Encryption, pseudonymization, tokenization, and anonymization: an overview of the main techniques to securely process data

17 March 2023 - 13 min. read

Data Security and Governance | *GDPR*

Data protection (aka improperly “privacy”) and security have become increasingly important today, especially with the rise of big data and the increased use of digital technologies.

Data Protection refers to the right of individuals to control their personal information and to keep it from being disclosed to others unlawfully. It involves **protecting personal information from being accessed**, **used**, **or disclosed** by unauthorized parties and ensuring that individuals have control over how their personal data is collected, used, shared, and stored.

Data security, on the other hand, refers to the **protection of data from unauthorized access**, theft, corruption, or destruction. It involves safeguarding information systems, networks, and databases from security breaches and ensuring that sensitive information is protected from both internal and external threats.

In other words, data protection is concerned with protecting personal information from being misused, lost, or compromised. In contrast, data security is concerned with protecting information from being compromised, regardless of whether the information is personal or not.

To ensure privacy and data security, organizations must **implement appropriate policies and procedures**, such as access controls, encryption, firewalls, and security monitoring, to prevent unauthorized access to personal and sensitive data. Both privacy and data security

are necessary to protect individuals and organizations from harm and are essential in today's digital age, where vast amounts of personal and sensitive data are being collected and stored online.

In this article, we will focus on **data protection in terms of confidentiality, especially in relation to GDPR requirements**, and we will explain data security concepts that can be used to grant users the privacy required by law.

Data security

Data security is essential for several reasons: first, it **helps protect individuals' privacy** by preventing unauthorized access to their personal information. Protecting an individual's privacy is particularly important in healthcare, finance, and government industries, where sensitive data such as medical records, financial information, and personal identification must be protected from unauthorized access.

It is also crucial for **maintaining data integrity**; by implementing security measures such as encryption and data backups, developers can ensure that data is not tampered with or lost due to hardware or software failures.

Finally, data security is essential for **maintaining the reputation and credibility of businesses**. Data breaches and cyber-attacks can have severe consequences for businesses, including financial losses, legal liabilities, and damage to reputation. By implementing robust data security measures, companies can demonstrate their commitment to protecting sensitive data and maintaining the trust of their customers.

The central aspect of data security is the ability to process and store data to minimize the risk of exposing sensitive information, allowing normal operations while restricting the availability of sensitive data to the least number of systems and eyes that the use case permits.

Maintaining **the right balance between security and operators' agility is crucial** because both are essential for the smooth functioning of an organization. Security measures are necessary to grant the privacy required by regulations. These measures help to ensure that an organization can operate without interruption and that confidential information is kept safe. However, if security measures are too restrictive, they can limit the ability of employees and other authorized personnel to perform their duties effectively, potentially resulting in decreased productivity and morale.

On the other hand, it is essential to ensure that employees and other authorized personnel can perform their duties effectively and efficiently. Employees must be free to access the

information and resources they need to do their jobs without unnecessary obstacles or limitations. If operators are too restricted, they may be unable to perform their duties effectively, leading to decreased productivity and morale.

Therefore, it is essential to maintain a **balance between security and operators' ability to access the information they need** to ensure that the organization can operate efficiently and effectively while protecting its assets and sensitive information.

Four fundamental techniques can be used to produce datasets that operators can safely manipulate based on the level of information they need: encryption, pseudonymization, tokenization, and anonymization. This article will overview the above-mentioned techniques, with generic considerations for their implementation in cloud architectures.

Let's begin the overview of the techniques by describing them without further ado.

Encryption

Cryptography is a widely used technique for securing data and ensuring confidentiality, integrity, and authenticity that involves transforming data into an unreadable format by unauthorized parties. This transformation is achieved using various encryption algorithms such as **Advanced Encryption Standard (AES)** and **Rivest–Shamir–Adleman (RSA)**. The data can only be decrypted using a specific key known only to the authorized parties. Cryptography can also be used to ensure data integrity and authenticity by using digital signatures and hash functions.

Under GDPR, companies must protect personal data from unauthorized access, modification, and disclosure. Cryptography can be used to achieve this by encrypting personal data and ensuring that only authorized parties have access to the decryption keys. This means that even if an unauthorized party gains access to the data, they cannot read it without the decryption key.

