Technology Primer: Overview of Technological Solutions to Support Privacy-Preserving Record Linkage

Version 1.0, December 7, 2017

Primer Writing Team: Dixie Baker, Mark Phillips, David van Enckevort, Peter Christen, Ken Gersing, Maximilian Haeussler, Cenk Sahinalp, and Adrian Thorogood

1. Background

This Technology Primer is intended to serve the dual purposes of:

1) Informing the policy work of the Global Alliance for Genomics and Health (GA4GH) Regulatory and Ethics Working Group (REWG) and the International Rare Diseases Research Consortium (IRDiRC) by identifying and characterising technology approaches and solutions designed to enable coded data records associated with the same individual to be linked without disclosing the individual's identity; and
2) Recommending to GA4GH and to IRDiRC existing and emerging privacy-preserving record linkage (PPRL) technology approaches and solutions that warrant further exploration and assessment with respect to the functional, performance, and scalability requirements of the data stewards, data service providers, and researchers who support the GA4GH ecosystem.

The primary motivation of the GA4GH in this endeavour is its conviction that because linkage enables the creation, availability, and precision of data, it therefore improves the quality of both research and the health care provided to people. The GA4GH believes that this reinforces the right to share in scientific advancement and its benefits as guaranteed by Article 27 of the Universal Declaration of Human Rights, as mobilized in the GA4GH Framework for Responsible Sharing of Genomic and Health-Related Data.

2. Problem Statement

The Privacy-Preserving Data Linkage (PPRL) project addresses two primary problems that lie at the intersection of biomedical research and clinical practice:

1. The de-duplication and linking of datasets for use by researchers, without disclosing the participant's identity; and
2. The re-identification of research participants for clinical purposes, such as to return results that may be useful in clinical diagnosis or treatment.

This Primer describes methods and technologies capable of providing highly reliable linking (also known as “matching”) of coded[1] data records associated with the same individual *without disclosing the identity of that individual*. Records to be linked may be held in a single dataset containing records from multiple sources, or may span datasets.

The overarching goals of avoiding duplication of records and accurately linking records associated with a single individual are quite similar to the goals of enterprise master patient indexing (EMPI) commonly used to manage patient identity within clinical settings. However, EMPI technology aims to accurately associate records with the proper *identity*, whereas PPRL seeks to link records *without disclosing personal identity attributes*.

For the purpose of this primer, we assume that the identifiability of the data associated with the coded records, whether taken alone or in combination, is nearly nonexistent, or is at least reduced to an acceptable level for the particular purpose. This must be assured, in practice, to avoid defeating the purpose of the exercise, but is beyond the scope of this document.[2]

Recognizing the potential value of the capability to re-identify an individual, when research results offer high value (perhaps life-critical) to appropriate diagnosis and treatment, the capability for authorized entities to re-link identity to coded data is highly desirable.

2.1 Use Cases

The PPRL Task Team is focusing on use cases that address record-linkage within a research context, wherein the records to be linked have been stripped of identifying data elements, and coded so as to conceal the identity of the individual to whom they pertain. The PPRL Task Team has identified and defined the following three use cases:

1. De-duplication of records associated with the same participant within the same database or when databases are aggregated
2. Discovery of additional data associated with the same individual across multiple coded datasets
3. Ensuring that each individual is uniquely represented in a study sample in order to increase research reliability and validity

2.2 Desired Features and Attributes

The objective is to identify and recommend for further consideration one or more approaches to enabling linkage of coded data across organizations such that besides the fact that records have been linked, no information about the identity of the individual to whom the data pertain is revealed or can be ascertained.

Consistent with the ethico-legal considerations discussed in the companion *Ethico-Legal Primer*, the

[1] The term “coded” refers to a record whose identifiers have been removed and replaced with a code that is generated independently of the values of identity attributes, making derivation of the participant’s identity impossible without access to the key linking the code to the identifying attributes. Also known as pseudonymisation.

[2] See e.g. techniques and metrics that provide this assurance, such as *k*-anonymity.

following desired features and attributes have been identified:

- The approach should recognize, with a high degree of confidence, coded records associated with the same individual.
- The approach should be applicable to any data type (e.g., text, clinical data, images, genomic data).
- The approach should use a linkage algorithm that does not require the knowledge of the individual's direct identifiers.
- The approach should not inherently fail to recognize records associated with the same individual due to spelling differences, typographical errors, missing and out-of-date data, and other minor irregularities.
- The approach should enable a participant to limit linkages to her data.
- The approach should use techniques that are resistant to re-identification attacks (e.g., frequency, dictionary, cryptanalysis), while enabling re-identification when required and authorized.
- The approach should enable an assessment of linkage quality and completeness.
- The approach should be scalable and distributable, allowing linkage of very large datasets across multiple organizations.
- The approach should have been implemented for use, and not simply theoretical.

3. Current State of Knowledge and Practice

Within research environments like those in which GA4GH and IRDiRC generally work, PPRL is most often used for the purpose of creating a research dataset in which all records pertaining to the same person are linkable, even as the identity of the person remains unknown to the researchers. Accomplishing this presents two different kinds of challenges in an international ecosystem that highly values privacy. The first relates to legal restrictions that sometimes regulate the collection, use, and disclosure of specific attributes including a person's name, gender, birthdate, and place of birth, such as in HIPAA.

The second is that the more records a data set contains pertaining to the same individual, the easier it will be to identify that individual – a problem called statistical disclosure control. This tendency simultaneously empowers Big Data analytics. This challenge relates to the attributes that can be used to identify an individual, such as name, gender, birthdate, and place of birth. Most countries and jurisdictions have laws regulating the collection, use, and disclosure of such data. Further, the GA4GH Privacy and Security Policy states that:

> "Any attempt to re-identify individuals should be strictly prohibited, except where expressly authorised by the Data Donor or authorised under the law. This obligation follows the Data through the data sharing chain. Data Stewards should monitor data usage on a regular basis to detect any such re-identification attempts."

So any PPRL approach adopted by GA4GH needs to carry with it techniques for protecting the linkages. The set of techniques for record linkage without revealing identity is the domain of "privacy-preserving record linkage (PPRL)." [1]

Experts studying identifiability distinguish between *direct identifiers* such as a person's personal unique identifier (PUID) and, in most contexts, their name, on the one hand, and *quasi*-*identifiers*

(QIDs), such as gender, date of birth, and address, on the other. If a one-to-one mapping of individual-to-code is likely possible, the code is considered a direct identifier. QIDs also, however, play an important role in PPRL.

To be most useful, the records to be linked need to include a common set (or subset) of attributes, and the attributes need to be expressed such that they are recognizable across data sets (e.g., spelling consistency, common metadata, controlled vocabulary). PPRL involving Big Data, such as genomic data, presents additional challenges, including scalability, linkage quality, and increased privacy risk [2]. In some contexts, these challenges can be addressed by pre-processing techniques, such as those described by Christen [9].

