ENISA'S LAST TECHNICAL ANALYSIS OF DATA PSEUDONYMIZATION ADVANCED MEASURES IN DATA PROTECTION AND PRIVACY

SERGIO GUIDA
Independent Researcher, Sr. Data Governance & Privacy Mgr.

ABSTRACT

Among technical and organisational measures for security and data protection by design, pseudonymisation can reduce the risks for data subjects and help controllers and processors meet their data protection obligations. Nevertheless, techniques that may work in one specific case to achieve data protection, may not be sufficient in other cases. So building on the basic pseudonymisation techniques, Enisa 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. There is not a single solution on how and when to apply pseudonymisation, different solutions might provide equally good results in specific scenarios, depending on the requirements in terms of protection, utility, scalability, but also comprise of a very complex process, both at technical, and organisational levels as well. Regulators (Data Protection Authorities and the European Data Protection Board) should promote risk-based data pseudonymisation through the provision of relevant guidance and examples.

Keywords: *Data protection and security by design, pseudonymisation, advanced cryptography.*

Summary: 1. Intro. 2. Advanced pseudonymisation techniques and some relevant operating aspects. 3. Conclusions and recommendations for all relevant stakeholders.

1. Intro.

Pseudonymisation is 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 its past work[1], in this report ENISA analyses advanced pseudonymisation techniques and specific use cases that can help towards the definition of the state-of-the-art in this field.

2. Advanced pseudonymisation techniques and some relevant operating aspects.

Among advanced pseudonymisation[2] techniques, based on cryptographic techniques[3], the following are disclosed:

---

[1] The first relevant report in January 2019 ("ENISA, 2019 – 1") presented 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 detailed analysis of the technical methods and specific examples and best practices for particular data sets(email addresses, IP addresses and more complex data sets).

[2] Cf. "Pseudonymisation is defined in article 4(5) of the GDPR as: 'the processing of personal data in such a manner 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'. In GDPR, the data controller is the entity responsible to decide whether and how pseudonymisation will be implemented. Data processors, acting under the instructions of controllers, may also have an important role in implementing pseudonymisation. Recital (29) GDPR states that the application of pseudonymisation to personal data can reduce the risks to the data subjects concerned and help data controllers and processors to meet their data protection obligations. Moreover, recital (30) states that measures of pseudonymisation should, whilst allowing general analysis, be possible within the

2.1 Asymmetric encryption.

Although symmetric encryption is most commonly used, asymmetric encryption has some interesting properties that could support data minimization, too, and the 'need-to-know principle'[4], while providing robust protection. There are two different entities involved during the pseudonymisation process:

(i) a first entity creates 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)[5].

They do not have to share the same knowledge: a data controller can make available its PK to its data processors, who can collect and pseudonymise the personal data using the PK, so the former is the only entity which can later compute the initial data from the pseudonyms. Such a scenario is related to the generic one of a data processor being the Pseudonymisation Entity, with the additional advantage, in terms of protecting individuals' identities, that the processors do not have

---

same controller, when that controller has taken appropriate technical and organisational measures and that additional information for attributing the personal data to a specific data subject is kept separately", as we can read in ENISA, Recommendations on shaping technology according to GDPR provisions. An overview on data pseudonymisation. November 2018 in https://www.aepd.es/sites/default/files/2019-09/recomendations-on-shaping-technology-according-to-GDPR-provisions -1.pdf.

[3] Cryptographic techniques, such as encipherment, digital signatures, key management and secret sharing schemes, are important building blocks in the implementation of all security services. "You can define encryption as a means by which to convert readable content (plaintext) into unreadable gibberish (ciphertext). Encryption is a mathematical operation that exists within the realm of cryptography. However, there's an important difference:

- *Cryptography is the overarching term for the field of cryptographic communications.*
- *Encryption, on the other hand, refers to the actual process of encrypting plaintext data into unreadable ciphertext.*

Basically, encryption is the process of transforming plaintext into ciphertext through the use of two important elements:

- ✓ *Algorithms — An encryption algorithm is a set of directions to help you solve a problem. More specifically, it's a set of mathematical instructions and processes that serve a specific purpose. Some algorithms are designed to work in either private or public channels. So, you can have asymmetric or symmetric encryption algorithms. In general, encryption algorithms are useful for encrypting data. When coupled with authentication measures, they also protect data integrity.*
- ✓ *Keys — A cryptographic key is a long, random and unpredictable string of letters and numbers that you use to encrypt or decrypt data. No matter whether you're talking about asymmetric vs symmetric encryption, the keys are important to protect",*

as we can read in Casey Crane, Asymmetric vs Symmetric Encryption: Definitions & Differences, December 7, 2020 in https://www.thesslstore.com/blog/asymmetric-vs-symmetric-encryption/.

