absio®

A Guide to Encryption for GDPR Compliance

720 836 1222 **sales / support**

sales@absio.com **email**
www.absio.com **web**

8740 Lucent Boulevard, Ste 101
Highlands Ranch, CO, 80129

TABLE OF CONTENTS

INTRODUCTION

Who This Guide is For

- **Controllers and/or processors** who have assessed the risk posed to their data subjects by a breach of their personal data and have determined that data should be secured with encryption
- **Processing solution vendors** currently offering or planning to implement encryption to satisfy customer demand for GDPR-compliant solutions
- **Others, such as consultants, insurers, or regulators,** in need of a comprehensive GDPR-specific gap analysis/audit tool to assess current or proposed encryption solutions intended to enable compliance with the GDPR

This guide is written in a plain-language/least-technical manner; expertise in encryption or the GDPR is not required.

What This Guide is For

- To enable readers to **understand how encryption does, and does not, assure compliance** with the GDPR
- To **conduct gap analyses and/or compliance audits** on current or proposed encryption solutions whose purpose is to enable compliance with the GDPR
- To **develop a comprehensive set of encryption-related technical requirements** for current or proposed data processing operations

This guide is suitable for any type of software platform, application architecture, data type, device type, processing method, or scale of operation.

The Purpose of Encryption in the GDPR

Pseudonymization[1] and encryption[2], a form of pseudonymization[3], are the only technical data security measures specifically described in the GDPR.[4] The purpose for their inclusion is to obligate controllers and processors handling personal data to prevent:

1. **Confidentiality Breach:** Access to identifiable personal data by unauthorized persons. Unauthorized access occurs in three ways:
    a. **Infiltration:** Unauthorized access to intelligible data stored where intended. Infiltrators can be insiders accessing data to which they are not privileged, or outsiders who have gained insider access to applications, devices, and/or networks, typically through compromised credentials or malware.
    b. **Exfiltration:** The transfer of intelligible data to unintended or unauthorized devices or locations, or unauthorized physical possession of an authorized data storage device. Data is exfiltrated by

malware, human infiltrators, and malicious or negligent insiders via internet connection, emails and texts, removable data storage, and by device loss or theft.

c. **Interception:** The capture of data while it is being transmitted from one device to another device.[5]

2. **Integrity Breach:** Accidental or malicious adulteration of personal data.
3. **Availability Breach:** Damage or destruction of personal data enabled by infiltration. Note that the use cases for preventing an availability breach through use of encryption is limited to those dependent on successful infiltration. Data can be damaged or destroyed by other means, such as storage device loss, theft, or damage, encryption key loss, unmitigated ransomware, or denial-of-service attack.[6]

Definitions

"Data" in the remainder of this guide refers to personal data as defined by the GDPR.[7]

"Breach" refers exclusively to the three forms of breach defined in the preceding section.

"Encryption solution" refers to third-party encryption solutions owned by an organization or those with whom it contracts, encryption-as-a-service, and applications or services with built-in or added encryption capabilities.

"User," unless otherwise stated, is a human or system who may be authorized or unauthorized (an **"infiltrator"**), but is nonetheless able to request and potentially gain access to a network, device, application, or data.

"Cleartext" is stored or transmitted data that has not been encrypted or that has been decrypted.

GDPR Guidance on Evaluating Encryption Solutions

The need to evaluate encryption solutions intended to protect data subjects and the accountable organization is explained in the Working Party 29, Guidelines on Personal data breach notification under Regulation 2016/679 issued October 2017. Paragraph II D "Conditions where notification is not required," paragraph 7, states:

> . . . a failure to comply with Article 33 will exist where a controller does not notify the supervisory authority in a situation ***where the data has not actually been securely encrypted***. Therefore, when selecting encryption software controllers should carefully weigh ***the quality and the proper implementation*** of the encryption offered, ***understand what level of protection it actually provides*** and ***whether this is appropriate*** to the risks presented. Controllers should also be ***familiar with the specifics of how their encryption product functions***. [emphasis added]

GDPR BREACH RISKS AND MITIGATIONS

Data Subject Risk

The GDPR requires regulated organizations to inventory and map the flow of data between persons and systems[8] and to assess the degree and kind of risk to the data subject.[9] The degree and kind of risk documented as a result of these processes is the basis for determining if encryption and/or pseudonymization is warranted.[10] This guide assumes that one or more of the sets of data recorded in the data inventory warrants encryption.

Controller/Processor Breach Risk

A reportable breach[11] of intelligible data may be one of the largest, if not the largest, GDPR-related financial risk facing controllers and processors. A data breach may result in an administrative fine based on a single infringement, but historically, data breach investigations often determine that the breach resulted from cascading failures[12], which may lead to multiple infringements and larger fines. Reported breaches will result in supervisory authorities assessing the risk to data subjects based on the use of encryption, or lack thereof, when calculating administrative fines.[13] In addition to administrative fines, insurance rate increases, and mitigation costs, the findings of supervisory authorities may precipitate or strengthen civil litigation, further increasing the total cost of a reported data breach.