3.1 Record Linkage

Record linkage is generally accomplished using one of three basic types of protocols [9]:

1) 1. Two-party protocols are used when only two database owners want to link their data.
2) 2. Three-party protocols are used when two parties are assisted by a trusted third party, enabling the linkage to occur without either party seeing the other's data.
3) 3. Multi-party protocols are used to link more than two data sets, and may involve a trusted third party.

Most record-linkage processes include the following activities, as shown Figure 1 [3].

1. Data Preprocessing — Real-world data often need to be prepared for matching by filling in missing data, removing unneeded values, translating data into a well-defined and consistent form, and resolving inconsistencies and typographical errors in data representations and encodings. As shown in Figure 1, data pre-processing can be conducted at each individual data source.
2. Indexing — Metadata are applied to records across all datasets in a consistent manner to enable efficient searching.
3. Comparison — Candidate record sets are compared using similar functions. Some PPRL approaches, especially for Big Data applications, include a blocking/filtering step to prune very likely non-matches to reduce the number of comparisons that need to be made.
4. Classification — Weight vectors of compared candidate record sets are given as input to a decision model that will classify them into matches, non-matches, and possible matches.
5. Clerical Review — The record sets that are classified as "possible matches" are manually assessed and classified into matches or non-matches.
6. Evaluation — Evaluation of the complexity, quality and privacy of the linkage to measure the applicability of a linkage project in an application before implementing it into an operational system.

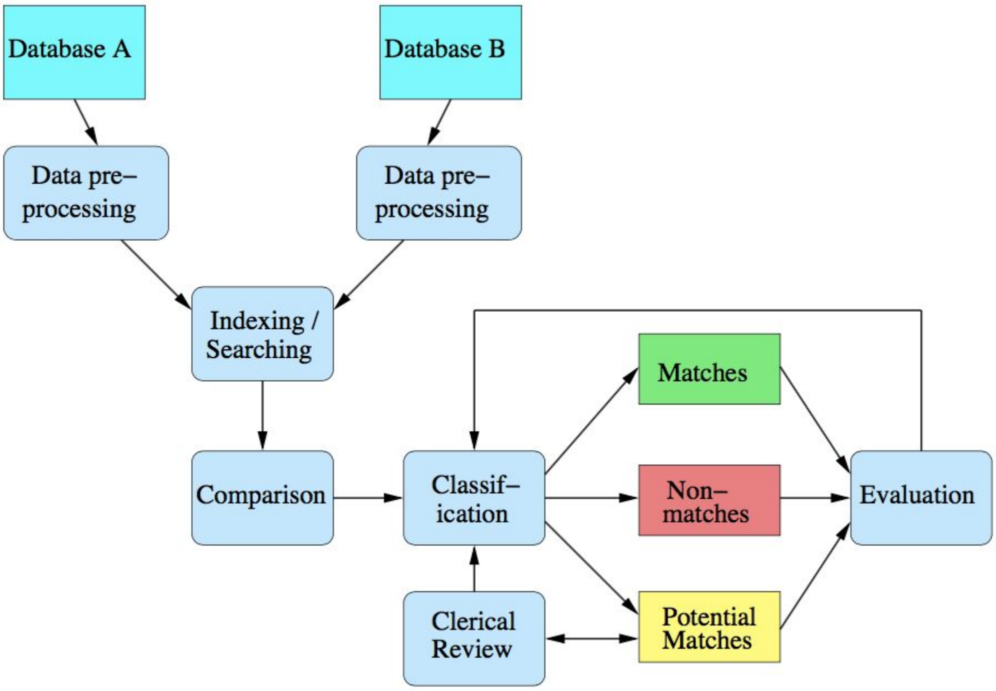


Figure 1. The record-linkage process typically includes the steps shown here.

To preserve privacy, data preprocessing may include removal of direct identifiers and quasi-identifiers, passing only pseudonymised data to the indexing step. Figure 2 shows how the nominal record-linkage process is modified to overlay a privacy-preserving context [3].

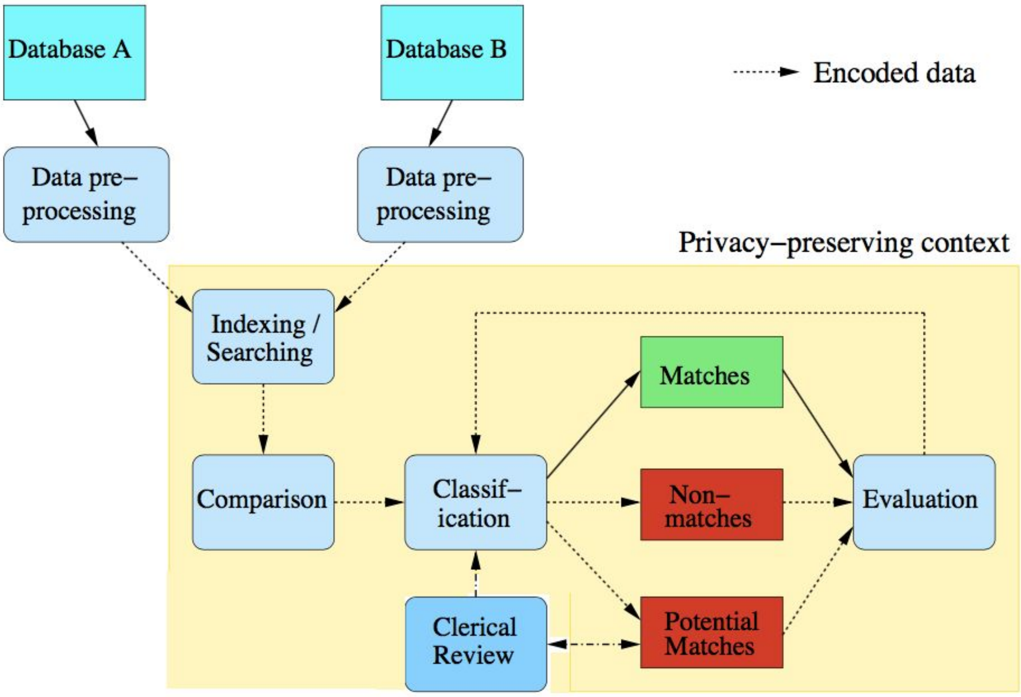


Figure 2. Privacy-preserving record linkage operates primarily using pre-processed, encoded data.

3.2 PPRL Techniques

PPRL techniques have evolved over time. First-generation techniques (mid-1990s) were primarily based on exact matching using simple hash encoding. The U.S. National Institutes of Health (NIH) Global Unique Identifiers (GUID) approach is an example of this technique [4]. These techniques are challenged by the fact that a single-letter difference in the attribute values used will yield dramatically different hash values.

Second-generation techniques (early 2000s) rely upon approximate matching and include comparisons of edit distances and other string-comparison functions. The principal limitation of these techniques is scalability. Third-generation techniques (mid-2000s) take scalability into account and often represent a compromise between privacy-protection and scalability; these techniques may allow for some information leakage [3].

A large number of matching approaches and protocols have involved some combination or extensions of the techniques discussed here [2]. Table 1 compares the advantages and disadvantages of these techniques.

