Recommended cryptographic measures

Securing personal data

September 20th, 2013

www.enisa.europa.eu

Contributors to this report

This work was commissioned by ENISA under contract ENISA P/18/12/TCD Lot 2 to the consortium formed for this work by KU Leuven (BE) and University of Bristol (UK).

- Contributors: Vincent Rijmen, Daniel De Cock (KU Leuven), Nigel P. Smart (University of Bristol)
- ENISA project manager: Rodica Tirtea

We would like to extend our gratitude to the reviewers for their comments, suggestions and feedback.

About ENISA

The European Union Agency for Network and Information Security (ENISA) is a centre of network and information security expertise for the EU, its member states, the private sector and Europe's citizens. ENISA works with these groups to develop advice and recommendations on good practice in information security. It assists EU member states in implementing relevant EU legislation and works to improve the resilience of Europe's critical information infrastructure and networks. ENISA seeks to enhance existing expertise in EU member states by supporting the development of cross-border communities committed to improving network and information security throughout the EU. More information about ENISA and its work can be found at www.enisa.europa.eu.

Contact details

For contacting ENISA or for general enquiries on this topic, please use the following details:

- E-mail: sta@enisa.europa.eu
- Internet: http://www.enisa.europa.eu

For questions related to this project, please use the following details:

- E-mail: sta@enisa.europa.eu

Table of content

1. Executive summary 5
2. Introduction 7
2.1. Policy context 7
2.2. Information security as a puzzle 10
2.3. Privacy and cryptography 11
2.4. Scope, structure, terminology 12
3. Identifying security requirements 13
3.1. Matching personal data lifecycle stages to security requirements 13
3.2. Security Requirements 14
4. Basic cryptographic techniques 17
4.1. Encryption 17
4.2. Data authentication 18
4.3. Hashing 18
4.4. Digital signing 19
5. Cryptographic primitives 20
5.1. Symmetric primitives 20
5.2. Asymmetric primitives 21
5.3. Strength of cryptographic primitives 22
5.4. Key management 23
6. Case study. Protective measures 26
6.1. Classification 26
6.2. Countermeasures 28
7. Concluding remarks 30
8. Appendix: data minimization 31
8.1. Local extraction 31
8.2. Attribute-based credentials 31
8.3. Private information retrieval 31
9. Bibliography 32

1. Executive summary

This document addresses the protection measures applied to safeguard sensitive and/or personal data, which has been acquired legitimately by a data controller. In this respect it discusses how information technology users, who have a basic knowledge of information security, can employ cryptographic techniques to protect personal data. Finally, it addresses the need for a minimum level of requirements for cryptography across European Union (EU) Member States (MSs) in their effort to protect personal and/or sensitive data.

In the following sections a mapping of security requirements for personal data to basic cryptographic techniques and cryptographic primitives will be proposed. It is of worth to note that information security measures and mechanisms could be deployed for the protection of personal data. Of course information security does not cover all the issues regarding personal data protection and privacy. In this light, we briefly present protection measures applied to safeguard sensitive or confidential data, which has been collected legitimately. **It should be noted that this document is complemented with a set of technical recommendations for algorithms, key sizes, parameters and protocols, which is part of another study published by ENISA [1] addressed to decision makers and specialists designing and implementing cryptographic solutions.**

In the introduction we present the policy context where the need for clearly defined technical measures (i.e. to make data unintelligible and to ensure only authorized access) adapted to the context and technology evolution is stated. In the same section, the context of information security as a puzzle is presented and then the document focuses exclusively on cryptographic measures. The clear link between privacy and cryptography is also presented, demonstrating how the second can play a role in personal data protection. As personal/sensitive data requires different protection measures in different stages of the lifecycle, we propose a short version of such a lifecycle description for the purpose of this paper.

Section 3 identifies security measures while Section 4 provides a short introduction to basic cryptographic techniques. Section 5 makes the link to cryptographic primitives, which are the building blocks in cryptography.

Section 6 proposes a case study where we provide protective measures for several types of attacks that can lead to data breaches. Such protective measures are considered in the context of the recently published Commission Regulation No 611/2013 [2][1] on the measures applicable to the notification of personal data breaches under Directive 2002/58/EC on privacy and electronic communications. The case study also provides a link to the complementing technical recommendations on cryptography [1].

Section 7 summarizes the conclusions and the recommendations of this work. It also provides suggestions for future work. Amongst the findings and recommendations are:

- The cryptographic measures are only one piece of a puzzle when we refer to privacy and data protection. Cryptographic measures provide a layer of protection and may reduce the impact of breaches.
- The relevant stakeholders (Data Protection Authorities, member states authorities, service providers) should recommend and implement security measures for protecting personal data and should rely on state-of-the-art solutions and configurations for this purpose. All

[1] *EC regulation (EU) No 611/2013 of 24 June 2013 on the measures applicable to the notification of personal data breaches under Directive 2002/58/EC on privacy and electronic communications [2]*

these stakeholders could use as reference for cryptographic measures the recommendations proposed in [1]. It is within ENISA's intentions to update this report at least on a yearly basis. The relevant stakeholders should refer to the most recent version of the recommendations.

- Specialized personnel are needed for the correct implementation of cryptographic protective measures. Such measures need also to be updated.
- ENISA should continue this work by addressing more specialized privacy protective measures and technologies relying on cryptography; some of these techniques were proposed by the research community, however they need to be deployed in the systems. ENISA should survey the existing technologies and promote their deployment.

2. Introduction

In our modern society, businesses and other organizations collect data and extract information for various purposes. For example, a shop owner needs to have a view on which items are sold in order to refill the stock. A more advanced use of the data may be to link external factors to the rate at which certain items are being sold, in order to predict further sales and optimize stock management.

Data from different sources may be aggregated to reveal more information. By analysing which items are bought by the same customer, it becomes possible to construct customer profiles, which can be used to send the customer targeted advertisements or to improve customer service in other ways. Users may object to excessive profiling, or consider the data collection or the data aggregation by themselves to be a violation of their privacy, but the latter concern is not the topic of this document. This document deals with protection measures applied to safeguard sensitive or confidential data, which has been collected legitimately.

2.1. Policy context

2.1.1. Data protection directive

The Data Protection Directive[2] 95/46/EC [3] on the "protection of individuals with regard to the processing of personal data and on the free movement of such data" covers in Article 17 the aspects of "security of processing". The article mentions that "1. Member States shall provide that the *controller must implement appropriate technical and organizational measures to protect personal data against accidental or unlawful destruction or accidental loss, alteration, unauthorized disclosure or access, in particular where the processing involves the transmission of data over a network, and against all other unlawful forms of processing*. […][S]uch measures shall *ensure a level of security appropriate* to the risks represented by the processing and the nature of the data to be protected. […] 2. The Member States shall provide that the controller must, where processing is carried out on his behalf, choose a processor providing sufficient guarantees in respect of *the technical security measures and organizational measures* governing the processing to be carried out, and must ensure compliance with those measures."

In this document we explain some of the technical measures that could be used for data protection while in the technical recommendations report we propose and recommend an 'appropriate' level of security criteria for current state-of-the-art in technologies and cryptology.

Article 6, covering principles relating to data quality, specifies also that personal data must be kept "in *a form which permits identification* of data subjects for no longer than is necessary for the purposes for which the data were collected or for which they are further processed". Data minimization techniques and anonymisation techniques could play a role here and they can be implemented using cryptography; we are only providing some links for this area, as further work is needed and this is beyond the scope of this document.

2.1.2. The proposed EC regulation on data protection

The data protection framework is currently undergoing a reform. In January 2012 the EC proposed a Regulation [4] to replace the existing Data Protection Directive.

[2] *Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, Official Journal L 281 , 23/11/1995 P. 0031 – 0050, [3]*

The proposed regulation builds on the existing data protection directive 95/46/EC. In Section 2 - Data security, Article 30, obliges the controller and the processor to implement appropriate measures for the security of processing, extending that obligation to processors, irrespective of the contract with the controller. The Commission will probably prepare delegated acts to "determinations of what constitutes the state of the art, for specific sectors and in specific data processing situations, in particular taking account of developments in technology and solutions for privacy by design and data protection by default" and also set implementation acts to "specifying the requirements […] to various situations, in particular to: (a) *prevent any unauthorised access to personal data*; (b) *prevent any unauthorised disclosure, reading, copying, modification, erasure or removal of personal data*; (c) *ensure the verification* of the lawfulness of processing operations."