[4] Cf. "This principle states that a user shall only have access to the information that their job function requires, regardless of their security clearance level or other approvals. In other words: a User needs permissions and a Need-to-know. And that Need-to-know is strictly bound to a real requirement for the User to fulfill its current role. As you might be able to tell by the choice of words the Need-to-know principle is typically enforced in military or governmental environments. Sometimes, in non-military scenarios, you will also find a slightly different description which states in weaker terms that access to data must be regularly reviewed to ensure that users only access data they strictly need for legitimate reasons. This is enforcement by regulation or rule rather than permissions and can be sufficient in the private sector. In information technology the Need-to-know can be implemented by using mandatory access control (MAC) as well as discretionary access control (DAC) in conjunction with a secondary control system" in https://techcommunity.microsoft.com/t5/azure-sql/security-the-need-to-know-principle/ba-p/2112393.

[5] Cf. "In asymmetric or public-key cryptography, two different keys are used, the first key (the public key) is used by the sender to encrypt the information, the second key is a private and secret key used by the recipient to decrypt the information. Therefore, the encryption key can be made public, a common secret is not needed to be agreed on by the parties in advance as the second secret key is only known by the recipient. .. This technique is used mostly for end-to-end encryption. Thus, in an asymmetric encryption scenario the private key has to be kept secret. The risk that a third party could obtain the key consequently arises e.g. if the secret key is stored at a cloud provider which also holds the public key or by man-in-the-middle attacks, if a third party misleads the other parties by pretending to be the respective counterpart. If all necessary security measures are complied with – in the sense of the relative approach – it is not reasonably likely that a man-in-the-middle attack occurs" in Gerald Spindler, Philipp Schmechel, Personal Data and Encryption in the European General Data Protection Regulation, 7 (2016) JIPITEC 163 par.1, in https://www.jipitec.eu/issues/jipitec-7-2-2016/4440.

the pseudonymisation secret. But, 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, i.e. a different pseudonym is derived each time for the same identifier, without changing the process or the pseudonymisation secret[6].

Many asymmetric encryption schemes support homomorphic encryption[7], which 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: e.g. the product of two pseudonyms created using Paillier's scheme[8] is the pseudonym of sum of the two identifiers. The generation speed and the size of the pseudonym obtained using asymmetric encryption can also be an issue, these parameters being strongly correlated to the size of the keys: for instance, in certain setups, the key size can be up to 2018 or 3096 bits.

A typical application is to make available healthcare data to research groups. In order to ensure that the identifiers (e.g. social security number or medical registration number) of a given patient are not linkable, 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; so 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[9] has been used.

2.2 Ring signatures and group pseudonyms.

Digital signatures constitute a main cryptographic primitive[10] towards ensuring both the integrity of the data as well as the authentication of the originating user, (the signer of the message), so that

---

[6] *See Section 5.2.3 "ENISA, 2019 – 2".*

[7] *Cf. "Homomorphic encryption makes it possible to analyze or manipulate encrypted data without revealing the data to anyone. Just like other forms of encryption, homomorphic encryption uses a public key to encrypt the data. Unlike other forms of encryption, it uses an algebraic system to allow functions to be performed on the data while it's still encrypted. Then, only the individual with the matching private key can access the unencrypted data after the functions and manipulation are complete. This allows the data to be and remain secure and private even when someone is using it .. Dr. Craig Gentry describes homomorphic encryption as a glovebox where anybody can get their hands into the glovebox and manipulate what's inside, but they are prevented from extracting anything from the glovebox. They can only take the raw materials and create something inside the box. When they finish, the person who has the key can remove the materials (processed data)" in Bernard Marr, What Is Homomorphic Encryption? And Why Is It So Transformative?, Forbes, Nov 15, 2019 in https://www.forbes.com/sites/bernardmarr/2019/11/15/what-is-homomorphic-encryption-and-why-is-it-so-transformative/.*