3.2.1 Secure Hash Encoding

A cryptographic hash function is a one-way algorithm that, given any size string of characters as input, will produce a unique, repeatable, fixed-size output. Hash functions are commonly used for a variety of tasks, including to confirm that a message string communicated over a network arrives at its destination without having been modified in any way during transmission. This is accomplished by running the string through a hash function before it is transmitted, and again, using the same hash function, once it has arrived at its destination, and verifying that the two resulting hash values are identical. The most common sets of hash algorithms used for this purpose are the Secure Hash Algorithms (SHA-1, SHA-2) and Message Digest (MD5).

A hash value (i.e., output from a hash function) does not disclose any of the content itself: because a "one-way" algorithm is used, an adversary cannot derive the content from the hash value. However, dictionary and frequency attacks are possible. For example, if an adversary believes that a given hash value was generated from the string "Preserve privacy" and knows the hash function that was used to generate the hash, they will be able to reproduce the hash value to confirm or refute this suspicion. In some circumstances, such as password authentication systems, it is possible to mitigate or overcome dictionary attacks by injecting a random known value into the hash input known as a "salt", giving rise to the widely-used salted hash technique.

Note that hash functions test only exact matches, and have no inherent capacity to handle near matches. Therefore, if portions of clinical records are used as the input to a given hash function, one of which identifies a condition as "autism" while another says "autistic spectrum," for example, the two will generate distinct hash values bearing no relationship to one another.

Hash functions play a major role in PPRL because of their ability to reliably confirm matching inputs without directly revealing any information about the content of those inputs.

3.2.2 Statistical Linkage Key

A statistical linkage key (SLK) is a derived variable generated from components of QIDs. The SLK-581, developed by the Australian Institute of Health and Welfare (AIHW) to link health datasets, is an example of a statistical linkage key. The format of the complete SLK-581 is:

XXXZZDDMMYYYYM.

The SLK-581 comprises four elements:

- The second, third, and fifth letters of the client's family name (XXX)
- The second and third letters of the client's given name (ZZ)
- Date of birth, represented as 2-numeral day (DD), 2-numeral month (MM), and 4-numeral year (YYYY)
- Sex, represented as male (M), female (F), and unknown (U) [6]

Additionally, the SLK-581 is submitted with a companion element indicating the accuracy of the date of birth [6]. However, SLK-based masking has been shown to provide limited privacy protection and poor sensitivity (reported in [2]).

Also, because an SLK is derived from identity elements, it would not meet the GA4GH requirements that the approach "not require knowledge of the individual's identity" and "use techniques that are resistant to re-identification attacks."

3.2.3 Secure Multiparty Computation

Encryption schemes that support PPRL approaches typically are those that enable secure multi-party computation (SMC), which are cryptographic methods in which multiple parties jointly compute a function while keeping their individual inputs private. During the computation, each participating party computes part of the function, and in the end, each party knows only the end result and its own input.

Two computing schemes that support SMC are commutative encryption and homomorphic encryption. A "commutative" encryption algorithm is one that enables multi-party computations to occur in any order, with the same result. "Homomorphic" encryption is encryption that allows computations to be carried out using encrypted data, generating an encrypted result that, when decrypted, is the same result that is produced when the same computations are performed using unencrypted data. The most commonly used SMC techniques for PPRL are secure set union, secure set intersection, and secure scalar product.

SMC techniques can impose high overhead costs, both for communications among the parties, and for computations that require strong encryption keys (i.e., thousands of bits long) [7]. Although several comparison and classification techniques for multi-party PPRL have been described in the literature, thus far they have failed to provide a practical solution either because they enable only exact matching or because they are computationally infeasible [2].

3.2.4 Bloom Filters

A Bloom filter is a data structure used to determine the likelihood that a data element is a member of a set, in a memory-efficient way. Specifically, a Bloom filter is a vector into which hash values are loaded. Multiple vectors are generated, and then compared, resulting in either a "definite no" or a "perhaps yes" match. The method is was first defined by Schnell, whose paper provides a detailed description of how Bloom filters work [1]. Bloom-filter encoding has been widely used as an efficient technique for matching records without sacrificing privacy [1, 2, 3].

To strengthen the reliability of the results, comparisons should be made using a number of hash functions. Bloom filters have been shown to be susceptible to cryptanalysis attacks, depending upon the number of hash functions used and the length of the Bloom filter. A number of variations on this approach have been proposed as countermeasures for such attacks and to improve linkage quality [2]. Because Bloom filters employ only vector comparisons of hashed digraphs, they are computationally efficient and practical. Another advantage of the Bloom filter approach is that it enables partial matches, with attendant quantitative measures of likelihood.

In 2013, Randall et al. reported results of a large study that compared linkage quality between those established using Bloom filters and traditional probabilistic methods using full, unencrypted personal identifiers. The study tested the linkage methods using ten years of New South Wales (NSW) and Western Australian (WA) hospital admissions data, comprising a total of over 26 million records. No difference in linkage quality was found when the Bloom filter results were compared to traditional probabilistic methods using full unencrypted personal identifiers [8].

A key advantage of Bloom filters is that they allow for data discrepancies. Bloom filters can be implemented using different identity fields, different hash algorithms, and different numbers of passes. Implementation decisions often involve a trade-off between strength of privacy and security, and quality of linkages. Thus implementation options are selected on a project-by-project basis. Another key advantage is that the data custodian remains in control of identities at all times; the linkage unit receives only a set of keys, and researchers receive only keys and content. Re-identification may be performed only by the data custodian [9].

Technique	Advantages	Disadvantages
Secure hash encoding	Irreversible; privacy-protective	Subject to dictionary and frequency attacks; Tests only exact matches
Statistical linkage key (SLK)	Simple to implement	Derived from identifiers
Secure Multiparty Computation (SMC)	Privacy protective Can be used to eliminate need for trusted third party	High or unknown computation and communication costs (solutions emerging, but not widely implemented)
Bloom filter (vectored hash values)	Tolerates data discrepancies – quantifiable “definite no”/”perhaps yes” Allows for design trade-off’s between privacy and linkage quality	Susceptibility to crypto-analysis depending on number of hash values

Table 1. Comparison of PPRL techniques

3.3 Challenges

3.3.1 Characteristics of Data Sets Affect Linkage Performance

van Grootheest et al. [10] studied record-linkage performance under simulated conditions and found that linkage performance is dependent upon several parameters.

1. Algorithm used. Probabilistic methods can identify more links than deterministic linkage algorithms, but an appropriate threshold must be chosen to avoid incorrect links.
2. Choice of linkage variables. When the use of a patient identifier is not possible, personal information such as sex, date of birth, name and address must be used.
3. Dataset size and overlap. In general, both the sensitivity and precision of probabilistic linkage decrease as the overlap becomes smaller and the datasets become larger.
4. Errors in datasets. Best linkage results are achieved when both datasets have been created or updated around the same time, and when the address history is recorded. Pre-processing can also help to standardise variables and remove common spelling mistakes.

