Federal Register Notice 87 FR 35250, https://www.federalregister.gov/documents/2022/06/09/2022-12432/request-for-information-on-advancing-privacy-enhancing-technologies, June 9, 2022

Request for Information (RFI) on Advancing Privacy Enhancing Technologies

Anonos

To: PETS-RFI@nitrd.gov

Re: RFI Response: Privacy-Enhancing Technologies

From: Gary LaFever, CEO, Anonos, Inc.

Mark Little, Chief Data Strategist and Head of Engineering, Anonos, Inc.

Type: Industry Response

1. *Specific research opportunities to advance PETs:* Information about Federal research opportunities that could be introduced or modified to accelerate the development or adoption of PETs. This includes topics for research, hardware and software development, and educational and training programs. This also includes information about specific techniques and approaches that could be among the most promising technologies in this space.

Encryption and access controls have become firmly established as standard practice for protecting data in transit and at rest. Processing of data with sensitive information, however, has suffered from a lack of PETs that are simultaneously practical to use, provide effective protection and preserve utility. Stated differently, organizations have not had access to PETs that provide efficient, scalable protection for data when in use.

Statutory Pseudonymisation, as first defined in Article 4(5) of the EU General Data Protection Regulation (GDPR), is rapidly becoming a de facto global standard for protection of data when in use and formally recognized by more than forty governments and non-governmental organizations (NGOs) around the world. Essentially identical statutory language is found in the EU GDPR, the UK GDPR, and the Data Protection regulations of Japan, South Korea, Brazil, and five US States (CA, VA, CO, UT, and CT), and formally acknowledged by the German Association For Data Protection And Data Security (GDD) and the World Economic Forum (WEF).

In each case, it has been embraced as a means for reconciling conflicts between maximizing data value and protection. Other countries and US states are looking to adopt similar provisions incorporating Statutory Pseudonymisation because of its unique ability to simultaneously maximize both data utility and data protection without being overly cumbersome, a significant advantage over other PETs (see submission to Item 2 below for further comments on this point).

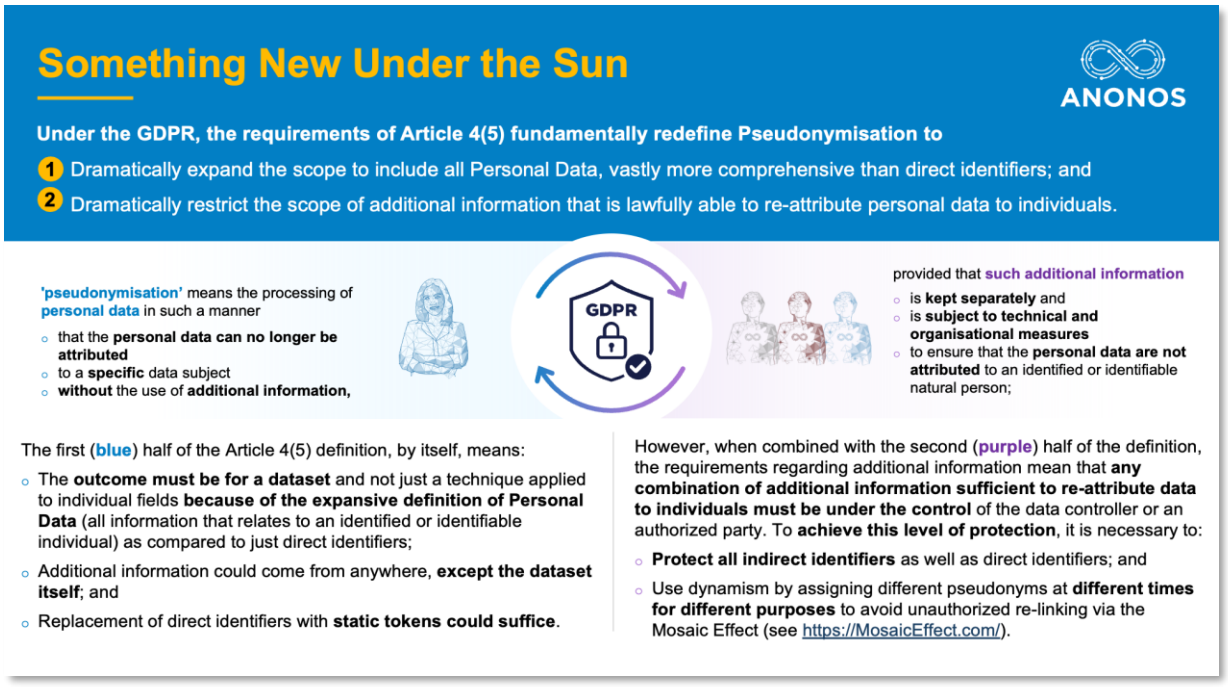
To understand how Statutory Pseudonymisation delivers this advantage, it is necessary to look at the differences between past common use of the term pseudonymization and the statutory construction of the new definition, and the resulting implications.

Prior to the ratification of the GDPR, there were no statutes or laws defining the term pseudonymization, although the term has been in common use for many years. Most data protection practitioners would characterize it as a technique for obscuring personally identifying information (PII) that replaces direct identifiers with static tokens.

In contrast the EU GDPR defines Statutory Pseudonymisation as:

> 'pseudonymisation' means the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organisational measures to ensure that the personal data are not attributed to an identified or identifiable natural person;[1]

The following graphic highlights how significantly different and more demanding the requirements are for Statutory Pseudonymisation than for the PET known variously as pseudonymization, hashing, tokenization, and key-coding.