Another critical aspect of **GDPR** is the right to erasure, also known as the right to be forgotten. **Companies must ensure that personal data is deleted upon request** by the individual concerned or at the termination of the retention period. Cryptography can be used to achieve this by securely erasing the decryption keys, rendering the encrypted data unreadable. That may be faster and less prone to errors than searching for all the user data occurrences and deleting them.

Pseudonymization

Pseudonymization is the process by which an individual is prevented from being identified through their data. The GDPR is particularly strict regarding pseudonymization: the **impossibility of tracing the identity** of the data owner by other parties than the data controller must be absolute.

This technique protects personal data by making it impossible to link to the original individual identity (without holding the pseudonymization algorithm or table) while allowing the data to be used for specific purposes. **Pseudonymization is often used when there is a need to share data but where it is essential to protect the privacy of individuals**.

Examples of where pseudonymization can be used include medical research, clinical trials, marketing, and social media analytics, especially when creating datasets for machine learning, reports, or statistics.

A good pseudonymization algorithm replaces a person's identifying information, such as their name, address, or date of birth, with a pseudonym or other artificial identifier.

The resulting pseudonym is unique to the individual but does not reveal any personally identifiable information. The original data, the algorithm, and/or the transcodification matrix are stored separately, allowing the system to operate normally.

One of the main advantages of pseudonymization is that it allows data to be shared while protecting the full confidentiality of the data subject's identity. This technique also reduces the risks associated with data breaches, as even if the pseudonymized data is stolen, it is impossible to link it back to specific individuals. However, **it is essential to note that pseudonymization does not guarantee absolute anonymity**. It is still possible to re-identify the data if someone can access the pseudonymized and original data.

Tokenization

Tokenization is another technique used for processing data securely and **involves replacing sensitive data with a unique identifier or token**. The original data is stored securely in an independent database, while the token represents the data in other systems. Tokenization is commonly used in payment processing, where it is essential to protect sensitive financial data, such as credit card numbers, while still allowing for transactions to be processed.

Tokenization typically involves using a **tokenization algorithm**, which generates a unique token for each piece of sensitive data. The token is usually a randomly generated string of characters or a hash value. The original data is then encrypted and stored securely in the secured vault.

One of the main advantages of tokenization is that it provides a high level of security for sensitive data, as the original data is never transmitted or stored in unencrypted form. This technique also **simplifies compliance with data protection regulations**. However, it is important to note that **tokenization does not provide absolute anonymity**, as it is still possible to link the token back to the original data if someone has access to both the token and the data vault.

Anonymization

Anonymization is a more extreme form of data processing that **involves the removal of all identifying information from data**. Anonymization is typically used when there is no legitimate need to retain identifying data and where it is essential to protect the privacy of individuals. Examples of where anonymization can be used include public health research, demographic analysis, and public opinion surveys.

Anonymization typically involves the removal of any identifying information from data, such as names, addresses, or other personal information.

According to most laws and regulations, anonymization produces the same effects as deleting the data or deleting the encryption key used to encrypt them because the crucial aspect is to make the identifying information impossible to obtain.

The resulting data is aggregated or summarized to provide insights without revealing personal information. There are various techniques to obtain data anonymization, such as data masking, generalization, or suppression.

Getting started with GDPR on AWS

Amazon Web Services (AWS) offers various services and infrastructure that can be used to implement GDPR request handling; what follows is a list of typical impacted services:

- **Amazon S3**: Amazon S3 is a highly scalable and secure object storage service that can store and manage personal data. With S3, you can easily set up lifecycle policies to automatically delete or archive data based on retention periods.
- **Amazon EC2/Fargate/Lambda**: Amazon EC2 provides scalable computing capacity in the cloud, making it possible to build and run applications that handle GDPR requests. Your application can be configured to run in a secure virtual private cloud (VPC) environment, ensuring that data is protected and the perimeter is well delimited and controlled.
- **Amazon RDS**: Amazon RDS is a managed relational database service that provides an easy way to set up, operate, and scale a relational database. RDS supports a variety of

database engines, including MySQL, PostgreSQL, and Oracle, making it easy to store and manage personal data. It also supports encryption at rest provided by AWS and optionally in transit encryption if supported by the DBMS.