Processing Application Provider Risk

Firms who provide applications or application-based services to processors, but that do not have access to data, are not regulated by the GDPR. However, they are still at risk. The GDPR makes no distinctions between processing data with applications developed in-house and applications provided by third parties. In either case, the controller/processor must be able to demonstrate that applications used to process data comply with all applicable GDPR requirements,[14] such as incorporating data protection by design and by default and use of risk-appropriate, state-of-the-art data security technology such as encryption.[15]

Given the increased liabilities arising from failure to comply with the GDPR, controllers, processors, and insurers will be prone to transfer risk to application providers by requiring them to purchase specialized cyber insurance, if available in the application providers country. They may require processing solution providers to attest to their applications' GDPR compliance and/or seek more favorable indemnification terms in licensing agreements. If so, willingness on the part of processing solution providers to accept additional risk, and/or to provide documentation detailing how their application supports GDPR data security compliance, may become a competitive benefit.

Non-Reportable Data Breaches

If a data breach is "unlikely to result in a risk to the rights and freedoms of natural persons," it does not have to be reported to a supervisory authority.[16] Likewise, reporting a breach to affected data subjects may not be necessary if the breached data was encrypted.[17] However, upon becoming aware of a breach, the breach must be evaluated immediately.[18] The risk to the data subject must be assessed and that assessment must be documented even if the breach is not reported.[19] The documentation should include a rationale for the decision to forgo reporting and documentation supporting assertions that at the time of the breach, encryption was:

- "state of the art"[20] and "appropriate"[21]
- regularly tested, assessed, and evaluated for effectiveness[22]
- if applicable, evaluated as part of a data protection impact assessment (DPIA)[23,24]
- incorporated into the incident response plan[25]
- reevaluated immediately after discovering the breach to determine if notification was required[26]

Non-Reported Breach Risk

The decision to not report a breach is not a guarantee it will not be reported at a later time. The criteria for not reporting is that risk to the data subject is unlikely—not impossible. The decision is necessarily somewhat subjective, relying on how the controller, and if applicable, their processor, defines "unlikely" in relation to the mechanics of the breach (e.g., how it happened and the set of data breached) and the efficacy and scope of encryption solution(s) in use at the time of the breach. Exfiltrated encrypted data is not attributable to the source of the data unless the data has been decrypted; discovery of attributable decrypted data is reportable. Decrypting exfiltrated encrypted data requires possession of the keys or brute-force decryption. Brute-force decryption, while theoretically possible, is unlikely if strong encryption (i.e., appropriate, state-of-the-art encryption algorithms) are employed. However, brute-force decryption of exfiltrated encrypted data is likely if weak or deprecated encryption algorithms were used.

The decision to not report a breach may be reversed when:

1. Controller/processor staff subsequently determine that the breach should have been reported
2. Law enforcement or other third parties subsequently discover the breach, an increasingly common occurrence[27]
3. A supervisory authority, in contravention to the controller, determines that risk to the data subject was likely

In all three cases, supervisory authorities will analyze the original rationale for deciding not to report the breach to determine if it was reasonable. If one or more of the following conditions are true at the time of the breach, supervisory authorities have ample cause to question the rationale for not reporting:

- Discovery that unauthorized users gained access to decrypted data
- Discovery by staff or a third party of exfiltrated decrypted data
- The encryption solution used deprecated or weak encryption algorithms[28]
- The encryption solution(s) were not regularly tested, assessed, and evaluated for effectiveness
- The breached data was initially stored as cleartext and encrypted at a later time, and the post-breach evaluation does not positively determine that only encrypted data was breached
- Of the data that was breached, some was stored as cleartext and some was stored encrypted, and the post-breach evaluation does not positively determine that only encrypted data was breached
- Pre- or post-breach evaluations are characterized by the supervisory authority as cursory, or pre- and/or post-breach evaluations were not conducted

ABOUT ENCRYPTION

Encryption-Enabled Data Security Functions

There are four encryption-enabled data security functions:

- **Protect stored data.** All formats of stored data can be encrypted.
- **Verify identity.** Cryptographic authentication is one means of assuring that someone, or something, is who they say they are. Humans, devices, operating systems, applications, databases, groups of files (e.g., directories or folders), individual files, and network transmissions can be authenticated, and they can be authenticated to each other. For example, a human user can be authenticated to an application, the application can be authenticated to a database, and the user authenticated to a file containing the results of a query of that database.
- **Preserve and verify data integrity.** Cryptography can preserve integrity by assuring that only authenticated users can access data and by verifying that stored or transmitted data has not been tampered with.
- **Protect data in transmission.** Network transmission encryption renders intercepted information payloads[29] in transmission unintelligible.

Combining Security Functions

Combining and coordinating encryption-based security functions strengthens security. For example, use of cryptographic authentication and folder encryption mitigates one form of breach—exfiltration due to device loss/theft—provided that all the data that should be in the folder is actually in the folder when the device was lost/stolen. When a user is logged in, even though they have been cryptographically-authenticated, folder encryption does not check file integrity or protect data in transmission, and it does not prevent data exfiltration by malware or by a malicious or negligent human user.