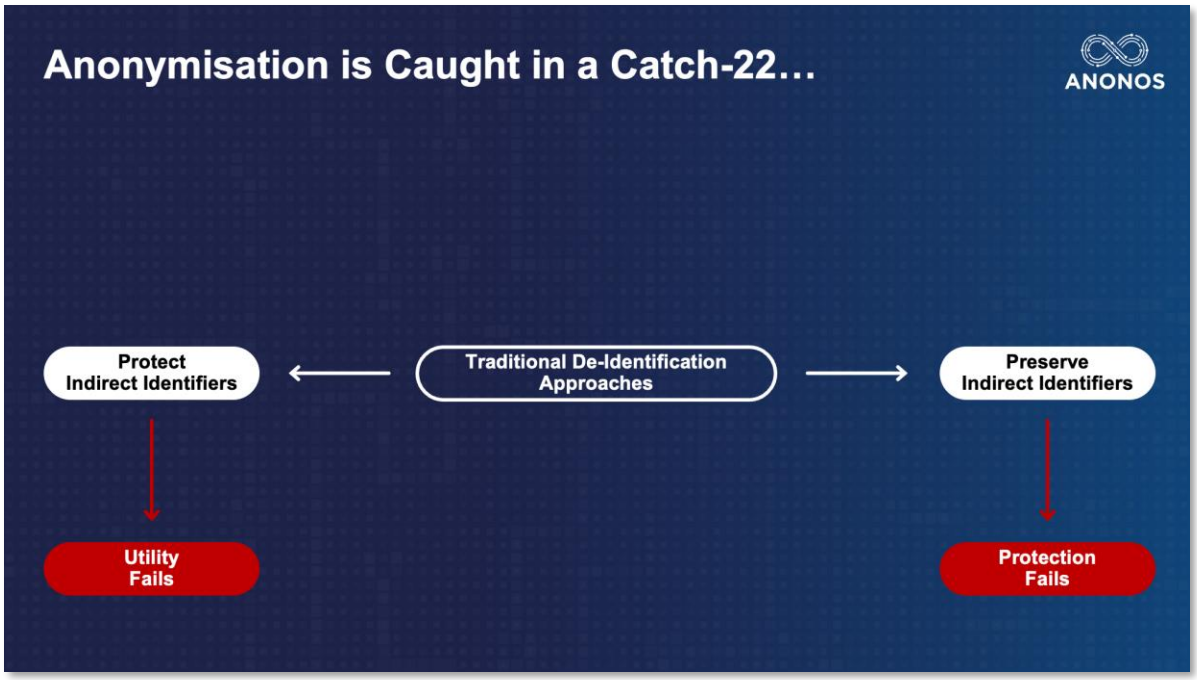


This language fundamentally changes the meaning of the term in two ways. First, it dramatically expands the scope of applicability to Personal Data as defined under the EU GDPR (all information that relates to an identified or identifiable individual) which is much more comprehensive than direct identifiers that are PII. Second, the scope of additional information that can be used to

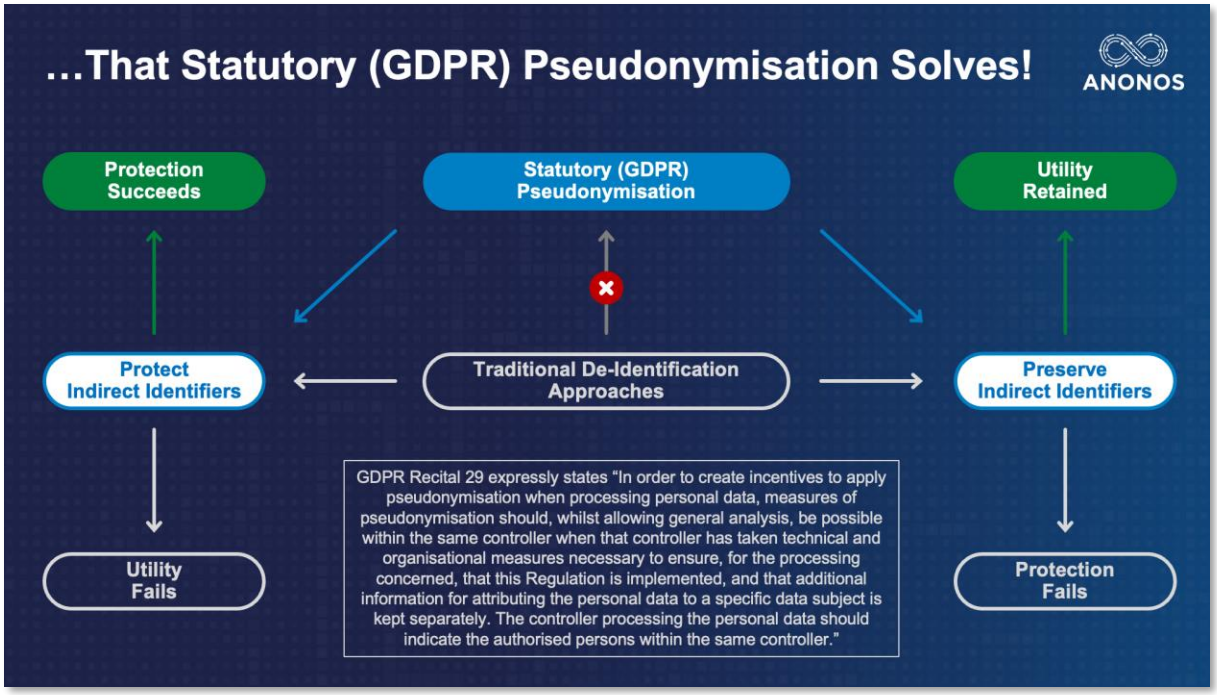
lawfully re-attribute personal information to an individual is dramatically narrowed so that all such information must be subject to technical and organizational controls limiting access to it.

Accordingly, it is clear that when defined this way, Statutory Pseudonymisation is no longer a description of a technique, but rather of an outcome describing the state of an entire data set, just as encryption or anonymisation transform cleartext into new states, *viz.,* encrypted and anonymized. As such, transformation or replacement of individual fields must take into account the potential to reattribute data to an individual not only for direct identifiers, but also for quasi-identifiers. In some cases, it may be necessary or advantageous from a data protection standpoint to protect certain indirect-identifiers and attribute fields as well.

Under the EU GDPR, Anonymous [2] data is not considered Personal Data, and thus falls outside the scope of the regulation. In theory this makes anonymisation an attractive option for processing personal data. However, in a world awash in data, as a data protection technique, organizations attempting to rely on anonymisation find themselves in a catch-22[3]. On the one hand, if in attempting to anonymize data quasi-identifiers are (by definition) irreversibly protected, utility for analytics is largely destroyed, as the original values cannot be recovered. However, if the quasi-identifiers are not protected in an effort to retain analytic utility, there is little chance the data set will meet any meaningful standard of anonymity. Moreover, for data to qualify as anonymous under the GDPR, even the party creating an “anonymous” data set must not be able to reverse the protection. In practice, no one deletes source data after creating an “anonymous” version of the data, which means reversal of the protection is trivial (with the possible exception of aggregate data, which has limited utility in most instances).



What is not widely appreciated about Statutory Pseudonymisation as a PET is that it solves both of these limitations. Because, by definition, any protection of quasi-identifiers can be reversed when authorized, they can be aggressively protected with no loss in analytic utility. Statutory Pseudonymisation is often (incorrectly) characterized as weaker protection than anonymisation because it is still personal data and does not move processing outside the jurisdiction of the law, however, nothing could be further from the truth. Statutory Pseudonymisation actually provides superior protection against unauthorized reidentification with better utility than so-called anonymous data.

