not forget that the organisational model and its underlying structural architecture are also very important parameters of success. In other words, there is no use of putting in place a robust pseudonymisation technique, without having ensured that the entities involved (and the relevant data flow scheme) will be able to support it. To this end, when discussing advanced techniques, we also need to put in place advanced scenarios, such as the case of the data custodianship model, which we explore in this report in the context of healthcare.

Data controllers and processors should consider possible scenarios that can support advanced pseudonymisation techniques, based – among other – on the principle of data minimisation.

The research community should support data controllers and processors in identifying the necessary trust elements and guarantees for the advanced scenarios (e.g. data custodianship) to be functional in practice.

Regulators (e.g. Data Protection Authorities and the European Data Protection Board) should ensure that regulatory approaches, e.g. as regards new technologies and application sectors, take into account all possible entities and roles from the standpoint of data protection, while remaining technologically neutral.

Establishing the state-of-the-art

Although a lot of work is already in place, there is certainly more to be done in defining the state-of-the-art in data pseudonymisation. For instance, it is important to work on more complex cases and their possible evolution, e.g. in the light of emerging technologies. While doing so, one should ask different types of questions, e.g. the focus to be pursued (horizontal/vertical), the parties to be involved, the process of maintaining the state-of-the-art, etc. To this end, research and application scenarios must go hand-in-hand, involving all relevant parties (researchers, industry, and regulators) to discuss joined approaches.

The European Commission, the relevant EU institutions, as well as Regulators (e.g. Data Protection Authorities and the European Data Protection Board) should support the establishment and maintenance of the state-of-the-art in pseudonymisation, bringing together all relevant stakeholders in the field (regulators, research community, and industry).

The research community should continue its efforts on advancing the existing work on data pseudonymisation, addressing special challenges appearing from emerging technologies, such as Artificial Intelligence. The European Commission and the relevant EU institutions should support and disseminate these efforts.

Towards the broader adoption of data pseudonymisation

Looking at the recent developments, e.g. in the field of international personal data transfers (CJEU Schrems II Judgment[34] and beyond, the need to further advance appropriate safeguards including supplementary measures for personal data protection becomes evident. In addition, the (increasing) need for open data access can only intensify the use of pseudonymisation in

[33] See Recital 7 of the GDPR.
[34] CJEU, Judgment of the Court (Grand Chamber) of 6 October 2015 (request for a preliminary ruling from the High Court (Ireland)) — Maximillian Schrems v Data Protection Commissioner (Case C-362/14).

the future, as one of the prominent solutions for personal data protection. It is, thus, important to provide already today the necessary information and motivation for broader adoption and real world usage of pseudonyisation in different application scenarios.

Regulators (e.g. Data Protection Authorities and the European Data Protection Board), the European Commission and the relevant EU institutions should disseminate the benefits of data pseudonymisation and provide for best practices in the field.

7. REFERENCES

Akil, M., Islami, L., Fischer-Hübner, S., Martucci, L. A., & Zuccato, A. (2020). Privacy-Preserving Identifiers for IoT: A Systematic Literature Review. *IEEE Access (DOI: 10.1109/ACCESS.2020.3023659)*.

Armknecht, F., Boyd, C., Carr, C., Gjøsteen, K., Jäschke, A., Reuter, C. A., & Strand, M. (2015). A guide to fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 1192.

Avitabile, G., Botta, V., Iovino, V., & Visconti, I. (2020). *Towards Defeating Mass Surveillance and SARS-CoV-2: The Pronto-C2 Fully Decentralized Automatic Contact Tracing System.* Cryptology ePrint Archive. Retrieved from https://ia.cr/2020/493

Badimtsi, F., Canetti, R., & Yakoubov, S. (2020). Universally Composable Accumulators. *T-RSA 2020: Topics in Cryptology* (pp. 638-666). Sprinfer LNCS.

Barić, N., & Pfitzmann, B. (1997). Collision-Free Accumulators and Fail-Stop Signature Schemes Without Trees. *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 1997)*, (pp. 480-494).

Bellare, M., Canetti, R., & Krawczyk, H. (1996). Keying Hash Functions for Message Authentication. *CRYPTO '96* (pp. 1-15). LNCS.

Benaloh, J., & de Mare, J. (1993). One-Way Accumulators: A Decentralized Alternative to Digital Signatures. *Workshop on the Theory and Application of of Cryptographic Techniques (EUROCRYPT 1993)* (pp. 274-285). Springer.

Biryukov, A., Dinu, D., & Khovratovich, D. (2016). Argon2: New Generation of Memory-Hard Functions for Password Hashing and Other Applications. *2016 IEEE European Symposium on Security and Privacy (EuroS&P).*

Biskup, J., & Flegel, U. (2000). On Pseudonymization of Audit Data for Intrusion Detection. *Designing Privacy Enhancing Technologies, International Workshop on Design Issues in Anonymity and Unobservability.* doi:10.1007/3-540-44702-4_10

Blum, M., Feldman, P., & Micali, S. (1984). How to Generate Cryptographically Strong Sequences of Pseudorandom Bits. *SIAM Journal on Computing, 13*.

