Renewal periods for cryptographic keys

Posea, S.

Award date:
2012

Renewal Periods for Cryptographic Keys

Simona Posea

August 2012

Technische Universiteit
Eindhoven
University of Technology

Eindhoven University of Technology
Department of Mathematics and Computer Science

Master's Thesis

Renewal Periods for Cryptographic Keys

Simona Posea

Supervisors:
dr. ir. L.A.M. (Berry) Schoenmakers (TU/e - W&I)
drs. D.M. Wieringa RE CISSP (Deloitte - S&P)
H. Hambartsumyan MSc. CISA (Deloitte - S&P)

Eindhoven, August 2012

Abstract

Periodically changing cryptographic keys is a common practice. Standards and publications recommending how long a key may be used before it should be renewed are available for guidance. These recommendations are rather based on experience or tend to only consider the key length as a parameter. In practice, the key length is often not the main concern. Compromise of cryptographic systems is usually, as far as we know, not caused by keys which are brute-forced, but by other factors such as inadequate key management. Possible additional parameters for determining the lifetime of a key could be the storage location, the key transport channel, the training and awareness of employees, etc.

This thesis addresses the key renewal period problem for symmetric keys by presenting a risk-based methodology for key renewal interval estimation. It provides a step by step guide to planning, conducting and maintaining key exposure risk assessment. Key renewal period's dependency on key exposure risks caused by insecure change and use of cryptographic keys is analyzed and general directions to protect key secrecy are provided. In particular, the methodology proposed deals with the limitations of existing key renewal intervals recommendations by considering the key management environment as a differentiator. For example, keys stored inside a hardware security module (HSM) are much safer compared to keys stored in databases, hence can be used for a longer period. The risk-based methodology adopted in this thesis not only guides the reader into estimating suitable key renewal intervals, but also allows for easy identification of key management processes affecting key confidentiality.

Acknowledgments

I take this opportunity to express my gratitude to all those who helped me through the completion of this dissertation.

I would like to thank my supervisor, Berry Schoenmakers, for all his feedback, careful guidance and patience through the entire process. The several discussions I had with him were invaluably helpful, giving me sound advice, a lot of new insights into the researched topic and assisting me in writing the report.

To my supervisors in Deloitte, Derk Wieringa and Henri Hambartsumyan, a very special thank you for the vast knowledge and experience shared and for their intense involvement in this research. Their dedication to excellence and their enthusiasm inspired and motivated me to make progress on my thesis.

My profound gratitude also goes to Nicola Zannone for reviewing my thesis, for providing valuable suggestions of improvement and for accepting to be in my assessment committee.

I am indebted to my colleagues in Deloitte for providing a stimulating and fun environment in which to learn and progress. I express my deepest appreciation to Sergio Hernando for several constructive discussions and for challenging me with very interesting questions about my research topic.

Last but not least, I would like to thank all my friends and family for being supportive during the whole experience.

Simona Posea
Eindhoven, August 2012

Contents

Glossary of Terms and Abbreviations

Terms and Definitions

Access vector	Measure to express how remote an attacker can be to attack a target: Local, Adjacent Network, Network [PM07].
Asset	Anything that has value to the organization [ISO04].
Asymmetric key cryptography	A cryptography system that uses two different keys for protecting and respectively processing protected data. One of the key is made public, while the other key is kept secret by its owner. It is also referred as public key cryptography. Well-known asymmetric algorithms include the RSA, DSA and ElGamal primitives.
Control	Means of managing risk, including policies, procedures, guidelines, practices or organizational structures, which can be of administrative, technical, management, or legal nature. The term is also used as a synonym for safeguard or countermeasure [Sta07].
Cryptographic boundary	An explicitly defined perimeter (i.e. set of hardware, software or firmware components) that establishes the boundary of all components of a cryptographic module [FIP09].
Cryptographic module	The set of hardware, software, firmware, or some combination thereof that implements cryptographic logic or processes, including cryptographic algorithms and key generation, and is contained within the cryptographic boundary of the module [FIP94]. Note that a cryptographic module can be a complete product, a sub-system of a larger product or a single component of a product.
Impact	The magnitude of harm that could be caused by a threat's exercise of a vulnerability or set of vulnerabilities [GSF02].
Information management policy	Policy that specifies what information is to be collected or created and how it is to be managed [BBCS10].

Information security policy	Aggregate of directives, regulations, rules and practices that prescribes how an organization manages, protects and distributes information [Kis11].
Key-generating module	A cryptographic module in which a given key is generated [BA01].
Key metadata	Information associated with a particular key that specifies the secure and appropriate usage and management of the key [BBCS10].
Key transport	A key establishment procedure whereby one party (the sender) selects and encrypts the keying material and then distributes the material to another party (the receiver) [BR11].
Key wrapping key	A key-encrypting key [BR11].
Policy	Overall intention and direction as formally expressed by management [Sta07].
Risk	Combination of the likelihood of an event and its impact [NIS11b].
Risk assessment	The process of identifying, prioritizing, and estimating risks to organizational operations (including mission, functions, image, reputation), organizational assets, individuals, other organizations, and the nation, resulting from the operation of an information system [GSF02].
Risk acceptance (retention)	Acceptance of the burden of loss or benefit of gain from a particular risk [Sta08].
Risk avoidance	Decision not to become involved in, or action to withdraw from, a risk situation [Sta08].
Risk reduction	Actions taken to lessen the likelihood, negative impact, or both, associated with a risk [Sta08].
Risk model	A key component of a risk assessment methodology (in addition to assessment approach and analysis approach) that defines key terms and assessable risk factors [NIS11b].
Risk transfer	Sharing with another party the burden of loss or benefit of gain, for a risk [Sta08].
Risk treatment	The process of selecting and implementing measures to modify risk [ENId].
Social engineering	The process of breaking corporate security by manipulating employees into divulging confidential information. It is mainly based on psychological tricks in order to gain trust [Res11].

Symmetric key cryptography	A cryptography system in which the same cryptographic key is used for both protecting information (e.g. applying encrypting) and processing protected data (e.g. decryption of ciphertext). This key is referred as "symmetric key", sometimes known as "shared key" as well. Symmetric keys are usually known by one or multiple entities. Examples of well-known symmetric algorithms are: AES, DES, Triple DES, Twofish, etc.
Unconditional security	The security is guaranteed against the adversary having unlimited (i.e. infinite) computational resources [Shi].
Vulnerability	A weakness of an asset or group of assets that can be exploited by one or more threats [ISO04].

Abbreviations

AES	Advanced Encryption Standard
ATM	Automated Teller Machine
CKMS	Cryptographic Key Management System
CSP	Critical Security Parameters
CVSS	Common Vulnerability Scoring System
CWE	Common Weaknesses Enumeration
DES	Data Encryption Standard
DSS	Data Security Standard
ECRYPT	European Network of Excellence in Cryptology
FIPS	Federal Information Processing Standard
FNISA	French Network and Information Security Agency
HSM	Hardware Security Module
ISO	International Standards Organization
LRC	Leakage Resilient Cryptography
NIST	National Institute of Standards and Technology
NSA	National Security Agency
NVD	National Vulnerability Database
OUP	Originator Usage Period
PCI	Payment Card Industry
RNG	Random Number Generator
RUP	Recipient Usage Period
VND	Vulnerability Notes Database

Chapter 1

Introduction

1.1 Context

Information systems together with the data they contain represent essential assets of an organization and need to be suitable protected. This need is particularly emphasized with the increasing number of interconnected computers that share the same networks, to which competing organizations and individuals may have access too. While cryptography was initially confined to the realms of the military, the expansion trend noticed in organizations makes it now ubiquitous [KK11]. Cryptography is now a common practice for protecting information, either in transit or at-rest, against unauthorized modifications and disclosure and for authenticating the identities of system users [BBCS10]. Effectively implemented cryptography can reduce the need to protect large amounts of information to the need to protect one or more cryptographic keys and key metadata.