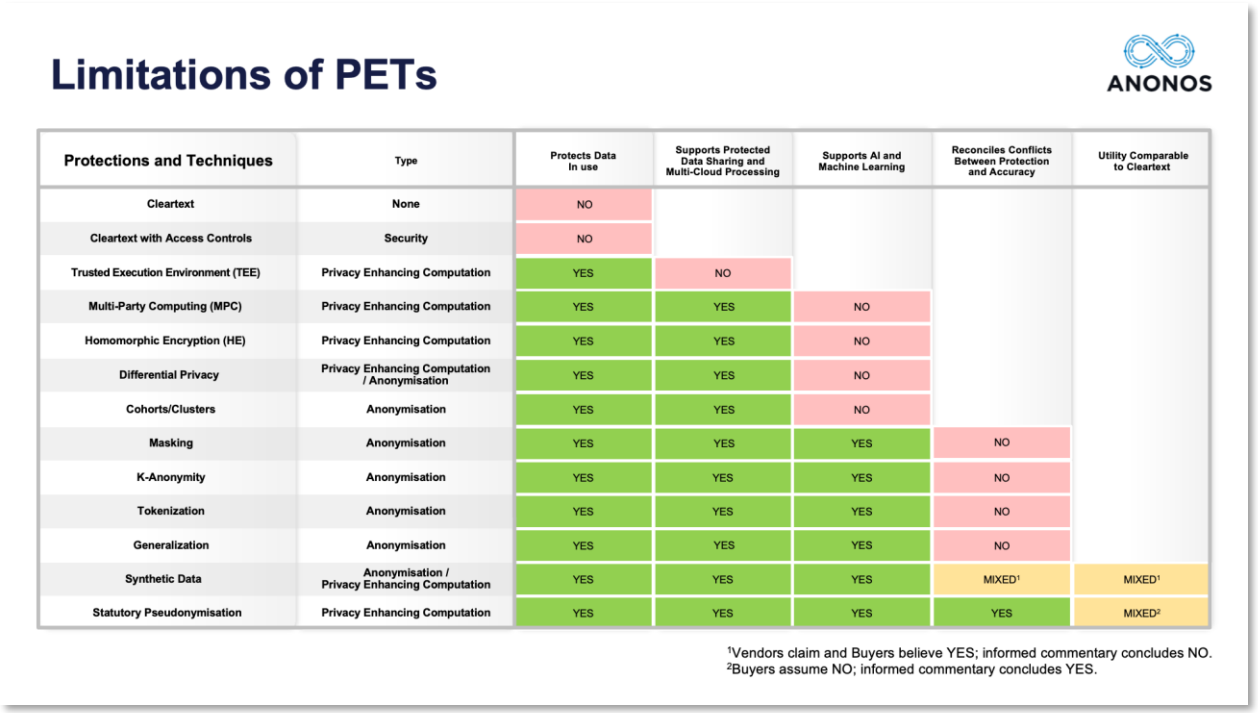


Extensive conversation with experts in EU data protection law – among them members of the European Data Protection Supervisor (EDPS) [4] and EU Member State Data Protection Authorities, including several involved in drafting the statutory language or subsequent regulatory guidance and recommendations (e.g., GDPR Article 4(5), EDPS recommendations for Schrems II compliant international data transfers) – confirms two things. First, that the above interpretation is exactly what was intended in the original construction of the GDPR statutory language and second, that the above analysis is not well understood outside the drafters of the language and a very small number of data protection experts.

In light of the foregoing, Federal research opportunities confirming the above, supporting software development, and providing education and training on the advantages and utility of Statutory Pseudonymisation as a PET should be a high priority.

2. Specific technical aspects or limitations of PETs: Information about technical specifics of PETs that have implications for their development or adoption. This includes information about specific PET techniques that are promising, recent or anticipated advances in the theory and practice of PETs, constraints posed by limited data and computational resources, limitations posed by current approaches to de-identification and deanonymization techniques, limitations or tradeoffs posed when considering PETs as well as technical approaches to equity considerations such as fairness-aware machine learning, security considerations based on relevant advances in cryptography or computing architecture, and new or emerging privacy-enhancing techniques. This also includes technical specifications that could improve the benefits or privacy protections, or reduce the risks or costs of adopting PETs.

The following chart evaluates the full range of data protection techniques, including both security-based approaches and traditional PETs against a series of criteria for evaluating the effectiveness of protection and the utility of the protected output. Rather than being a traditionally red/green or stoplight chart that evaluates all PETs against all criteria, this is a knockout chart. PETs are evaluated against the criteria sequentially from left to right, and once a PET fails to meet a criterion it is dropped from further consideration.



Limitations of PETs

ANONOS

Protections and Techniques	Type	Protects Data In use	Supports Protected Data Sharing and Multi-Cloud Processing	Supports AI and Machine Learning	Reconciles Conflicts Between Protection and Accuracy	Utility Comparable to Cleartext
Cleartext	None	NO				
Cleartext with Access Controls	Security	NO				
Trusted Execution Environment (TEE)	Privacy Enhancing Computation	YES	NO			
Multi-Party Computing (MPC)	Privacy Enhancing Computation	YES	YES	NO		
Homomorphic Encryption (HE)	Privacy Enhancing Computation	YES	YES	NO		
Differential Privacy	Privacy Enhancing Computation / Anonymisation	YES	YES	NO		
Cohorts/Clusters	Anonymisation	YES	YES	NO		
Masking	Anonymisation	YES	YES	YES	NO	
K-Anonymity	Anonymisation	YES	YES	YES	NO	
Tokenization	Anonymisation	YES	YES	YES	NO	
Generalization	Anonymisation	YES	YES	YES	NO	
Synthetic Data	Anonymisation / Privacy Enhancing Computation	YES	YES	YES	MIXED[1]	MIXED[1]
Statutory Pseudonymisation	Privacy Enhancing Computation	YES	YES	YES	YES	MIXED[2]

[1]Vendors claim and Buyers believe YES; informed commentary concludes NO.
[2]Buyers assume NO; informed commentary concludes YES.

Cleartext with Access Controls

Access controls are an essential component of data security. However, no matter how granular they are (e.g., attribute-based, tasked-based or even zero-trust) they are still binary; once granted, access is to clear-text. As a result, they do not provide protection for data in use.