[8] *"From a conceptual view point, encryption with the Paillier scheme consists of a fixed basis modular exponentiation with the message as exponent, and the generation of a noise factor used to mask the message. For fixed basis exponentiation, it is well-known that the performance can be increased by pre-computing powers of the fixed basis. By fine-tuning this and other known methods, we reduce the complexity of this step considerably. For the generation of the noise factor, we apply a new method that consists of using pre-computed noise to generate new noise factors. This reduces the bottleneck of noise computation to a few modular multiplications. Together, these methods achieve the considerable increase in encryption performance mentioned above" in Christine Jost, Ha Lam, Alexander Maximov, Ben Smeets, Encryption Performance Improvements of the Paillier Cryptosystem 2015 in https://eprint.iacr.org/2015/864.pdf , page 2.*

[9] *"ElGamal is a public key system which uses modular exponentiation as the basis for a oneway trap door function. The reverse operation, the discrete logarithm, is considered intractable. ElGamal was never patented, making it an attractive alternative to the more well known RSA system. Public key systems are fundamentally different from symmetric systems, and typically demand much larger keys. 1024 bits is the minimum recommended size for ElGamal, and even larger keys are recommended for some applications" in Bryce D. Allen, Implementing several attacks on plain ElGamal encryption, https://lib.dr.iastate.edu/cgi/viewcontent.cgi?article=2577&context=etd , page 7.*

[10] *"Cryptographic primitives are well-established, low-level cryptographic algorithms that are frequently used to build cryptographic protocols for computer security systems. These routines include but are not limited to, one-way hash*

anybody can verify the validity of the signature associated with a known signer. Several advanced digital signature techniques are known, each aiming to fulfill the requirements of a specific application. The so-called 'ring signature' 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 Pk and a relevant secret (private) key Sk. 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 not to identify which user is the actual signer: for instance, a ring signature could be used to provide a verifiable signature from "a high-ranking official", without revealing who exactly is that official. Actually, it's 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) is under the sole control of the data subject.

Group pseudonyms have been used in many contact tracing protocols (like Pronto-C2[11]) proposed during the COVID-19 pandemic[12]: each time two data subjects meet, a pseudonym is created with a contribution from each data subject, so they both have computed the same pseudonym. Each data subject has a list of group or encountered pseudonyms and, 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 has to be randomized, so that when two data subjects meet again, they get a new group pseudonym to avoid any malicious traceability.

2.3 Chaining mode.

Not very often a secure cryptographic hash function[13] is expected to be an appropriate pseudonymisation technique, while authentication codes and keyed-hash functions are being

---

functions and encryption functions" in https://medium.com/geekoffee/ensuring-integrity-authenticity-and-non-repudiation-in-data-transmission-using-node-js-af73c2404153.

[11] Cf. Gennaro Avitabile, Vincenzo Botta, Vincenzo Iovino, Ivan Visconti, Towards Defeating Mass Surveillance and SARS-CoV-2: The Pronto-C2 Fully Decentralized Automatic Contact Tracing System. IACR Cryptol. ePrint Arch. 2020: 493 (2020) in https://eprint.iacr.org/2020/493.pdf :"In the 70s Merkle, Diffie and Hellman invented public-key cryptography. Starting with Merkle's puzzles, Diffie and Hellman proposed a key exchange protocol [DH76] (i.e., the Diffie-Hellman protocol) where two parties can establish a secret key K by just sending one message each on a public channel. A message consists of a group element in a setting where the so called Decision Diffie-Hellman assumption holds. In our view, the most natural way to realize a privacy-preserving ACT system consists of having as pseudonym a group element that corresponds to a message in the DH protocol. This natural idea was also proposed to the DP-3T team .. In order to actually realize such form of ACT system, one needs to solve the following two main problems. Anonymous call: realizing a mechanism that allows an infected party to use K in order to call the other party in a secure and privacy-preserving way. Shortening pseudonyms: making sure that the size of a group element fits the number of available bits in a BLE identifier beacon", page 7. "Notice that the approach of Pronto-C2 is therefore completely different from the one adopted in the DP-3T systems. Indeed, while in the DP-3T systems the pseudonyms of the infected person are broadcast to everyone (or added to a Cuckoo filter by the server that then transmits the filter) we instead ask the infected party to send a message that is understandable uniquely by the party with which she was in close proximity", page 8.