An important principle in cryptography, due to Auguste Kerckhoffs, states that a cryptographic system should be secure even when everything about the system, except for the key, is public. A cryptographic key can be seen as the combination to a safe: if you have the right combination, you can open the safe, but it is generally hard to open it otherwise. Therefore, keys must be protected for as long as they are authorized for use by legitimate entities. Depending on the key type, key use can consist in one or both of the following: data protection (e.g. encrypting data) or processing of protected data (e.g. decrypting ciphertexts). Given the advances in technology and the error-prone human nature, protecting keys for a large amount of time tends to be challenging. To prevent key disclosure, companies should consider changing keys periodically. How long a key is authorized for use should depend on the circumstances under which the cryptographic control is being used, and the perceived risk [Sta07].

The need for changing cryptographic keys

Generally, the greater amount of time a key is used, the more likely it is to be exposed and the higher the potential consequences associated with its exposure [Kri11]. To address this problem, best practices recommend renewing data protection keys at specific intervals, which we further refer to as *key renewal periods* or *key renewal intervals*. Note that changing data protection keys will further lead to changing associated processing keys.

Periodically changing cryptographic keys has the role to [BR11]:

- Limit the amount of information protected by a given key that is available for cryptanalysis;
- Limit the amount of exposure when a single key is compromised;
- Limit the use of a particular algorithm to its estimated effective lifetime;
- Limit the time available for computationally intensive cryptanalytic attacks (in applications where long-term key protection is not required).

All these contribute to reducing the likelihood of key exposure and the potential impact caused by such an event. For example, consider a company working on an innovative product to be released in the future. The product developers exchange information protected with a cryptographic key that has been already in use for a couple of years. To avoid high costs of replacement, the management does not consider renewing the key soon. Such a scenario gives advantage to business competitors, who are highly motivated to steal the product specification, make some small changes such that it does not raise suspicions and launch it in the market before the real developers do so. It is likely that compromising the cryptographic key leads to business competitors having unauthorized access to the whole product specification. Having periodically changed the key would have considerably reduce the likelihood of such an attack. Compromising one key would only partially reveal product details and therefore, having the complete product specification would require increased effort.

Note that this example is just one of the multiple scenarios in which adversaries might compromise cryptographic keys for malicious purposes and it is meant to highlight the need for replacing keys every time they do not offer anymore an acceptable level of security. This need is emphasized by unexpected situations as well, e.g., suspected or proved key exposure incidents and employees with knowledge of the key leaving the company.

Problem identification

Based on the discussion above, the reader may be tempted to think that changing keys often is the solution to their secrecy. In reality, the costs associated with changing cryptographic keys tend to be very high [BR11]. Reasons include potential need for revocation and replacement of a large number of keys (e.g. where there are very large number of geographically and organizationally distributed key holders) or decryption and re-encryption

with the new key, very large, distributed databases. Therefore, expenses of implementing security measures meant to allow for longer key renewal periods might be justified.

In addition, cryptographic key replacement is usually a complex process and getting it right may be difficult. Inadequate key management might lead to cryptographic keys being partially or completely exposed even before being used. Hence, when selecting key renewal intervals, practitioners end up in a decision point that needs to consider two incompatible trends:

- On one hand, the greater amount of time keys are used, the more likely it is to be exposed.
- On the other hand, the more often keys are renewed through weak key replacement procedures, the more likely it is that keys will leak.

Finding the balance point between these two needs is what makes the selection of key renewal periods a complex process.

1.2 Objectives and Contributions

The main objective of this thesis is to provide a foundation for performing effective key renewal period estimation. We consider cryptographic keys that are to be generated, distributed and deployed as replacement keys. The question to answer becomes:

> ***How to estimate for how long should a newly installed key be used for protecting information?***

To answer the question, we start by clearly stating the key renewal period problem and highlighting the challenges it brings. Furthermore, the thesis contributes by proposing a risk-based methodology for key renewal interval estimation. In particular, we address the limitations of existing key renewal intervals recommendations in literature by taking into account different key environments. Note that the goal of the thesis is not to provide numerical estimations for key renewal intervals, but to analyze the behavior of the interval depending on the key exposure risks caused by changing and respectively using the cryptographic key.

While presenting the methodology, we provide practitioners with practical guidance for performing each of the steps required. The ultimate goal is to increase awareness among organizations regarding the significance of key renewal periods for protecting cryptographic keys.

1.3 Scope

For this research we limit our scope to renewal periods for symmetric cryptographic keys, as highlighted in Figure 1.1.

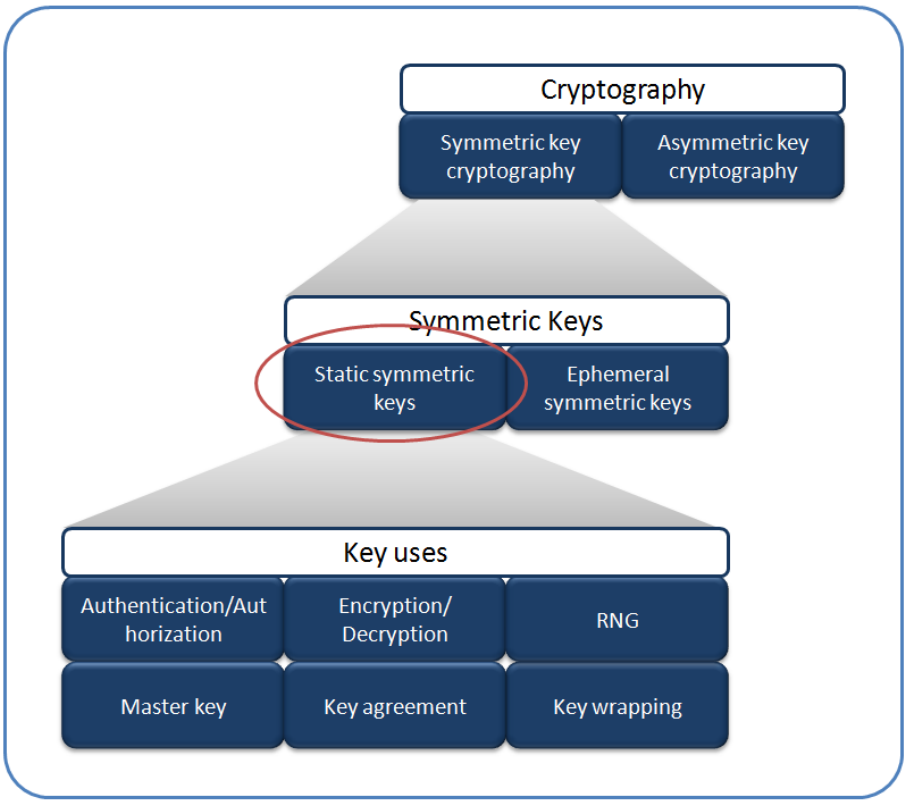

Figure 1.1: Key types and uses. The figure focuses only on static symmetric keys, the scope of this thesis.