Encryption

Encryption is the *sine qua non* for protecting data at rest and in transit. But to use data, for example in computation or analytics, it must be decrypted, at which point it is no longer protected at all.

Homomorphic Encryption (HE)

Research has been ongoing for many years in an effort to find a way to improve processing speed to a level even approaching commercial viability. Most information touting progress talks only of "improvements" and not actual processing throughput results, for good reason. Estimates suggest processing speeds that are 5 to 10 orders of magnitude slower than processing cleartext. That implies that computations that would take one millisecond in clear text would take anywhere from 1.5 minutes to nearly 4 months.

Multi-Party Computing (MPC)

A relatively new technique that is frequently (mis)represented as "encryption in use," presumably for marketing purposes. The justification seems to be that more precisely, the encoding of data done to enable the shared computations is fairly characterized as a cryptographic technique, as is encryption. But as commonly used, encryption is not understood to be the encoding done in MPC, which results in cleartext values. In any case, MPC remains cumbersome, as it requires tremendous bandwidth for the communication and coordination required between the computing parties, which can be both expensive and results in processing speed penalties, limiting its use to niche applications.

Trusted Execution Environment (TEE) / Confidential Computing Environment (CEE):

Among the newest of new techniques, this approach sets up an on-processor enclave of a portion of system memory, and in some implementations, part of the CPU itself. Data is stored and moved around the processor in encrypted form until inside the enclave, where it is decrypted using a key only available within the enclave. Implementation is technically challenging, and often requires rewriting applications to work in the TEE. Additionally, the enclave is by definition a silo. Thus, this approach is not well-suited for data sharing and combining and multi-cloud or hybrid-cloud applications.

Differential Privacy and Cohorts/Clusters

By definition, these techniques provide results that are aggregated, and do not provide the record-level output necessary for most uses of data.

Anonymization

The following techniques, Masking, Generalization, Tokenization, K-Anonymity, Noise Introduction, and Synthetic Data all are used, typically by combining several together, in an effort to Anonymize data. However, in the effort to do so, they all fail to resolve the intractable trade-off between privacy

and utility that is inherent in anonymisation. In a big data world, they fail to deliver the privacy promised by anonymisation, and efforts to push them to their limits to do so ends up destroying the utility of the protected output.

Masking

This technique protects direct identifiers by masking or overwriting one or more characters. It requires the data, its use, and its users are all restricted/sequestered to prevent other unprotected fields in a record from being combined with the information in additional data sources to enable an individual to be distinguished from others or identified via linkage attacks (see https://MosaicEffect.com). This requirement to restrict access is inconsistent with the architectural requirements of increasingly prevalent use cases that require free flowing data and involve dynamically changing data sources, processes, and processors.

Generalization

This technique attempts to protect against reidentification by reducing the granularity of the original data. Classic examples include converting age to age ranges by range binning, or by rounding numerical values. Masking can also be used for generalization such as masking one or more trailing digits of a zip or postal code to create values that represent larger areas. By itself, this technique does little to protect identity as it is not useful for direct identifiers. It is often put into practice in an effort to achieve a specified level of k-anonymity (see below).

Tokenization

(Hashing/Key-Coding/Pre-GDPR Pseudonymization): These techniques: (i) only protect direct identifiers and (ii) protect those direct identifiers by replacing them with a recurring (persistent) token, making them effective only for limited, static use cases. They require that the data, its use, and its users are all restricted/sequestered to prevent other unprotected fields in a record from being combined with the information in additional data sources to enable an individual to be distinguished from others or identified via linkage attacks (see https://MosaicEffect.com). This requirement to restrict access is inconsistent with the architectural requirements of increasingly prevalent use cases that require free flowing data and involve dynamically changing data sources, processes, and processors.

K-Anonymity

K-anonymity techniques are intended to prevent a data subject from being singled out by grouping them with at least “k”-1 other individuals who share the same values for a specified subset of attributes in a data set. This subset of attributes, which are commonly referred to as quasi-identifiers because of their ability to, when used in combination, reveal identity. The quasi-identifiers are generalized as necessary (using techniques such as range binning, rounding, and masking) to ensure that all possible subgroups defined by the values of the quasi-identifiers have at least k individuals in them. In most cases, to achieve that status for all records in the data set, the required generalization severely degrades the utility of the data. To mitigate the degradation, a decision is made to be less aggressive in the generalization, and then suppress values or entire

records in those subgroups where k falls short of the specified level. Note however that this also results in degradation of data utility, as a result of distortion in the output dataset statistical properties relative to those in the original source data.

Noise Introduction

This technique involves intentionally changing values in a data set so that they are less likely to be useful in revealing identity while at the same time avoiding excessive degradation in data utility due to distortion of the statistical relationships among attributes. This technique explicitly trades off utility (i.e., accuracy) for privacy, and tends to fall short on both accounts.

Synthetic Data

The failure of synthetic data to adequately protect against identity disclosure is now well documented in academic papers. The current state of the art appears to be ~ 1% of data subjects at risk of identity disclosure, which is likely to be judged to be far short of the regulatory requirements for anonymous data. Efforts to reduce this risk inevitably come at the expense of accuracy, as maximizing accuracy leads to overfitting and duplicating unique records in the source data. Some organizations report accuracy rates of as low as 70%. An additional challenge relates to the incorporation of incremental records to an existing source data set, or the addition of additional tables. In order to properly preserve the statistical properties between records, fields and tables, these situations almost always will require regenerating the models used to create synthetic data.

Statutory Pseudonymisation

While it does address the forgoing criteria, there is no getting around the fact that a Statutorily Pseudonymised data set looks quite different from its cleartext source. For many aspects of analytics, particularly actual computation in algorithms this is not an issue at all, as the pseudonyms simply process as nominal or categorical strings. For analytics involving active participation by a person (e.g., exploratory data analysis, BI reports, feature engineering, results interpretation, etc.) this is clearly not the same as working with cleartext. That said, appropriate organizational controls used in conjunction with authorized reversals of pseudonyms to cleartext when necessary to advance processing means the issue is not insuperable, but more akin to a change in workflows.

---

[1] EU GDPR Article 4(5). Note that other jurisdictions have as a rule borrowed this construction verbatim (or nearly so).
[2] EU GDPR Recital (26).
[3] A problem in which the solution is denied by the problem itself. See https://www.merriam-webster.com/dictionary/catch-22.
[4] See attached report provided to the EDPS following a meeting with them and providing a mathematical proof of how data is protected using Statutory Pseudonymisation.