Furthermore, Articles 31 and 32 introduce an obligation to notify personal data breaches, building on the personal data breach notification in Article 4(3) of the e-privacy Directive 2002/58/EC. Some exceptions are mentioned regarding notification to the data subject if there are "appropriate technical protective measures" in place to "render the *data unintelligible* to any person who is not authorized to access it".

The principles of data protection by design and by default is enforced in Article 23 while Article 79, obliges each supervisory authority to sanction the administrative offences listed in the catalogues set out in this provision, imposing fines up to maximum amounts (up to 1 000 000 EUR or, in case of an enterprise up to 2% of its annual worldwide turnover), with due regard to circumstances of each individual case. Even if this Regulation is still at proposal stage, it sets higher expectations for personal data protection.

The regulation is currently being discussed in the European Parliament and some of the above-mentioned provisions could be amended. The proposed regulation could be applicable two years after entering into force.

2.1.3. Directive 2002/58/EC on privacy and electronic communications

The text of the Directive specifies the need for minimisation of processing of personal data and recommends development and deployment of techniques for anonymisation " (9) The Member States, providers and users concerned, together with the competent Community bodies, should *cooperate in introducing and developing the relevant technologies* where this is necessary *to apply the guarantees* provided for by this Directive *and taking particular account of the objectives of minimising the processing of personal data and of using anonymous or pseudonymous data* where possible."

Article 4 of Directive 2002/58/EC[3] on privacy and electronic communications, specifies the security requirements "1. The provider of a publicly available electronic communications service *must take appropriate technical and organisational measures to safeguard security of its services*, if necessary in conjunction with the provider of the public communications network *with respect to network security*. Having regard to the state of the art and the cost of their implementation, these measures shall ensure a level of security appropriate to the risk presented."

Directive 2002/58/EC is amended by directive 2009/136/EC[4]. The amendments cover the above-mentioned article 4, specifying at least the following measures: "ensure that personal data can

[3] *Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications) available at: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32002L0058:en:HTML ; text in italics by ENISA.*

[4] *Directive 2009/136/EC of the European Parliament and of the Council of 25 November 2009 amending Directive 2002/22/EC on universal service and users' rights relating to electronic communications networks and services, Directive*

be *accessed only by authorised personnel* for legally authorised purposes", "*protect* personal data stored or transmitted against accidental or unlawful destruction, accidental loss or alteration, and unauthorised or unlawful storage, processing, access or disclosure", and, "ensure the *implementation of a security policy* with respect to the processing of personal data".

Also, new text is added in the above-mentioned article 4, covering exceptions regarding notification of the subscriber "[n]otification of a personal data breach to a subscriber or individual concerned shall not be required if the provider has demonstrated to the satisfaction of the competent authority that it *has implemented appropriate technological protection measures, and that those measures were applied to the data concerned* by the security breach. *Such technological protection measures shall render the data unintelligible to any person who is not authorised to access it*." The article also identifies ENISA, next to other bodies, as advisor for the technical implementing/protective measures.

With the current work, we provide technical recommendations for protecting personal data and present ways how cryptography could help in reaching these objectives.

2.1.4. The Commission Regulation on the measures applicable to the notification of personal data breaches under Directive 2002/58/EC on privacy and electronic communications

Article 4 of the Commission Regulation [2] [5], provides recommendations for "Technological protection measures". In particular, notification of a personal data breach to a subscriber or individual concerned is exempted if "the provider has demonstrated to the satisfaction of the competent national authority that it *has implemented appropriate technological protection measures, and that those measures were applied to the data concerned* by the security breach."

Such technological protection measures "*shall render the data unintelligible* to any person who is not authorised to access it".

"Data shall be considered unintelligible if:

1. it has been *securely encrypted with a standardised algorithm*, *the key used to decrypt the data has not been compromised in any security breach, and the key used to decrypt the data has been generated so that it cannot be ascertained by available technological means by any person who is not authorised to access the key*; or

2. it has been replaced by its *hashed* value calculated with a standardised cryptographic keyed hash function, the key used to hash the data has not been compromised in any security breach, and the key used to hash the data has been generated in a way that it cannot be ascertained by available technological means by any person who is not authorised to access the key.

The Commission may, after having consulted the competent national authorities via the Article 29 Working Party, the European Network and Information Security Agency and the European Data Protection Supervisor, publish an indicative list of appropriate technological protection measures, referred to in paragraph 1, according to current practices."

2002/58/EC concerning the processing of personal data and the protection of privacy in the electronic communications sector and Regulation (EC) No 2006/2004 on cooperation between national authorities responsible for the enforcement of consumer protection laws, available at: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2009:337:0011:0036:en:PDF ; text in italics by ENISA.

[5] *EC regulation (EU) No 611/2013 of 24 June 2013 on the measures applicable to the notification of personal data breaches under Directive 2002/58/EC on privacy and electronic communications, available at: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2013:173:0002:0008:en:PDF ; Italics in the cited text by ENISA.*

In recital (16) of the EU Commission's regulation, the EC recognizes the need to update such specific technological protection measures over time as technology advances.

With this document and with the accompanying recommendations that ENISA is publishing, we provide input to the Commission on the cryptographic measures above-mentioned in article 4. A case study is presented in Section 6 addressing protective measures. Furthermore ENISA intends to continue this work by updating the technical recommendations on a yearly basis.

2.2. Information security as a puzzle

In this section we enumerate some of the measures required for securing information (including personal data). Securing information is a difficult problem, which requires measures at very different levels and techniques from many different domains [4], [5].

2.2.1. Risk assessment

The first step in the design of a system to provide information security is the risk assessment. Roughly, we can distinguish the following steps [5]:

1. Identify and value assets: computer equipment and software are assets, as is the information stored on the computer equipment. Also reputation is an asset.
2. Identify threats: list the possible negative effects on the assets. There are different ways to categorize threats.
3. Identify vulnerabilities: examine the system for weaknesses that could be used to damage assets.
4. Evaluation of the risk: risk is determined as a combination of the value of assets, the number and severity of the threats and the number and severity of vulnerabilities.

After the risk assessment has been completed, a security policy can be designed, and subsequently the technical measures to implement the security policy can be decided. Regular auditing is necessary to verify whether the lists of assets, threats and vulnerabilities are still correct and complete.

2.2.2. Complementary security measures

The following list is not exhaustive, but includes the main groups of security measures that would be required to complement the logical security measures outlined in this paper.

1. Physical security: walls, doors, windows. Note that blocking physical access to computers is less effective than in the past. Because of the spreading of wireless communication channels and the multitude of mobile devices, physical access controls can be more easily circumvented. Physical security also includes tamper resistance. Tamper-resistant hardware is designed such that the information they store is not accessible through external means. Only software routines installed by the system administrator can access the information, and only in well-specified ways.
2. Organizational security: organizational security includes separation of responsibilities and duties by definition of roles, splitting of keys, etc.
3. Personnel security: personnel security includes vetting of candidates, education and awareness raising activities, procedures for handover of keys and access tokens in case of contract termination, etc.

2.2.3. Logical security

Logical security includes all types of security measures that deal with the "virtual world." Good logical security measures reduce information security problems to security problems in the physical world. Logical security by itself can never suffice to solve information security problems, because it always assumes the existence of secure locations, servers . Again, the following list is not exhaustive.

1. Network security: the vast majority of data flows over the Internet, whose basic structure was not designed for the way in which it is used. This leads to many vulnerabilities, which are e.g. countered by firewalls and intrusion detection systems.
2. Operating system security: operating systems are used to implement identification and authentication mechanisms as well as security policies on computers. They tend to be complex and therefore prone to bugs. Operating system security includes timely updates, correct configuration and possibly trimming down of the functionalities.
3. Software security: Malware is software that has been designed with a malicious purpose. Also non-malicious software can lead to vulnerabilities if it contains features or bugs that can be exploited by attackers, e.g. buffer overflows or bad behaviour when confronted with malformed inputs.
4. Logging: since security problems inevitably show up at some point, it is important to keep track of actions that have taken place. Secure logging of all security-relevant events is essential for auditing.
5. Cryptology: cryptology employs mathematical functions to transform sensitive data into sequences that resemble random noise and that can be decoded only when the correct key is applied. Cryptology can also be used to construct strong check sums and other services, which are discussed in more detail further in this document. An important advantage of cryptology is that because it is rooted in mathematics, one can construct mathematical proofs of security (in certain models), which can give a high level of trust for a relatively low price.