Combining and coordinating the use of all four functions provides a higher level of data security. For example, data is:

- Encrypted when initially stored,
- Only accessible to multi-factor cryptographically-authenticated users,
- Cryptographically integrity-checked each time it is accessed,
- Transmitted as encrypted data through an encrypted connection, and
- If exfiltrated, unable to be accessed on unauthorized devices even if user credentials have been compromised.

In this scenario, infiltration, exfiltration, adulteration, and interception are all prevented.

Data Formats

Encryption solutions may be suitable for one data format but not another. To provide adequate coverage, the type of encryption solution(s) employed should protect all applicable data formats:

- **Unstructured Data.** Examples include documents, images, media files, and data extracted from a database and stored as individual files. Content in unstructured data is not organized to optimize processing. To make unstructured data reusable, minimal metadata[30], such as a filename and file type extension, is appended to the payload.
- **Semi-Structured Data.** Semi-structured data is stored as files. Data is semi-structured when one or both of the following two conditions is true. First, the content may be organized to make it easier to process.[31] Second, additional metadata has been appended to the files en masse to make the data easier to manage.[32]
- **Structured Data.** Content is organized to optimize processing (i.e., rows and columns in a relational database).

— **Data Stream.** To transmit data, digital content is converted to a sequence of data packets that are transmitted to a specified location. Those packets can be consumed at their destination but not saved (e.g., streaming video or audio), or they are reassembled and saved in their original format (downloaded). Metadata appended to the packets facilitates routing, streaming, and reassembly.

Data Format Conversion

During processing, applications may convert data from one type to another. Not all encryption solutions protect all data formats. Examples include:

— Unstructured data can be converted to semi-structured data by appending metadata. When metadata is appended by a classification system, moving the data to a common storage location often follows.

— Unstructured data can be converted to a binary large object (BLOB) and stored in a database field, and later extracted from the database and converted back into a file.

— Structured data can be converted to unstructured or semi-structured data by storing the results of a database query as a file, or sending database query results in a stream to a web interface, where it may be stored on the machine hosting the web interface.

— Unstructured, semi-structured, and structured data are all converted to a stream during transmission.

TYPES OF ENCRYPTION SOLUTIONS

Cryptographic Authentication

Cryptographic authentication is usually a component of a larger system: a device, operating system, application, and/or an identity and access management system (IAM)[33]. Cryptographic authentication is coupled to authorization; authentication verifies identity whereas authorization determines what data an authenticated user can or cannot access.

Network Traffic Encryption

Network traffic encryption renders the payload portion of a stream of data flowing from point to point unintelligible if intercepted. The metadata portion is transmitted as cleartext. Network traffic encryption is built into operating systems and/or applications.

Database Encryption

An entire database, or only selected portions of a database, may be encrypted. Most modern databases offer built-in functions that can encrypt data as it is written. Typically, built-in database encryption requires integration with compatible applications and a separate encryption key manager. Applications can also encrypt data and then store it in a database, or a separate encryption application or service can sit between an application and a database to encrypt/decrypt data as it is read in or out.

Disk Encryption

Full-Disk Encryption (FDE) automatically encrypts the entire drive. Self-Encrypting Drives (SEDs) are hard drives with encryption built into the disk controller that automatically encrypts the entire drive.

Folder Encryption

Folder encryption protects content stored in specified folders or directories.

Device Encryption

Device encryption usually encrypts all the data on a device; some also encrypt the operating system.

File Encryption

Files selected by a user (human and/or system) are individually encrypted with their own key.

Encryption Key Managers

Key managers generate, backup, recover, update, delete, and otherwise manage encryption keys.

Centralized key managers are usually comprised of a hardware security module (HSM)[34] or a virtual security module and server-based key management software. To authenticate users, encrypt or decrypt data or check data integrity, applications connect to the central key manager to fetch the appropriate keys. Encrypt/decrypt and other key management functions are not available offline, and network transmission encryption, except for transmission between the application and the key manager, is managed by other means.

Flexible key managers can be centralized or decentralized; keys can be generated on any device. Keys and encrypted data can be stored where needed. Brokering software enables flexible key managers to distribute, synchronize, and backup keys and data for applications that run on multiple devices and platforms, and/or brokering software enables multiple unrelated applications to securely exchange encrypted data with each other. When installed on the device storing the data, flexible key managers enable offline encryption and decryption.

In-Application Encryption

Some applications have built-in encryption-enabled security functions. For those that do not, there are multiple solutions that enable software developers to incorporate cryptographic authentication, data encryption, cryptographic integrity assurance, and transmission encryption into new and existing applications.[35]

Additionally, some solutions support cryptographically-secured metadata. Metadata may be cryptographically bound to data so that it is only available when data is decrypted. Metadata associated with a particular piece of data may be stored in an encrypted database, usually to support proper separation of duties[36], that is, enabling data to be managed without decrypting the payload. Keys for in-application encryption can be fetched from a centralized key manager or generated in-application using a flexible key manager.

ENCRYPTION SOLUTION LIMITATIONS AND VULNERABILITIES

Depending on the mechanics of a given breach, encryption-enabled security functions alone may not prevent a reportable breach. The limitations and vulnerabilities should be taken into account when evaluating current or proposed encryption solutions. Information pertaining to limitations and vulnerabilities recounted in this section is incorporated into the gap analysis/audit questions provided at the end of the guide.