Attached Reference Document:

Anonos Data Embassy Overview Presented to European Data Protection Supervisor

ANONOS

Data Embassy and Variant Twins

This document describes the technical and mathematical underpinnings enabling Anonos software to overcome the well-established axiom:

“Data can be useful or perfectly anonymous, but never both.”

With Anonos software, people are no longer forced to choose whether they want data utility or protection.

Anonos enables them to have both.

March 2022

Data Embassy and Variant Twins:
Overview and Mathematical Underpinnings

****We recommend watching the following 9-minute video summary before reading this document https://www.anonos.com/9minutes**

Overview

Data Embassy Value Proposition

Anonos Data Embassy software uses technologically enforced protection to transform cleartext data into variable-resolution, use-case specific outputs called **Variant Twins.**[1] This fine-grained approach delivers:

- **Proactive Security/Privacy,** where data is fully protected **during use,** even in the event of a breach.
- **Compliance:** GDPR, CCPA, and more. Schrems II compliant surveillance-proof processing in US operated clouds, regardless of the location of servers.
- **Full Compatibility** with a wide range of both primary and secondary uses of data.
- **100% Accuracy,** verified by external experts, relative to processing unprotected cleartext by enabling the relinking of the results of protected processing to source data under controlled conditions for authorised purposes only.

The technological controls embedded in Variant Twins travel with the data, enabling fully decentralised processing. **Anonos Data Embassy allows for maximum data utility, compliance with international privacy laws, and mitigated liability risk upon breach.**

Data Embassy Use Cases

Secondary Uses of Data

- Analytics, Business Intelligence and Reporting
- Machine Learning
- Artificial Intelligence
- Data Combing and Sharing

Primary Uses of Data

- Customer and Employee Support
- Transaction Processing
- Right to be Forgotten/Delete My Data

Location and Jurisdiction Independence

- On Premises
- Private Cloud
- Public Cloud and other International Transfers
- Multi-Cloud and Hybrid Cloud

Scalable Enterprise-Grade Protection

Data Embassy deploys with features and capabilities necessary for use at scale by global enterprises.

- **Data Protection Rules:** Configurable templates enable digitisation and technological enforcement of an organisation's privacy policies so that they are no longer "just words" in a document. Configure and approve once per use case, automatically apply over and over.
- **Tagging:** Enables tags for field identifier types, statistical data types, use cases, jurisdictions, Variant Twin recipients, and more, enabling rapid and automated configuration of data transformers.
- **Group and Role-Based Permissions:** Establishes group and role-based permissions allowing users to authenticate and authorise controls for the necessary separation of responsibilities, segregation of duties, and "need-to-know" restrictions essential for demonstrating the technical and organisational controls over approvals for both protecting data and reversing protections.
- **Approvals:** Implements approvals for protection rules, policy deviation requests, data transformer configuration, Variant Twin creation and controlled relinking/reversal.
- **Auditability:** Maintains immutable records of all system activity through User IDs.
- **Design Studio with Privacy Engineering Tools and Aids:** Provides users tools for risk scoring, k-anonymity analysis, cardinality analysis, multiple preview modes, field/rule linking of data, to maintain referential integrity when required.
- **No Code Configuration:** Fast, easy, and scalable.
- **Fully Documented APIs:** Supports custom development via scripting, automation, and integration with existing data pipelines.
- **Modern Component Stack:** Kubernetes, Docker, Cassandra, PostgreSQL, Spark, Kafka, Kotlin, React, Vault, Keycloak; supports batch, streaming, high throughput, and high availability.
- **Improved Productivity:** Customers report a four-fold increase in the approval of projects, each in 25% of the time–achieving an overall 16x productivity gain in making high-value data available for use.

Compound Cryptographic Security

The use of multiple data transformation techniques, including cryptographic algorithms, enhances security and reduces the risk of access by an unauthorised party. Data Embassy integrates multiple transformation techniques, including:

- Omission of direct identifiers and replacement with random or deterministic pseudonyms (i.e., the same input value always results in the same output value) when it is not possible to omit such direct identifiers.
- Conversion of numerical fields to categorical whenever possible (e.g., age 25 to an age range of 20-29), while taking care to ensure that cardinality (number of unique values in a field) is neither too high (risking "fingerprinting") nor too low (more easily guessed via brute force attacks).
- Replacement of the values in **all** categorical fields (direct identifiers, quasi-identifiers, indirect identifiers, and attributes) with deterministic pseudonyms.
- Scope delimiting the referential integrity of deterministic pseudonyms to the minimum needed for each given use case.
- Enforcement of patented dynamism by replacing data elements to be transformed with different pseudonyms at different times for different purposes (i.e., the same input value is assigned different pseudonyms for different projects, so outputs from one project cannot be used to "attack" outputs for another project). This is accomplished by using different initialisation vectors (commonly referred to as keys) and at least two cryptographic algorithms (see Appendix). This approach restricts the ability to combine or correlate otherwise seemingly unassociated information to defeat unauthorised reidentification via inference attacks and linkage attacks using the Mosaic Effect.
- Application of k-anonymity to suppress records with uncommon combinations of quasi-identifier values (even when replaced with pseudonyms) to defeat singling out attacks.

Because of the integrated complexity of these multiple layers of protection, when properly implemented, it would not be possible to "attack" and reverse the protections to reattribute data to specific individuals without access to the keys and algorithms used to encode the data.

Despite the integrated complexity of this multi-layered protection, as with all processing before passing it to a processor for computation, the resulting output is ultimately converted into binary (0's and 1's), resulting in the same processing efficiency as unprotected cleartext – i.e., unlike homomorphic encryption or other methods of protecting the underlying data, the use of Anonos' Variant Twins does not increase the need for computing resources.

The net effect of the above is protection for data **in use** that is analogous to SSL/TLS (Secure Sockets Layer/Transport Layer Security) protection for data **in transit**, since Data Embassy prevents unauthorised use of information, including potential personal details.

Data Embassy implements a Zero Trust Security Policy known as the principle of least privilege, or PoLP, that limits a user's access rights to only what is strictly required for an assigned role, using centralised controls that technologically embed protections into data, **effectively restricting access rights to decentralised data flows to only the level of identifying data (including none) required for each authorised use case.**

Cleartext Utility

The layered approach to replacing cleartext source data with a cryptographically secure Variant Twin is fully compatible with most analytics, machine learning and AI processing, while delivering the accuracy of cleartext.