[12] For an in-depth discussion, see Sergio Guida, Un Framework per il Contact Tracing in Italia tra esigenze scientifiche, possibilità tecnologiche e rispetto di Diritti e Libertà Individuali in termini di Data Protection, European Journal of Privacy Law & Technologies, September 16, 2020 in http://www.ejplt.tatodpr.eu/Tool/Evidenza/Single/view_html?id_evidenza=62.

[13] "A cryptographic hash function is an algorithm that takes an arbitrary amount of data input—a credential—and produces a fixed-size output of enciphered text called a hash value, or just "hash." That enciphered text can then be stored instead of the password itself, and later used to verify the user. Certain properties of cryptographic hash functions impact the security of password storage:

- ✓ *Non-reversibility, or one-way function. A good hash should make it very hard to reconstruct the original password from the output or hash.*
- ✓ *Diffusion, or avalanche effect. A change in just one bit of the original password should result in change to half the bits of its hash. In other words, when a password is changed slightly, the output of enciphered text should change significantly and unpredictably.*
- ✓ *Determinism. A given password must always generate the same hash value or enciphered text.*

preferred – which include the use of a secret key. However, more advanced techniques can be obtained by appropriately chaining hash functions, a layered approach in which several somehow intermediate pseudonyms are temporarily generated, in order to finally obtain the pseudonym, which is the output of the last hash function. Each layer is computed by a different entity who holds a secret used to obtain an intermediate pseudonym: K1 is used to obtain the temporary value X=HK1(ID). Value X is then transmitted to the second entity which computes Y=HK2(X). Finally, the last entity computes the Pseudo=HK3(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 K1, K2, K3.
Apparently pseudonym resolution requires to have the three entities to cooperate and just that 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 knows the original identifiers) is not able to match the final pseudonyms with the identifiers. For example, the recipient of the final 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 articulated structures where, depending on the application scenario, each entity can apply a different pseudonymisation technique in this chaining approach.

2.4 Pseudonyms based on multiple identifiers or attributes.

Pseudonymisation is the processing of an identifier into a pseudonym, i.e. a one-to-one mapping, but to add new properties it can be the processing of several identifiers, many-to-one mapping.
The identifiers can be

- *Homogeneous: they have the same type (only phone number for instance) and they are related to different individuals or*
- *Heterogeneous: they match different attributes of a single individual (social security number, phone number, first name and last name).*

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 (cf. "ENISA, 2019 – 1"). Now to ensure some additional properties of such pseudonyms which correspond to many-to-one-mappings, more sophisticated approaches are needed: cryptographic accumulators[14] 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 □ L to provide a proof that a given y actually has been accumulated within z. Such a proof is called a witness w.
Here follows an example based on Merkle Tree[15], a binary tree constructed through hash functions (which could be seen as a generalisation of hash chains). This structure properly suits the purposes

---

- ✓ *Collision resistance. It should be hard to find two different passwords that hash to the same enciphered text.*
- ✓ *Non-predictable. The hash value should not be predictable from the password,*

as we can read in https://www.synopsys.com/blogs/software-security/cryptographic-hash-functions/.

[14] Cf. "A cryptographic accumulator is a short binding commitment to a set of elements and allows for, short membership proofs for any element in the set and/or, non-membership proofs for elements not inside the set. These proofs, also called witnesses (witness to element being accumulated in the accumulator), can be verified against the commitment. They are often used as communication-efficient authenticated data structure for remote databases, where individual elements with their proofs can be retrieved and efficiently verify integrity of the database. Accumulators can be categorized into Static and Dynamic. Unlike static variants, dynamic accumulators allow for addition/deletion of elements at cost independent of the size of the accumulated set. A dynamic accumulator is universal, if it supports both membership and non-membership proofs" in Amit Panghal, Cryptographic Accumulators: Part1, Medium, May 6, 2019 in https://medium.com/@panghalamit/cryptographic-accumulators-part1-3f23172d3fec.