Decryption in Memory

Encrypted data, with rare exception[37], must be decrypted in memory to be used. Breaches attributed to memory-scraping malware (i.e., malware that harvests and exfiltrates decrypted data temporarily stored in a system's memory for processing) should be reported unless mitigated by other measures.

Post-Exfiltration Decryption

It is possible that exfiltrated encrypted data may be decrypted if 1) usable keys are also exfiltrated, or 2) weak or deprecated encryption algorithms are used. Breaches discovered by detection of decrypted exfiltrated data or exfiltration of data encrypted with weak or deprecated algorithms should be reported. Encryption keys themselves can be stored encrypted. Exfiltrated copies of encrypted keys may present little threat.

Cryptographic Authentication

Cryptographic authentication may be a component of one or more authentication factors provided by a device, application, authentication system or service. For data security to be effective, the use of multiple authentication factors is more important than the means of authentication. The data security protections afforded by strong content encryption, cryptographic integrity checking, and network transmission encryption do not prevent a logged-in user from exfiltrating or altering data. Therefore, multi-factor authentication is highly recommended to protect against access using compromised credentials.[38]

Compromised Credentials and Exfiltrated Data

Exploits which exfiltrate compromised credentials and data are among the most common. In this scenario, encrypted data has been exfiltrated and stored on an unauthorized device. If a user in possession of compromised credentials can log in to an application capable of decrypting the exfiltrated data, a reportable breach has occurred even though the exfiltrated data was encrypted. This risk can be mitigated if there is an additional authentication factor that is not available to the user in possession of the compromised credentials. This is usually an authentication factor applied by an administrator to authorized devices. Since the administrator-applied authentication factor will not have been applied to the device storing the exfiltrated data, the user in possession of compromised credentials will be unable to log in to the application to decrypt the exfiltrated data.

Decrypt and Save as Cleartext

If a user can log in, decrypt stored data, and save it as cleartext to disc or removable media, transmit it, or enable it to be consumed by applications other than those intended for processing,[39] cleartext data can be exfiltrated unless mitigated by other means.

Copy or Send Encrypted Data

If a user can move or copy encrypted data to disc or removable media, or transmit it, encrypted data can be exfiltrated unless mitigated by other means.

Database Encryption

Database encryption protects content if the database is exfiltrated, but it does not prevent logged in users from querying the database, viewing, and exporting cleartext data unless otherwise mitigated. Data integrity is maintained through functions built into the database, which may or may not utilize cryptographic integrity checking. Non-relational databases[40] process stored semi-structured data. A separate encryption solution designed to integrate with a NoSQL database may be required to encrypt stored semi-structured data.

Full-Disk Encryption, Self-Encrypting Drives, Device Encryption, and Folder Encryption

Disk, drive, device and folder encryption prevent exfiltration of data stored on lost or stolen devices if the user is logged out of the device, or in the case of folder encryption, logged out of the protected folders/directories. When a user is logged in, data can be exfiltrated unless otherwise mitigated. These forms of encryption do not prevent a user from copying or moving encrypted content. Cryptographic data integrity checking may or may not be provided.

File Encryption

File encryption limits exfiltration of decrypted data to objects in memory (i.e., files that are open). Exfiltration of decrypted content may be mitigated by other means such as by a data loss protection system or controls within the application. Encrypted files are protected on lost or stolen devices when the user is logged out. File encryption relies on a user to recognize that content should be encrypted and to take specific actions to encrypt the file. Cryptographic data integrity checking may or may not be provided. File encryption does not necessarily prevent a user from copying and moving encrypted content.

Key Management

Key management vulnerabilities include:

- **Key security:** Unauthorized users with keys can decrypt data. Keys may or may not be stored encrypted.
- **Key backup and recovery:** If keys are lost or corrupted and cannot be recovered, the data is unrecoverable (an availability breach).
- **Key rotation:** If keys cannot be rotated, then a potential or actual key compromise cannot be mitigated.
- **Key update:** Encryption algorithms lose strength or become more vulnerable over time. To maintain key strength, the strength and/or type of encryption algorithm used must be updateable.
- **Key access control:** Whoever has access to the key has access to the data. If that's a third party, they are a processor.

Key Granularity

For unstructured and semi-structured data, the amount of data encrypted per key varies from a single key protecting a large data repository or database to a unique key per individual file. For structured data, databases can be encrypted in their entirety or encrypted per row, column, or field. The computational workload to brute-force decrypt data protected by a given key is the same if that key is protecting two kilobytes or two gigabytes of data. The more granular the keys, that is, the more unique keys there are for a given volume of data, the less likely brute-force decryption is to succeed.

Network Traffic Encryption

Depending on the use case, it may not always be possible to assure that network traffic encryption is properly configured and up-to-date on all the points that risky data may traverse. Common issues include misconfiguration, partial use of available security components, improper implementation, and use of deprecated protocols to enable backward compatibility.

Network traffic encryption is usually the sole protection against the interception of intelligible data afforded to all internet-connected devices—thus the perpetual effort by criminals and certain nation-state intelligence services to defeat it. Risk is mitigated by transmitting encrypted data instead of cleartext data, so that a network-traffic-encryption compromise only yields encrypted data.