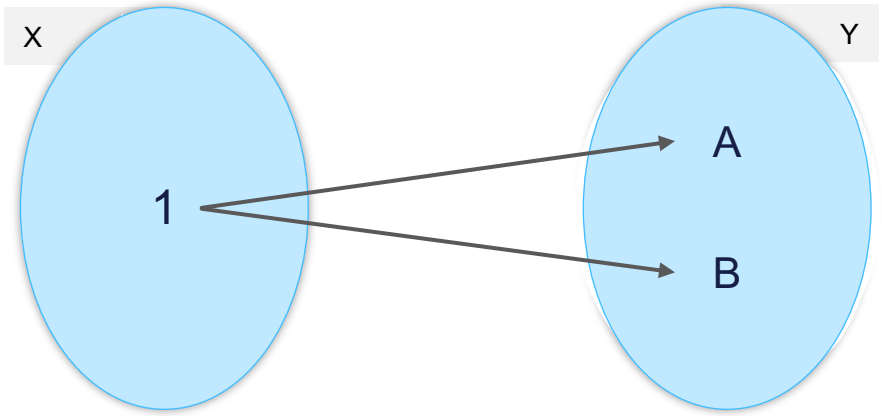
The reason for this surprising result is that replacing categorical values with deterministic pseudonyms has **zero** impact on the functionality of those data elements in the underlying algorithms. The algorithms only care that the values of the strings comprising categorical variables are (1) consistently used and (2) distinguishable. For example, in a model, Male/Female, M/F and a098dfae19 / fjfud630rmf7 all work equally well, but only the last pair serves to prevent reidentification attacks.

However, the latter is not an option for a quasi-identifier like sex (nor for indirect identifiers nor attributes either) when attempting to anonymise data. The reason is that since anonymisation must be irreversible, much of the utility or insights that might be derived from the protected fields are irretrievably destroyed. However, with GDPR-compliant pseudonymisation, this type of protection is (1) necessary to meet the statutory requirements but (2) is definitionally allowed to be reversed under controlled conditions, ensuring the utility and insights are not lost, but are instead preserved.

Mathematical Underpinnings Of Variant Twin Cryptographic Security

Primer on Mathematical Functions

The concept of a function is very important in mathematics. You can think of a function as being a machine that takes in a certain set of inputs, and for each input item, generates one (and only one) output. Mathematically speaking, a function f is a mapping that takes elements from a set X and maps them to elements in a set Y. To emphasise that an element y in Y is the output of an element x in X, we commonly write $y = f(x)$.

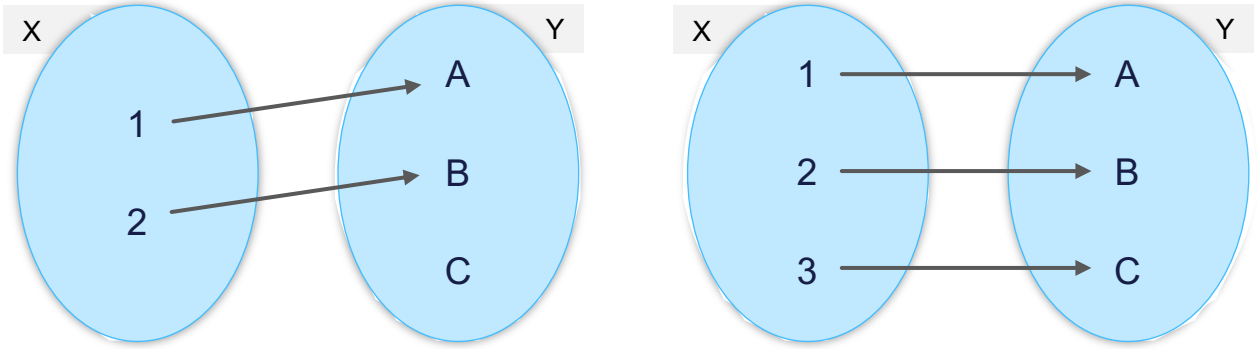


A one-to-many mapping, which is not a function.

Let f be a mapping from a set X to a set Y. If f maps an element of X to more than one element of Y, then f is said to be a *one-to-many*, which is not a function. An easy way to remember this is the phrase “he has had one too many, so he cannot function.” The below image shows an example of a one-to-many mapping, which is not a function.

If f is a function from a set X to a set Y which maps each element of X to one (and only one) element in Y, then f is said to be *one-to-one* (or more mathematically sophisticated, *injective)*.

An easy way to remember the term *injective* is to think of getting an injection or vaccination. The vaccination (injection) maps one (unvaccinated) person, to one (vaccinated) person. The below image demonstrates two injective functions.



An injective mapping, which is a function.

A function f from a set X to a set Y is said to be *onto* (or more mathematically sophisticated, *surjective*) if f maps an element of X to each element of Y. In other words, if for every y in Y, we can always find (at least one) element x in X such that $y = f(x)$, then f is surjective. The below images demonstrate two surjective functions.

A surjective mapping, which is a function.

We now introduce a special type of function, called a *bijection*. A function f which is both injective and surjective, is called a bijection. In other words, a bijection is a function between two sets X and Y where each element of both sets gets mapped directly with one and only one element of the other set. In other words, a bijection can be thought of as a relabeling of elements.

Bijections are very convenient functions to work with, in the sense that they have a particularly useful property: they are always invertible. What this means is that, if you have two sets X and Y and you know that the elements are connected via a bijective function f, if you have the element $y = f(x)$, and know the function f, you can always "work back" and find the element x which uniquely mapped to y. This is proven by the following theorem.

Theorem A: Let $f: X \to Y$ be a bijection. Then f is invertible.

Proof:

First, we must show that f^{-1} is the inverse of f:

Let	$x \in X$ and $y = f(x)$
By definition	$f^{-1}(y) = x$
By substitution	$f^{-1}\big(f(x)\big) = x$
By simplification	$x = x$

Now,

Let	$f: X \to Y$ be a bijection and let $y \in Y$
Since f is surjective, there exists an element $x \in X$ such that $y = f(x)$	
Let	$x = f^{-1}(y)$
Since f is injective, we know that x is unique	
Let	$y \in Y$ and $x = f^{-1}(y)$.
By definition	$f(x) = y$
By substitution	$f\big(f^{-1}(y)\big) = y$
By simplification	$y = y$

It also turns out that, if you have a function f between two sets X and Y which is invertible, then f is necessarily a bijection. This is proven by the following theorem.

Theorem B: Let $f: X \to Y$ have an inverse. Then f is a bijection.

Proof:

Let	$f: X \to Y$ be invertible and have inverse $f^{-1}: Y \to X$.

We must show that f is surjective and injective.