2.3. Privacy and cryptography

Cryptography is an essential technical means to provide privacy and privacy-related services. Gürses defines three privacy paradigms: confidentiality, control and practice [6].

1. In the first paradigm, privacy is ensured by keeping personal data confidential, i.e. protecting it so that unauthorised people can't access or modify it. This definition of privacy is clearly and strongly linked to classical cryptographic schemes: electronic data secure can be kept secure by using strong cryptography (and strong keys).
2. The second paradigm, which is more common in legal texts, adds the ability to control what happens with personal data. This is also called the right to informational self-determination. In this definition, several advanced cryptographic schemes can play an important role, e.g. techniques to reduce the amount of personal data that is released to the strictly required minimum. We discuss briefly some of these cryptographic techniques in the appendix.
3. The third paradigm defines privacy as transparency on the ways in which information is collected, aggregated and used. Cryptography plays here a much less visible role, but it is still present underneath as the method to enforce the policies formulated with respect to the treatment of personal data.

2.4. Scope, structure, terminology

This document focuses on elementary cryptographic mechanisms: encryption, authentication, hashing and electronic signatures. These cryptographic mechanisms are useful in protecting personal data.

Protective measures are identified in the document, as case studies, for several of types of attacks that can lead to data breaches. Such protective measures are considered in the context of the current legal framework covering the measures applicable to the notification of personal data breaches.

Throughout this report we introduce and we refer to some important cryptographic primitives. For background information on cryptographic primitives and an introduction to cryptography in general, we refer to [7] [8]. More detailed information on how these primitives should be used and which primitives should be chosen in order to achieve a desired security level is given in [1].

When using any cryptographic mechanisms, in order to achieve a desired level of security, we recommend referring to the detailed information provided in [1].

Since the survey[6] conducted by ENISA shows that the deployment of even basic cryptographic mechanisms is still incomplete, this document starts with an introduction to the basic mechanisms and treats the more advanced concepts in a separate section.

The Appendix mentions briefly some more advanced cryptographic mechanisms that can be useful in the context of privacy protection. Although some of the mechanisms described in the appendix are known in the research community since more than a decade, their applications outside a research environment are still rare.

For the future, we recommend that ENISA considers a survey of deployment and best practices of the more advanced cryptographic methods that can be used to improve personal data protection.

[6] *ENISA study on the Use of Cryptographic Techniques in Europe, 2011, available at:* *http://www.enisa.europa.eu/activities/identity-and-trust/library/the-use-of-cryptographic-techniques-in-europe*

3. Identifying security requirements

3.1. Matching personal data life cycle stages to security requirements

For the purpose of this paper we describe in this section the life cycle of personal data[7]. This description may not cover all relevant stages and possible interactions, however, this description allows the identification and the presentation of adequate information security means to be deployed for the purpose of protecting personal /sensitive data.

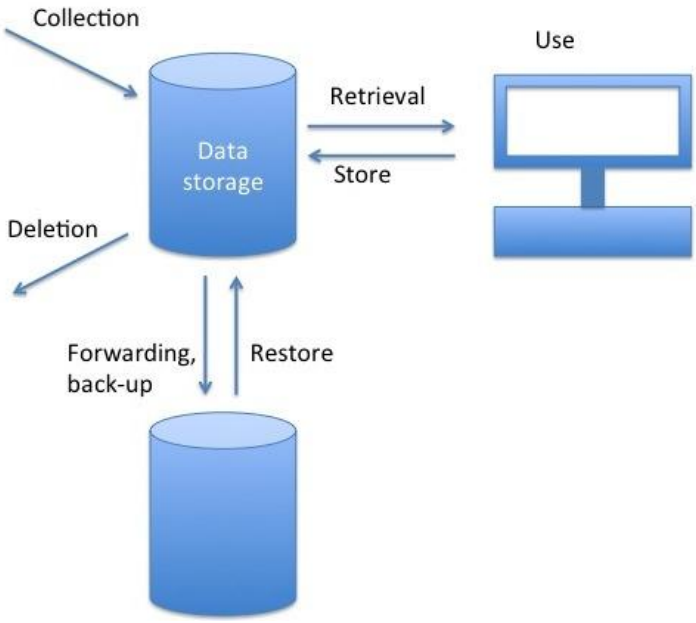

Figure 1. Life cycle of personal data

We distinguish the following phases in the life cycle of personal data.

1. Collection and storage: Personal data items are collected and combined; a data record is created and stored. Even if the original items separately are not sensitive, together they may become a sensitive data record. In particular, in the context of ubiquitous computing, many different sensors will collect personal data that is not sensitive by itself. However, when the data of the sensors is pooled, it becomes sensitive. During this phase, the data needs to be protected against unauthorized reading access. This is called ensuring the confidentiality of data. Furthermore, the data needs to be protected against unauthorized alterations. This is called ensuring the integrity

[7] *Personal data is defined in data protection legislation (i.e. Directive 95/46/EC [3]) as " […] any information relating to an identified or identifiable natural person ('data subject'); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity" (Article 2). Sensitive data is considered "personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, trade-union membership, and the processing of data concerning health or sex life"(Article 8).*

or authenticity of the data[8]. In order to determine whether an entity requesting access is authorized to perform a certain operation, we need to have entity authentication.

2. Data can be forwarded for purposes of back-up or archiving, or in order to share the data with other organisations. During transmission, the data needs to be protected against unauthorised reading access and unauthorised modification. This can be achieved with the same security services as during the collection and storage phases.
3. When personal data is needed (and the access request is legitimate), it has to be retrieved from the database: the data has to be available. Furthermore, the authenticity of the data needs to be verified. Upon detection of an unauthorised alteration, the data needs to be discarded, restored or corrected. Note that there are cryptographic mechanisms to verify the authenticity of data, but there is no cryptographic mechanism to achieve restoration or correction. Hence, other mechanisms have to be employed in order to ensure that data can be restored or corrected. If the access request is not legitimate, then the data has to be blocked. When implemented,non-repudiation services ensure that users can be made accountable for the actions they performed. Accountability can also be achieved by authentication and logging mechanisms.
4. While the retrieved data is processed, it needs to be protected against unauthorized reading access and unauthorized modification. This can be achieved with the same security services as during the collection and storage phases. Providing these security services in practice may be more difficult during the use phase, because typically the data will reside in an environment that is more easily accessible by attackers.
5. When data is no longer needed (or if personal data has been collected without legitimisation) it has to be removed completely. This service is called secure deletion. In practice, secure deletion is a difficult process, because parts and traces of data may have been copied to various places in the system. There is no cryptographic mechanism to achieve secure deletion (but there are secure deletion techniques that work on standard media that do not use cryptographic techniques), but cryptographic mechanisms may be employed to reduce the number of places in the system where the data has been stored in a readable form. On the other hand, cryptography uses secret keys, which are confidential data themselves, and which need to be deleted securely when they are no longer used.

From this example, we can distil several security requirements, which we explain next.

3.2. Security Requirements

Data protection is often described as a process based on three pillars: Confidentiality, Integrity and Availability (CIA). Although these three requirements are definitely very important, the problem of information security has more dimensions than those three.

3.2.1. Confidentiality

Confidentiality is the property that data is unknown to unauthorised persons. Confidentiality can be achieved by the cryptographic mechanism of data encryption.

The data that is kept confidential, may also be the name of a person. In that case, we refer to the property as anonymity. Note that simply encrypting the name does not anonymise the data. Data could still be re-identified.

[8] *In this paper we make the simplifying assumption that integrity mechanisms implicitly provide data origin authentication, and vice versa.*

3.2.2. Integrity

Data integrity, also referred to in this paper as data authenticity, is the property that data has not been modified in an improper manner. If data is written on a medium that allows erasure and rewriting, then in principle an attacker can always modify the data on this medium. Cryptographic mechanisms to protect the authenticity of data are called (data) authentication mechanisms. They are usually restricted to the *detection* of modifications. (Cryptographic mechanisms for the *correction* of modifications will be discussed at the end of this document.)

Instead of data, we may also wish to authenticate the name or identity of a person, usually as the first step of an access control mechanism. This is called entity authentication.

3.2.3. Availability

Data has to be easily available whenever it's needed. If the data is present in authenticated and encrypted form in a database, but for some reason it can't be exported from the database and used, then the property of availability is violated. There is no cryptographic technique that can be applied to the data itself in order to make it more available. However, cryptography can be used to protect the methods to access the data (access control).