Additionally, many organizations share data with other organizations in the normal course of operations. Although the GDPR does not state it explicitly, further guidance and case law may hold senders of personal data more explicitly responsible for securing the data to the point of reception. This risk can be mitigated by transmitting encrypted data through an encrypted connection, and decrypting the data to be processed and/or stored on the receiver's end.

Cryptographic Authentication

Cryptographic authentication may be a component of one or more authentication factors provided by a device, application, or authentication system or service, but in terms of effective data security, what is most important is the use of more than one authentication factor. Strong content encryption, cryptographic integrity checking, and transaction encryption are rendered moot by weak authentication. Multi-factor authentication is highly recommended.[41]

Partiality and Temporality

For a given set of stored data, encryption may be of limited or no benefit if in routine processing a portion of the data is encrypted and the remainder is not, or if data is routinely stored as cleartext and encrypted sometime later. Some commonly overlooked forms of partiality and/or temporality include:

- Shadow IT[42], which can be defined in the context of the GDPR as the consumption of data by an application that utilizes the data for operations that are not included in the records of processing activities[43]
- Cleartext temporary files that persist until manually deleted
- Cleartext data awaiting classification and subsequent encryption based on class
- Cleartext data stored in browser caches or downloaded via a web user interface and stored locally. This is common for web and cloud-native applications. Data in the cloud may be encrypted, but locally-stored data is usually not.
- Cleartext data is "forward positioned," that is, it is stored in web caches[44] and/or content delivery networks[45] on servers geographically closer to users in order to reduce latency and increase performance. This is common for cloud-native applications.
- Metadata that may include identifiable personal information such as names, GPS locations, timestamps, IP addresses, and more
- Database queries stored as cleartext

Partiality and temporality risks can be mitigated if all instances of a given set of data are encrypted when they are initially stored and not decrypted thereafter except by authenticated users. Partiality and temporality may be mitigated on a breach-by-breach basis if there are forensic means able to determine that breached data was encrypted, not cleartext.

Unconstrained Application Storage and Transmission Functions

Many applications provide users with multiple methods of storing decrypted data in user-selected locations and devices, as well as multiple methods of transmitting decrypted data. Risks can be mitigated through the appropriate use of technical and organizational controls designed to constrain storage and transmission methods and potential storage locations.

Subversion by User

If applying encryption requires users to evaluate content for risk and/or to make additional manual efforts to encrypt, encryption may be subverted. Two common examples are: 1) relying on users to properly evaluate the risk of the data and then requiring them to move it, mark it, or exchange keys to encrypt it, and 2) enabling users to log in to processing applications via open Wi-Fi. This type of risk may be mitigated by organizational or technical measures that minimize or eliminate reliance on the quality and consistency of user decisions.

GDPR-MANDATED DATA MANAGEABILITY

Data Subject Rights

In addition to securing data, compliance with the GDPR requires that for each individual data subject, their data:

- must be accurate[46]
- must be rectified upon request[47]
- must be erased upon request[48]
- may be retained but its processing halted or resumed upon request[49]
- may be retained only for the period of time permitted for lawful processing[50]
- when provided to other organizations, must include a record of what was provided and to whom it was provided[51]

Compliance with these rights requires the ability to find, alter, classify, delete, and maintain a record of how each individual data subject's encrypted data was processed, throughout that data's lifecycle. Meeting this requirement for data processed and stored in a relational database is usually straightforward since transaction logging is usually built into the database.

Compliance with these rights when data is stored as unstructured and semi-structured data is more complex because transaction logging is an external process. It is common for status and classification metadata to be appended to individual files, or stored in a separate database, or both. An individual data subject's information may be stored on multiple devices and accessed by multiple applications, which creates a requirement for synchronizing files, logs, and metadata. Take this into consideration when evaluating encryption solutions for unstructured and semi-structured data.

Searching Encrypted Data

Meeting the requirements for data manageability may require the ability to search unstructured and semi-structured data. When searching within large bodies of unstructured or semi-structured data, such as a search for all the personally identifiable information related to an individual data subject, usually, a full-text search database derived from the data is searched, not the data itself. When files are created or modified, their contents are indexed to update the full-text search database. If so, the encryption solution must be able to interact with the indexing and full-text search database solutions. Take this into consideration when evaluating encryption solutions for unstructured and semi-structured data.

Integrating Encryption Solutions and Encrypted Data with Other Systems

When multiple encryption solutions are contemplated, and encrypted data needs to be handled by more than one encryption solution or processing application, system interoperability is a necessity. Keys may or may not function across encryption solutions even if the solutions adhere to a common key management interoperability protocol.

Additionally, encrypted data may need to be decrypted for use by other systems. Common use cases include other processing applications, decryption for security analysis (e.g., data loss prevention systems) and to add or edit metadata and in many cases, move the data to a new location afterwards (e.g., content classification systems). Identify and incorporate integration needs in the evaluation process.