Depending on the amount of time they are intended to be used, symmetric keys can be static (also known as long-term keys) or ephemeral (e.g., restricted to one communication session). Since by definition the use of ephemeral keys is limited to very short time intervals, the key renewal period problem does not apply. Therefore, we focus here on long-term symmetric keys, whose unauthorized disclosure tends to cause a significant impact on the business, due to the generally high amount of data they protect.

Depending on the security function they support and the intended use as displayed in the bottom part of Figure 1.1, several types of symmetric keys are identified [BR11]:

- **Symmetric authentication key**: Provides assurance of the integrity and source of messages, stored data or communication sessions. The authentication process requires the use of an agreed algorithm and a symmetric key known by both the authenticated entity and the authenticator.

- **Symmetric authorization key**: Used to grant privileges to an entity based on a symmetric cryptographic method. The symmetric key needs to be known by both the responsible for monitoring and granting access privileges and the authorization requester.

- **Symmetric data encryption key**: Used with symmetric algorithms to ensure data confidentiality by encrypting it.
- **Symmetric key wrapping key**: Used with symmetric algorithms to encrypt other keys. For this reason, they are known as *key encrypting keys* as well.
- **Symmetric master key**: Used with symmetric algorithms to derive other symmetric keys.
- **Symmetric random number generation key**: Used for generating deterministic random numbers.
- **Symmetric key agreement key**: Used to establish keys (e.g. data encryption keys, key wrapping keys, MAC keys) as part of the key establishment process.

Note that the analysis in this thesis is applicable to any long-term symmetric key, independent on its type.

1.4 Target Audience

This thesis is intended to serve individuals independent on their key management expertise and technical knowledge, responsible for:

- Key confidentiality protection;
- Purchase of technical products to be used within key management processes;
- Design, development, implementation or performance of any of the key management processes;
- Operational activities on hardware and software used to support key management;
- Security risk assessment and monitoring.

It is also addressed to the general reader interested in gaining more knowledge in the key renewal interval topic.

1.5 Outline

The remaining of this thesis is organized as follows:

- **Chapter 2** surveys previous related work.
- **Chapter 3** covers basic key management concepts. It includes an overview of the key management lifecycle and identifies the main key management stakeholders.

- **Chapter 4** is a key chapter that provides the methodology used for addressing the key renewal period problem.
- **Chapter 5** provides practitioners with general guidance for identifying and assessing vulnerabilities in the key management process.
- **Chapter 6** presents conclusions and directions for future work.

Chapter 2

State of the Art and Related Work

This chapter surveys standards and literature providing guidance on key renewal periods.

To the author's knowledge, the NIST Special Publication for key management [BR11] is the only one to include detailed key renewal interval recommendations for specific key types. Note that NIST does not refer to them directly as key renewal intervals, but uses the term "Originator Usage Period" (OUP) to denote the time period a symmetric key is authorized for protecting data [BR11]. Recommending OUPs is equivalent to recommending key renewal intervals.

In fact, NIST recommendations are broader, focusing on the entire cryptoperiod. Cryptoperiods are defined as the time interval a cryptographic key is authorized for use. One may hesitate when differentiating between the cryptoperiod and the renewal interval of a key, being tempted to think of a complete equivalence between the two. Figure 2.1 highlights this difference by introducing the "Recipient Usage Period" (RUP). RUP denotes the interval within which a key is only available for processing previously protected data. In some cases, the two defined usage periods may overlap (e.g. assume a cryptographic key used for protecting the communication beween two entities; the recipient is expected to process the data at a short interval after being sent by the originator) or be completely disjoint (e.g. a voting-type system, when only after all the data has been received, one can start processing the information).

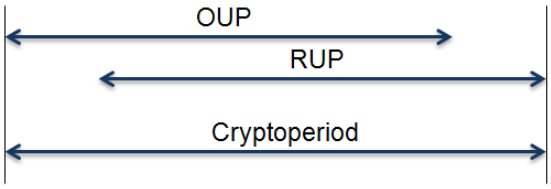

Figure 2.1: Cryptoperiod [BBCS10].

Illustrated in Table 2.1, NIST recommendations are only rough order of magnitude guidelines. They seem to be based on experience rather than on a well-defined mathematical model. The fragments below are extracted from [BR11] and support the previous statement:

> *"Note that the cryptoperiods suggested are only rough order of magnitude guidelines.*
> *...*
> *Based primarily on the consequences of compromise, a data encryption key that is used to encrypt large volumes of information over a short period of time (e.g., for a link encryption) should have a relatively short originator usage period. An encryption key used to encrypt less information could have a longer originator usage period."* [BR11]

Key type	OUP	RUP
Private Signature Key	1-3 years	
Public Signature Key	Several years (depends on key size)	
Symmetric Authentication Key	$\leq$ 2 years	$\leq OUP$ + 3 years
Private Authentication Key	1-2 years	
Public Authentication Key	1-2 years	
Symmetric Data Encryption Key	$\leq$ 2 years	$\leq OUP$ + 3 years
Symmetric Key Wrapping Key	$\leq$ 2 years	$\leq OUP$ + 3 years
Symmetric and asymmetric RNG Keys	Upon reseeding	
Symmetric Master Key	About 1 year	
Private Key Transport Key	$\leq$ 2 years	
Public Master Key	1-2 years	
Symmetric Key Agreement Key	1-2 years	
Private Static Key Agreement Key	1-2 years	
Public Static Key Agreement Key	1-2 years	
Private Ephemeral Key Agreement Key	One key agreement transaction	
Public Ephemeral Key Agreement Key	One key agreement transaction	
Symmetric Authorization Key	$\leq$ 2 years	
Private Authorization Key	$\leq$ 2 years	
Public Authorization Key	$\leq$ 2 years	

Table 2.1: Recommended cryptoperiods for specific key types [BR11]. Symmetric keys of interest for this research are highlighted in red.

The guideline has a static character and does not differentiate between key usage environments. To determine actual cryptoperiods for specific environments, NIST recommends a set of so-called key exposure risk factors to be considered [BR11]:

- The strength of cryptographic mechanism, referring to the algorithm used, the key length, block size and operation mode;

- Embodiment of the mechanism (e.g. software implementation on a PC, Security Level 4 as described by FIPS 140 certification);
- The operating environment of the cryptographic module (e.g. secure limited access facility, open office or publicly accessible terminal);
- The volume of information flow or the number of transactions;
- The security life of data;
- The security function[1] (e.g. data encryption, key production or derivation, key protection);
- The re-keying method (e.g. keyboard entry, remote re-keying within a PKI, re-keying using a key loading device with no human direct access to the key information);
- The key update or key derivation process;
- The number of copies of a key;
- The number of nodes in the network that share a common key;
- The threat to the information, describing whom the information is protected from and what are their perceived technical capabilities and financial resources to mount an attack;
- Key revocation and replacement cost, e.g. changing keys that protect data at rest might require data re-encryption with the new key, increasing replacement costs. This often leads to keys protecting stored data being less often changed than keys protecting data in transit.