Bundesministerium des Innern. (2017). *Guidelines for the legally secure deployment of pseudonymization solutions in compliance with the General Data Protection Regulation.* Retrieved from https://www.gdd.de/downloads/white-paper-pseudonymization/

Camenisch, J., & Lehmann, A. (2015). (Un)linkable Pseudonyms for Governmental Databases. *22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15)* (pp. 1467–1479). ACM. doi:https://doi.org/10.1145/2810103.2813658

Camenisch, J., & Lehmann, A. (2017). Privacy-Preserving User-Auditable Pseudonym Systems. *2017 IEEE European Symposium on Security and Privacy (EuroS&P).* IEEE. doi:10.1109/EuroSP.2017.36

Camenisch, J., Kohlweiss, M., & Soriente, C. (2009). An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials. *International Workshop on Public Key Cryptography (PKC 2009)* (pp. 481-500). Springer LNCS.

Chase, M., & Miao, P. (2020). *Private Set Intersection in the Internet Setting From Lightweight Oblivious PRF.* Cryptology ePrint. Retrieved from https://eprint.iacr.org/2020/729.pdf

Chi-Chih Yao, A. (1986). How to generate and exchange secrets. *27th Annual Symposium on Foundations of Computer Science (SFCS 1986).* IEEE.

EDPS. (2016). *EDPS Opinion 9/2016 on Personal Information Management Systems: Towards more user empowerment in managing and processing personal data.* Retrieved from https://edps.europa.eu/sites/edp/files/publication/16-10-20_pims_opinion_en.pdf

EDPS. (2020). *A Preliminary Opinion on data protection and scientific research.* EDPS. Retrieved from https://edps.europa.eu/sites/edp/files/publication/20-01-06_opinion_research_en.pdf

Elgamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory, 31*(4), 469 - 472.

ENISA. (2019 - 1). *An overview on data pseudonymisation: Recommendations on shaping technology according to GDPR provisions.* Athens: ENISA. Retrieved from https://www.enisa.europa.eu/publications/recommendations-on-shaping-technology-according-to-gdpr-provisions

ENISA. (2019 - 2). *Pseudonymisation techniques and best practices: Recommendations on shaping technology according to data protection and privacy provisions.* Athens: ENISA. Retrieved 2019, from https://www.enisa.europa.eu/publications/pseudonymisation-techniques-and-best-practices

Fazio, N., & Nicolosi, A. (2002). *Cryptographic Accumulators : Defintions, Constructions and Applications.* Retrieved from http://www.inf.ufsc.br/~martin.vigil/paper.pdf

Feige, U., Fiat, A., & Shamir, A. (1988). Zero-knowledge proofs of identity. *Journal of Cryptology*, 77-94.

Framner, E., Fischer-Hübner, S., Lorünser, T., Alaqra, A. S., & Pettersson, J. S. (2019). Making secret sharing based cloud storage usable. *Information & Computer Security, 27*(5), 647-667.

Goldreich, O., Micali, S., & Wigderson, A. (1987). How to play ANY mental game. *Nineteenth annual ACM symposium on Theory of computing (STOC '87)*, (pp. 210-217).

Goldwasser, S., Micali, S., & Rackoff, C. (1985). The knowledge complexity of interactive proof-systems. *Seventeenth annual ACM symposium on Theory of computing (STOC '85)*, (pp. 291-304).

H. Li, L. P. (2019). Blockchain Meets VANET: An Architecture for Identity and Location Privacy Protection in VANET. *Peer-to-peer Networking and Applications, Springer, 12*, 1178–1193.

Hazay, C., & Lindell, Y. (2008). Efficient Protocols for Set Intersection and Pattern Matching with Security Against Malicious and Covert Adversaries. *Theory of Cryptography (TCC 2008)* (pp. 155-175). Springer LNCS.

He, S., Ganzinger, M., & Hurdle, J. K. (2013). *Proposal for a data publication and citation framework when sharing biomedical research resources.* Stud Health Technol Inform. Retrieved from https://pubmed.ncbi.nlm.nih.gov/23920975/

Information Commissioner's Office (ICO). (2012). *Anonymisation: managing data protection risk code of practice.* Retrieved from https://ico.org.uk/media/1061/anonymisation-code.pdf

Joye, M. (2013). On Elliptic Curve Paillier Schemes. *Algebraic Informatics (CAI 2013)* (p. 6). Springer LNCS.

Kasem-Madani, S., Meier, M., & Wehner, M. (n.d.). Towards a Toolkit for Utility and Privacy-Preserving Transformation of Semi-structured Data Using Data Pseudonymization. *International Workshop on Data Privacy Management (DPM 17, CBT 17))* (pp. 163-179). Springer.

Kolesnikov, A., Kumaresan, R., Rosulek, M., & Trieu, N. (2016). Efficient Batched Oblivious PRF with Applications to Private Set Intersection. *ACM SIGSAC Conference on Computer and Communications Security* (pp. 818-829). ACM.

Krawczyk, H. (2010). Cryptographic Extraction and Key Derivation: The HKDF Scheme. *Advances in Cryptology – CRYPTO 2010* (pp. 631-648). Springer LNCS.