3.3.2 Mutability of Personally Identifiable Information

Personally identifiable information (PII), such as date of birth and name at birth, usually are considered immutable; that is, they will never change. However from the perspective of clinical and research information systems, PII should be considered mutable, since mistakes and discrepancies in recording the information often occur. Where a clinical information system will keep an audit trail of changes to these data, any system that may receive the data is unlikely to have access to this change record.

Language is another source of discrepancies in PII. For instance, place of birth names might have different spellings, even within a single country, e.g., "Brussels," "Brüssel," "Brussel," "Bruxelles." Some mechanisms are available to make a system that uses PII as input more robust against mistakes in the spelling of a name. For instance, calculating a hash of each datum separately could detect a potential match if multiple but not all of the hashes match. Another approach is to normalize or codify the data, which can be especially useful in eliminating differences based on language. For example, in the example above, replace "Brussels" with a standardized code. Any system that uses mutable data as the input for linkage runs the risk of losing the ability to link if the input data changes over time.

For reasons including these, some have turned to biometric data as more immutable for the purpose of linkage. This approach can pose its own challenges, however. First, any use of biometric data that are reliably immutable (e.g., fingerprint, DNA) poses a significant privacy risk. Second, if a PPRL relies on biometric data that are expensive or cumbersome to collect, such as whole-genome sequences, then in an international context, the scheme will necessarily exclude any potential member project that lacks the capacity to collect or generate those data on demand.

3.3.3 Computational Practicality

As noted in section 3.1 above, some approaches to PPRL are technically interesting, but may be computationally impractical. In particular, approaches involving asymmetric encryption of large quantities of data are computationally intensive. Hardware-based encryption, which implements encryption algorithms in hardware rather than software, dramatically improves the efficiency of encryption solutions; however, software-based solutions are more cost-effective and therefore more widely used.

3.3.4 Susceptibility to Attack

PPRL approaches are susceptible to several adversarial models and attack methods. Adversaries include "honest but curious" parties who follow protocol, while being curious to find out about another party's data, and malicious parties, who undertake concerted attack efforts. Malicious parties may be motivated by revenge for perceived wrong doing by the data holder, disagreement with a policy or practice, or a desire to demonstrate knowledge or expertise. Attack methods include dictionary attacks wherein an adversary encodes a list of known values using existing encoding functions until a match is found; frequency attacks in which a frequency distribution of encoded values is matched with a distribution of known values; cryptanalysis attacks, a frequency attack using encrypted or hashed values; and collusion, in which two or more parties work together to learn another party's data [3].

3.3.5 Revocation of Identifier, and Participant Rights

Another PPRL consideration is the potential to modify the linkage mechanism once established. For example, one might want to reassign new pseudonyms to those individuals affected by a security breach.

Because of the irreversibility of hash values, some have suggested using a hash value as a pseudonym.

However, if all hash values in a data set are generated using the same hash function, then assigning a new hash value to one person, or a subset of people, would affect everyone whose hash values were generated using the same hash function as the affected individuals. Also, if an adversary holds the name of a person, and the hash function used to generate the hash value associated with that person, matching pseudonym to identity is trivial. For these reasons, hash values are generally not used as pseudonyms. Instead, when hashing used in linkage systems, the hash values themselves are kept secret and associated with an independent, randomly generated identifier that is distributed and used as a pseudonym. Because the hash value is used only as a linkage back to the individual, a pseudonym can be revoked or changed without affecting every individual in the data set.

Another prospective consideration when designing a PPRL system is the degree to which it allows participants to exercise their rights to access and control their own data, such as their right to revoke their consent to the use or storage of their data, either in a particular project or in their inclusion in the PPRL scheme itself.

3.4 Approaches Currently in Use within GA4GH Ecosystem

3.4.1 U.S. National Institutes of Health Global Unique Identifier

The U.S. National Institutes of Health (NIH) developed the Global Unique Identifier (GUID) Tool as a customised, client-server, software application used to generate a Global Unique Identifier (GUID) for each study participant. The GUID is a subject pseudonym designed to allow a researcher to share data specific to a study participant without exposing personally identifiable information (PII), and to match participants across labs and research data sets.

To generate a GUID:

1. The researcher holding the PII executes the GUID client locally.

2. The researcher enters the following PII data elements: sex, first name, last name, middle name, date of birth, and city, municipality, and country of birth.

3. GUID client uses the PII data elements to generate a one-way hash value.

4. GUID client sends the one-way hash value to the GUID server at NIH.

5. If the hash value matches the hash value associated with an existing GUID, then that GUID (pseudonym) is returned to the researcher.

6. If the hash value does not match the hash value associated with any existing GUID, a new GUID is randomly generated and returned.

The GUID system offers the following advantages:

- No PII ever leaves the researcher’s computer.
- Because the GUID is randomly generated from a hash value that is generated outside the GUID server, there is no way to infer the identity of the individual based on the GUID alone.
- The same individual's information will result in the same GUID across time, location, and research study. This allows researchers to match shared data from that participant regardless

of source, without ever sharing or viewing PII. [4]

The disadvantage is that an attacker with access to the data elements associated with an individual, can use the GUID client to query the GUID server for the GUID associated with that individual.

This scheme uses the secure hash encoding matching scheme, and involves the use of a trusted third party that generates and manages GUIDs.

3.4.2 Mainzelliste (Germany)

Developed at the Johannes Gutenberg University Mainz, Mainzelliste is an open-source, RESTful service for pseudonymisation. A user inputs PII (e.g., name, date of birth) and receives back a pseudonym generated using data unrelated to the identifiable elements. The pseudonymisation service maintains a database of identifiable data strings matched to pseudonyms. Upon receiving the identifiable elements, the service performs a lookup to determine whether a pseudonym already exists. The lookup runs a linkage algorithm to account for near matches (e.g., typographical errors). If a match is found, the service returns the existing pseudonym. If no match is found, the service generates and returns a new pseudonym [5, 11].

Linkage is possible even in the event of typos or alternate spellings. Mainzelliste allows for the possibility of using in-house phonetic codes and string comparisons for linkage, thereby allowing names from other linguistic backgrounds to be fault-tolerantly compared. Currently, weight-based record linkage is supported, but the modular concept allows for retrofitting an in-house algorithm. The possibility to manually rework uncertain assignments further supports the automatic matching process [3].

3.4.3 European Patient Identity Management

The European patient-identity management solution (EUPID) approach addresses the risk associated with the GUID system, as described above, by using *context-specific* data elements and hashing algorithms to generate a *context-specific* pseudonym for each individual. The context-specific pseudonyms then are linked within the EUPID system and associated with a linkage pseudonym, without revealing the context-specific pseudonyms included in the association.

The EUPID approach, originally developed by the European Network for Cancer Research in Children and Adolescents (ENCCA), was designed to meet the following requirements:

- Prevent duplicate registration of patients.
- Preserve the capability to re-identify subjects by a trusted third party in special cases.
- Support the capability to use different pseudonyms for the same patient in different contexts, while preserving the capability for a trusted third party to link datasets pertaining to the same patient and stored under different pseudonyms — while assuring that patient identification in any single context is nearly impossible from another context.
- Avoid creating a transparent universal patient ID.
- Assure that the approach can be implemented in a distributed computing environment.