[15] Cf. "A Merkle tree is a data structure that is used in computer science applications. In bitcoin and other cryptocurrencies, Merkle trees serve to encode blockchain data more efficiently and securely. They are also referred to as 'binary hash trees' .. Breaking Down Merkle Tree: in bitcoin's blockchain, a block of transactions is run through an algorithm to generate a hash, which is a string of numbers and letters that can be used to verify that a given set of data

of pseudonymization: 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. 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 wi for the corresponding IDi). Generally, each contributor knows IDi and the corresponding witness wi (including the corresponding key ki) A contributor can later reveal IDi and wi 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 impossible to revert the tree, i.e. recover any values ID1, ID2, ID3 or ID4 while knowing only its root (i.e. the accumulated pseudonym). If a subset of identifiers, ID1 and ID3 for instance, has been revealed, it is still not possible to recover the other identifiers ID2 and ID4. It is only possible to know that ID2 and ID4 have accumulated into z if and only if their corresponding witnesses w2 and w4 have been revealed.
Similar structures can be generalized as 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. The value at each internal node can be seen as an intermediate pseudonym, depending on one or more individual's attributes. 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.

2.5 Pseudonyms with proof of ownership.

In special cases, pseudonymisation may 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): for example, in cases where the data controller does not have access to original identifiers but only to pseudonyms [16], 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.
Although the pseudonym is a type of an identifier, if its association with a specific data subject cannot be appropriately established, to avoid that the data controller cannot satisfy relevant requests, 'pseudonyms with proof of ownership' are required: a pseudonym P is created by a data subject from a given identifier ID and later transferred to a data controller. The hiding property states that the data controller must not be able to recover any information from the pseudonym P: 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 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

---

is the same as the original set of transactions, but not to obtain the original set of transactions. Bitcoin's software does not run the entire block of transaction data – representing 10 minutes' worth of transactions on average – through the hash function at one time, however. Rather each transaction is hashed, then each pair of transactions is concatenated and hashed together, and so on until there is one hash for the entire block. (If there is an odd number of transactions, one transaction is doubled and its hash is concatenated with itself). Visualized, this structure resembles a tree" in Jake Frankenfield, Merkle Tree, Investopedia, Feb 18, 2020 in https://www.investopedia.com/terms/m/merkle-tree.asp.
[16] This is a rather common situation in experimental clinical studies (RCT, 'randomized controlled trial', typically) that take place in various centers, often in different countries. It may happen that the patient resident in a State wants to request his/her data from the Study Coordinator at the 'Sponsor', which can be located in a different country: in such cases, the pseudonymisation techniques used by the data controller(s) must allow access to the data to take place with the appropriate guarantees but without any problem or delay for the data subject (patient).

properties, hiding and binding, can be achieved by 'cryptographic commitment scheme'[17], having 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), he/she needs to succeed a challenge/response protocol and to open the commitment; the data controller asks the subject to sign using its private key SKa, providing also all the values needed to let the data controller verify the pseudonym: it includes, apart from the signature R, the public key PKa and the value k.
To check if any request made by the data subject related to a specific pseudonym is legitimate or not, the data controller must ensure that PKa matches the pseudonym and that the signature is correct using PKa.

2.6 Secure multiparty computations.

A secure Multiparty Computation[18] (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. Applications include 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, by means of a 'oblivious Pseudorandom Function'[19](PRF) F;

