Privacy Patterns for Pseudonymity

Alexander Gabel[1] and Ina Schiering[2]

[1] Ostfalia University of Applied Sciences
Wolfenbüttel, Germany
ale.gabel@ostfalia.de
[2] i.schiering@ostfalia.de

1 Minimal Pseudonym Scope

Summary: Restrict the linkability of a pseudonym by limiting the usage to the smallest possible scope for the purpose of data processing (data minimization).

Context: It is often not necessary for a pseudonym to have a very broad scope in the general case. Even if linkability across different scopes is necessary, usually not every party (e.g. an attacker) should be able to link pseudonyms trivially.

Problem: Pseudonyms are usually used to protect an identity from being disclosed. However when using only a single unique pseudonym for an identity, it becomes increasingly traceable and it may be linked across several databases and scopes. With more information about an identity, re-identification of a pseudonym becomes increasingly likely. Also in case of a data breach, datasets with potentially different information about an identity, which refer to the same pseudonym become linkable for attackers.

Forces/Concerns:

- Controllers may want linkability across different scopes for some services
- Users may prefer not to be tracked across multiple scopes

Solution: To prevent linkability across different scopes using a pseudonym, one may limit the use of a pseudonym to a small scope. For different scopes, different pseudonyms are used, which cannot be linked without additional information. A scope may be depend on a...

- role (e.g. shopping or video on demand)
- relationship (Company A or B)
- location
- time frame
- transaction (one-time use).

Furthermore combinations may be useful (e.g. role-relationship), depending on the use case. The controller needs to balance the purpose of the service and privacy of users. The scope has to be chosen according to the principle of data minimization. Selective Linkability can also be established via *Recoverable Identity* or *Pseudonym Converter*, which might decrease the risk in case of a data breach.

Consequences:

Benefits:

- In case of a data breach, pseudonyms across different scopes may not be linked trivially.
- Pseudonyms only refer to (small) partial identities, which cannot be linked trivially.

Liabilities:

- Additional complexity may be necessary, if linkability of pseudonyms in different scopes is necessary under certain conditions (e.g. by applying a *Pseudonym Converter*).

Examples:

- A user may use different **relationship-pseudonyms** [18], to limit linkability across different organizations. For example a user may want to use a different pseudonym for a dating website and for their business profile.
- A user may use different **role-pseudonyms** [18]. For example a user may use distinct pseudonyms (i.e. usernames) for a shopping service and for a video on-demand service at the same company.
- When generating pseudonyms for medical data of a person for secondary use (i.e. research), the same person may be referred to via different pseudonyms for each research purpose/organization (**relationship-pseudonym**) to prevent linkability.
- The pseudonym of a car in a car-to-x network may change depending on **location** and **time-frame**.
- The pseudonym of a user's smartphone may change depending on the **location**.
- A payment system may use a different pseudonym to refer to a user for each **transaction**.

[Known Uses]:

- Pommerening and Reng use a different pseudonym for each secondary use project of electronic health record (EHR) data [19].
- Mano et. al exchange pseudonyms of users when they meet at the same hub and propose a privacy verification algorithm [13].
- Rottondi et al. use a time-limited pseudonym to prevent linkability of smart meters over a longer time window in a smart grid system [23].
- The GSM standard uses a so-called Temporary Mobile Subscriber Identity (TMSI) as a pseudonym for a mobile station (MS). A different TMSI should be assigned when the location area of the MS changes. A network operator might also assign new TMSIs periodically inside the same location area. However, there were discovered some weaknesses in the TMSI reallocation procedure and its implementation [1].
- Tokenization is recommended by the Payment Card Industry Data Security Standard (PCI DSS) to protect sensitive information, such as the primary account number (PAN), from being leaked, by replacing it with a surrogate value. Different tokens can be created for the same PAN, depending on the handling party (relationship-based) or the timespan (cryptoperiod of the tokenization key) [17].

[Related Patterns]:

- Extends *Pseudonymous Identity*, as it improves the existing solution of protecting identities behind a pseudonym by giving it a small scope, thus making it more difficult to re-identify.
- Complements *Recoverable Identity*, as the small scope leads to less data being linked to the real identity in case of re-identification. It is complemented by *Recoverable Identity*, as it may help to prevent misuse when many pseudonyms make it hard to track/block a user.
- Used by *Pseudonym Broker*, as pseudonyms are different for each organization and time frame.
- Used by *Data Fragments*
- Used by *Data owner-based Pseudonymisation*
- Required by *Pseudonym Converter*

2 Recoverable Identity

Summary: The identity behind a pseudonym is recoverable under certain conditions.

Context: Pseudonym system are usually designed such that the re-identification of a pseudonym (i.e. determining the identity behind a pseudonym) is reasonably hard. In many cases it is sufficient to be able to link different transactions via pseudonyms. However in some cases, it might be necessary to recover the identity behind a pseudonym, for example in case of misuse of the system. Then e.g. only a trusted party/combination of multiple trusted parties should be able to recover the identity behind a pseudonym, in case of misuse.