The EUPID scheme is illustrated in Figure 3, from Nitzlnader and Schreier [12], which provides detail regarding the methodology and how the EUPID linkage is generated and used in actual practice. A key feature is context-specific pseudonymisation, which maintains identity linkages locally, while enabling re-identification of a linked data set through a three-party collaboration involving the local context, the linkage agent, and a trusted third party. EUPID combines several of the PPRL matching techniques discussed in Section 3.2 to identify linkages (hash encoding, statistical linkage key, encryption), and an optional trusted third party to enable re-identification as authorized.

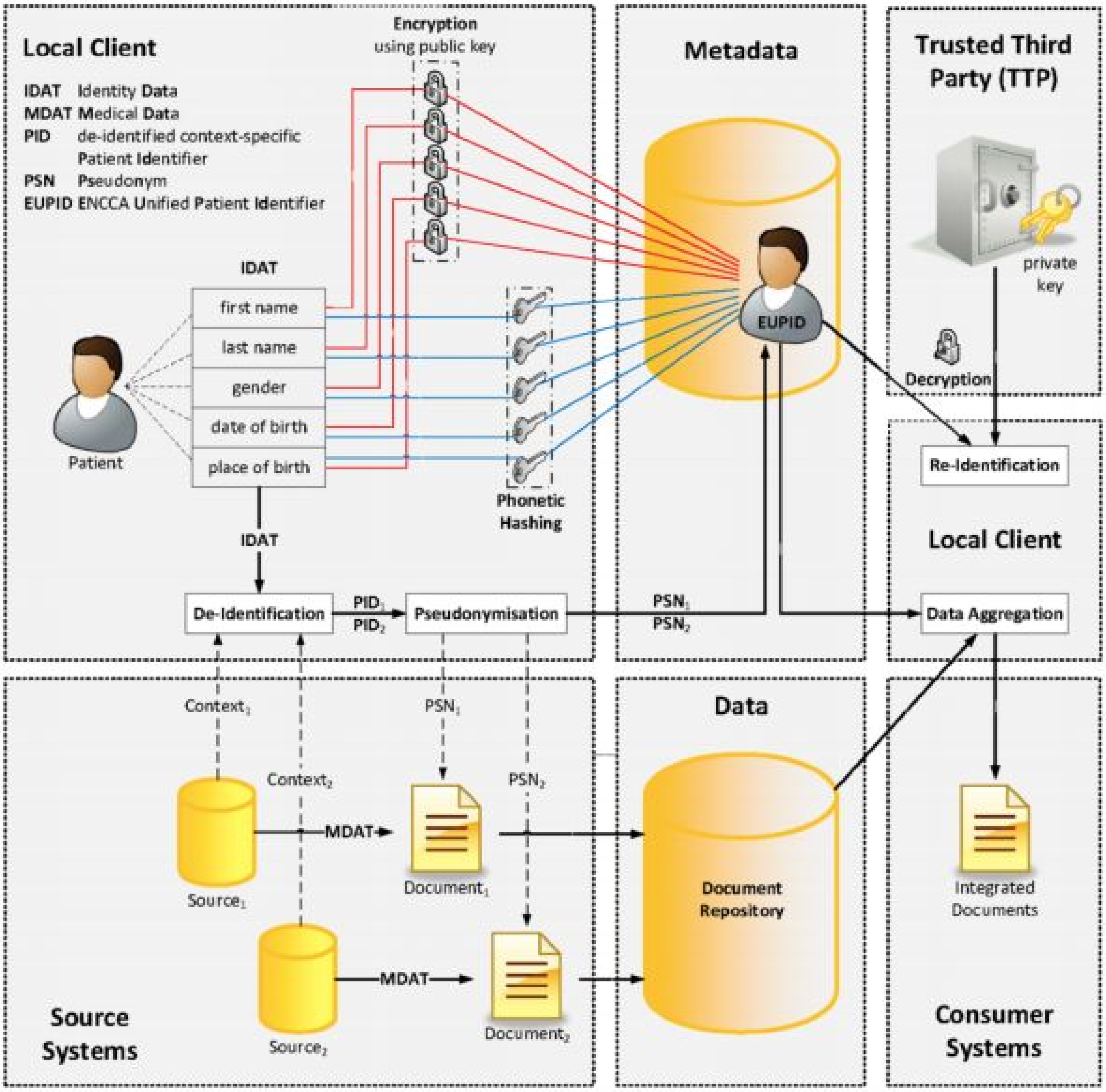


Figure 3. The ENCCA Unified Patient Identifier (EUPID) approach enables the use of context-specific pseudonyms (PSNs), while preserving the capability for a trusted third party to link PSNs pertaining to the same individual, through the use of indexed EUPIDs [12]..

The PPRL Task Team has concluded that the EUPID system may be well suited for the GA4GH federated environment, and further investigation has been completed in collaboration with the EUPID project team. More detail regarding the EUPID system is given in Appendix A.

5. Discussion

PPRL techniques and approaches generally use either a direct identifier (PUID) or a quasi-identifier (QID). The use of this information is regulated throughout the world under applicable jurisdictional laws and institutional policies. The mere centralization of data presents privacy challenges. In addition, considerations such as the need to enable participants to withdraw from a research study and the various policies relating to the return of results must be factored into the design of a PPRL approach and implementation.

Any PPRL approach must consider privacy risks inherent in its methods. A hash value generated using PUID or QID cannot be used as a pseudonym because the hash value is derived from personal data, which often is prohibited by applicable law or regulation. Also, use of a hash value as a pseudonym that does not allow re-identification and so may be illegal in some contexts due to the right of participants to withdraw, to access their own data, and to receive the results of research performed using their data returned when so desired.

Some of the technical approaches examined generate a random or quasi-random pseudonym, and store an association between the generated pseudonym and a hash value derived from PII. In this way, no participant can be identified on the basis of the pseudonym alone. However, a unique linkage between a PII-based hash value and a pseudonym leaves open the possibility of using the linkage system to reverse-discover the pseudonym associated with a known individual.

The Bloom filter approach may be less vulnerable to this type of reverse-discovery attack if a large number of hash functions and a sufficiently long filter are used [3]. The Bloom filter approach produces quantitative values reflecting the strength of the match, and has been shown to produce linkages comparable to those produced by traditional methods using unencrypted identity attributes [10].

The hope behind these initiatives is to simultaneously safeguard privacy while also furthering open data ideals such as the FAIR principles (not to be confused with FIPP, the Fair Information Practice Principles, discussed below), which demand that data be **F**indable, **A**ccessible, **I**nteroperable, and **R**eusable [13]. Since their emergence in the scientific context, FAIR principles have been recognized as particularly important where health-related data are concerned.