On the other hand, if data doesn't need to be available, it can be made physically not accessible and it will be kept secure by physical security means also without the application of cryptography. Availability can be traded-off for security. Some cryptographic methods might lead to a reduction in availability, e.g. if attackers can abuse them to overload a server. Hence, availability is a concern in the design and the deployment of cryptographic systems.

A good design moves the trade-off curve and achieves an appropriate balance between availability on the one hand and confidentiality and authenticity on the other. Availability can also be linked to performance, which is related to the speed of the cryptographic algorithms used to ensure security (a low speed of the cryptographic algorithms implies then a low availability).

The next two properties are more advanced in nature.

3.2.4. Forward confidentiality

This property is a stronger form of confidentiality. It is applicable to systems that have a key architecture where keys with a long lifetime (long-term keys) are used to construct keys with a short lifetime (short-term keys), and all data is encrypted with the short-term keys. The concept forward security is defined as follows [9, 10]:

> A protocol is said to have perfect forward secrecy if compromise of long-term keys does not compromise past session keys.

Forward confidentiality is important in the assumption that long-term keys may eventually be broken or leaked. In a system with forward confidentiality, data that was secured with short-term keys that were generated before the long-term key was broken, remains secure.

According to Murphy's law, accidents are bound to happen at some time. In the absence of forward secrecy, when a cryptographic key gets lost, stolen or broken, all the data or keys that were encrypted with this key, are no longer secure. In contrast, in a system that has forward security, the loss of some keys will endanger only a fraction of the data.

3.2.5. Non-repudiation

Non-repudiation is the property that an entity cannot take a certain action and later deny having taken this action. The action can be for example buying something over the Internet, or agreeing to a certain contract. In the physical world, non-repudiation is often implemented by means of manual signatures. Also in the electronic world, non-repudiation is achieved by constructing

'receipts' that can be linked back to the person who took the action using cryptographic means and show that someone has taken certain actions. Where non-repudiation on the receiver side is required, there is also a protocol element involved.

4. Basic cryptographic techniques

We present here four basic cryptographic techniques useful while protecting personal data. More advanced techniques are presented in the appendix.

4.1. Encryption

Encryption is the process that makes data unintelligible by transforming it into ciphertext, thereby protecting it against unauthorized access (reading). Encryption processes use a cryptographic key. It is understood that outsiders may get access to the ciphertexts. A good encryption process ensures that the reverse conversion of ciphertexts to the original data is possible only for entities that possess the required cryptographic key (decryption key).

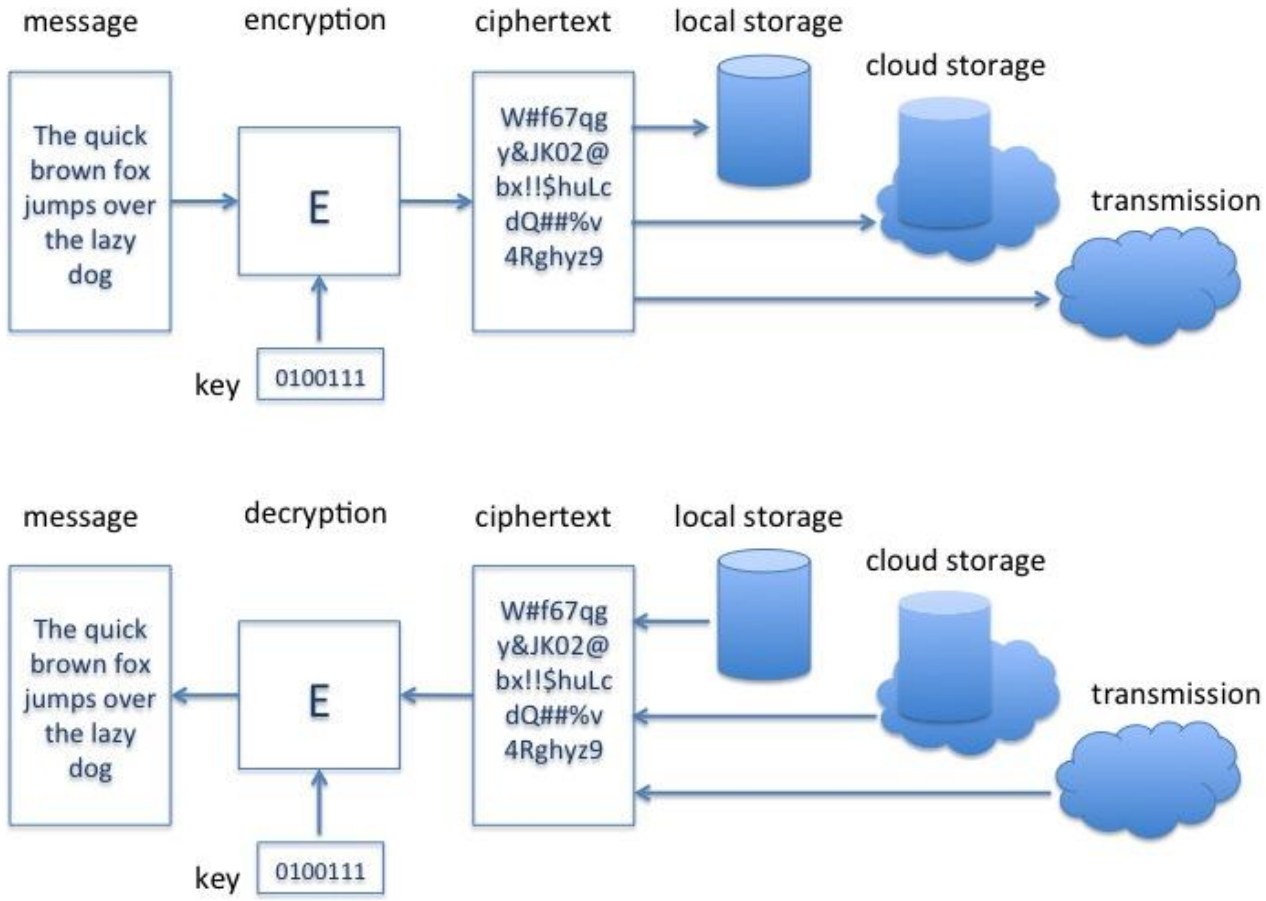

Figure 2. Encryption and decryption processes. Data->encryption->ciphertext, ciphertext->decryption->data

Note that while encryption and authentication are two different cryptographic operations, which can operate independently, many practical systems need both. Hence, modern encryption schemes usually provide the combination, i.e. authenticated encryption.

4.2. Data authentication

Data authentication (which implies integrity protection) is the process that makes undetected modification of data impossible or unfeasible by adding a tag (intuitively: a cryptographic check sum) or a digital signature to the data. Modern data authentication processes use a cryptographic key. It is understood that outsiders may get access to both the data and the tag. Any modification of the data will invalidate the tag. Computing the new tag is possible only for entities that possess the required cryptographic key (the authentication key). The validity of the authentication tag is checked by the verification algorithm, which uses a verification key.