Problem: If identity recovery is necessary, it should usually only be possible in very specific and constrained cases.

Forces/Concerns:

- Users may fear, that their identity is recovered in cases where it is not necessary (e.g. the user did not misuse the system), resulting in compromise of their privacy. Therefore the trusted party which is able to recover pseudonyms should transparently show and enforce their policies. The party should be trusted by both the controller and the users.
- The controller may want to identify users, e.g. for legal or payment purposes.

Solution: Restrict the ability of identity recovery via organizational and technical constraints.

Implementation:

- Use a Trusted Third Party for Identity Recovery
  - e.g. store a the pseudonym mapping in a Pseudonym Table
  - e.g. use an encryption scheme to encrypt identifiable data inside of a pseudonym
- Use a (thresholded) secret sharing scheme to allow identity recovery only with $n > t$ operators
- Use a (thresholded) secret sharing scheme to allow identity recovery only when there is enough ($n > t$) evidence for misuse
- Use anonymous credentials / *Attribute-based Credentials* with a trusted inspector authority, which is able to recover identities from pseudonyms.

Consequences:

Benefits:

- The identity behind a pseudonym is only recoverable in very specific, constrained cases.
- Misuse of the system by pseudonymous users may be limited, as users are informed about the possibility of identity recovery in such cases.

Liabilities:

- Users may have less trust in the system, if the policy for identity recovery or the technological barriers are too lax.

Examples:

- In a Pseudonymous Messaging system where users are communicating via email, pseudonymous users may be re-identified by a trusted third party, if they abuse the system, e.g. for illegal purposes. The pseudonymiser (the entity which translates real email addresses to pseudonymous ones) encrypts the original identity inside the pseudonymous e-mail address and is therefore the only entity which is able to recover an identity from a pseudonym only.
- In a smart grid system, each smart meter uses a pseudonym, which is generated by encrypting identifiable information (e.g. an ID known to the grid operator) using the public key of a trusted third party (TTP). The TTP may recover identities behind pseudonyms in case of misuse using its private key.

[Known Uses]:

- Hussain et al. use a secret sharing scheme to allow only the combination of all revocation authorities to recover the identity behind the pseudonym of an online electric vehicle (OLEV) in case of a legal need, such as refusing to pay after electricity consumption [11].

- Rottondi et al. allow the Configurator, a trusted party of a Smart Grid system, the recovery of identities by decrypting the identity as part of the pseudonym using its private key [23].
- Biskup and Flegel use a secret sharing scheme to allow re-identification of pseudonyms in an intrusion detection system only when there is enough evidence (i.e. enough events from a certain identity within a time-frame) [3].
- Attribute-based Credential Systems like IBM Idemix [12] or Microsoft U-Prove [16] allow re-identification of users via a separate Inspector authority. The ABC4Trust framework allows the usage of Idemix or U-Prove as backends [2].

[Related Patterns]:

- Complements *Pseudonymous Messaging*, as it may help to prevent misuse of the messaging service.
- Complements *Minimal Pseudonym Scope*, as it helps to re-identify users in case of misuse.
- Similar to *Pseudonym Converter*, as both patterns allow a trusted third party (TTP) to selectively link a pseudonym. In case of the *Pseudonym Converter*, a TTP can link pseudonyms, while in *Recoverable Identity* the TTP can link a pseudonym to an identity.

3 Data hidden from Pseudonymiser

Summary: Data being pseudonymised is not readable by the Pseudonymiser (entity which assigns pseudonyms).

Context: The pseudonymiser (i.e. the entity which creates pseudonyms and assigns them to identities) is usually only responsible for assigning pseudonyms, but does not need to have access to additional data. For example a pseudonymisation entity for medical data may not need access to the assigned medical reports etc.. Additionally, pseudonyms may be generated based on unique IDs instead of identifiable information (e.g. name).

Problem: When assigning a pseudonym to an identity the pseudonymiser might learn additional information about the identity, which may be unwanted and unnecessary.

Forces/Concerns:

- The pseudonymiser needs some kind of reference to the original identity. However, information about the person (such as the name or further information) may not be necessary.
- A secure channel between a data source and the party which receives pseudonymised data might be needed.

Solution: Hide data assigned to an identity by e.g. applying cryptographic measures before pseudonymisation.

[Implementation]:

- Before sending an identity and data to a pseudonymiser, encrypt the assigned data using public key cryptography. The pseudonymiser will receive a tuple $(ID, Enc(data))$ from the data source as pseudonymisation request and will send a tuple $(Pseudonym, Enc(data))$ to a party from which the real identity should be hidden. The receiving party is able to decrypt the hidden data using its private key.
- Use a secret sharing scheme to split the assigned data into parts, which are then pseudonymised by multiple distinct pseudonymisers. The receiving party is able to reconstruct the data if all parts are received, but each pseudonymiser on its own is unable to do so.
- If the pseudonymiser for a specific reason needs to have access to the assigned data, the additional use of de-identification methods to remove identifiable data (e.g. name, ID card number, birth data, . . .) is strongly recommended.