It is important to recognize that the very act of linking records pertaining to the same individual may make them easier to identify. Indeed, this is why many research programs require a minimum "bin size" or "cell size" -- i.e., that a minimum number of individuals be represented in any bar in a histogram ("bin") or any single "cell" in a spreadsheet. Linking two records within a "bin" essentially reduces the bin size, and increases privacy risk. In addition, a trusted third-party (used in all 3 of the implementations discussed above) becomes an attractive target for attackers.

A strong data-linkage approach should conform to the "principle of least privilege" wherein each entity has access to only those data and system privileges it needs to perform its assigned functions. EUPID takes a step in the right direction through its context-based pseudonymisation and collaborative approach to re-identification. But all three of the approaches discussed in Section 3.4 include a single point of failure. Methods that distribute trust across entities, such as multi-party computation (MPC) and federation, are potential avenues for addressing this vulnerability.

As in other areas of data-sharing, no technical solutions on their own can ensure both data privacy and data sharing. Any workable international PPRL solution will require strong privacy policy, enforceable through the combined use of technical methods, like those discussed above, and robust organizational and governance measures, perhaps taking inspiration from the Fair Information Practice Principles (FIPPs) that undergird most international privacy law.

6. Recommendations

After considering the methods and approaches discussed above, the PPRL Task Force concluded that the EUPID approach held the most promise for the emerging, global GA4GH and IRDiRC research environment. In particular, the Task Force was impressed with EUPID's use of context-specific identity attributes, hashing functions, and pseudonyms to localize privacy risk, and its use of phonetic hashing to enable robust linking. The model in principle can be federated and scaled to accommodate other consortia and data-sharing efforts. In addition, the model's "re-identification" capability could be offered as an optional module for contexts that require the capability to learn the identity of a research participant under special circumstances, and with appropriate authorisation.

The PPRL Task Force is collaborating with the EUPID project to deepen its understanding of the EUPID model and to further explore its use. A security review is planned as well as an investigation of the feasibility of using secure multi-party computation (SMC) as part of the federation model.

7. References

[1] Schnell, R. Privacy-preserving data linkage. Chapter 9 in Methodological Developments in Data Linkage. Katie Harron, Harvey Goldstein, Chris Dibben, Eds. pp. 201-225. Wiley. 2016. ISBN: 978-1-118-74587-8.

[2] Vatsalan, D. Z. Sehili, P. Christen, E. Rahm. Privacy-preserving record linkage for big data: Current approaches and research challenges. 2016. (in press).

[3] Christen, P. Privacy-preserving record linkage. ScaDS Leipzig, July 2016. Available from http://users.cecs.anu.edu.au/~christen/publications/christen2016scads.pdf (accessed 12/07/17).

[4] National Institutes of Health. Global Unique Identifier (GUID). Available from https://data-archive.nimh.nih.gov/guid/ (accessed 12/07/17).

[5] Mainzelliste. Accessible at https://mainzelliste.de (accessed 12/07/17).

[6] SLK-581 Guide for Use. Available from https://www.aihw.gov.au/getmedia/e1d4d462-8efa-4efa-8831-fa84d6f5d8d9/aodts-nmds-2016-17-SLK-581-guide.pdf.aspx (accessed 12/07/17).

[7] Hall, R. and S. E. Feinberg. Privacy-preserving record linkage. Available from https://www.cs.cmu.edu/~rjhall/linkage_survey_final.pdf (accessed 12/07/17).

[8] Randall, S.M., A.M. Ferrante, J.H. Boyd, J.K. Bauer and J.B. Semmens. Privacy-preserving record linkage on large real-world datasets. J Biomedical Informatics. 50 (2014) 205–212. Available from http://www.sciencedirect.com/science/article/pii/S1532046413001949 (accessed 12/07/17).

[9] Boyd, J and A. Ferrante. Center for Population Health Research, Curtin University. Perth, Western Australia. Personal communication with PPRL Task Team. 27 April 2017.

[10] van Grootheest, G, M.C.H. de Groot, D.J. van der Laan, J.H. Smit, and B.F.M. Bakker. Record linkage for health studies: Three demonstration projects. 2015. Available from http://www.biolink-nl.eu/public/2015_recordlinkageforhealthstudies.pdf (accessed 12/07/17).

[11] Medical Free/Libre and Open Source Software. Mainzelliste. Available from https://www.medfloss.org/node/1114 (accessed 12/7/16).

[12] Nitzlnader, M. and G. Schreier. Patient identity management for secondary use of biomedical research data in a distributed computing environment. eHealth2014 – Health Informatics Meets eHealth. A. Hörbst et al. (Eds.). 2014. The authors and IOS Press. doi:10.3233/978-1-61499-397-1-211. This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License at https://eupid.eu/assets/downloads/nitzlnader2014.pdf (accessed 12/07/17).

[13] FORCE11. Guiding Principles for Findable, Accessible, Interoperable and Re-usable Data Publishing version b1.0. https://www.force11.org/node/6062 (accessed 12/07/17).

8. Definitions

(Where available, these terms draw on the GA4GH Data Sharing Lexicon)

AIHW	Australian Institute of Health and Welfare
Anonymisation	The irreversible delinking of identifying information from associated data. In a clinical setting, data collected without associating an identity.
API	Application Programming Interface
Coding	A type of pseudonymization in which identifying information about each participant is replaced with a code, so as to make identification of the participant impossible without access to the key linking the code to the identifying information.
Collusion	Attack in which two or more parties work together to learn another party’s data.
De-identification	The process of removing data elements from PII in such a way that the identity of the individual to whom the data pertain cannot be derived from the resulting data.
Dictionary attack	An brute force attack vector whereby the attacker systematically trying all combinations in a list
EMPI	Enterprise master patient indexing
ENCCA	European Network for Cancer Research in Children and Adolescents
EUPID	ENCCA Unified Patient Identifier

Frequency attack	Attack in which a frequency distribution of encoded values is matched with a distribution of known values.
GA4GH	Global Alliance for Genomics and Health
GUID	Global Unique Identifier
HIPAA	US *Health Insurance Portability and Accountability Act of 1996*
HIPAA Privacy Rule	The HIPAA regulation that sets for the terms under which protected health information (PHI) may be used and shared, including the conditions under which PHI may be considered de-identified
IRDiRC	International Rare Diseases Research Consortium
MD5	Message Digest hash algorithm
MPI	Master Patient Index
NIH	U.S. National Institutes of Health
PHI	Protected Health Information under the U.S. HIPAA law
PII	Personally Identifying Information - (Identifiable/Personal Data): Data that alone or in combination with other data may reasonably be expected to identify an individual. PII includes both PUIDs and QIDs.
PPRL	Privacy-Preserving Record Linkage, i.e. techniques for record linkage without revealing identity
Pseudonymisation	Any of several techniques which produce output characterized by the fact that the individuals to whom data relates can only be identified with access to a mechanism crafted for that purpose. Pseudonymization techniques include coding, encryption, and tokenization.
PUID	Personal Unique Identifier

QID	Quasi-identifier
RESTful Service	A software style that allows interoperable access to and manipulation of resources using web-based Hypertext Transfer Protocol (HTTP)
REWG	GA4GH Regulatory and Ethics Working Group
SHA	Secure Hash Algorithm
SLK	Statistical linkage key, a derived variable generated from components of QIDs
SMC	Secure multi-party computation
SWG	GA4GH Security Working Group

Appendix A: ENCCA Unified Patient Identifier (EUPID) System

The principal high-level components of the EUPID system, and how data flow among these components, are described in Section A1. More technical detail describing the EUPID components and interactions shown in Figure 3 (section 3.4.3 above) is given in section A2.