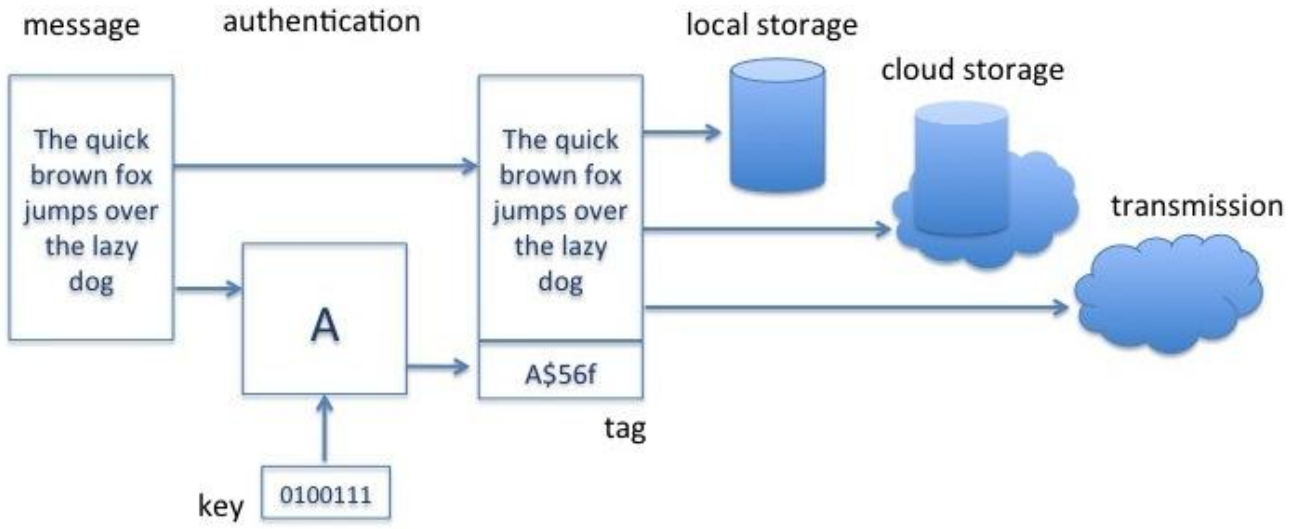

Figure 3. Authentication process. Data->authentication->data+tag

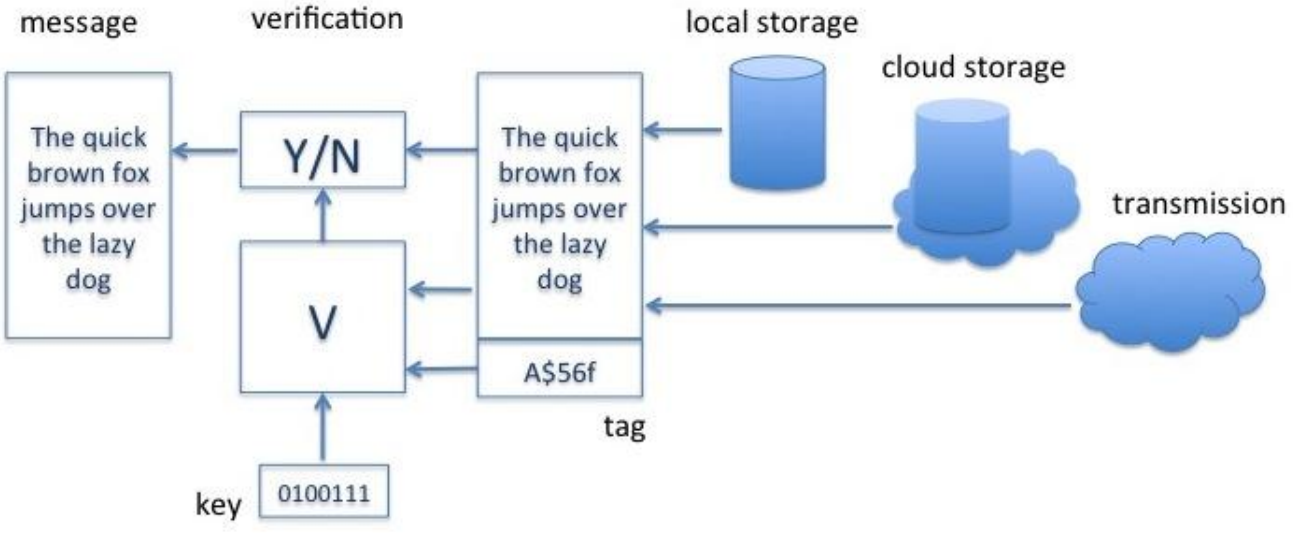

Figure 4. Verification process. Data+tag->verification->yes/no;

4.3. Hashing

Hashing is a process that computes a short and unique representation of a piece of data in an irreversible way. Since hashing is not reversible, it can't be used as encryption (because

decryption would be impossible). Hashing doesn't use a cryptographic key as input. Hence, it is *not* an authentication mechanism by itself. However, the HMAC constructions build a message authentication mechanism by applying hashing to a combination of a secret key and a message.

Hashing is used to produce "digital fingerprints" of messages. For example, in order to speed up digital signature operations, not the whole message is signed, but only its fingerprint. Similarly, in Operating System (OS)) security, often the authenticity of important files is protected by hashing the files and comparing the resulting fingerprints to some reference values. Note that in such a scheme it is important that the authenticity of the reference values themselves is protected.

Hashing is also used in order to produce pseudo-random noise.

4.4. Digital signing

Digital signing is a process that produces a short string that depends on the content of the message and on a secret known only to the signer. The validity of a signature can be verified without knowledge of the secret. Digital signing combines data authentication and entity authentication (identification).

The most popular way to perform digital signing is by using asymmetric cryptography, but there are also constructions that use only hash functions (authentication trees).

5. Cryptographic primitives

In this section we introduce some important cryptographic primitives. More detailed information on how these primitives should be used and which primitives should be chosen in order to achieve a desired security level is given in [1]. For background information on cryptographic primitives and an introduction to cryptography in general, we refer to [7] [8].

5.1. Symmetric primitives

Symmetric primitives are so named because (in principle) all parties use the same keys: encryption and decryption operations use the same keys, or keys that can easily be derived from one another. The same holds for data authentication and verification based on symmetric primitives.

5.1.1. Block cipher

A block cipher replaces blocks of text by other blocks. The blocks have a fixed length, which is called the block length of the cipher. The substitution depends on the key. For a good block cipher, without knowing the key it is not possible to see a relationship between the input and the output. In security proofs for cryptographic schemes and protocols that use block ciphers, it is often assumed that a block cipher applies a random substitution. This is called the *ideal-cipher model*.

If the key of a block cipher is kept constant, then two equal blocks of text at the input of the block cipher lead to two equal blocks at the output, hence patterns in the input leak through. Block ciphers are never used `as is', but always in *modes of operation*. There exist several standardized modes of operation that can be used for encryption. Also modes of operation for authentication have been standardized. The modern solution is to use a mode of operation that performs both encryption and authentication. This is called *authenticated encryption*. For more information on block ciphers, we refer to [11].

5.1.2. Stream cipher

A stream cipher can be described as a pseudo-random generator with cryptographic strength. The secret key determines the seed of the generator and/or some parameters that influence the state update of the generator. However, the output of a stream cipher satisfies more criteria than an ordinary (non-cryptographic) pseudo-random noise generator. On top of showing no statistical deviations from truly random bit sequences, the output should not allow to determine the secret key. If you don't know the key, the output should be unpredictable, even when many samples have been observed.

Stream ciphers encrypt data by adding or xor-ing their pseudo-random output to the plaintext. For the same value of the key, a stream cipher will generate the same output stream. This means that if the same key is used to encrypt two messages, and attacker can easily decrypt. In order to solve this issue, modern stream ciphers take two inputs: the secret key and the initialisation vector (IV). The pair (key,IV) should be non-repeating, i.e. the application developer should make sure that for the same key, no IV is used twice.

Note: the stream cipher RC4 does not take an IV. Hence, special care needs to be taken when this cipher is used.

For more information on a selection of modern stream cipher proposals, we refer to [12].

5.1.3. Hash function

A hash function performs hashing: it computes a short and unique representation of a piece of data in an irreversible way. A hash function doesn't use a secret key. A cryptographic hash function *h* satisfies the following security requirements:

1. Preimage resistance: for any given output *y*, it should be infeasible to find an input *x* such that *h(x)=y*.
2. Second preimage resistance: for any given input *x*, it should be infeasible to find a second input *y* such that *h(x)=h(y)*.
3. Collision resistance: it should be infeasible to find two inputs *x, y* such that *h(x)=h(y)*.

In security proofs for cryptographic schemes and protocols that use hash functions, it is often assumed that a hash function outputs random strings, with as restriction that the same input to the hash functions always leads to the same output. This is called the *random-oracle model*.

Note: some textbooks define next to unkeyed hash functions also a hash function that use a secret key as input. In this document, hash functions that use a key, are called Message Authentication Codes (MACs)

5.1.4. Message Authentication Code

A message authentication code (MAC) takes as input a message and a secret key and outputs a short string (*tag*). Many MAC constructions take a third input: the *nonce*. A nonce is a non-repeating value, similar to the IV in modern stream ciphers. If nonces are repeated, the security of the MAC is reduced. For some MACs, nonce misuse can facilitate a key recovery attack.

The security criteria for a MAC are similar to those for a hash function. For an attacker who doesn't know the secret key, it should be infeasible to produce preimages, second preimages or collisions (cf. above). These criteria can be replaced by the following stronger criterion: for an attacker who doesn't know the secret key, it should be infeasible to distinguish the output of a MAC from random noise. An additional security requirement is that it should be infeasible to recover the secret key, even after many message-tag pairs have been observed.

5.2. Asymmetric primitives

With asymmetric primitives, the keys used for decryption can't be derived easily from the keys used for encryption. Similarly, the keys used for data authentication can't be derived easily from the keys used for validation. Only the keys used for decryption and for data authentication need to be kept secret. They are called "private keys." The keys used for encryption and for validation are called "public keys."