Consequences:

Benefits:

- The pseudonymiser does not learn additional information about an identity. Identities may be referred to as unique random identifiers, such that other identifiable data (such as a person's name) is also not available to the pseudonymiser.

Liabilities:

- Additional complexity of the system may arise depending on how the hiding mechanism is implemented.

Examples:

- A medical clinic may need to pseudonymise patients' medical data to be used in a research project. Instead of sending complete patient records with identifiable data (name, birth date etc.) to a pseudonymiser, only a list of randomly generated unique IDs is sent to the pseudonymiser. The pseudonymiser then converts each ID to a unique pseudonym and sends the resulting list (with the same order as the original list) to the research organization. Furthermore the clinic sends de-identified medical records (same order) to the research organization. The research organisation may then refer to a patient using the pseudonym from the list, while the pseudonymiser does not have any access to the medical data.
- Instead of sending the medical data separately, a clinic may also encrypt it for the research party and send it to encrypted to the pseudonymiser, who is unable to read the encrypted data.

[Known Uses]:

- Pommerening and Reng hide associated medical data for the pseudonymiser by encrypting it for the receiving research organization [19].
- Noumeir et al. perform de-identification of radiology data before sending it to a pseudonymisation system to reduce the risk of identification [15].
- Rottondi et al. use a secret splitting scheme in a smart meter system to let several pseudonymisation nodes pseudonymise shares of a smart meter (producer) reading, ensuring that these nodes cannot read the data, while the receiving node (consumer) can do so, when receiving all secret shares [23].
- Rahim et al. perform pre-pseudonymisation of patient identifiers in addition to encryption of the assigned medical data to completely hide identifiable information from the pseudonymisation server [20].

[Related Patterns]:

- Used by *Pseudonym Broker*, as the data assigned to a pseudonym is sent to a database or to a portal without any interaction with the Trusted Third Party, which acts as the pseudonymiser.
- Conflicts with *Data owner-based Pseudonymisation*, because the data owner (i.e. the pseudonymiser) already has knowledge of the data and it is not useful to hide that data.
- Complements *Pseudonymous Messaging*, as it hides the message content from the party which performs the pseudonymisation of the messages, providing additional privacy.

4 Pseudonym Converter

Summary: A separate entity, the Converter, is able to translate a pseudonym from one scope to a pseudonym in another scope.

Context: When pseudonyms in different scopes refer to the same identity, but there is a need to exchange data between those scopes without weakening the unlinkability of the pseudonyms (from the perspective of each scope). A scope in this pattern is typically a separate organization or department (i.e. domain-specific or relationship-based pseudonyms).

Problem: Unlinkability between different pseudonyms introduced for example by applying *Minimal Pseudonym Scope* prevents exchanging data regarding the same identity in different scopes. Depending on the use case however, selective exchange of data might be needed without weakening the unlinkability property of the pseudonyms.

Forces/Concerns:

- A naive approach might be to let a trusted third party perform the conversion between pseudonyms by storing a mapping table, however it may then be able to learn about all requests and correlations [5].
- Furthermore a data breach on the converter side may leak the relations of all pseudonyms.
- Auditability of data flows may be required at least for data owners.

Solution: Create a separate entity, the *Pseudonym Converter*, which is the only entity able to link pseudonyms in different scopes. When data in scope A needs to be linked to scope B (e.g. A requests data from B regarding a pseudonym in A), this link is established by the converter, if the request was legitimate. The converter forwards the request to scope B with a pseudonym mapped to scope B. Using a PET, as the one proposed by Camenisch and Lehmann [4] allows this procedure to be completed in a privacy-preserving way (blindly), i.e. without leaking information to the converter.

Consequences:

Benefits:

- The converter allows exchange of data regarding a pseudonym without sacrificing the unlinkability of different pseudonyms.
- The converter might allow or deny conversion requests based on a policy against misuse.

Liabilities:

- The converter adds additional complexity to the system in both organizational and technological areas.
- Depending on the implementation, a converter might need a certain degree of trust and might be a single point of failure.

Examples:

- In an e-health system, a patient may be represented by different pseudonyms depending on the health care organization handling the related data. For example a patient's general practitioner (GP) may use a different pseudonym for the same patient, than a hospital. When the GP needs data about the patient from the hospital, a conversion request with the patient's pseudonym in the GP's scope can be sent to the converter. The converter then decides whether the request is valid under its policy and forwards the request with (blindly) converted pseudonym to the hospital, which may then send data (without the hospitals pseudonym) back to the GP.

[Known Uses]:

- Camenisch and Lehmann propose a Pseudonym Converter system for a privacy-friendly exchange of distributed pseudonymised data even against a fully corrupt converter [5].
- Camenisch and Lehmann further improve their system by additionally allowing users to audit the flow of their personal data while maintaining all privacy properties [4].
- Verheul et al. propose the use of Polymorphic Pseudonymisation [26]. They use a trusted Transcryptor or Tweaker as an entity which is able to convert a pseudonym blindly into another pseudonym for a different entity.