Let	$y \in Y$ and $x = f^{-1}(y)$.
Then	$f(x) = f\big(f^{-1}(y)\big) = y$
So	f is surjective

Next

Let	$x_1, x_2 \in X$ such that $f(x_1) = f(x_2)$.

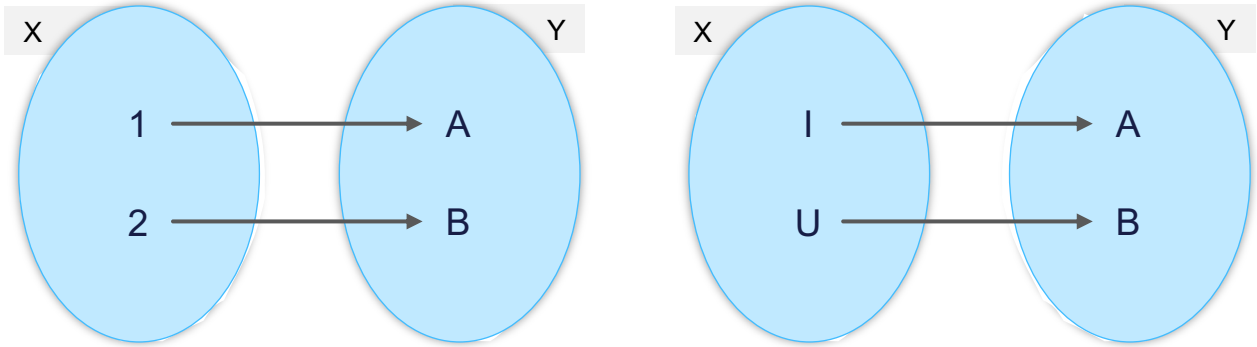
We must show that $x_1 = x_2$.

Let	$y = f(x_1) = f(x_2)$
Then	$x_2 = f^{-1}\big(f(x_2)\big) = f^{-1}(y) = x$
But also	$x_1 = f^{-1}\big(f(x_1)\big) = f^{-1}(y) = x.$
So	$x_1 = x_2$ and f is injective.

The above two theorems, in combination, tell us that the concept of a function being invertible is the same concept as a function being a bijection. A function f is a bijection if and only if it is invertible. For this reason, the terms *bijective* and *invertible* for functions are, in fact, interchangeable.

However, an important thing to note is that one is only able to "reverse engineer" elements from a set Y back to the original input elements in the set X if we have (1) information on what the input set X originally is, as well as (2) information on the what the bijective function f looks like.

For example, consider the following two bijections:



In both scenarios, we have bijections from two different sets X_1 and $X_2$to the set $Y = \{A, B\}$.
If we only provide someone with the set Y, and no information on the bijection or X, then it is not possible to reverse engineer the elements of Y to determine which input elements were used to arrive at the elements of Y.

This is a particularly useful concept and is often used in cryptography, as one can encode elements of X into Y and sharing only the encoded elements of Y. Only those who have the necessary information, X and the bijection itself, are able to accurately reverse engineer the elements of Y to obtain the original elements of X.

Bijections have long been used in cryptography. In each case, the framework is as follows:

> Suppose that person A wants to share the value x with person B confidentially. An external person C would like to discover what this value is.
>
> A encodes x using a cryptographic algorithm that takes x and what is commonly called a key (or in some contexts an initialisation vector/value (IV)) to create y.
>
> In some cases (symmetric encryption) the key must be treated as a secret and shared confidentially with B. In other cases (public-private key encryption) two keys are used, a public one for encrypting (that B gives to A) and a private one (that B alone holds) for decrypting. This equivalent to the operation $y = f(x)$ above.
>
> B would use the appropriate key depending on the scenario above to obtain the original value of x. This is equivalent to the operation $x = f^{-1}(y)$ discussed earlier.
> However, Person C is not able to recover x. Why?

Recall that in the earlier proofs, the function for recovering x is $x = f^{-1}(y)$. However, C is not in possession of source data, f, or of f^{-1} ("additional information" necessary to reverse protection). As a result, C is unable to compute x directly using f, nor to derive f from f^{-1}.

Note that even if they know the specifics of the cryptographic algorithm(s) used (which are often publicly published by an A to foster credibility in their data protection practices), that is at most partial information on f, as C in not in possession of the source data, keys/IVs or lookup tables necessary to have complete knowledge of $f$or f^{-1}.

Finally, the adoption of the various cryptographic algorithms used is grounded on security proofs predicated on validation of the absence of an efficient algorithm for determining x in the absence of knowledge necessary to reverse the protection. Here "no efficient algorithm" means that with current

or projected processing technology, it would take years, decades or even longer for a brute force attack to be successful.

Expressed another way, the only possible way for C, who is not in possession of the source data or the "additional information", to reverse the protection is to try to try all possible values of x and all possible values for keys/IVs in the cryptographic algorithm, and for each guess, checking whether the result is y, which is not computationally feasible, by design, based on the specified length of keys/IVs. Note that this is true for any one field in the input/output dataset, and best practice is to use unique keys/IVs for each individual field.

Linking the Math to Data Embassy

Data Embassy uses these properties of functions, and the security of cryptographic algorithms in several ways.

First, for each record in a source data set, a single, random pseudonym is generated and attached to the source record. This is a pseudonym" for the record as a whole and serves as a pointer, or look-up value used when reversing the protections, and each of the elements in the record is thus mapped to the record-level pseudonym. In the language of functions, as to the individual elements in the record, this mapping is many-to-one, so it is surjective, but not injective. However, the mapping from the record as a whole to the record-level pseudonym is exactly one to exactly one and thus a bijection.

Because the record-level pseudonym is randomly generated, there is no formula or calculation available for returning from the pseudonym to the cleartext record. Instead, a Master Index (i.e., a look-up table) is created, with restricted access, which serves as the additional information (i.e., the inverse function) necessary to reverse that part of the process.

Then, a subset of the source record ("selected fields"), including the record-level pseudonym, and typically omitting at least any direct identifiers, as well as any other fields not needed for the intended use of the data in protected form (the Variant Twin). These selected fields are then mapped to corresponding fields in the output Variant Twin. This is an exactly one to exactly one mapping and thus the function that does so is a bijection.

Note, however, that this function is actually a composite of many different functions, as each field to be included in the Variant Twin has its own protection configuration. Some fields (e.g., the record level pseudonym, and numerical indirect-identifiers and attributes not suitable for conversion to categorical) are simply copied in cleartext form. In most cases, the remaining fields (quasi-identifiers, indirect identifiers and attributes that were natively or converted to categorical) are deterministically pseudonymised. Of importance, unless action is taken to expand the scope of the determinism beyond field-level, by default, every pseudonymised field will have its own unique, randomly generated key/IV.