---

[17] *"The notion of commitment is at the heart of almost any construction of modern cryptographic protocols. In this context, making a commitment simply means that a player in a protocol is able to choose a value from some (finite) set and commit to his choice such that he can no longer change his mind. He does not however, have to reveal his choice - although he may choose to do so at some later time. There are many ways of realizing this basic functionality, some are based on physical processes, e.g. noisy channels or quantum mechanics, while others are based on distributing information between many players connected by a network. We will say a bit more about this later, but for now we will concentrate on the scenario that seems to be the most obvious one for computer communication: commitments that can be realized using digital communication between two players.*
As a very simple example of this kind of commitments, consider the case where P has a pair of RSA keys, where V (like anyone else) knows the public key with modulus n and public exponent e. To commit to a bit b, P can build a number xb, which is randomly chosen modulo n, such that its least significant bit is b. Then he sends the encryption C = x e b mod n to V . We do not prove anything formal about this scheme here, although that is in fact possible. But it should be intuitively clear that P is stuck with his choice of b since the encryption C determines all of xb uniquely, and that V will have a hard time figuring out what b is, if he cannot break RSA. Thus, at least intuitively, the binding and hiding requirements are satisfied" in Damgård, Ivan & Nielsen, Jesper, Commitment Schemes and Zero-Knowledge Protocols (2007). Lecture Notes in Computer Science – LNCS in https://www.researchgate.net/publication/245702839_Commitment_Schemes_and_Zero-Knowledge_Protocols_2007, pages 1-2.

[18] *Cf. "Secure multi-party computation (MPC) enable a group to jointly perform a computation without disclosing any participant's private inputs. The participants agree on a function to compute, and then can use an MPC protocol to jointly compute the output of that function on their secret inputs without revealing them. Since its introduction by Andrew Yao in the 1980s, multi-party computation has developed from a theoretical curiosity to an important tool for building large-scale privacy-preserving applications" (page 5) .. "The term secure computation is used to broadly encompass all methods for performing computation on data while keeping that data secret. A computation method may also allow participants to confirm the result is indeed the output of the function on the provided inputs, which is known as verifiable computation" (ibidem) .. "The goal of secure multi-party computation (MPC) is to enable a group of independent data owners who do not trust each other or any common third party to jointly compute a function that depends on all of their private inputs. MPC differs from outsourced computation in that all of the protocol participants are data owners who participate in executing a protocol" (page 8) in David Evans, Vladimir Kolesnikov and Mike Rosulek, A Pragmatic Introduction to Secure MultiParty Computation. NOW Publishers, 2018. (This version: April 15, 2020) available at https://www.cs.virginia.edu/~evans/pragmaticmpc/pragmaticmpc.pdf.*

[19] *"Before defining pseudorandom function, we first recall the definition of a random function. We can describe a random functions in two different ways: a combinatorial description—as a random function table—and compuational description—as a machine that randomly chooses outputs given inputs and keeps track of its previous answers. In the combinatorial description, the random function table can be view as a long array that stores the values of f . So, f(x) returns the value at position nx. The problem with random functions is that (by definition) they have a long description length. So, we cannot employ a random function in our encryption scheme. We will next define a pseudorandom function, which mimics a random function, but has a short description. Intuitively, a pseudorandom function (PRF) "looks" like a random function to any n.u. p.p.t. adversary. In defining this notion, we consider an adversary that gets oracle access to either the PRF, or a truly random function, and is supposed to decide which one it is interacting with. More precisely, an oracle Turing machine M is a Turing machine that has been augmented with a component called an*

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 Fk(v) (without learning the key k) and S does not learn the input v. So, if the private lists consist of personal data, a secure MPC protocol actually provides the means for sophisticated pseudonymisation schemes: assuming that xi, yj are e-mail addresses of users, 1 ≤ i, j ≤ n, the outputs of the PRF function Fk(xi) and Fk(yj) are actually pseudonyms, where the key k is the pseudonymisation secret. Such techniques for private set intersection may be the proper solutions in terms of personal data protection in situations where comparison of two different lists from two different data controllers are required without revealing anything else than their common entries, such as, for example, if two health insurance companies wish to ensure that no one has taken out the same insurance with both of them. It is used also for advertising purposes, i.e. measuring adv 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.

2.7 Secret sharing schemes.

Secret sharing schemes can be seen as specific instances of secure Multiparty Computation (MPC) protocols. They are well known cryptographic techniques, aiming to appropriately split a secret information D into n parts D1, D2, . . ., Dn 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 and this is an approach to securely manage a secret cryptographic key: if on the one hand, storing the key in a single, well-guarded location is unreliable in terms of single misfortune or corruption, on the other hand 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 available: 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 Adi Shamir stated, "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".