[Related Patterns]:

- Requires *Minimal Pseudonym Scope*, as it needs different scopes between which the conversion is performed.
- Similar to *Recoverable Identity*

5 Data fragments

Summary: Split data of a single identity into small fragments and assign each fragment its own pseudonym. Only authorized entities are given the knowledge of which pseudonyms belong together.

Context: Whenever a collection of pseudonymised data records are under risk of re-identification by inference attacks due to the informative value of combined fields.

Problem: A record of data about an identity may contain enough information to re-identify it, even if primary identifiers are removed from the record. For example the combination of the attributes gender, ZIP code and birth date may uniquely identify 87% of the US-American population [25]. Furthermore it may be unwanted in a system to enable anyone with access to the dataset (e.g. also insiders like administrators) to be able to link sensitive data.

Forces/Concerns:

- Using server-side encryption may not help, if insiders such as administrators have access to the encryption keys.
- Encrypting the data using end-to-end encryption (i.e. unauthorized entities do not have access to the keys) might help, however when the dataset is large the performance penalty may be unacceptable/impractical.
- De-identification of the data using techniques from the area of Statistical Disclosure Control may work for some scenarios. However, such techniques may remove data needed for the use case.

Solution: Instead of storing related data with a single pseudonym, split the data into small fragments, which are hard to re-identify by themselves, and assign each fragment its own unique pseudonym. Only authorized persons or systems get the knowledge of which pseudonyms (i.e. which data fragments) belong to the same identity. It is also possible to reveal only partial information about which fragments belong together, to limit access to certain parts of data records. The pattern may furthermore be combined with de-identification to desensitize potentially identifiable data such as birth dates (e.g. mask day and month of birth) before transmitting the data.

Consequences:

Benefits:

- Enables unlinkability of data fragments by default, while authorized entities are able to link subsets of fragments.
- May significantly reduce the risk of insider attacks, as insiders are unable to link fragments or establish a relation to an identity.
- In case of a data breach, data fragments remain unlinkable for attackers without additional knowledge.
- Computationally efficient, as data fragments do not necessarily have to be encrypted

Liabilities:

- Increases complexity of the system, as knowledge about pseudonyms needs to be managed.

Examples:

- In an e-health system where health records or metadata of records from patients are stored centrally, instead of storing data referring to the same person in a linkable way, data fragments may be used to split health records into small fragments. For example each medical result is stored as a separate fragment. Only the data owner (i.e. the patient) has the knowledge which pseudonyms/data fragments belong to her. When the data owner wants to share fragments with a doctor, new pseudonyms pointing to the fragments can be generated and shared with the doctor.

[Known Uses]:

- PIPE (Pseudonymisation of Information for Privacy in E-Health) uses data fragments for electronic health records. By default only the patient (data owner) is able to access her health records. Access to the pseudonyms is managed through a central metadata storage which is encrypted with the users keys. The data owner may decide to give access to some records to

selected medical personnel by creating additional pseudonyms referring to fragments [22,10,14]. Identifiable and non-identifiable data is also unlinkable by default, so the system may be employed for secondary use, e.g. in research.
- Stingl and Slamanig describe a concept for an e-health portal, which uses unlinkable and undetectable partial identities of a patient to keep separate health records for participating parties (i.e. dentist and general practitioner access different partial identities) [24].
- The Fraunhofer ISST designed a concept for the German electronic health card (eGK), which uses ticket-based authorization and challenge-based authentication to allow fine-granular access control to data fragments, which are unlinkable by default [6].
- Biskup and Flegel use a secret sharing scheme to assign each event in an intrusion detection system a unique pseudonym, which keeps events unlinkable until enough evidence for re-identification is available [3].
- Camenisch and Lehmann propose the use of “data snippets”, which are stored with unlinkable pseudonyms. A central entity is able to link those snippets and may provide de-identified subsets of the original record to authorized parties. They suggest the use a central *Pseudonym Converter*, which is able to convert pseudonyms in a blind way while providing auditability for users [4].

[Related Patterns]:

- Uses *Minimal Pseudonym Scope*, as every data fragment gets its own pseudonym, therefore the scope of a pseudonym is very limited.
- Refines *Data owner-based pseudonymisation*, as it allows the data owner in a more specific context (i.e. shared repository of data) to perform the pseudonymisation and therefore provide a more privacy-preserving solution in comparison to a trusted third party solution.
- Extended by *Encrypted Link*

6 Encrypted Link

Summary: To authorize access to *data fragments* in a way that is not detectable by third parties, encrypt pseudonyms, pointing to *data fragments.*

Context: *Data fragments* are stored in a shared repository. A user wants to share a link to a *data fragment* with some other party.

Problem: A user wants to share a *data fragment* in a shared repository with some other party, without unauthorized parties with access to the repository being able to observe that there is a link between the data and the other party.

Forces/Concerns:

- When access rights are managed in a traditional access control system, Administrators of the repository (or similar insiders) may have access to permissions specified by a user.