- **Amazon Kinesis**: Amazon Kinesis is a real-time data streaming service that can be used to collect, process, and analyze data from various sources. With Kinesis, you can quickly and efficiently process GDPR requests as they come in, ensuring they are handled promptly and efficiently.

In addition to these services, AWS also provides the infrastructure to support GDPR compliance, including:

- **AWS Identity and Access Management (IAM)**, which you can use to fine-grained control who has access to the Cloud resources, S3 objects, and AWS-managed cryptographic keys.
- **AWS Key Management Service (KMS)**: KMS is a managed service that makes it easy to create and control the encryption keys used to protect personal data stored in AWS. With KMS, you can create and manage keys, define key policies, and control access to the keys.
- **AWS CloudTrail**: CloudTrail is a service that enables you to log, continuously monitor, and retain audit-related events that occur in your AWS account.

Overall, AWS provides a robust set of services and infrastructure that can be used to implement GDPR request handling. By leveraging these services, you can ensure that personal data is stored, processed, and transmitted securely and complies with GDPR requirements.

To implement the techniques previously described into a Cloud application, you should usually:

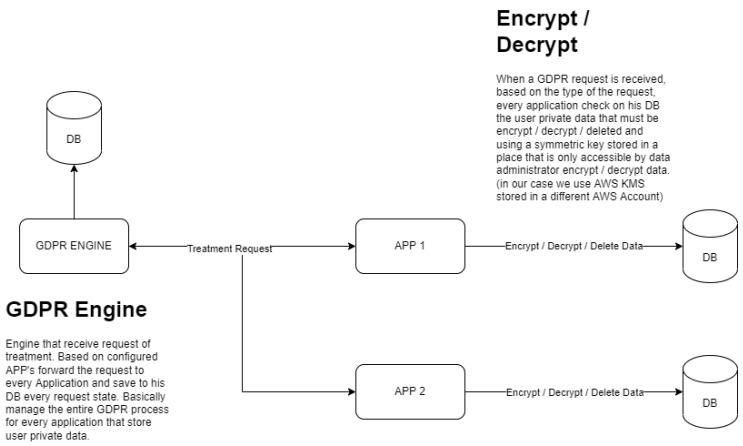
- Identify the sensitive data that must be protected, including personally identifiable information (PII), financial information, health information, or any other sensitive data.
- Choose a technique appropriate for the type of data you are working with and the level of protection required.
- Depending on your chosen technique, you may need to add specific features to your code. For example, if you use encryption, you must use an encryption library to encrypt the sensitive data before storing it in your database. Similarly, if you use tokenization, you must generate a token for each piece of sensitive data and store the tokens in your database instead of the original data (whenever possible)

There is also another approach: if you are building a microservices-based application, you can build a specific service to manage data protection and all regulation-specific features.

A sample design: centralized GDPR system

The following is an example of how we **build a GDPR service to handle data encryption and specific request related to GDPR compliance**.

This system is designed to **serve multiple applications with multiple data sources**. The main requirement is centralizing all GDPR-related work and user request handling, such as the "right to be forgotten."



A GDPR service to handle data encryption and specific request related to GDPR compliance

The Core GDPR System centralizes, manages, and forwards every user request to the configured APPs. It must implement an API callback that receives the response of the processed request from the application. This is because the time to process a GDPR request could be high due to the number of records and sensitive data that must be processed.

Every APP must implement an API that receives the GDPR Request and, based on the type of request, encrypt / decrypt / delete sensitive data from its DB. This Job must always reply to the GDPR System with the detail and the status of the processed request.