A1. Top-Level Description

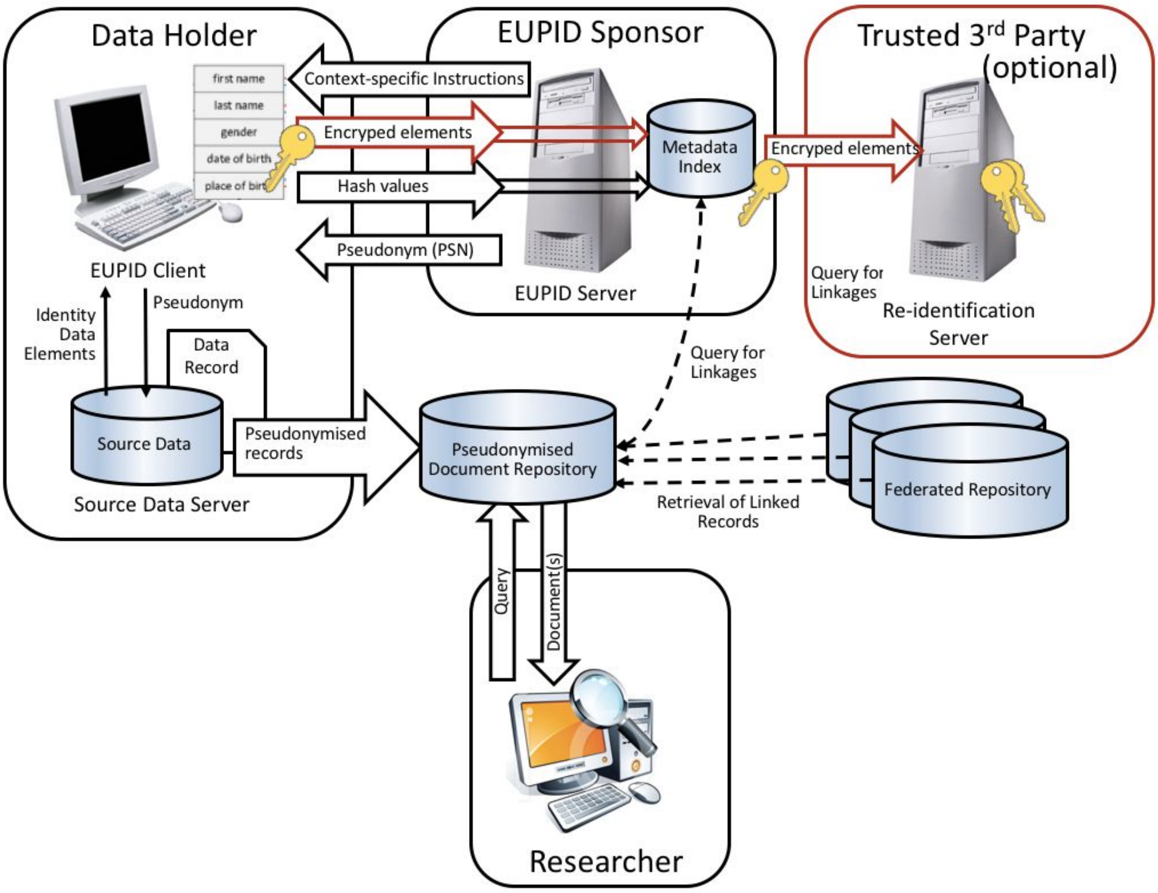


Figure A1. EUPID record-linkage components and data flows. Optional re-identification component and data flows shown in red.

When a data holder applies for authorization to use the EUPID Service, once approved, the data holder downloads the EUPID client software into a secured computer, and selects a trusted person to perform operations using the EUPID client. .

To help preserve individual privacy, the EUPID software operates on a "per context" basis. For example, one "context" might be hospital admissions, another "context" might be a cancer research lab, etc. The EUPID service provides the client software a set of these context definitions, each including a context identifier, a list of identity elements to use, and the method to use to generate the "hash values" the system requires. If the context supports re-identification, the context definition will include instructions for encrypting each identity element.

When the data holder wants to convert an identifiable data set to a coded data set, the EUPID client operator selects a context.suitable for the data set and re-identification requirements.   The operator extracts the prescribed identity elements, and runs each of them through the hashing software. If re-identification is planned, the operator runs the identity elements through the encryption software as well.

The EUPID client operator then transmits hash values (and optionally encrypted identity elements) to the EUPID server along with the identifier of the context definition used. The EUPID server returns a pseudonym (PSN) for each record in the data set. The data holder removes identifiers from each record, labels the record with the assigned pseudonym, and makes the coded records available to authorized researchers.

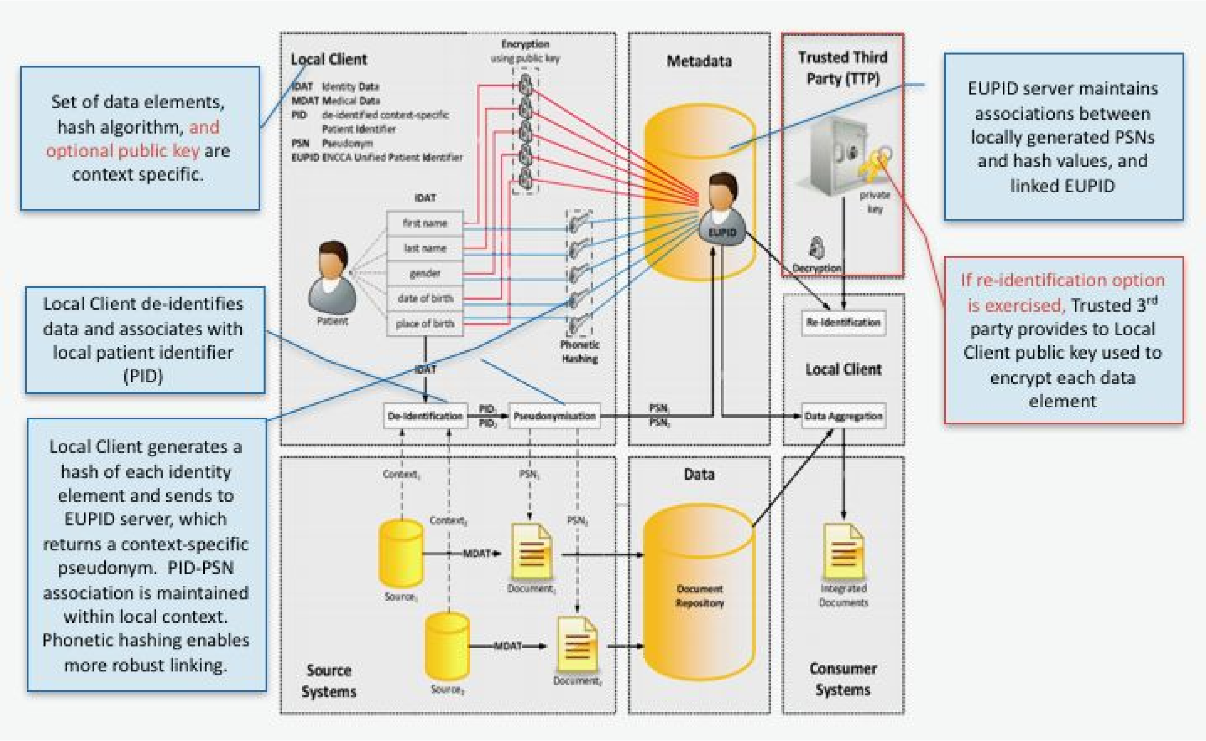
The EUPID server attempts to match the hash values of the identity elements of each context with those of other contexts stored in the server. This process also attempts to detect duplicate records associated with the same individual. Hashes of the identity elements, encrypted identity elements, and pseudonym associations are stored in a metadata index to enable searches for privacy-preserving data linkage (PPRL).