Note that the former list is not exhaustive and can be further expanded. Other additional relevant factors we identify are:

- The embodiment of the key storage mechanism;
- The employee training and awareness;
- The network infrastructure security.

Except for NIST, the PCI-Data Security standards (PCI-DSSs) which apply to any company processing or transmitting credit card transaction data require encryption keys to be changed at least annually [BCEK07]. Numerous scientific reports and governments provide key length recommendations to ensure key resistance to guessing attacks by a specific year. One may think of this guidance as recommending "cryptoperiods expressed

[1]Note that despite of being mentioned by NIST as an influencing factor, we believe that the key type itself does not play a significant role in estimating key renewal periods. Instead, it is the criticality of data and processes that are protected with, that become a differentiator.

in absolute time", instead of time intervals. NIST [BR11], Lenstra et al. [LV01], FNISA [FNI10] and ECRYPT [OEFC11] offer valuable advice on key length selection. All these recommendations are aggregated in [Gir11], to accelerate the key length selection process.

NIST and FNISA come up with simple recommendations. NIST proposes minimum 80-bits of strength to be used by 2010, 112-bits by 2030 and at least 128 beyond 2030. Similar, FNISA considers 100-bit keys offering adequate protection between 2010-2020. After 2020, organizations are advised to increase the key size to at least 128 bits. Note that the standards do not bring solid justification for their recommendations.

ECRYPT recommendations are based on a study performed in [BDR+96] in early 1996. Blaze et al. [BDR+96] used technology (general purpose computers, FPGAs and special-purpose chips) to estimate how long 56-bit DES keys and 40-bit RC4 keys would take to be brute-forced. They further used the results to identify key lengths such that adversaries with various financial resources can guess them in a few months. ECRYPT applies Moore's law[2] to the results in [BDR+96] to reflect technology evolution by 2011. The key-size recommendations are as shown in Table 2.2. However, these recommendations only offer basic protection, i.e. ensure key protection only for a few months. To address this issue, as well as the need of different levels of protection depending on the application type, ECRYPT refines the recommendations on eight security levels (see Table 2.3).

Attacker	**Budget**	**Hardware**	**Key size**
Hacker	0	PC	53
	$400	FPGA	58
Small organization	$10k	FPGA	64
Medium organization	$300k	FPGA/ASIC	68
Large organization	$10M	FPGA/ASIC	78
Intelligence agency	$300M	ASIC	84

Table 2.2: Minimum symmetric key-size in bits for various attackers (2011) [OEFC11].

Finally, Lenstra et al. [LV01] recommend key lengths to be chosen as stated in Appendix B. The model is based on the following considerations:

- 56-bit Data Encryption Standard (DES) was considered secure by any reasonable meaning until 1982. Further, all the computations are related to the DES algorithm, for which $5 * 10^5$ Mips-Years[3] or using [$50 million, 2 days, 1980]-hardware[4] are necessary to break it;

- The effect of changes in the computational environment is modeled using Moore's law;

[2]Moore's law predicts that the computing power per chip doubles every 18 months, and thus the cost of any fixed attack effort drops by a factor of 2 every 18 months.

[3]A Mips-Year is defined as the amount of computation that can be performed in one year by a single DEC VAX 11/780. Half a million Mips-Years is roughly 13.500 months on a PC [LV01].

[4]The notation is adopted from Lenstra et al. [LV01] and denotes a 50 million dollar key searching machine that takes two days to brute-force a DES key in 1980.

Security Level	Security (bits)	Protection	Comment
1	32	Attacks in "real-time" by individuals	Only acceptable for auth. tag size
2	64	Very short-term protection against small organizations	Should not be used for confidentiality in new systems
3	72	Short-term protection against medium organizations, medium-term protection against small organizations	
4	80	Very short-term protection against agencies, long-term protection against small organizations	Smallest general-purpose level, 2-key 3DES restricted to 2^{40} plaintexts/ciphertexts, ≤ 4 years protection
5	96	Legacy standard level	2-key 3DES restricted to $\approx 10^6$ plaintext/ciphertexts, ≈ 10 years protection
6	112	Medium-term protection	3-key 3DES, ≈ 20 years protection
7	128	Long-term protection	Good, generic application-indep. recommendations, ≈ 30 years protection
8	256	"Foreseeable future"	Good protection against quantum computing unless Grover's algorithm applies

Table 2.3: Security levels [OEFC11]. "x" years protection should be considered starting with 2011, the year when the study was published.

- It takes ten years on average for an expected two-fold increase of budget;
- No efficient cryptanalytic attack is expected to be proposed in the near future.

A comparison of the aforediscussed recommendations for years 2011-2035 is illustrated in Figure 2.2. It can be noticed that the model of Lenstra et al. [LV01] is the only one to provide fine-grained year-dependent recommendations for key size selection. Moreover, the key lengths it proposes are remarkably lower than other recommendations discussed, such that the reader may be tempted to think that they do not provide sufficient security. In reality, the model of Lenstra et al. [LV01] tends to be conservative, i.e. the attack time estimates are considerably lower than the time required by hardware actually built and publicly announced. For example, the [$130.000, 4.6 days, 1998]-hardware obtained through the model for breaking a 56-bit key is about 24 times faster than the Deep Crack

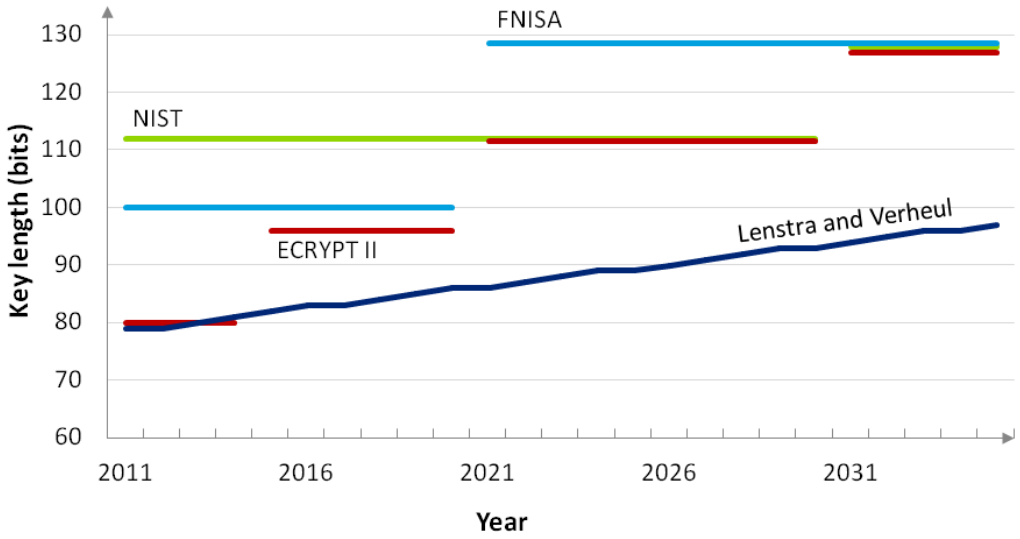

Figure 2.2: Key length recommendations provided in [LV01, OEFC11, FNI10, BR11].

machine physically created in the same year for breaking DES. Moreover, the [$10.000, 6.4 days, 2006] COPACOBANA machine brute-forces DES keys $\approx$ 100 times slower than suggested in [LV01]. However, this does not mean that better hardware cannot be build. It may be that given its recommendations, NSA already secretly possesses a fast hardware satisfying the estimates in the Lenstra et al. model.