Performance

Good performance is not a GDPR requirement, but it is a practical one. The impact of encryption on performance may run from negligible to significant depending on the solution chosen, proper configuration, changes in scale (such as the volume of stored data), database size, processor load, and, if keys are fetched from elsewhere, connection speed and reliability. Consider performance at initial and larger scales in the evaluation process. Performance questions are included in the evaluation process.

GDPR ENCRYPTION GAP ANALYSIS/AUDIT

In simple language, to assure GDPR compliance through encryption, two fundamental questions must be answered: where is the data, and is it protected? The gap analysis/audit questions that follow are designed to assure appropriateness and coverage, that is, to make certain that appropriate, state-of-the-art, encryption-enabled security functions have been applied to all instances of a given set of data. The questions are applied per application because applications, based on input from humans and/or other applications, are the mechanisms that actually create, alter, send, receive, store, move, copy, and manage access to data. Applications and encryption solutions operate within specific subsets of all possible combinations of operating system, device type, application architecture, and data format.

Coverage cannot be assured without discovering and evaluating all the applications able to process a given set of data and then discovering all the instances where data is stored or transmitted. Appropriateness cannot be assured without identifying risks attributable to the application, the data, and the encryption solution and then mitigating those risks.

Answering the questions develops a list of all instances of data in storage and in transmission, all instances of access requests and how they are authenticated, how data integrity is preserved and verified, and instances where additional organizational or technical measures in addition to encryption may be needed to justify not reporting a breach. Documenting these questions, answers, and how encryption is implemented provides demonstrable GDPR accountability.

Application Inventory Questions

Applications to be evaluated may be sourced from the data inventory and flow mapping exercises conducted to comply with record-of-processing requirements. Gap analysis questions may surface applications that have not been included in the data inventory. If so, these applications should be included in the data inventory.

In the Device Type and Operating System sections, record the operating system/device type pairs by choosing one from the operating system list and one from the device type list until all applicable operating system/device type pairs for a given application are listed in the inventory. Client-server applications will always have at least two listings—the server will be one listing and each client another.

1. What is the name of the application?
2. What is the current version of the application?
3. Is the application a general purpose or a custom application?[52]
4. Architecture: What is the application architecture? (choose one)
    a. Web App[53]
    b. Client-Server[54] (server and client(s) are listed separately)
    c. Peer-to-Peer[55]
    d. Cloud-Native[56]
    e. Standalone[57]

5. Device Type: What type of device is the application installed on? (choose one)
    a. Server
    b. Desktop
    c. Mobile
    d. IoT
6. Operating System: What Operating System is installed on the device that is running the application? (choose one)
    a. Windows Desktop
    b. Windows Server
    c. Windows Phone
    d. Linux
    e. OS X
    f. iOS
    g. Android

Risk Questions

The purpose for answering these questions is to identify risks to be mitigated. Once identified, risks can be ranked and mitigations can be selected and documented. When answering the questions:

- Expect some "back and forth," that is, answering a question will prompt a return to a previous question in order to add or edit the answer.
- Since the list covers multiple types of applications, some questions may not be applicable and there may be some repetition and overlap between some questions and answers.
- The term "users" is inclusive of humans and systems, authorized or unauthorized, who are able to request and potentially be granted access.
- An "unintended location" is any device, in or out of network, on which data can be accidentally or maliciously stored in violation of the organization's intentions.

Data Stored by the Application

While answering these questions, maintain a list of the locations where data can be stored. Almost every application is capable of storing data in multiple locations.

1. Does this application utilize a database stored on the device? If yes, describe where on this device the application normally stores the database.

2. Do other applications access this database? If yes, list them in the application inventory.
3. Is adulteration of the data stored in the database prevented and its integrity verified? If yes, describe how.
4. Are users requesting access to the application that manages the database authenticated? If yes, describe the authentication method(s).
5. Are users requesting direct access to this application’s database authenticated? If yes, describe the authentication method(s).
6. Are users requesting access to data storage on this device authenticated? If yes, describe the authentication method(s).
7. Does this application provide its logged-in users the ability to choose where the database is stored or to make a copy of the database? If yes, are there measures that prevent logged-in application users from moving or saving a copy of the database to an unintended location? If yes, describe them.
8. Are there measures that prevent users with access to data storage on this device from copying and saving the application and/or the database to an unintended location? If yes, describe them.
9. Assume that a malicious user has compromised credentials and exfiltrated data stored on a device under his control. Are there measures that prevent the malicious user from accessing the data? If yes, describe them.
10. Are logged-in users of this application able to query the database and store the results as unstructured or semi-structured data? If yes, include query results stored as files in questions pertaining to unstructured and semi-structured stored data.
11. Does the application store unstructured or semi-structured data on this device?
12. Where on the device does the application normally store unstructured and/or semi-structured data? List all locations, including those discovered answering other questions.
13. Does this application utilize a web-browser-based interface? If yes, can personally identifiable information be stored as cleartext in the browser cache or does the application enable storage of extracted data from a web server to be stored on the local machine? If yes, describe what data is stored and where it is stored.
14. Does this application utilize a server-side web cache? If yes, describe what data is stored and where it is stored.
15. Does this application utilize a content-distribution network? If yes, describe what data is stored and where it is stored.