Lamport, L. (1981). *Password authentication with insecure communication.* Communications of the ACM. doi:https://doi.org/10.1145/358790.358797

Lehmann, A. (2019). ScrambleDB: Oblivious (Chameleon) Pseudonymization-as-a-Service. *Proceedings on Privacy Enhancing Technologies , 2019*(3), 289-309. doi:https://doi.org/10.2478/popets-2019-0048

Li, H., Pei, L., Liao, D., Sun, G., & Xu, D. (2019). Blockchain Meets VANET: An Architecture for Identity and Location Privacy Protection in VANET. *Peer-to-peer Networking and Applications, 12*, 1178–1193.

Lindell, Y. (2020). *Secure Multiparty Computation (MPC).* Cryptology ePrint. Retrieved from https://eprint.iacr.org/2020/300.pdf

Liu, J. K., & Wong, D. S. (2005). Linkable Ring Signatures: Security Models and New Schemes. *Computational Science and Its Applications – ICCSA 2005* (pp. 614-6123). Springer LNCS.

Merkle, R. (1987). A Digital Signature Based on a Conventional Encryption Function. *Conference on the Theory and Application of Cryptographic Techniques (CRYPTO 1987)* (pp. 369-378). Springer.

Montenegro, G., & Castelluccia, C. (2004). Crypto-based identifiers (CBIDs): Concepts and applications. *ACM Transactions on Information and System Security*, 97-127.

Nyberg, K. (2005). Fast accumulated hashing. *International Workshop on Fast Software Encryption (FSE 1996)* (pp. 83-87). Springer.

Paillier, P. (1999). Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 1999)* (pp. 223-238). Springer LNCS.

Paillier, P. (2000). Trapdooring Discrete Logarithms on Elliptic Curves over Rings. *Advances in Cryptology — ASIACRYPT 2000* (pp. 573-584). Springer LNCS.

Pinkas, B., Schneider, T., & Zohner, M. (2019). *Scalable Private Set Intersection Based on OT Extension.* Retrieved from https://eprint.iacr.org/2016/930.pdf

Pommerening, K., Schröder, M., Petrov, D., Schlösser-Faßbender, M., Semler, S., & Drepper, J. (2006). *Pseudonymization Service and Data Custodians in Medical Research Networks and Biobanks.* Retrieved from https://dl.gi.de/bitstream/handle/20.500.12116/23646/GI-Proceedings-93-715.pdf

Rabin, M. O. (1979). *Digitalized Signatures and Public-Key Functions as Intractable as Factorization.* Massachusetts Institute of Technology.

Rivest, R. L., Shamir, A., & Adleman, L. M. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 120-126.

Rivest, R., Shamir, A., & Tauman, Y. (2001). How to Leak a Secret. *International Conference on the Theory and Application of Cryptology and Information Security* (pp. 552-565). Springer LNCS.

Shamir, A. (1979). *How to share a secret.* Communications of the ACM. doi:https://doi.org/10.1145/359168.359176

Su, J., Schukla, A., Goel, S., & Narayanan, A. (2017). De-anonymizing Web Browsing Data with Social Networks. *26th International Conference on World Wide Web (WWW '17)* (pp. 1261-1269). ACM.

Tartary, C. (2008). Ensuring Authentication of Digital Information Using Cryptographic Accumulators. *International Conference on Cryptology and Network Security (CANS 2009)* (pp. 315-333). Springer LNCS.

Verheul, E., Jacobs, B., Meijer, C., Hildebrandt, M., & de Ruiter, J. (2016). *Polymorphic Encryption and Pseudonymisation for Personalised Healthcare – A Whitepaper.* Cryptology ePrint Archive. Retrieved from https://eprint.iacr.org/2016/411.pdf

Weber, S. G. (2012). On Transaction Pseudonyms with Implicit Attributes, Report 2012/568. *Cryptology ePrint Archive*.

XX-00-00-000-XX-X

ABOUT ENISA

The European Union Agency for Cybersecurity, ENISA, is the Union’s agency dedicated to achieving a high common level of cybersecurity across Europe. Established in 2004 and strengthened by the EU Cybersecurity Act, the European Union Agency for Cybersecurity contributes to EU cyber policy, enhances the trustworthiness of ICT products, services and processes with cybersecurity certification schemes, cooperates with Member States and EU bodies, and helps Europe prepare for the cyber challenges of tomorrow. Through knowledge sharing, capacity building and awareness raising, the Agency works together with its key stakeholders to strengthen trust in the connected economy, to boost resilience of the Union’s infrastructure, and, ultimately, to keep Europe’s society and citizens digitally secure. More information about ENISA and its work can be found at www.enisa.europa.eu.

ENISA
European Union Agency for Cybersecurity

Athens Office
1 Vasilissis Sofias Str
151 24 Marousi, Attiki, Greece

Heraklion office
95 Nikolaou Plastira
700 13 Vassilika Vouton, Heraklion, Greece

enisa.europa.eu

ISBN 978-92-9204-465-7
DOI 10.2824/860099