Despite the significant value they add by guiding organizations into selecting sufficiently secure keys, the discussed key-length recommendations only take brute-force attacks into account. To address the lack of comprehensive models for estimating key renewal periods, we propose a risk-based methodology that considers the entire key management environment. This methodology is discussed in Chapter 4.

Chapter 3

Key Management

Cryptographic key management denotes the complete set of operations to nurture and sustain encrypted data and its associated keys [She11]. This chapter focuses on key management aspects. It includes an overview of the key management lifecycle and identifies key management stakeholders.

3.1 Key Management Lifecycle

The key management lifecycle defines the operations to be performed in order to create, maintain, protect and control the use of a cryptographic key [THA]. These operations are illustrated in Figure 3.1 and further discussed below.

3.1.1 Key Generation

Key generation is the process of creating keys for cryptography, either by mathematically processing the output of random number generators (RNGs) and other possible parameters, or based on one or more previously generated keys [BA01]. It must be performed such that it conforms to the key requirements specification and it ensures sufficient key entropy. The higher the entropy, the more unlikely is that adversaries will succeed in guessing the key.

For management purposes, it is common to bind each key generated with specific metadata elements, describing the appropriate management of the key. Example of metadata elements are: label, identifier, owner identity, lifecycle state, length, type, cryptographic algorithm using the key, scheme or modes of operation, product used to create the key, security policy applicable, access control list, usage count, parent key, sensitivity, date-times [BBCS10]. It is important to identify sensitive key metadata and to protect it along with the key.

Key generation techniques include [BA01]:

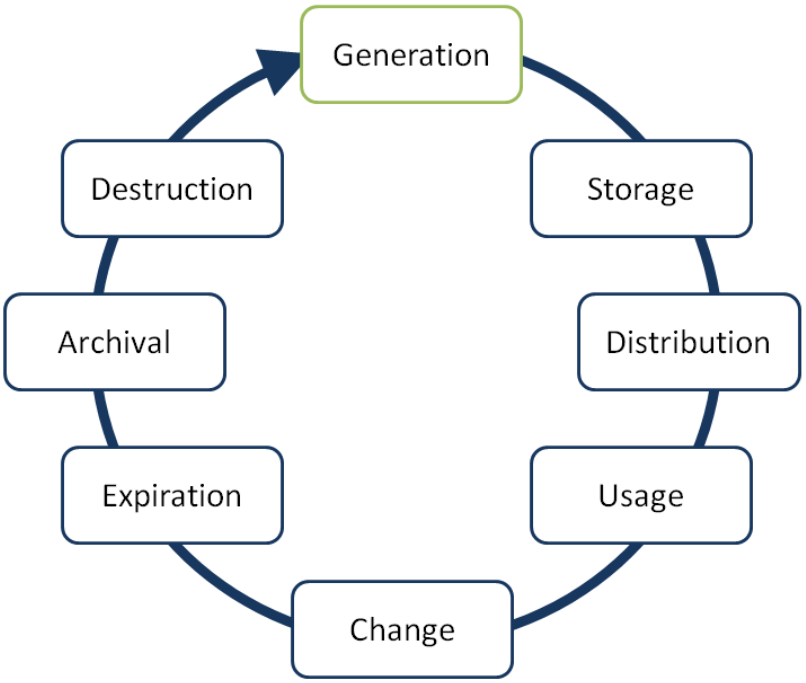

Figure 3.1: Key management lifecycle.

- **"Direct generation" of symmetric keys.** This technique makes direct use of the output of a RNG, e.g. generated random numbers could become new cryptographic keys.

- **Key derivation.** Obtaining keys based on a pseudo-random function and other information, some of which is secret (e.g. master key, password/passphrase) is referred in literature as key derivation. Note that key derivation from passwords is still a questionable practice, due to the usually reduced entropy of passwords [BA01].

- **Key update.** The key update technique is very similar to the key derivation previously discussed. However, the updated key and the old one are not mutually independent, but usually related i.e. the new secret is derived from the old key based on a key derivation function [BA01].

- **Key establishment through key agreement scheme.** Key agreement schemes are procedures in which the secret key is given by a function whose input consists in information gathered from participants [Kel]. The key is never transferred over communication channels and no one except for the involved parties should be able to derive it. Key generation and key distribution are accomplished within a single process. This makes key agreement schemes a practical solution to the widely-discussed key distribution problem.

According to a survey study performed by ENISA [ENI11] in 2012 to which thirteen European countries have participated, most e-government applications rely on Diffie-Hellman (DH) for key agreement. The security of DH depends on the discrete logarithm problem, which assumes that it is infeasible to calculate $K = g^{xy} \ mod \ p$ given a large prime p and the public parameters g^x and g^y [DH76]. Secure Shell (SSH), a network protocol for secure data communication, as well as Internet Protocol Security (IPSec), a protocol which incorporates secure communications into IP network layer itself, support the DH key agreement scheme to negotiate the session encryption key [Car01].

Key generation takes place inside a cryptographic module. Such a module denotes the set of hardware, software, firmware, or some combination thereof that implements cryptographic logic or processes [FIP94].

3.1.2 Key Storage

To preserve their secrecy, cryptographic keys must be stored in a location that is restricted to authorized entities and software, and from where they can be easily and quickly retrieved when needed. It is best practice to store keys separate from the location holding protected data.

Depending on the protection level required, operational storage solutions can be:

- **Regular devices**, e.g. tapes, disks, flash disks, removable hard drives, etc.;
- **Secure dedicated devices**, e.g. HSMs like: smartcards, PCI cards, USB tokens, etc.

To facilitate key recovery in case of loss or unintentional destruction, it is best practice that keys are replicated and stored on backups [Net]. Except for the storage devices previously mentioned, backup keys can be stored as **hard-copies** (e.g. printed media) as well. We want to stress that replicating keys intensively might determine an increase of the key exposure risk. The reason behind stands in the strength of the backup system being reduced to the strength of the weakest backup storage.

3.1.3 Key Distribution

Key distribution refers to a protocol in which a secret key is transported from an entity who either generates or owns the key to another entity intended to use the key [BR11].

When individuals cannot be trusted with secrets, practitioners may prefer to first share the key using secret sharing schemes and further distribute the key shares. A secret sharing scheme is a method to split a key among trusted entities in such a way that at least k out of n entities need to cooperate in order reconstruct it. It is often referred in literature as a *(k,n)* threshold scheme. An example of a trivial *(n,n)* sharing scheme is based on performing XOR operations. More specifically, shares are calculated such that XOR-ing all

of them constructs the secret. Another well-known scheme is the Shamir's secret sharing [Sha79], based on polynomial interpolation.

The challenge behind key distribution is to exchange the key or its shares without allowing a possible eavesdropper to gain any non-negligible information that may partially or completely disclose it. Ideally, keys are protected end-to-end while they are distributed between systems, no media vulnerability allowing for key disclosure.

Key distribution can be manually, preferably as at least two separate key shares using different communication channels, or electronically, recommended in encrypted form [VIS08].