We briefly introduce here some mathematical problems on which asymmetric cryptographic schemes (digital signature, asymmetric encryption) are based.

5.2.1. Factoring/RSA problem

Each integer number can be written in exactly one way as a product of prime numbers. Factoring refers to finding this product. While most integers can be factored easily, cryptography uses integers for which factoring is difficult, for example: integers that are the product of two large primes of approximately the same size.

If the modulus of an RSA public key can be factored, then the private key can be determined as well. However, there are other ways to decrypt messages that were encrypted with RSA. The RSA problem is defined as recovering the input of the RSA modular exponentiation without prior

knowledge of the secret key. Most factoring-based cryptography can be secure only if the RSA problem is difficult.

5.2.2. Discrete logarithm

The discrete logarithm problem is most easily explained with integer numbers, but it can be defined over any finite Abelian group, e.g. point addition on an elliptic curve over a finite field. For integer numbers, the discrete logarithm problem is: given h, g, n find x such that $h=g^x \bmod n$.

The famous Diffie-Hellman key agreement protocol can be secure only if the discrete logarithm problem is difficult, but it is not proven that there are no other ways to recover the secret key that is produced by this protocol. Several Diffie-Hellman problems have been defined. Most discrete-logarithm-based cryptography is based on the difficulty of one of the Diffie-Hellman problems.

5.2.3. Pairing

A pairing is a map, taking inputs from two groups and outputting an element in a third group, which has some additional properties (bilinearity, non-degeneracy). In cryptography, the three groups are usually defined by point addition over three elliptic curves. The security of pairing-based systems relies on the difficulty of several hard problems.

5.3. Strength of cryptographic primitives

5.3.1. Key sizes

Shannon proved that from an information-theory point of view, perfect confidentiality is possible only if the entropy (size) of the secret key is at least as large as the entropy (size) of the message [13]. Information-theoretically secure authentication requires that a new secret key be used for every message. At the same time, there are very fast and simple to implement schemes known that achieve perfect confidentiality (the Vernam cipher or One-Time Pad) [14] or authentication (authentication codes) [15] [15], provided that the key material is available.

In practice, large keys are more difficult to handle than small keys. Large keys are more difficult to generate and to store securely. In secure hardware, the storage cost per bit is still high enough to be an important design constraint. For asymmetric-cryptography mechanisms, larger keys lead also to longer electronic signatures and larger minimal sizes for the ciphertext.

One can argue that practical cryptography is in fact the discipline that seeks to provide non-perfect but still good security while minimizing the size of the keys. One of the main purposes of the Algorithm and Key Size Report [4] is to inform the users about the smallest key sizes that can be used. As explained in the report, these minimal key sizes depend on the general state of the technology and on the specific cryptographic primitive.

5.3.2. Shortcut attacks

For symmetric-cryptography primitives, it is usually understood that the key size of the mechanism equals its effective key size, which is loosely defined as the logarithm of the expected number of computer operations that are required to recover the key (Triple DES is a noteworthy exception: with a key size of 168 bits, it has an effective key size of 112 bits only.). Since in principle an attacker can always try out all possibilities for the key (exhaustive key search), it is clear that the effective key size of a cryptographic mechanism can never be larger than its actual key size.

Any algorithm that can recover the secret key with a smaller expected number of computer operations, is a shortcut attack, even though for algorithms with large keys, it might pose no immediate practical threat.

Asymmetric-cryptography primitives are always used with key sizes that are much larger than their effective key size, which is determined by the computational requirements of the best shortcut attack known.

5.4. Key management

Note that in this document, the word "key" always refers to a "cryptographic key," i.e. a parameter used in a cryptographic algorithm. More information about key management techniques can be found in [9, p. Chapter 13].

5.4.1. Secret keys and public/private key pairs

In the oldest form of cryptography, the decryption key is equal to the encryption key, and the verification key is equal to the authentication key. Hence, an entity that should be able to decrypt and/or verify the authenticity of data, uses the same set of keys as the entity that encrypts and/or authenticates the data. Because of this symmetry, this is called symmetric cryptography.

By necessity, symmetric cryptography is based on secret keys. In order to keep the application secure, there needs to be a mechanism to protect the confidentiality and the authenticity of the keys.

Asymmetric cryptography starts from the insight that for a typical application, the security requirements on the keys are asymmetric:

1. In order to control the confidentiality of data, we need to control only the access to the decryption key. In principle, the encryption key may be known by other entities.
2. In order to guarantee the authenticity of data, we need to control only the access to the authentication key. In principle, the verification key may be known by other entities.

In asymmetric cryptography, the symmetry between the keys is removed. It is infeasible to derive the decryption key, respectively the authentication key, from the encryption key, respectively the verification key. In order to keep the application secure, there still needs to be a mechanism to protect the confidentiality and the authenticity of the decryption key and the authentication key. These keys are called the private keys. For the encryption key and the verification key, however, only the authenticity needs to be protected. Since these keys may be made public, they are called public keys. This is the reason why asymmetric cryptography is also called public-key cryptography.

5.4.2. Objectives of key management

Cryptographic mechanisms reduce the problem of data security to the problem of key management. This is known as Kerckhoffs' principle: "the security of a cryptosystem should not rely on the secrecy of any of its workings, except for the value of the secret key." It follows that a good key management is essential in order to benefit from the introduction of cryptography. We distinguish the following objectives of key management:

1. Protecting the confidentiality and authenticity of secret and private keys, as well as protecting secret and private keys against unauthorized use.
2. Protecting the authenticity of public keys.
3. Ensuring the availability of secret and public keys.

5.4.3. Key generation

Secret keys and private keys need to be unpredictable. Symmetric primitives usually don't have additional requirements for the secret keys, except that some primitives have a small fraction of weak keys, which shouldn't be used. Asymmetric primitives usually have additional requirements, both on their private and public keys. For example, they often require the generation of primes that need to satisfy extra properties. Generating secret keys or private keys with a sufficient amount of entropy turns out to be a very challenging task in practice. For the most recent illustration of randomness-related problems, we refer to [16].

5.4.4. Key registration/certification

Keys need to be associated with their owner (user). For example, public keys are linked to their owner by means of (public-key) certificates. Through the issuing of a certificate, a registration authority/certification authority guarantees that a certain key belongs to a certain user, and for what purposes the owner may use the key. A certificate also has a validity period. Certificates are usually public documents. Their authenticity is ensured by means of a digital signature, placed by the certification authority.

5.4.5. Key distribution & installation

Keys need to be distributed to their users. For systems based on symmetric cryptography, both the sender and the receiver need to obtain a copy of the key, hence the key needs to be transported securely (protection of confidentiality and authenticity) at least once. All the copies of the key need to be installed and stored securely. For systems based on asymmetric cryptography, the private key is often generated in place, such that no transport is needed. (In secure hardware, the functionality to export the private key of an asymmetric key pair is usually deliberately not implemented.) Otherwise, it needs to be protected like a symmetric key. The public key still needs to be transported, but only the authenticity (and hence the integrity) needs to be protected, which is achieved by the use of certificates.

In order to reduce the number of keys that need to be stored locally, one can use Key Distribution Centers, centrally managed key servers. Users share long-term keys with the Key Distribution Centers and trust the servers to provide them with the keys of the other users when they need them. Key Distribution Centers can manage both secret and public keys.

5.4.6. Key use

The goal of key management is to put keys in place such that they can be used for a certain period of time. During the lifetime of a key, it has to be protected against unauthorized use by attackers. The key must also be protected against unauthorized uses by the owner of the key, e.g. even the owner of the key should not be allowed to export a key or to use it in an insecure environment. This protection is provided by storing the key on secure hardware and by using secure software, which includes authorization checks.

5.4.7. Key storage

By using secure hardware, it is possible to store keys such that they can never be exported, and hence are very secure against theft or unauthorized use. However, sometimes keys get lost and it might be desirable to have a backup copy. Organizations might require backups of keys in order to be able to access data after employees leave. Similarly, expired keys might be archived in order to keep old data accessible. Finally, under certain conditions law enforcement agencies might request access to certain keys. Technical systems that implement access for law enforcement agencies are called key escrow mechanisms.

Backup, archival and escrow of keys have in common that they complicate key management, because they increase the risk for loopholes for unauthorized access to keys.