Such a technique can be used to split an identifier into distinct pseudonyms: let us assume that the Pseudonymisation Entity substitutes (through a mapping procedure)the user's identifier 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.

In case of need, these pseudonyms can be used for re-identification, too, but only if a well-determined number of the recipients agree to exchange their different pseudonym for the same entity pseudonyms. For example, this approach could be used to pseudonymise auditing log files of a system so as to ensure that pseudonymisation reversal will occur only if a suspicious activity is present: then, 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

---

oracle: the oracle receives requests from M on a special tape and writes its responses to a tape in M. We now extend the notion of indistinguishability of distributions, to indistinguishability of distributions of oracles. The intuition about why f is a pseudorandom function is that a tree of height n contains 2n leaves, so exponentially many values can be indexed by a single function with n bits of input. Thus, each unique input to f takes a unique path through the tree. The output of f is the output of a pseudorandom generator on a random string, so it is also pseudo-random" in Raphael Pass, Abhi Shelat, A Course in Cryptography, 2010, pages 94 - 97 in https://www.cs.cornell.edu/ courses/cs4830/2010 fa/lecnotes.pdf.

identification of users from their relevant log data is possible. Another secret sharing scheme has been used to protect vehicular identity privacy in a Vehicular Ad Hoc Network (VANET), constituting a main application field for the Internet-of-Things (IoT), which generally poses demanding challenges to the personal data protection.
Because of the inherent properties of secret sharing schemes, we can conclude that the pseudonymisation secret is somehow shared across multiple entities (this could be a case of joint controllership). In fact, in the above scenario, each pseudonym also plays, in a sense, the role of a part of the secret of the pseudonymization: indeed, in a (k, n) scheme, combination of any k such shares (but no less) suffices to extract the original identifier.
This idea of sharing the secret of pseudonymisation can also be applied in other pseudonymisation techniques, for example, assuming a technique where the pseudonymization secret itself is a secret key.
Since

- ✓ *the secret of the pseudonymisation is inextricably linked to the additional information necessary to attribute the pseudonymous data to a specific data subject, and*
- ✓ *the GDPR establishes that, pursuant to Article 4 (5), such additional information must be kept separately and be subject to technical and organizational measures,*

it is clear that secret sharing schemes can provide the means to achieve these goals. Therefore it is understood how complex the task of selecting the optimal parameters and settings for sharing secrets can become to securely store the secret data shares, while satisfying all the requirements of the end user and all the 'boundary conditions' .

3. Conclusions and recommendations for all relevant stakeholders.

3.1 Defining the best possible technique

A risk based approach[20] to pseudonymisation is fundamental to 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 data controllers and processors to define the best possible option. For instance, although simple hash would not provide adequate data protection in most cases, appropriate elaboration of this technique (as in the case of chaining mode or Merkle trees) can significantly increase the protection level.
Regulators (e.g. Data Protection Authorities and the European Data Protection Board) should promote risk-based data pseudonymisation by providing relevant guidance and examples.

3.2 Advanced techniques for advanced scenarios

Technical solutions are of course a critical element for achieving correct pseudonymisation, but the organizational model and underlying structural architecture are 'critical success factors' (CSF) as well. In practice, before implementing a solid pseudonymisation technique, you need to make sure that the entities involved (and the associated data flow scheme) will be able to support it. Data controllers and processors should always imagine scenarios capable of supporting advanced pseudonymisation techniques, based, inter alia, on the principle of data minimization.

---

[20] Not only taking into account specifically the risks for rights and freedoms of natural persons, as required by the GDPR, but also in a broader sense:"A holistic approach proceeds from an accurate overview of the risk landscape—a governing principle that first of all requires accurate risk reporting. The goal is to empower organizations to focus their defenses on the most likely and most threatening cyber risk scenarios, achieving a balance between effective resilience and efficient operations. Tight controls are applied only to the most crucial assets. The holistic approach lays out a path to root-cause mitigation in four phases: 1. Identify risks and risk appetite. 2. Analysis and evaluation. 3.Treatment. 4.Monitoring. .. Among the most important instruments for fostering discipline throughout the organization are scheduled status updates to senior management on top cyber risks, treatment strategy, and remediation" in Jim Boehm, Peter Merrath, Thomas Poppensieker, Rolf Riemenschnitter, and Tobias Stähle, Cyber risk measurement and the holistic cybersecurity approach, McKinsey, November 2018 Risk Practice, in https://www.mckinsey.com/~/media/McKinsey/BusinessFunctions/Risk/OurInsights/Cyberriskmeasurementandtheholisticcybersecurityapproach/Cyber-risk-measurement-and-the-holistic-cybersecurity-approach-vf.pdf, pages 3 to 5.