Manual Key Distribution

Manual distribution techniques are the simplest to communicate keys [Opp01]. Keys are exchanged between parties by use of a courier or some other physical means. Before being transported, keys and metadata must be bound out of a cryptographic module, the process being referred in literature as key output. Similarly, they need to be entered into the destination cryptographic module, process known as "key entry". Key entry can be done either manually by typing it into a terminal, external pad or similar devices or automatically by importing the key from a transport device.

Despite their simplicity, manual key distribution techniques are expensive, time consuming and present scalability problems which make them suitable only when the number of communication peers is reduced. Therefore, when possible, electronic key distribution is preferred to manual distribution [PM02].

Electronic Key Distribution

Nowadays most keys are distributed using a protocol via an automated process, relieving individuals from this task [Har09]. Referred as electronic key distribution, the protocol makes use of an online mechanism to transport keys between entities. However, the key needs protection for exposure during transmission. It generally implies prior distribution of a symmetric key wrapping key or a public key transport key, either between the two entities or between each of them and a key distribution center. This leads to a concept of a key hierarchy, where the top-level or master-key has no key to protect it [PM02]. Any public key encryption scheme can be used as key exchange protocol [ENI11]. The secrecy of the communicated key is then reduced to the secrecy of the key used to extract it.

Special attention must be given to **quantum key distribution (QKD) protocols** which rather than relying on the intractability of a mathematical property, they rely on the following fundamental laws of quantum mechanics [ENIa]:

- It is physically impossible to create a copy of an unknown quantum state (this statement is known as the non-cloning theorem).
- It is impossible to distinguish with certainty two non-orthogonal quantum states, without destroying information.

Given the assumption that the properties hold, QKD is unconditionally secure, i.e. is secure even in the presence of an adversary with unlimited computational power [PPS04]. These type of protocols can be used to exchange very long secret keys in such a way that no eavesdropping could be successful. Therefore, they can be used to generate keys having the same length as the message to be protected, allowing for the use of the one-time-pad (OTP) encryption scheme.

However, portraying QKD as the ideal solution to the distribution problem tends to be exaggerated. This is because there is still the need for an authentic channel to be used by the communicating parties, otherwise man-in-the-middle attacks becoming possible [LWW+10]. Only if the authentication mechanism and the protocol used are unconditionally secure, is the key exchange system unconditionally secure. Examples of key distribution protocols include: BB84 [BB84] (the first quantum key distribution protocol), GP00 [GP00] and B92 [Ben92].

Today, QKD is no longer confined to laboratories, but it is commercially available. Nevertheless, due to its high cost, the main area of application is very high assurance systems, particularly related to national security and government use-cases [Hog09]. In addition, it is not possible over unlimited distances[1], reducing its applicability considerably.

3.1.4 Key Usage

A key which evolved through all the phases discussed up to this moment is considered to be ready for use, i.e. cryptographic operations are to be performed within a cryptographic module. Four types of cryptographic modules are identified by NIST [FIP01]:

- **Software modules**: The cryptographic mechanism is implemented at the software-level and runs in a modifiable operational environment (e.g. normal PC/server, etc.).
- **Firmware modules**: The cryptographic boundary of the module delimits the firmware solely components that execute in a limited or non-modifiable environment.
- **Hardware modules**: The cryptographic boundary of the module is specified at the hardware perimeter. Firmware may be included in the cryptographic boundary too.
- **Hybrid modules**: The cryptographic boundary delimits the composite of a software or firmware component and a disjoint hardware component.

To avoid key disclosure while in use, it is best practice to perform all cryptographic operations in a physically/logically isolated, dedicated environment offered by HSMs. It is also recommended to test the new key for a pre-determined time interval, ensuring that it operates successfully.

[1] In 2009 quantum key exchange was demonstrated up to 250 km in optical fibers [SWV+09].

3.1.5 Key Change

Whenever there is a concern that the key has been compromised or it is likely to get exposed due to the long period it has been used, a new key should be deployed. Key change is concerned with activating a previously deployed key and, if required, providing utility to re-key sensitive data protected by the previous one.

Unless keys have not leaked as a result of security incidents, companies need to consider renewing them after a predefined usage time interval. In fact, this interval is exactly the key renewal interval discussed in this thesis.

3.1.6 Key Expiration, Archival and Destruction

Assuming a new key is in place to be used, the old key can be deactivated. An expired key is not used anymore to apply cryptographic protection to information, but it may be used to process already protected information. In some cases, it may be needed to archive keys before their scheduled end of life, e.g., businesses with data retention requirements must do so in order to allow for decryption of legacy data. Note that some cryptographic devices automatically archive expired keys in a secure way. Archived keys can be stored on same type of media as key backups.

When not used anymore for processing cryptographically protected information, the key can be finally destroyed.

To support secure management of cryptographic keys, a Cryptographic Key Management System (CKMS) consisting of security policies, procedures and assets used to protect and manage cryptographic keys and bound metadata [BBCS10] should be designed and properly developed. Each of the key management system operations should be performed only by authorized individuals, referred in literature as key stakeholders. The following section identifies the main key management stakeholders and briefly describes their roles and attributions.

3.2 Key Management Stakeholders

In the key management context, a stakeholder denotes any person or entity playing a role in the key management system. Various key stakeholders interact differently with the cryptographic key, depending on the roles they have. In [BBCS10] several stakeholder categories have been identified, including:

- **System authority**: Executive-level management responsible for ensuring that key management operations are securely performed.
- **System administrator**: Entity responsible for performing related management of a cryptographic management system other than its keys. In addition, the system administrator establishes the personnel authorized to use the cryptographic module.

- **Cryptographic officer**: Entity responsible for performing key management operations and cryptographic initialization on the cyrptographic module, e.g. initiate key generation process.
- **Key custodian**: Entity responsible for distributing and loading key or secret shares into the cryptographic key management system.
- **Key owner**: Entity authorized to share and use the key and whose identity is associated with the cryptographic key.
- **System user**: Users authorized to use the cryptographic key when required by an application.

Note that the list above is not meant to be comprehensive but only highlights the main stakeholder categories. Key recovery agents, audit administrators and domain authorities are other examples of roles that may require access to cryptographic keys and/or metadata bound to cryptographic keys.

Chapter 4

Methodology for Key Renewal Period Estimation

This chapter presents the key renewal period estimation methodology proposed in this thesis. We identify and discuss the activities necessary for preparing, conducting and maintaining the estimation of key renewal intervals.

The methodology we consider is based on key exposure risk assessment, the process of identifying, prioritizing, and estimating risks to cryptographic keys [GSF02]. Key renewal intervals are estimated based on the identified risks and when applicable, proposed as a response to them.

Figure 4.1 summarizes the risk assessment methodology as proposed by NIST. It comprises three general phases [NIS11b]: **prepare**, **conduct** and **maintain**. The preparation phase is concerned with establishing a context for the key exposure risk analysis. Questions like "What is to be assessed?", "How is the assessment to be performed?", "Where input information is obtained from?" are to be answered before the risk assessment to effectively start. During the conduct phase the risk of key being disclosed as a result of insecure key management processes is assessed through a five-step process. Finally, changes in the key usage environment or technological improvement increasing the attacker's capabilities may alter the previously identified risks, making them inconsistent with reality. Therefore, risk analysis should be maintained on an ongoing basis and changes should be reflected in the key renewal period.