When a researcher[3] queries a pseudonymised document repository, any documents that match the query parameters and that are available within that repository are returned. The local repository may query the EUPID server for linked documents stored elsewhere. As allowed by policy, the EUPID server may return context-specific pseudonyms associated with the same individual. The local repository retrieves documents from federated repositories.

To enable re-identification, when required, an optional re-identification module operated by a trusted third party holds the link between the encrypted data elements held by the EUPID server, and the identities of the individuals. When an authorized user requests re-identification, the trusted third party retrieves the encrypted data elements from the EUPID server, decrypts them, and makes the identity available.

[3] Note that the term "researcher" is not restricted to individuals associated with research institutions, and includes, for example, clinicians investigating potential diagnoses, and citizen scientists performing independent investigations.

A2. Technical Detail of EUPID Pseudonymisation


Set of data elements, hash algorithm, and optional public key are context specific.

Local Client de-identifies data and associates with local patient identifier (PID)

Local Client generates a hash of each identity element and sends to EUPID server, which returns a context-specific pseudonym. PID-PSN association is maintained within local context. Phonetic hashing enables more robust linking.

EUPID server maintains associations between locally generated PSNs and hash values, and linked EUPID

If re-identification option is exercised, Trusted 3rd party provides to Local Client public key used to encrypt each data element

Figure A2. Annotated expansion of EUPID graphic given in section 3.4.3 and published in [12].

Figure A2 provides more technical detail about the EUPID components and how they interact. As noted in the previous section, each data holder applies to participate in the EUPID program. Once approved, the data holder is able to download the EUPID Client software and install it on a local machine, within a protected environment. The EUPID Server also returns a list of available contexts, along with the associated configuration for each context – including the set of identity data elements and hashing methods the EUPID client should use with the associated context. If the context includes re-identification, the encryption methods to use, and a context-specific public key (generated by the trusted third party) are included.

When the data holder needs to pseudonomise data for a given context (e.g., research project), the trusted operator of the client software selects an appropriate context and extracts from the source data the identity data elements required for the chosen context. The client runs the extracted identity elements through the phonetic hashing function to generate a hash value for each identity element, and transmits the hash values to the EUPID server along with a context identifier.

If the context requires re-identification, the client uses the context-specific public key to encrypt each identity element. The client includes the set of encrypted elements along with the hash values and context identifier in its transmission to the EUPID server. The EUPID server stores the hash values, encrypted elements, and context identifiers in its metadata index associated with a context-specific pseudonym (PSN) that is returned to the client. For each data record, the local client generates a hash value of the total set of identity elements extracted from that record, and associates this internal hash value with the PSN provided by the EUPID server. The data holder strips the record of identifying information, relabels the record with the PSN, and stores the pseudonymised record in a data repository made available to researchers. The EUPID program does not dictate who runs the pseudonymised repositories – they could be local, centralized, or federated across multiple data holders.

The EUPID Server runs algorithms against the set of hash values to detect potential duplicate records, and to identify other records associated with the same individual, but labeled with different context-specific PSNs. For example, a single individual may be represented in a research project context and in a clinical context. When matches are found, the EUPID server assigns a common, linkage pseudonym (an EUPID) to the matching records, while preserving the context-specific pseudonyms. To preserve the individual's privacy, the EUPID server holds these linkages in its metadata index and does not share them unless allowed to do so by policy and individual consents.

When a researcher queries a pseudonymised document repository, any records or documents held in that repository and matching the query parameters are returned. If the repository is federated with other pseudonymised repositories and record linkage is authorized, the repository may query the EUPID server for other PSNs associated with the same individual. As authorized, the EUPID server returns a set of context-specific PSNs, and the repository then may retrieve additional documents from other contexts within the same repository or from federated repositories, as authorized under applicable law, institutional policies, and individual consents. Note that this step is highly dependent upon policies and sharing agreements among repositories. Such agreements and policies are established outside the EUPID system.

Also, note that the EUPID sponsor has not yet implemented an application programming interface (API) to enable query of the EUPID metadata for linkages. These areas may represent opportunities for GA4GH collaboration with the EUPID sponsor.

Re-identification is enabled through the use of an optional re-identification module that is entrusted with a trusted third party. As mentioned above, some of the "contexts" provided to the data holder are designed for use in cases in which the data holder wishes to preserve the option to re-identify the individual. These contexts will include, in addition to the context identifier, a list of identity data elements, and hashing algorithm, a public encryption key and the name of the encryption algorithm to use to encrypt each of the identity data elements.

The trusted third party generates the keys, and sends the public key and encryption algorithm to the EUPID sponsor, while retaining the private key. The EUPID server packages the public key and encryption algorithm in a context definition that supports re-identification, and returns to the trusted third party the context identifier of the package containing the public key.

When a data holder wants to enable re-identification, the EUPID client selects a context designed for this purpose. In this case, in addition to generating a hash value for each identity data element, the EUPID client encrypts each element using the public key and the prescribed encryption algorithm, and transmits to the EUPID server both the hash values and the encrypted values, along with the context identifier. The EUPID server stores this information in the metadata index.

When an authorized user requests re-identification of a pseudonymised record, the trusted third party component queries the EUPID server for the encrypted identity elements associated with the PSN included in the query. The EUPID server returns the encrypted identity elements, along with the context identifier associated with the PSN. The trusted third party decrypts the identity elements using the private key that matches the public key used to encrypt them, and the associated encryption algorithm. The trusted third party then returns the identifiers to the authorized requester.

Note that the EUPID system does not dictate who is authorized to request re-identification, as this is a policy decision based on law, institutional policies, and individual consents.