Solution: Encrypt a pseudonym, pointing to the *data fragment* with the public key of the receiving party. Store the relation between the authorized party and the encrypted pseudonym accessible for the authorized party in the repository. Only the authorized party is able to decrypt the pseudonym and associate the data fragment that has been shared with it. Depending on the use case, information regarding associated identities (e.g. creator, 'sender' or related user of the data), can be added to the relation described above in an encrypted manner.

Consequences:

Benefits:

- Unauthorized users (including administrators) are unable to link the shared data fragment to the party it has been shared with.
- The link and the data can be stored in a repository, instead of sending it separately, which can improve auditability if used consistently.

Liabilities:

- The complexity of the system increases, as authorizations have to be encrypted and encryption keys have to be managed.

Examples:

- A patient in an e-health system has several records about her/his medical history stored in an associated repository. To keep the data unlinkable for unauthorized parties such as system administrators, the data is split into many small *data fragments*. Each fragment has its own pseudonym pointing to it. To store the information, which data fragments belongs to him, the patient stores multiple encrypted links in the repository. The links are assigned to the patient, however the content (i.e. the pseudonym pointing to a data fragment) is encrypted, such that only the patient may know which data fragments belong to her. When the patient wants to share a data fragment with a doctor, a new pseudonym is created, also pointing to the data fragment. Then the pseudonym is encrypted with the public key of the doctor and stored in the repository, assigned to the doctor. When the doctor accesses the repository, the assignment in combination with a matching private key allows the doctor to read the pseudonym and associate the data. Additionally, the data itself may also be encrypted.

[Known Uses]:

- A concept for the German electronic health card (eGK) developed by the Fraunhofer ISST stores medical data in a central way. Each record regarding a patient (e.g. prescription, result etc.) is stored as a separate *data fragment* (therefore unlinkable by default) in a virtual file system. A data owner (patient) may authorize e.g. a doctor by encrypting access information, a so-called "ticket toolkit" including the pseudonym of the data fragment, for the doctor. The data fragment itself is additionally encrypted with a separate symmetric session key. Users may revoke access rights by deleting the corresponding ticket toolkit [6].
- PIPE (Pseudonymisation of Information for Privacy in E-Health) allows patients to share certain *data fragments* with others by creating a new pseudonym pointing to a certain record, which is then shared in an encrypted fashion with the authorized person. Additionally a link to the patient's identifiable data may be provided and linked (encrypted) to the record pseudonym. This step however may be left out when pseudonymous data should be searched for example with a research organization. The data itself is stored in plaintext, however is not linkable to a patient without additional information. Access rights may also be revoked by the patient, by removing the shared pseudonym [22,10,14].
- Stingl and Slamanig describe a concept for an e-health portal, which allows users to share data fragments with other users by encrypting a pseudonym to the fragment with the other user's public key and storing this information in a central table together with the authorized user's ID. Additionally information regarding the sender, receiver, creator and corresponding patient of the *data fragment* may be attached or left out (also encrypted for the receiving party). Access may be revoked by deleting the corresponding reference entry [24].

[Related Patterns]:

- Extends *Data fragments*, as it provides a mechanism to share the links between data fragments in a privacy preserving way.
- Similar to *Private link*, as both patterns allow a data owner to share a certain item of interest with another party while hiding this exchange from others. The *Encrypted Link* pattern however provides a stronger guarantee, as it also protects from insiders such as administrators. Furthermore *Private Link* is not an alternative to *Encrypted Link* because of this reason.
- Uses *Encryption with user-managed keys*, as it depends on the fact that the server which stores the links does not have access to the encryption keys, but only the data owner (user) has access to them.
- Leads to *Buddy List* or a similar pattern, as encrypting a link for a particular party requires the user to find the right parties to share a link with and associate the corresponding public keys, which could be implemented using *Buddy List*.

7 Anonymisation Network

Summary: Hide the network identity of a communication partner by adding anonymisation nodes between communication partners.

Context: Nodes are communicating in a network.

Problem: When a node sends data to another node directly, the receiving node may track and possibly identify a node by it's address in the network, which may be unwanted if the node should stay anonymous or pseudonymous (in a given scope). Using other anonymisation/pseudonymisation techniques, usually the network metadata is the last piece of information which helps to re-identify a node.

Forces/Concerns:

- The solution may need to satisfy certain resource or performance constraints.
- For small networks, hiding an identity may be more difficult.

Solution: Instead of letting communication partners directly communicate, add intermediary anonymisation nodes, which increase unlinkability between the communication partners. Depending on the implementation and size of the network, different degrees of anonymity or pseudonymity can be reached.

[Implementation]

- A single anonymisation node (a proxy) can hide the address of several nodes, creating an *Anonymity Set*. However, the proxy itself is able to identify each node and to track the communication meta data (e.g. who communicates with whom, amount of data being transferred, etc.). The proxy may also replace the node identity with a pseudonym, which may be limited to a certain *Minimal Pseudonym Scope*.
- Mix networks, as suggested by David Chaum [7], use a fixed cascade/chain of proxies together with *Layered Encryption*, to hide the path of messages from several nodes and prevent tracing of the communication for each proxy.
- *Onion routing* similarly uses *Layered Encryption*, however the chain of proxies is always different and may change after some time.
- Lightweight anonymisation networks may be used in constrained environments. For example anonymisation networks may be simplified, if data only needs to be sent, but not received. For details, see the "known uses" section.