The risk assessment results are fed as input to the risk response process, consisting in: **control identification**, **evaluation of alternatives**, **control decision** and **control implementation** [NIS11a]. Key renewal intervals are estimated as part of the control identification step. Alternatives like strengthening the key management environment are to be considered as well. It is the responsibility of the organization to decide which risks are to be treated and what controls are to be considered. Control decision is mainly based on the risk tolerance of the organization as well as the financial resources it is willing to

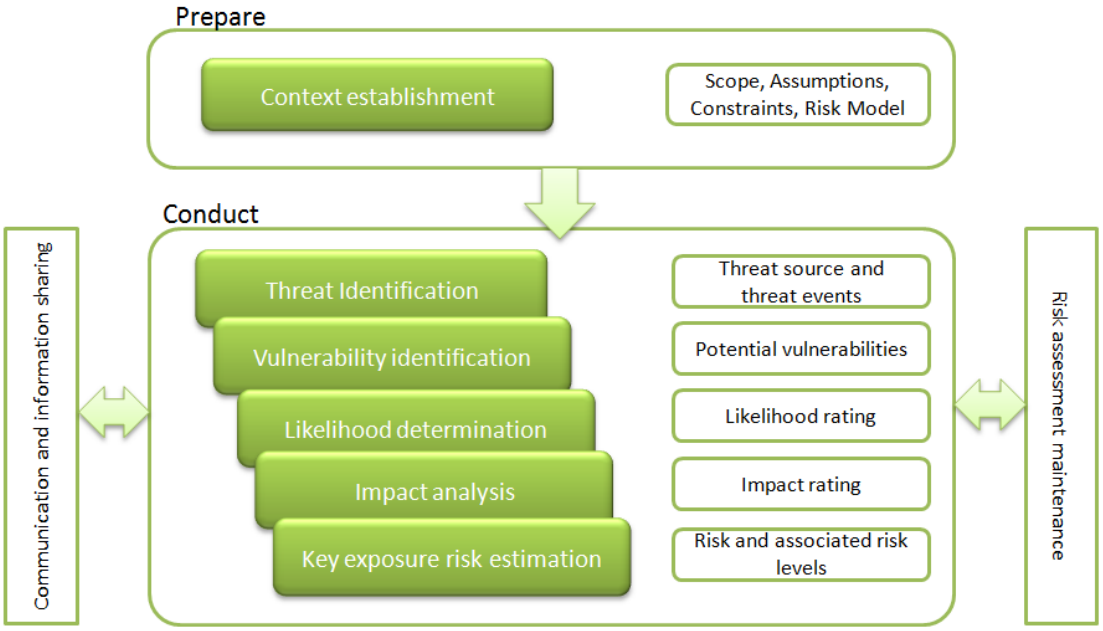

Figure 4.1: Risk assessment process [NIS11b].

invest in. In order to make effective use of recommendations, security policies should be accordingly refined and the key renewal interval as well as other identified controls should be enforced in practice.

To valorize the benefits they bring, risk assessment and response should be performed within the context of an integral risk management process. Their interrelation with other risk management components, **risk framing** and **risk monitoring** is illustrated in Figure 4.2. Risk framing has the role to inform other components with respect to the risk management strategy applied by the organization. It describes practices in assessing, responding and monitoring risks [NIS11a]. Changes in the technological and key management environment that may degrade the accuracy of previously estimated key renewal intervals are recorded by the risk monitoring component and are to be further communicated to the risk assessment component. Readers familiar with the ISO 27005 standard for information security risk management might have already noticed the intense similarities between the "plan-do-check-act" (PDCA) cycle (see Annex A) in [Sta08] and the risk management process of NIST considered in this thesis. Our choice for the last one is mainly motivated by the extensive suite of key management guidelines NIST provides, which we further use as a theoretical foundation for the thesis. Adopting the risk management process proposed by NIST allows us to preserve consistent terminology.

The guidance provided in this thesis focuses on the key exposure risk assessment and response components. However, organizations should ensure that a comprehensive and clear risk frame is available to support the processes. Moreover, companies are advised to keep

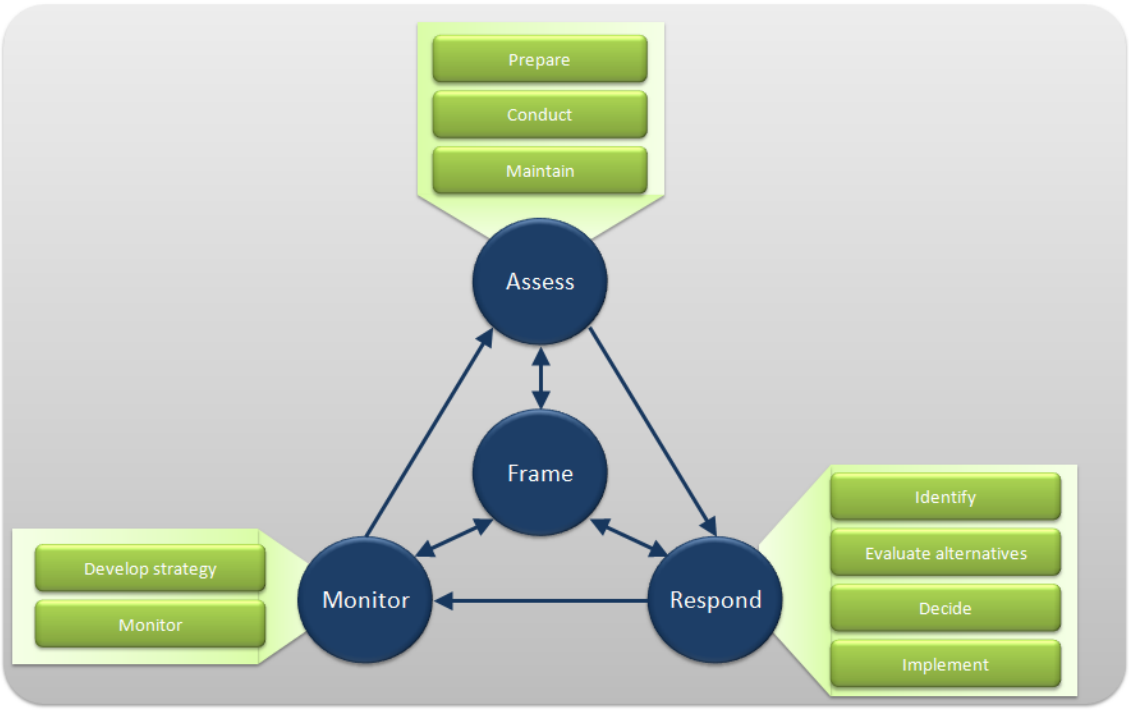

Figure 4.2: Risk management process [NIS11a].

an eye on the risk factors to spot potential fluctuations. We further proceed by analyzing each of the risk assessment and response steps mentioned before.

4.1 Context Establishment

Context establishment is the preparatory step of the risk assessment, defining the purpose, scope, risk tolerance, assumptions and constraints, sources of information to be used as input, the risk model and assessment approach to be applied. We discuss all these tasks in the current section.

4.1.1 Purpose and Scope

The purpose of the risk assessment depends on whether it is initially performed or it is an updated assessment due to potential changes. Initial risk assessments have the purpose to identify potential key exposure risks, further input to the key renewal interval estimation process. Later risk assessments have the purpose to identify changes in the key exposure risk factors and to reflect them in the key renewal period.