By design, access to the source data/master index and keys/IVs is restricted via technical and organisational controls. As a result, unless authorised, recipients of a Variant Twin do not have access to them and thus have no access to, again using the mathematical vocabulary introduced above, X or f^{-1} and thus will be unable to reverse the protection directly. The only option left would be to attempt a brute force attack: for a given y , try all possible values of x, and for each x, all possible values for the key/IV and see if the result matches y. When properly implemented, in particular with regard to key/IV generation and length, (see Appendix), each of the chosen techniques has the property that

there is no “efficient algorithm” for doing so. Note that the techniques selected for use in Data Embassy are noted by ENISA, and considered to be quantum computing secure.

However, an authorised user does have access to both X and f^{-1}, the additional information necessary to reverse the protection. For individual fields in the Variant Twin, including the record-level pseudonym f is a bijection, and thus always invertible to the original values in the source data set. Additionally, because of the inclusion of the record-level pseudonym, which attaches to all fields in the source, not just those selected for inclusion in the Variant Twin, access to omitted fields (in particular direct identifiers) is preserved. This extended relinking capability, which goes beyond mere reversal, is one of two key reasons why Data Embassy Variant Twins can ensure that 100% of the utility of cleartext is preserved. The second, of course, as noted earlier, is that the pseudonymised categorical values are fully compatible with advanced analytical techniques such as machine learning and AI, delivering 100% accuracy when compared to processing cleartext.

[1] Anonos systems, methods and devices are protected by a portfolio of granted international patents including, but not limited to: AUS 2018258656 (2021); US 11,030,341 (2021); CA 2,975,441 (2020); EU EP 3,063,691 (2020); US 10,572,684 (2020); CA 2,929,269 (2019); US 10,043,035 (2018); US 9,619,669 (2017); US 9,361,481 (2016); US 9,129,133 (2015); US 9,087,216 (2015); and US 9,087,215 (2015). See https://www.anonos.com/patents for more information.

APPENDIX - Data Embassy Pseudonymisation Algorithms

Anonos Data Embassy software pseudonymises data containing sensitive or regulated (e.g., personal) data. It enables users to process source data sets to generate two outputs:

1. A pseudonymised output (called a Variant Twin) in which case the data elements from one or more of the fields from the source data set are replaced with high entropy tokens.
2. A mapping output, called a Master Index, which contains the required information to recover the original data behind the high entropy tokens to reverse the pseudonymisation process.

Data elements can be transformed in a number of ways as an alternative to, or prior to, being replaced with pseudonyms:

1. Generalised using binning, masking, or rounding
2. Rescaling
3. Concatenation of two or more fields

Anonos Data Embassy can replace source data elements with four types of pseudonyms.

1. Reversible Deterministic
2. Reversible Non-Deterministic
3. Non-Reversible Deterministic
4. Non-Reversible Non-Deterministic

Pseudonym Type Definitions

Deterministic Pseudonyms

Deterministic pseudonyms replace recurring instances of the same data element value with the same pseudonym each time. Deterministic pseudonyms thus preserve referential integrity between pseudonyms over a defined scope. Possible examples of scope include within a single column, between two columns in a single table (e.g., country of origin and country of residence), across tables within a single database or between organisations, and across different databases.

When using deterministic pseudonyms with categorical fields, best practice is to use the narrowest scope consistent with the intended use case that preserves analytic utility, while still ensuring adequate protection against linkage and inference attacks. For example, by generalising age to age ranges using binning and then replacing each age-range value in an input data set with deterministic pseudonyms will result in a Variant Twin where it will be known that certain records have the same age-range, but not what that age-range is.

Random (Non-Deterministic) Pseudonyms

Pseudonyms that are random (non-deterministic) do not preserve consistency or referential integrity. Each recurrence of the same input value will be assigned a different unique pseudonym. The most common use is to create unique record-level pseudonyms that are used in controlled relinking. A second less common use is in the generation of test data, particularly if format preservation is used. This type of pseudonym is created using pseudorandom number generators that make use of the operating system kernel's entropy pool.

Reversibility

Pseudonyms can be reversible or non-reversible. These terms describe the options for recovering the original cleartext behind a pseudonym.

Non-reversible pseudonyms are generated using a technique called keyed hash message authentication code (Keyed HMAC). The algorithms are designed to be infeasible to reverse computationally. Instead, during the protection process, lookup tables called Master Indexes are created that map original cleartext to the generated pseudonym, which can be used to recover the cleartext when authorised.

Reversible pseudonyms are generated using symmetric encryption algorithms that take the source data element in cleartext and an encryption key as inputs to generate a ciphertext that is used as the pseudonym. The pseudonym can be directly reversed by using the key to decrypt the ciphertext.

Cryptographic Algorithms and Key Generation Details

The following table provides additional details regarding the specific algorithms used to generate each of the four types of pseudonyms.

Deterministic Reversible	Deterministic Non-Reversible	Non-Deterministic Reversible	Non-Deterministic Non-Reversible
Deterministic Authenticated Encryption using AES SIV IETF RFC 5297 with two 256-bit keys derived from PBKDF2, one for the initialisation value and one as the encryption/decryption key.	Keyed HMAC using SHA256 with a 256-bit initialisation value derived from PBKDF2.	AES GCM using a 256-bit initialisation value derived from PBKDF2.	Java SecureRandom API in conjunction with the Operating System kernel's entropy pool.

SecureRandom Value Generation

Data Embassy uses the Java SecureRandom API coupled with the OS kernel's entropy pool in order to convert source data elements into non-deterministic, non-reversible pseudonyms that are completely unrelated to the underlying data.

The actual code that generates these values is:

```
byte[ ] bytes = new byte[DEFAULT_SIZE]; PRNG.nextBytes(bytes);

return Hex.encodeHexString(bytes);
```

The DEFAULT_SIZE is the size of a pseudonymous token in bytes (16) and PRNG is an instance of SecureRandom.

The Java SecureRandom class provides a cryptographically strong random number generator (RNG) which complies with the specification FIPS 140-2, as mentioned in the official Java documentation.

Data Embassy uses this same strategy to generate the row-level pseudonyms for each record in the source data. Through Anonos' patented Controlled Relinking approach, authorised users can use these row-level pseudonyms to get back to the cleartext values for any field in the source data or produce a newly protected Variant Twin with new configuration values.