The advanced security requirement non-repudiation requires that the owner of a key is the only one who has access to the key. For example, keys that are used for advanced electronic signatures have to be under the sole control of the user [17]. Archival, backup or storage of such keys is difficult.

5.4.8. Revocation/validation

Cryptographic keys expire and are replaced. Sometimes it can happen that keys have to be taken out of use before the planned end of their lifetime, e.g. if secret keys leak to outsiders or if developments in cryptanalysis make schemes insecure. This process is called revocation. In centralized systems, revocation can usually be achieved relatively easily, but in distributed systems special measures have to be implemented to avoid that people use or rely on keys that have been expired early.

In the context of revocation, validation has a very specific meaning. It means to check whether a cryptographic operation, e.g. placing a digital signature, was performed with a key that is valid, or was valid at the time the operation took place.

5.4.9. Key destruction

When the lifetime of the key has expired, it has to be removed from the hardware. This requires a secure deletion process. In most operating systems and applications, the deletion of a file only clears a logic flag. It doesn't result in actual removal of the data until the disk space used to store the file is reclaimed and overwritten by another application. On many file storage media, even after a file has been overwritten, it is possible to recover the original file, using some moderately advanced equipment. This is called *data remanence*. Various techniques have been developed to counter data remanence. At the logical level, one can overwrite the disk space repeatedly with certain bit patterns in order to make recovery difficult. At the physical level, one can degauss (on magnetic media) or employ other operations that restore the storage media in pristine state, or one can destruct the storage media. More information can be found in [10].

6. Case study. Protective measures

The purpose of this section is to give recommendations on what technological protection measures are appropriate in the sense of Commission Regulation (EU) No 611/2013 of 24 June 2013 on the measures applicable to the notification of personal data breaches under Directive 2002/58/EC of the European Parliament and of the Council on privacy and electronic communications [2]. To this end, for the purpose of this case study, we classify data breaches in 5 types and describe the technological measures that can be considered as appropriate to make the data unintelligible as described in Section 2.1.4.

Note that this section considers technological measures only. Technological measures form only one piece of the security puzzle. A complete security solution requires also the implementation of risk management procedures, policy and organizational controls, etc. For recommendations on these non-technological measures, we refer to the Introduction section.

6.1. Classification

For the purpose of this work we consider 5 types of attacks and briefly outline the technical countermeasures. In Section 6.2 we explain the technical countermeasures in more detail.

The first type corresponds to the second case described in Article 4, item 2: data has been replaced by a hashed value (text cited in Section 2.1.4, based on [2]).

6.1.1. Type 0: hashed-data-only leak

Description:

The data itself was not exposed, but only hashed values. Hashing is by definition a non-invertible operation. Even if the key of the hash function is leaked, the non-invertibility of the hash function implies that an attacker can't derive the data directly from the hashed value alone. Note however that an attacker may be able to use the leaked hashed values in order to check if the hashed data is equal to some value that the attacker has guessed or obtained by other means.

Examples:

A file containing hashed passwords was leaked. Note that in case of a predictable password, the attacker can use the file with the hashes to verify his guesses.

Protection:

If the hash function used is a secure cryptographic keyed hash function and if the key of the hash function was not leaked, then leakage of the hashed values doesn't cause any leakage of data. Even if the key of the hash function is leaked, leakage of the hashed value only allows the attacker to confirm some guessed value for the data.

The next 4 types all correspond to the first case described in Article 4, item 1: data has been encrypted (see Section 2.1.4 for the cited text). Since the level of technological protection required in order to maintain security in the case of attacks depends on the severity of the attack, we subdivided this case in several different types. The types are presented here in order of increasing severity of the attacks (correspondingly, increasing level of technological protection).

6.1.2. Type 1: logical leakage

Description:

The data was exposed, but only at a logical level.

Examples:

An unauthorized copy of files was made; a write-once device was lost or stolen; the data was sent over an insecure line, i.e. over the Internet or over a WiFi; someone hacked into a WiFi.

In this case, the data loss is very similar to the loss of data printed on paper: one can quickly determine uniquely which data was exposed. This case is the easiest case to protect by means of cryptography, because the situation corresponds to the textbook example for the use of cryptography, namely it is equivalent to the situation that only the *communication channel* is attacked. There are no backdoors or side-channels to consider.

Protection:

The data is secured against this type of data breach if it is protected by means of state-of-the-art authenticated encryption and if only the ciphertext was exposed. In particular, the decryption key should still be secret.

6.1.3. Type 2: hardware leakage

Description:

An attacker has data-storing hardware in his possession, allowing him to apply physical attacks in order to bypass certain security controls applied at the logical level. Possibly, the attacker has a hardware lab available.

Examples:

Loss of laptops in powered-down state, but also loss of hard disks, USB sticks, any device that once stored sensitive or confidential data, even if it was deleted.

Protection:

The data is secure against this type of data breach if it is protected by authenticated encryption and if all unencrypted copies of the data (sensitive or confidential) that have ever been on this hardware, have been deleted securely. (Secure deletion is described below.) The same protection must be applied to the cryptographic keys that have been used. In case of a stolen laptop, this means that also the swap file and memory dump (hibernation) must have always been encrypted. From these requirements follows that the top-level cryptographic key should not be stored on the device, but should be input manually or read from a removable token.

6.1.4. Type 3: break-in on live device

Description:

An attacker has access to a user device that is working on sensitive or confidential data. The device is active during the attack. The attacker doesn’t have the device in his possession and can’t apply physical attacks.

Examples:

The attacker has access to a computer while a user is logged in; a system has been hacked; a virus with surveillance capabilities is active on the device.

Protection:

In order to be secure against this type of attack, the data has to be protected by means of authenticated encryption. The data should not be present in unencrypted form during the attack. Cryptographic keys should not be present in unencrypted form on the device during the attack.

Since that is opened by an application, resides in unencrypted form in the RAM, it is not protected against this attack. Data can only be considered secure if one can guarantee that the data was not opened for the entire duration of the attack. If cryptographic keys are present in RAM, then they are also to be considered as leaked, as well as all data that was encrypted using these keys. The amount of data leaked during this type of attack can be reduced by keeping as many keys as possible inside secure hardware and using a proper key architecture such that keys which have to reside in RAM during their usage, don't encrypt large amounts of data.

6.1.5. Type 4: full user impersonation

Description:

The attacker has the information and/or hardware tokens allowing him to impersonate the user.

Examples:

A user loses his password or access token.

Protection:

The amount of data leaked during this type of attack can be reduced by keeping as many keys as possible inside secure hardware and using a proper key architecture such that keys which have to reside in RAM during their usage, don't encrypt large amounts of data. Furthermore, the access control system should be designed such that the data encryption keys are not present on the access tokens.

6.2. Countermeasures

We explain the technical countermeasures that were mentioned in the previous section.

6.2.1. Secure deletion of files

In most operating systems and applications, the deletion of a file only clears a logic flag. It doesn't result in actual removal of the data until the disk space used to store the file is reclaimed and overwritten by another application. On many file storage media, in particular magnetic storage media, even after a file has been overwritten, it is possible to recover the original file, using some moderately advanced equipment. This is called *data remanence*. Various techniques have been developed to counter data remanence. At the logical level, one can overwrite the disk space repeatedly with certain bit patterns in order to make recovery difficult. At the physical level, one can degauss (on magnetic media) or employ other operations that restore the storage media in pristine state, or one can destruct the storage media. More information can be found in [10].

6.2.2. Authenticated encryption

Encryption ensures that data becomes unintelligible except to users who possess the decryption key. Data Authentication ensures that users who possess the validation key can detect if unauthorized modifications were applied to the data during transit or storage. Since the 2000s, there have been several developments leading to the current opinion that encryption of data should *always* be accompanied by data authentication. Principally, because there are no applications where it is of interest to keep data confidential, but have no guarantees about the correctness of the data, and also practically, because encryption systems without authentication

are more vulnerable to attacks. Authenticated encryption is the technical term for the combination of both operations.

For more information on encryption, data authentication and other cryptographic methods, we refer to [18]. For recommendations on schemes and ciphers to use in order to provide authenticated encryption, we refer to Section 4 of [1].

6.2.3. Secure hardware

Secure hardware refers to a processor dedicated to handle sensitive or confidential data. Examples are a smartcard, a hardware security module (HSM) or a trusted platform module (TPM). Secure hardware can be programmed to deliver only a specific small set of services to a general computing platform. By restricting the inputs that it accepts and the outputs that it delivers, it can be made more secure that a general-purpose computer that is always under threat of viruses and other malware.