Consequences:

Benefits:

- It is more difficult to trace a node hidden by an anonymisation network using network metadata, given that anonymisation nodes along the path can be trusted.

Liabilities:

- The complexity of the network increases.
- The bandwidth may decrease and latency may increase, due to intermediate nodes and increased packet sizes (e.g. due to encryption).
- The ability to re-identify users in case of misuse may not be possible (or very difficult) unless all anonymisation nodes along the path cooperate.

Examples:

- A user in an e-health system is able to access pseudonymous medical data regarding her/him. In order to authenticate against the system a pseudonym/password combination is used. However the service provider may be able to identify the user using his/her IP address and resulting location information. Therefore the user uses onion routing to hide the address from the service provider.
- A doctor in an e-health system can access information regarding pseudonymous patients given the knowledge of their pseudonym. The doctor authenticates using attribute-based credentials to the service anonymously and acquires medical data regarding a pseudonym of a currently visiting patient. To keep unlinkability between the doctor and patient, a mix cascade is used in combination with protection against browser fingerprinting. This hides most of the remaining pieces of information which could be used to re-identify the doctor.

[Known Uses:]

- JonDoNym[3] uses mix cascades to support users in anonymously browsing the web.
- Tor[4] uses *Onion Routing* to protect the privacy of a user using a decentral peer-to-peer network of onion routers.
- Finster and Baumgart use a lightweight anonymisation network based on probabilistic peer-to-peer routing in a smart metering system to report measurements. Their network does only allow sending of messages, therefore they suggest to use a bloom filter in order to detect lost messages [8].
- Rottondi et al. propose a set of Privacy Preserving Nodes (PPN) as intermediaries (i.e. proxies) to hide the network address of smart meters (producers) from the data processors (consumers). To furthermore hide the data from

[3] https://anonymous-proxy-servers.net/index.html

[4] https://www.torproject.org/

the PPNs, secret splitting is applied, such that every measurement is splitted into shares which are sent to multiple PPNs, before finally reaching the consumers. The PPNs assign a relationship-based pseudonym to each share, such that each consumer can rebuild the measurement after every share has been received [23].
- Henrici et al. adapt the concept of *Onion Routing* to RFID systems using a hash-based pseudonymisation approach. A pseudonym is composed of multiple receiver identifiers, hidden such that only the respective sender is able to get the corresponding receiver identifier [9].
- Zhao and Li integrate a proxy service provider into a video streaming service system to conceal the network address of users from the video streaming service provider and decouple payment information/identity from watching history [27].

[Related Patterns]:

- Refines *Anonymity Set*, as it creates an anonymity set of nodes in a network.
- Refined by *Onion Routing*, as it is a specific solution for anonymisation networks based on the presence of many nodes and the possibility for asymmetric encryption (i.e. enough performance).
- Lead to by *Data owner-based Pseudonymisation*

8 Data owner-based Pseudonymisation

Summary: Generate and assign pseudonyms on the data-owner side instead of using a third party, to keep the link between pseudonym and the data owner hidden from other parties.

Context: When the link between the pseudonym holder and the pseudonym should be (initially) unknown for everyone except the pseudonym holder (data owner).

Problem: When using a (trusted) third party as pseudonymiser, the pseudonymiser usually learns the link between pseudonym holder and the pseudonym. However, in cases where this link should be initially hidden (i.e. only known to the data owner), a trusted third party as pseudonymiser is inappropriate.

Forces/Concerns:

- The data owner's identity has to be hidden while the data owner is specifying the pseudonym. For example while the data owner registers at a service with his/her identity, the pseudonym should not be specified visibly to the service provider, as otherwise the link would be trivial, even though the pseudonym is specified by the data owner.

- If no constraints (e.g. through cryptographic protocols) are enforced during pseudonym generation, recovering a pseudonym to the identity may be impossible/very hard for the service provider and other parties. The service provider however might fear the misuse of the service, while users are basically not identifiable.
- In most cases pseudonyms should be unique in the scope, but if the data owner generates the pseudonyms, there needs to be some way to guarantee this.
- The service provider may need to limit the number of pseudonyms per user.

Solution: Let the data owner create a randomly chosen pseudonym (i.e. not relatable to the user) and assign data to it. Uniqueness of the pseudonym can be made very likely using long enough randomly generated pseudonyms or guaranteed by a pseudonym-managing entity in an anonymous way (anonymous authentication and communication necessary). Additional guarantees such as single pseudonym per user, revocability or provable ownership may be implemented using privacy-enhancing technologies (e.g. *Attribute-based Credentials*).

[Implementation]:

- *Attribute-based Credentials* can include a user-chosen pseudonym as issuer-hidden attribute and furthermore can be used to implement revocability, provable ownership or single pseudonym per user.
- Blind signatures may be used to hide the user-chosen pseudonym from an authority, while receiving a valid signature for the pseudonym.
- A public key can be used as a digital pseudonym [18], which further allow users to prove ownership using a corresponding private key.