The research community should indicate to data controllers and processors the elements of trust and the necessary guarantees so that the advanced scenarios are then functional in practice.
Finally, regulators should ensure that regulatory approaches, e.g. relating to new technologies and application sectors, consider all possible subjects and roles from the data protection point of view, while remaining technologically neutral.

3.3 Establishing the state-of-the-art

Much remains to be done to define the state of the art, for example, to work on more complex cases and their possible future evolution in the light of emerging technologies. To this end, research and application scenarios need to go hand in hand, involving all stakeholders (researchers, industry and regulators) to discuss joint approaches.
The European Commission, relevant EU institutions, as well as Regulators should support the creation and maintenance of the state of the art in pseudonymisation, bringing together fields (regulators, research community and industry).
The research community should continue its efforts to advance existing work on data pseudonymization, addressing the special challenges that arise from emerging technologies, such as AI.

3.4 Towards the broader adoption of data pseudonymisation

Advancements such as

- *CJEU Schrems II Judgment[21]:"Case C-311/18 Data Protection Commissioner v Facebook Ireland Ltd and Maximillian Schrems - Judgment of the Court (Grand Chamber), 16 July 2020. The Court of Justice invalidates Decision 2016/1250 on the adequacy of the protection provided by the EU-US Data Protection Shield[22]";*
- *the increasing need for open data access;*
- *big data: "in the case of big data, it is difficult to guarantee the confidentiality of a person or a group of people when databases from different sources are combined, even when the data were entered in an encoded or anonymized form[23]"*

are enlighting the need to further advance appropriate safeguards including supplementary measures for personal data protection, including broader adoption and real world usage of pseudonymisation in different application scenarios.

---

[21] Cf."In the view of the Court, the limitations on the protection of personal data arising from the domestic law of the United States on the access and use by US public authorities of such data transferred from the European Union to that third country, which the Commission assessed in Decision 2016/1250, are not circumscribed in a way that satisfies requirements that are essentially equivalent to those required under EU law, by the principle of proportionality, in so far as the surveillance programmes based on those provisions are not limited to what is strictly necessary. On the basis of the findings made in that decision, the Court pointed out that, in respect of certain surveillance programmes, those provisions do not indicate any limitations on the power they confer to implement those programmes, or the existence of guarantees for potentially targeted non-US persons. The Court adds that, although those provisions lay down requirements with which the US authorities must comply when implementing the surveillance programmes in question, the provisions do not grant data subjects actionable rights before the courts against the US authorities", we can read in the 'RESUME'' at http://curia.europa.eu/juris/document/document_print.jsf?docid=228728.

[22] For detailed analysis and commentary, see Sergio Guida, Caso Schrems II: Corte di Giustizia dell'Unione europea invalida la Decisione della Commissione sull'adeguatezza della protezione fornita dallo 'scudo UE-USA per la privacy' (Privacy Shield), in Data Protection Law-Rivista di Diritto delle nuove tecnologie, privacy e protezione dati, July 27,/2020 in https://www.dataprotectionlaw.it/2020/07/27/caso-schrems-ii-corte-di-giustizia-dellunione-europea-invalida-la-decisione-della-commissione-sulladeguatezza-della-protezione-fornita-dallo-scudo-ue-usa-per-la-privacy-privacy/.

[23] Cf Pilar Sanz, Key Points for an Ethical Evaluation of Healthcare Big Data. Processes. 7. 493. (2019) in https://www.researchgate.net/publication/334852654Key_Points_for_an_Ethical_Evaluation_of_Healthcare_Big_Data.