For example, one can set up a system where all cryptographic keys are stored and used on secure hardware only, i.e. they never leave the secure hardware. All data is encrypted-authenticated and decrypted-validated on the secure hardware. In this way, if an attacker gets physical access to system, he can only obtain the data that is present in decrypted form outside the secure hardware at the time of his attack. All data inside the secure hardware remains safe, as well as all the data outside the secure hardware but encrypted-authenticated with keys present only inside the secure hardware.

For performance reasons, it may not be feasible to perform all cryptographic operations inside the secure hardware. In that case, a proper key architecture will limit the exposure in case of a breach.

6.2.4. Key architecture

The amount of data encrypted-authenticated with one key should be limited, for various reasons. Firstly, most authenticated encryption schemes suffer from security degradation if large amounts of data are encrypted-authenticated with the same key [refer to D1 or to D5]. Secondly, by introducing different keys, the damage can be reduced in those cases where a single key gets lost, stolen or broken.

A key architecture will consist of at least two levels of keys. Only the keys at the lowest level (*session keys*) are used to encrypt-authenticate or decrypt-validate a well-defined subset of the data.

Keys at higher levels are used only to encrypt-authenticate or to derive the keys at lower levels. Keys should never be stored in unencrypted form outside secure hardware. Higher-level keys should be present on a restricted set of devices; they should not be present on devices that are used to encrypt-authenticate or decrypt-validate the data.

By adopting a schedule of regularly updating all keys (including the higher-level keys!), and securely deleting the old keys when they are no longer needed, i.e. when all data encrypted with these keys has expired, it is possible to achieve forward secrecy.

6.2.5. Access control system

An access control system uses passwords in combination with tokens (e.g. smartcards, USB sticks etc) containing cryptographic keys. The cryptographic keys used for the authenticated encryption of the data should be different from the keys used for access control. It should not be possible to derive encryption keys from access control keys or vice versa.

7. Concluding remarks

The protection of information is a multi-faceted problem. Since cryptography is based on mathematics, one can prove rigorously that it can eliminate certain risks. This is an important strength of cryptography. It is only true, however, if state-of-the-art cryptographic methods are used. It is therefore recommended to use only standardised algorithms, and to update products when the standard changes.

Cryptography can protect the secrecy and the integrity of data. Historically, these two properties have been ensured by means of different techniques. Currently it is recommended to always accompany secrecy protection techniques by integrity protection techniques, or to use techniques that simultaneously provide secrecy protection and integrity protection. (I.e., the use of encryption techniques without integrity protection is no longer recommended.)

When using any cryptographic mechanisms, in order to achieve a desired level of security, we recommend referring to the detailed information provided in [1].

The Appendix mentions briefly some more advanced cryptographic mechanisms that can be useful in the context of privacy protection. Data minimization techniques and anonymisation techniques could play a role here and they are relying on cryptography; we are only providing some links for this area, as further work is needed and this is beyond the scope of this document.

For the future, we recommend that ENISA considers a survey of deployment and best practices of the more advanced cryptographic methods that can be used to improve personal data protection.

8. Appendix: data minimization

The principle of *data minimization* is an important trend in the context of the protection of personal data. In brief, it can be summarised as follows: the best protection for personal data is to not have collected the data at all. Data minimization can be seen as an example of the principle "prevention is better than reaction." Here, "prevention" means that one tries to avoid storing personal data as much as possible, thereby reducing the need for "reaction", i.e. securing stored data.

The following three cryptographic techniques are all examples of data minimization techniques.

8.1. Local extraction

Local extraction is often used in "Privacy by design" solutions, where the motivation is to avoid privacy issues by eliminating central storage. An example for a pay-as-you-drive electronic toll collection system is given in [19]. In this system, there is no need to send information about the whereabouts to a central database. Instead, a trusted tamper-resistant unit on board of the vehicle computes the toll to be paid and transmits this information to a central server. Through a cryptographic technique known as commitments, the on-board unit guarantees that its computations are based on correct information. A second example is a smart metering system described in [20]. Here, instead of sending detailed data about a user's energy consumption, the smart meter computes the amount of money owned and passes on this information to the utility company.

8.2. Attribute-based credentials

In a "classical" PKI system, the public keys of the users are distributed by means of certificates. Next to the public key, these certificates also contain personal information about the user, often more personal information than is strictly needed. Attribute-based credentials can be used to reduce the amount of personal information revealed down to what is strictly necessary for the given application. For example, if a user needs to be of a certain age in order to get access, then it is not necessary to reveal the user's exact birth date: a simple assertion that the user has at least the required age, is sufficient.

8.3. Private information retrieval

A private information retrieval protocol allows a user to retrieve an element from a database without the database owner knowing which element of the database is retrieved [21].

9. Bibliography

[1] ENISA, "Algorithms, Key Size and Parameters Report. 2013 Recommendations," 2013. [Online]. Available: https://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/algorithms-key-sizes-and-parameters-report

[2] EU, "Commission Regulation (EU) No611/2013 of 24 June 2013 on the measures applicable to the notification of personal data breaches under Directive 2002/58/EC of the European Parliament and of the Council on privacy and electronic communications," 2013. [Online]. Available: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2013:173:0002:0008:en:PDF.

[3] EU, "Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data," 1995. [Online]. Available: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML. [Accessed 13 08 2013].

[4] European Commission, "Proposal for a regulation of the European Parliament and of the Council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation), COM(2012)," 01 2012. [Online]. Available: http://ec.europa.eu/justice/data-protection/document/review2012/com_2012_11_en.pdf. [Accessed 16 07 2013].

[5] ENISA, "Recommendations on technical implementation guidelines of Article 4," 30 April 2012. [Online]. Available: http://www.enisa.europa.eu/activities/identity-and-trust/risks-and-data-breaches/dbn/art4_tech. [Accessed 8 8 2013].

[6] D. Gollmann, Computer Security, Third edition ed., Wiley, 2011.

[7] F. S. Gurses, Multilateral privacy requirements analysis in online social network services, Leuven: KU Leuven, 2010.

[8] N. Smart, Cryptography, an introduction, 3rd Edition ed., 2009.

[9] C. Paar and J. Pelzl, Understanding cryptography: a textbook for students and practitioners, Heidelberg: Springer, 2009.

[10] A. J. Menezes, S. A. Vanstone and P. Van Oorschot, Handbook of applied cryptography, CRC, 1996.

[11] R. Kissel, M. Scholl, S. Skolochenko and L. Xing, "Guidelines for media sanitization," Gaithersburg, 2006.

[12] L. R. Knudsen and M. Robshaw, The block cipher companion, Heidelberg: Springer, 2011.

[13] M. Robshaw and O. Billet, Eds., New stream cipher designs, vol. LNCS 4986, Springer, 2008.

[14] C. E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal,* vol. 28, pp. 656-715, 1949.

[15] G. S. Vernam, "Secret signaling system". U.S. Patent 1310719, 22 7 1919.

[16] E. N. Gilbert, F. J. MacWilliams and N. J. A. Sloane, "Codes which detect deception," *Bell Systems Technical Journal,* vol. 53, no. 3, pp. 405-424, 1974.

[17] A. K. Lenstra, J. P. Hughes, M. Augier, J. W. Bos, T. Kleinjung and C. Wachter, "Ron was wrong, Whit is right," 2012. [Online]. Available: eprint.iacr.org/2012/064. [Accessed 23 05 2013].

[18] *Community framework for electronic signatures,* 1999.

[19] V. Rijmen, "Methodology and security measures for securing personal data," ENISA, 2013.

[20] W. de Jonghe and B. Jacobs, "Privacy-friendly electronic traffic pricing via commits," *Formal Aspects in Security and Trust,* vol. LNCS 5491, pp. 143-161, 2008.

[21] A. Rial and G. Danezis, "Privacy-preserving smart metering," in *Proceedings of the 10th annual ACM workshop on privacy in the electronic society*, 2011.

[22] B. Chor, E. Kushilevitz, O. Goldreich and M. Sudan, "Private information retrieval," *Journal of the ACM,* vol. 45, no. 6, pp. 965-981, 1998.

ENISA
© European Union Agency for Network and Information Security
Science and Technology Park of Crete (ITE)
Vassilika Vouton, 700 13, Heraklion, Greece

Athens Office
ENISA, 1 Vasilissis Sofias
Marousi 151 24, Athens, Greece

PO Box 1309, 710 01 Heraklion, Greece
info@enisa.europa.eu
www.enisa.europa.eu