Consequences:

Benefits:

- The pseudonym can not be related to the holder trivially by anyone except the holder (unless additional patterns such as *Recoverable Identity* are applied).
- The data owner may decide to provide certain trusted parties with knowledge of the link between identity and pseudonym.
- Compared to a system based on a trusted third party, there is a highly decreased risk of attacks from insiders or database leaks, as the pseudonym link is not known to anyone except the data owner.

Liabilities:

- The data owners identity may still be discovered, for example due to side channel information, such as browser fingerprints, network addresses, or time-based information.
- The service provider (or a trusted third party) may not have a list of all registered pseudonyms, unless each user has reported the chosen pseudonym to the service.

Examples:

- The simplest example of this pattern is the often encountered pseudonymous registration in online services. However, the pseudonyms chosen by the users (nicknames) can typically be related to them and linked with other services, as they are not randomly chosen. To keep improve the pseudonymity, pseudonyms should therefore be generated randomly. This could be implemented in the software used by the user.
- When a doctor stores medical information about a currently visiting patient in an e-health system, the patient may choose a randomly generated pseudonym (possibly in cooperation with the doctor) under which to store the data. The important fact is that the pseudonym is not generated centrally at the e-health system or a separate third party for pseudonymisation, but in a de-central manner by each patient/doctor. Thus the link between identity and pseudonym is only known by the patient (data owner) and his/her doctor.
- In an e-health system a patient may choose to upload medical data to a research system for analysis, while being pseudonymous using a self-generated random pseudonym. When results or a chance finding are available, instead of notifying the pseudonymous user via identifiable information, such as the email address, the user may instead check at regular intervals if new information regarding his/her data is available. This could also be automated in a software. The identity of the user has to be protected by additional measures such as Anonymisation Networks.
- A user is able to authenticate at a video streaming service using Attribute-based Credentials. Each time he/she visits the service, a different pseudonym may be used, such that the watching history is unlinkable, while the service provider is still able to verify that the user has paid the subscription fee (e.g. through an attribute containing the expiration date. The user can prove that the current date is before the expiration date, without revealing that date).
- A user registers using her/his identity at a voting service. In the registration process, the user chooses a randomly generated pseudonym, which is blind-signed by the voting service. Afterwards the user may re-visit the service to vote once, revealing the (un)-blinded pseudonym with a valid signature of the voting service provider. The user stays pseudonymous, as the identity is unlinkable to the pseudonym, as the pseudonym was blinded during registration.

[Known Uses]:

- PIPE (Pseudonymisation of Information for Privacy in E-Health) uses *Data Fragments* where each fragment has its own pseudonym assigned by the data owner. Only the data owner is able to link data fragments to her/his identity by default [22,10,14].
- Finster and Baumgart propose a smart metering system, where each smart meter initially generates its own public key as pseudonym. This public key

is then blind-signed by the grid operator. When sending measurements, a lightweight Anonymisation Network is used in order to hide the (otherwise identifying) network address from the grid operator. Smart meters and the grid operator can verify that the measurements are from a valid device, as they are signed with the private key of the corresponding smart meter, whose public key has been (blind-)signed by the grid operator [8].

- Rawassizadeh et al. present LiDSec (Lightweight Data Security), a tool which allows the data owner to decide which elements of a data set to suppress, pseudonymise or to anonymise. The pseudonymisation is performed on the device of the data owner and mappings between pseudonyms and identities can be stored optionally in a different file. The tool furthermore can check whether hidden values of the dataset are also contained in other attributes (e.g. a name is also present in a message). This approach gives the data owner a large degree of control about the data set which may afterwards be sent to a service provider for further processing [21].

[Related patterns]:

- Leads to *Anonymisation Networks*, as data owners may be easily tracked by the network address, unless it is hidden by some kind of anonymisation network.
- Refined by *Attribute-based Credentials*
- Refined by *Data Fragments*
- Uses *Pseudonymous Identity*, as the data owner creates a random pseudonym for the hidden identity.

Acknowledgement This work was supported by the Ministry for Science and Culture of Lower Saxony as part of SecuRIn (VWZN3224).