16. In the course of processing, does this application or the operating system create and store temporary files containing personally identifiable information? If yes, describe where the temporary files are stored.
17. Are users requesting access to stored unstructured or semi-structured data produced by this application authenticated? If yes, describe the authentication method(s).
18. Does this application require its logged-in users to evaluate data to determine if it should be encrypted, and then take action to encrypt it? If yes, are there organizational or technical measures that assure that data is evaluated properly and that data that should be encrypted actually is encrypted, and that the encryption takes place in a timely manner? If yes, describe them.
19. Is any part of the unstructured and/or semi-structured data stored by the application on this device initially stored as cleartext then encrypted at a later time? If yes, describe the process that triggers encryption and quantify the time interval between initial storage and encryption.
20. Is any part of the unstructured and/or semi-structured data stored by the application on this device normally stored as cleartext and another part stored encrypted? If yes, describe where each part is stored and the data stored in each part.
21. Is adulteration of the information stored by this application as unstructured and/or semi-structured data prevented and its integrity verified? If yes, describe how.
22. Does this application provide its logged-in users the ability to choose where unstructured and/or semi-structured data can be saved or to make copies of the data? If yes, are their measures that prevent logged-in application users from moving, saving, or transmitting copies of unstructured and/or semi-structured data to an unintended location? If yes, describe them.
23. Do other applications consume this application’s stored unstructured and/or semi-structured data? If yes, list them in the application inventory. Include other processing applications, applications that analyze stored data for purposes of classification or security, or that consume the data for any other purpose.
24. Does this application provide its logged-in users the ability to decrypt data and save it or transmit it as cleartext? If yes, are there measures that prevent them from saving or transmitting decrypted data as cleartext? If yes, describe them.
25. What forms of removable storage are available to this application on this device? (choose all that apply)
    a. USB flash drive
    b. Portable hard drive
    c. Optical media

d. Memory card

e. Other

26. For each type of removable storage selected, are there measures that 1) prevent the saving of cleartext data to removable storage, or 2) prevent encrypted data saved to removable media from being decrypted by an unauthorized user on an unintended device? If yes, describe them.
27. Does the data stored on this device by this application contain personally identifiable metadata? If yes, are there measures to mitigate that risk? If yes, describe them.

Network Transmission Encryption Questions

28. Does this application provide its logged-in users the ability to transmit stored data? If yes, describe how up-to-date and properly-implemented network transmission encryption is assured from this device to the destination.
29. Are there measures to protect the data if network transmission encryption is absent, out-of-date, improperly implemented, allows use of deprecated encryption algorithms, or is otherwise compromised between this device and the destination? If yes, please describe them.
30. When this application transmits data, is metadata that includes personally identifiable information transmitted? If yes, list the information contained in the metadata.

Data Manageability Questions

31. Can encrypted unstructured and/or semi-structured data stored by this application be searched?
32. Can all the unstructured and/or semi-structured data stored by this application on behalf of particular data subjects be rectified?
33. Can all the processing that uses unstructured and/or semi-structured data stored by this application on behalf of a particular data subject be suspended and resumed?
34. Can metadata be associated with, and/or appended to, encrypted unstructured and/or semi-structured data stored by this application?
35. Can encrypted unstructured and/or semi-structured data stored by this application on behalf of a particular data subject be deleted upon request, or when other specified conditions are met?
36. Are records kept when the encrypted unstructured and/or semi-structured data stored by this application on behalf of a particular data subject are rectified, erased, suspended from processing, resumed for processing, or transmitted to third parties?

Encryption Solution Quality Questions

For each current or proposed encryption solution, answer the following questions.

37. Where are encryption keys generated?
38. Does fetching keys require a connection?
39. Are keys fetched through a connection encrypted prior to transmission?
40. Where are encryption keys stored?
41. How are encryption keys secured?
42. How is access to encryption keys controlled?
43. How are encryption keys backed up and recovered?
44. How are encryption keys rotated?[58]
45. How are encryption keys updated?[59]
46. What are the type(s) and strength(s) of the encryption algorithm(s) utilized?
47. How is this encryption solution tested, and what is the interval of testing?
48. Is this encryption solution appropriate, state of the art, and does it protect data by design and default? If yes, explain why.

[1] Article 4(5), Article 6 (4)(e), Article 25(1), Article 32(1)(a), Recital 28, Recital 29, Recital 75

[2] Article 6(4)(e), Article 32(1)(a), Article 34(3)(a), Recital 83

[3] Pseudonymization may be accomplished by means other than encryption such as tokenization. Encryption is a type of pseudonymization. WP216 considers encryption as a form of pseudonymization, as does this analysis and much other technical literature pertaining to pseudonymization. A plain reading of Article 4(5) suggest that encryption is an acceptable type of pseudonymization.

[4] Other data security measusres are permitted and encouraged. Recital 28

[5] Common breach mechanism for copying a stream of data moving through a wired or wireless connection include compromised credentials, compromised network traffic encryption, and inadvertent use of unencrypted connections and unencrypted metadata.