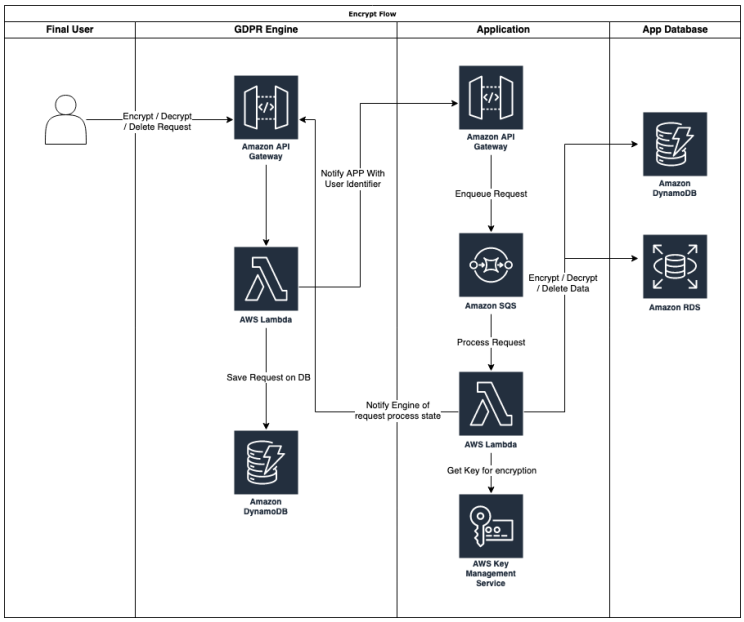
Every request has the identifier of the user that wants his sensitive data deleted from the system.

Encrypt / Decrypt uses a symmetric key that must be stored in a secure place that only Data Administrator can access. (In an AWS Environment, we typically use an AWS KMS key

stored in a different AWS account, accessible only by the backend application code and the Data Administrator).

Encrypt / Decrypt / Delete Flow

Let's see how the encryption flow works:



As we can see from the Flow chart above, a user can **generate an encrypt, decrypt, or delete request to the GDPR Engine**. In our case, we decided to create this engine taking advantage of the AWS Serverless Managed services like:

- AWS API Gateway, which allows us to expose API
- AWS Lambda, which will run the code that manages all the requests.
- AWS DynamoDB as the data store for the GDPR Engine.

When a treatment request is received, the GDPR engine saves the request in the DynamoDB table and checks the configured APPs that must be notified, and for each one, it calls the provided API.

When the application receives the request, it pushes it to an AWS SQS queue. This is because, in applications that store and process a large amount of data with user-related information, we can receive many GDPR-related requests. Depending on the data structure and volume, a single request could take some time to process.

Using a queue, we can **decouple the processing service from the main application**, allowing them to scale independently. Decoupling with SQS can also be helpful in building robust retry systems and ensuring that each request is retained until fulfilled.

It is also possible to **detect failures, and redirect failed requests to a dedicated queue** (dead letter queue) to change the consumer computing power, for example, switching from a Lambda Function to a container. In addition, we can detect second-level failures and alert an operator.

For more information on decoupling services using SQS, please refer to our previous article.

Every request is then processed based on his type (encrypt, decrypt, delete):

- **Encrypt** will use a Customer Managed KMS key that is stored in a different account accessible only by our Application Backend and Data Administrator to encrypt user-sensitive information.
- **Decrypt** will use the same KMS key to decrypt the user-sensitive information.
- **Delete** will remove or replace with a random string the user-sensitive data.

When the request is processed, regardless of the positive or negative status of the request processing, it will notify the GDPR engine, which will save on the DynamoDB table the result.

Conclusions

This was a high-level overview of the main techniques available to securely protect and process data in modern Cloud-Native applications.

Many specific technologies, services, and design patterns can be used to build GDPR-compliant applications, and this article is only the first introduction to the themes.

If you are interested in this topic, drop us a message or leave a comment and stay tuned for other articles about GDPR and AWS!

About Proud2beCloud

Proud2beCloud is a blog by beSharp, an Italian APN Premier Consulting Partner expert in designing, implementing, and managing complex Cloud infrastructures and advanced services on AWS. Before being writers, we are Cloud Experts working daily with AWS services since 2007. We are hungry readers, innovative builders, and gem-seekers. On Proud2beCloud, we regularly share our best AWS pro tips, configuration insights, in-depth news, tips&tricks, how-tos, and many other resources. Take part in the discussion!

Alessio Gandini

Cloud-native Development Line Manager @ beSharp, DevOps Engineer and AWS expert.Since I was still in the Alfa version, I’m a computer geek, a computer science-addicted, and passionate about electronics all-around.At this moment, I’m hanging out in the IoT world, also exploring the voice user experience, trying to do amazing (Io)Things.Passionate about cinema and great TV series consumer, Sunday videogamer

---