References

1. Arapinis, M., Mancini, L.I., Ritter, E., Ryan, M.: Privacy through Pseudonymity in Mobile Telephony Systems. In: Proceedings 2014 Network and Distributed System Security Symposium. Internet Society, San Diego, CA (2014)
2. Baignères, T., Bichsel, P., Enderlein, R.R., Knudsen, H., Damgård, K., Jensen, J., Neven, G., Nielsen, J., Paillier, P., Stausholm, M.: D4.2 final reference implementation. Deliverable, ABC4Trust Consortium (August 2014), https://abc4trust.eu/download/D4.2%20Final%20Reference%20Implementation.pdf
3. Biskup, J., Flegel, U.: On pseudonymization of audit data for intrusion detection. In: Designing Privacy Enhancing Technologies. pp. 161–180. Springer (2001)
4. Camenisch, J., Lehmann, A.: Privacy-preserving user-auditable pseudonym systems. In: 2017 IEEE European Symposium on Security and Privacy (EuroS P). pp. 269–284 (April 2017)
5. Camenisch, J., Lehmann, A.: (Un)linkable pseudonyms for governmental databases. In: Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security. pp. 1467–1479. CCS '15, ACM, New York, NY, USA (2015), http://doi.acm.org/10.1145/2810103.2813658
6. Caumanns, J.: Der Patient bleibt Herr seiner Daten Realisierung des eGK-Berechtigungskonzepts über ein ticketbasiertes, virtuelles Dateisystem. Informatik-Spektrum 29(5), 323–331 (October 2006), https://link.springer.com/article/10.1007/s00287-006-0101-0
7. Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM 24(2), 84–90 (February 1981), http://doi.acm.org/10.1145/358549.358563
8. Finster, S., Baumgart, I.: Pseudonymous smart metering without a trusted third party. In: 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. pp. 1723–1728 (July 2013)
9. Henrici, D., Gotze, J., Muller, P.: A hash-based pseudonymization infrastructure for RFID systems. In: Second International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU'06). pp. 6 pp.–27 (June 2006)
10. Heurix, J., Neubauer, T.: Privacy-preserving storage and access of medical data through pseudonymization and encryption. Trust, Privacy and Security in Digital Business pp. 186–197 (2011)
11. Hussain, R., Son, J., Kim, D., Nogueira, M., Oh, H., Tokuta, A.O., Seo, J.: PBF: A new privacy-aware billing framework for online electric vehicles with bidirectional auditability. Wireless Communications and Mobile Computing 2017 (2017)
12. IBM Research - Zürich: Specification of the identity mixer cryptographic library version 2.4.43, https://abc4trust.eu/index.php?option=com_content&view=article&id=187, accessed on 1st August 2018
13. Mano, K., Minami, K., Maruyama, H.: Privacy-preserving publishing of pseudonym-based trajectory location data set. In: 2013 International Conference on Availability, Reliability and Security. pp. 615–624 (September 2013)
14. Neubauer, T., Heurix, J.: A methodology for the pseudonymization of medical data. International Journal of Medical Informatics 80(3), 190–204 (March 2011), http://linkinghub.elsevier.com/retrieve/pii/S1386505610002042

15. Noumeir, R., Lemay, A., Lina, J.M.: Pseudonymization of radiology data for research purposes. Journal of Digital Imaging 20(3), 284–295 (September 2007), https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3043895/
16. Paquin, C., Zaverucha, G.: U-prove cryptographic specification v1.1 (revision 3) (December 2013), https://www.microsoft.com/en-us/research/publication/u-prove-cryptographic-specification-v1-1-revision-3/
17. PCI Security Standards Council: Tokenization product security guidelines. Tech. Rep. 1.0, PCI Security Standards Council (April 2015), https://www.pcisecuritystandards.org/documents/Tokenization_Product_Security_Guidelines.pdf
18. Pfitzmann, A., Hansen, M.: A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management (2010)
19. Pommerening, K., Reng, M.: Secondary use of the EHR via pseudonymisation. Studies in Health Technology and Informatics 103, 441–446 (2004)
20. Rahim, Y.A., Sahib, S., Ghani, M.K.A.: Pseudonmization techniques for clinical data: Privacy study in sultan ismail hospital johor bahru. In: 7th International Conference on Networked Computing. pp. 74–77 (September 2011)
21. Rawassizadeh, R., Heurix, J., Khosravipour, S., Tjoa, A.M.: LiDSec- a lightweight pseudonymization approach for privacy-preserving publishing of textual personal information. In: 2011 Sixth International Conference on Availability, Reliability and Security. pp. 603–608 (August 2011)
22. Riedl, B., Neubauer, T., Goluch, G., Boehm, O., Reinauer, G., Krumboeck, A.: A secure architecture for the pseudonymization of medical data. In: The Second International Conference on Availability, Reliability and Security, 2007. ARES 2007. pp. 318–324 (April 2007)
23. Rottondi, C., Mauri, G., Verticale, G.: A data pseudonymization protocol for smart grids. In: 2012 IEEE Online Conference on Green Communications (GreenCom). pp. 68–73 (September 2012)
24. Stingl, C., Slamanig, D.: Berechtigungskonzept für ein ehealth-portal. na (2007)
25. Sweeney, L.: Simple demographics often identify people uniquely. Health (San Francisco) 671, 1–34 (2000)
26. Verheul, E.R., Jacobs, B., Meijer, C., Hildebrandt, M., de Ruiter, J.: Polymorphic encryption and pseudonymisation for personalised healthcare. IACR Cryptology ePrint Archive 2016, 411 (2016)
27. Zhao, X., Li, H.: Privacy preserving authenticating and billing scheme for video streaming service. In: Cyberspace Safety and Security. pp. 396–410. Lecture Notes in Computer Science, Springer, Cham (October 2017), https://link.springer.com/chapter/10.1007/978-3-319-69471-9_29