[6] Article 33(1), Working Party 29, Guidelines on Personal data breach notification under Regulation 2016/679, 1. Personal data breach notification under the GDPR, A. Basic security considerations, Page 5

[7] Article 4

[8] Article 30, Recital 84

[9] Article 9, Article 35, Recital 79, Recital 83, Recital 91

[10] Article 30(5), Article 32(1)(a), Article 35(1) and (3), Recital 76, Recital 83, Recital 84

[11] Article 33, 34

[12] https://en.wikipedia.org/wiki/Cascading_failure

[13] Working Party 29 Guidelines on the application and setting of administrative fines III. Assessment criteria in article 83 (2), (g) the categories of the personal data affected by the infringement; Page 14-15

[14] Article 28(1), Recital 81

[15] Article 35(7)(d), Recital 78, Recital 84, Guidelines on Data Protection Impact Assessment (DPIA) and determining whether processing is "likely to result in a high risk", Page 8 §1

[16] Article 33(1), Working Party 29, Guidelines on Personal data breach notification under Regulation 2016/679, D. Conditions where notification is not required, Page 16 §1

[17] Article 34(3)(a)

[18] Working Party 29, Guidelines on Personal data breach notification under Regulation 2016/679, IV. Assessing risk and high risk, A. Risk as a trigger for notification, Page 19

[19] Working Party 29, Guidelines on Personal data breach notification under Regulation 2016/679, Accountability and record keeping, A. Documenting breaches, Page 23

[20] Article 32(1)(d), Recital 78

[21] Article 5(1)(e) and (f)Article 32(1), Article 28(1), (3)(e), (4) Recital 78, Recital 156

[22] Article 32(1)(d). For general purpose software, seek GDPR-specific guidance from the manufacturer. For custom software, consider periodic penetration testing.

[23] Article 35(7)(c) and (d), Recital 76

[24] Working Party 29, Guidelines on Personal data breach notification under Regulation 2016/679, Factors to consider when assessing risk, Page 20 §2, §3

[25] Working Party 29, Guidelines on Personal data breach notification under Regulation 2016/679, Introduction, Page 5 §1

[26] Working Party 29, Guidelines on Personal data breach notification under Regulation 2016/679, A. Risk as a trigger for notification, Page 19 §2

[27] https://www.csoonline.com/article/3159738/data-breach/how-data-breaches-are-discovered.html

[28] https://www.globalsign.com/en/blog/glossary-of-cryptographic-algorithms/

[29] https://en.wikipedia.org/wiki/Payload_(computing)

[30] Metadata is data that is associated with a particular instance of data in order to make it easier to find and work with the underlying data. For example, filename, file type, author, date created, date modified and file size are basic document metadata. Metadata can be contained in the particular instance of data or held separately (usually in a database) but linked to the particular instance of data as needed.

[31] Examples include XML and JSON, both commonly used as the primary store of human-readable information generated by web applications.

[32] A GDPR-specific example: A Word document is analyzed by a classification system and found to contain sensitive information of a certain type that should be retained no longer than 2 years. Metadata tags are appended to the documents so its storage location, distribution restrictions, and deletion can be managed by automated means. It is now semi-structured data.

[33] https://en.wikipedia.org/wiki/Identity_management

[34] https://en.wikipedia.org/wiki/Hardware_security_module

[35] Modern software is highly modular and is becoming more so by the day simply because it is less expensive to buy parts than to make them. The majority of code in modern applications is comprised of reusable third-party components: libraries, application programming interfaces (APIs), and software development kits (SDKs). Components are assembled with "glue code" that enables them to interoperate. As modularity increases, the level of effort to incorporate component-based, encryption-enabled security functions into applications decreases.

[36] https://en.wikipedia.org/wiki/Separation_of_duties

[37] https://en.wikipedia.org/wiki/Homomorphic_encryption

[38] https://en.wikipedia.org/wiki/Multi-factor_authentication

[39] Common examples include shadow IT applications, and in spreadsheets, email, and collaboration tools.

[40] https://en.wikipedia.org/wiki/NoSQL

[41] https://en.wikipedia.org/wiki/Multi-factor_authentication

[42] https://en.wikipedia.org/wiki/Shadow_IT

[43] Article 30, Recital 82

[44] https://en.wikipedia.org/wiki/Web_cache

[45] https://en.wikipedia.org/wiki/Content_delivery_network

[46] Article 5(1)(d), Recital 39

[47] Article 16

[48] Article 17

[49] Article 21

[50] Article 5(1)(e)

[51] Article 13(1)(e), Article 15(1)(c), Article 30(1)(d), Recital 61, Recital 63

[52] General purpose software includes Microsoft Office and Adobe Acrobat, and any other software that is obtainable by anyone. Typically, custom software is built to specification for its owner only, or for a limited set of licensed users.

[53] Server-side processing with a Browser UI.

[54] Server-side processing and client-side processing (thick client).

[55] Peer-to-peer (P2P) networks are decentralized. Each node on the network has the same capabilities. Common in IoT applications.

[56] Software composed of loosely-coupled, scalable, cloud services.

[57] An application that only processes information on the device on which it is installed.

[58] The changing of keys on a schedule or in response to a potential leak or compromise.

[59] Encryption algorithms lose strength or become more vulnerable over time. The key strength and/or type of encryption algorithm may need to be updated.