The scope determines the boundary for risk assessment and identifies cryptographic keys to which renewal interval decisions apply. Time horizons for the assessed risks are to be defined as well.

4.1.2 Assumptions and Constraints

Assumptions and constraints under which the assessment becomes effective need to be clearly stated. Rather than listing all the assumptions here, for the key exposure risk assessment in this thesis we state assumptions gradually, while analyzing each of the assessment stages. Constraints are mainly represented by resources, knowledge and expertise available for performing the assessment. They depend on the organization in scope, mainly determined by the budget the organization is willing to allocate for assessing key exposure risks. The less constrained the assessment process is, the more accurate results are expected to be.

4.1.3 Information Sources

Both internal and external information sources should be considered for getting insight into specific threats to organization. Internal information can be extracted from incident reports, security logs, monitoring results, vulnerability assessments, etc. External information can originate from research centers, security service providers, etc. [NIS11b].

4.1.4 Risk Model and Risk Assessment Approach

The risk model applied in this thesis is illustrated in Figure 4.3. It distinguishes a set of elements (*threat sources, threat events, vulnerabilities, likelihood of threat occurrence, impact* and *risk*) to be evaluated during the "conduct" phase of the risk assessment (see Figure 4.1) and pictures the relationships among them. We define and investigate each of these elements in later sections of the current chapter.

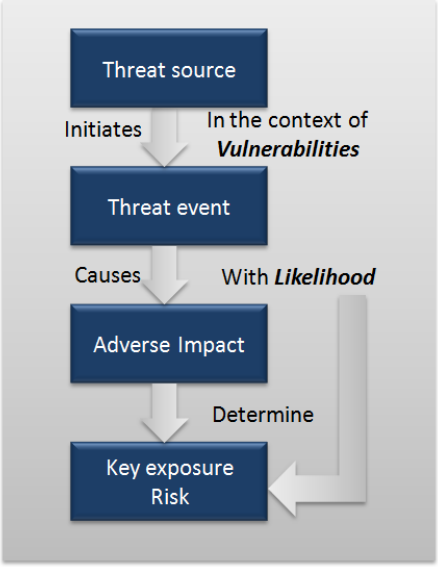

Figure 4.3: Risk model.

To assess risk and its factors three types of approaches (quantitative, qualitative and semi-quantitative) have been presented in literature. Qualitative approaches evaluate risk and risk factors using non-numeric scales, e.g. low, medium high. They bring the main advantage of being relatively easy to understand and to apply by all relevant personnel [Sta08]. However, unless the rating scale is clearly defined or illustrated with appropriate examples, the assessment results may vary depending on evaluators [NIS11b]. The range of values is also generally small, making a fine-grained comparison between identified risks a difficult task [NIS11b].

Quantitative assessment estimate risk and risk factors using numerical values, allowing for manipulation of information in consistent and reproducible ways. Large range of values are supported as well. Although a quantitative risk assessment approach can greatly simplify decision making and provide accurate final results, it can be difficult to apply due to lack of agreed metrics of vulnerabilities and accurate measures [EYZ10].

Semi-quantitative approaches combine the benefits of qualitative and quantitative analysis. Risks are estimated using numerical bins or scales, while qualitative equivalents are provided as well. Several standards propose the use of semi-quantitative approaches for risk assessment, including [Sta08], [GSF02], [NIS11b] and [ENIb]. Some of these standards recommend classifying risk and risk factors on five levels (e.g. Very low (0-4) , Low (5-20), Medium (21-79), High (80-95) and Very high (96-100) [GSF02]). In practice, such a scale tends to raise problems to practitioners, since the boundary between adjacent levels remains slightly uncertain. Therefore, three-level scales (e.g. Low, Medium, High) are often preferred. This motivates our choice for further assessing the key exposure risk based on the semi-quantitative approach proposed by NIST in [NIS11b]. The likelihood, impact and risk factors scales are described in detail in Sections 4.4, 4.5 and respectively 4.6.

4.2 Threat Identification

"Know the enemy, and know yourself, and in a hundred battles you will never be in peril", S.Tzu [Tzu88]

Threat identification attempts to determine both types of threat sources and threat events that may harm the organization by exposing the cryptographic key. *Threat sources* are actors with the intent targeted at the exploitation of a vulnerability or that may accidentally exploit a vulnerability [NIS11b]. *Threat events* (or simply *threats*) denote events or situations initiated or caused by a threat source that cause adverse impact [NIS11b]. Guidance for threat and threat source identification as part of the key disclosure risk assessment is provided in this section.

4.2.1 Threat Source Analysis

Threat sources may be of environmental or human nature. Environmental threats are usually cause of business disruption, being less relevant to key disclosure. The advances in

technology and the access by the public to specialized tools and knowledge make humans potentially dangerous threat sources, either deliberate or accidental. We expect that accidental threat sources do not directly lead to key exposure, but make it easier for adversaries to find out the key. Therefore, in this thesis we mainly focus on deliberate threat sources. Often referred in literature as *attackers* or *adversaries*, deliberate threat sources are driven by malicious goals. *Goals* reflect the intentional desire of adversaries, here the compromise of cryptographic key confidentiality or shortly key disclosure, without specifying how goals are to be satisfied [EYZ10].

Clearly defining the threat source from whom the key and metadata bound to it need to be protected is of major importance for estimating key disclosure risks. For example, it is often easier for employees to get access to cryptographic modules inside the company than for outsiders required to bypass access controls to enter the organization's building. Threat sources possessing high-computational power devices have also more chances to succeed in brute-forcing a key than those using an off-the-shelf PC.

Several taxonomies for adversarial threat sources have been proposed in literature during the time, some of them being summarized in Table 4.1. Note that we selected taxonomies originating from widely recognized organizations and experts in the field. We also included the taxonomy used in [VX10] due to its quantitative approach. Although they allow for fine-grained threat source description, these taxonomies are rather difficult to use in practice. Most companies would not even be able to differentiate between adjacent adversary classes and therefore assessing the key exposure risk for such threat source categories would not add a significant value. Moreover, some of the taxonomies do not distinguish between insiders and outsiders, a major differentiator factor when assessing the key exposure risk.

Author	Approach	Criteria for threat source classification
ISO [Sta08]	Qualitative	Motivation. Five threat source categories identified: Hacker/cracker, computer criminal, terrorist, industrial espionage, insiders.
Schneier [Sys99]	Qualitative	Motivation, access, skills, risk aversion and funding.
NIST [NIS11b]	Semi-quantitative	Capability, intent and targeting.
Vavoulas et al. [VX10]	Quantitative	Determent to cost, attack difficulty and detectability.

Table 4.1: Threat source taxonomies.

Abraham et al. [ADDS91] come up with a simpler classification, dividing threats into clever outsiders, knowledgeable insiders and funded organizations (see Table 4.2). The classification addresses the issues previously discussed such that we consider it suitable for the key exposure risk assessment. We also assume that all stakeholders except for resource users are honest and have key and bound metadata protection as common goal. Though, we do not exclude the possibility of accidental errors caused by stakeholders to occur.

Class	Threat source characteristics
Clever outsiders	Intelligent attackers, with limited knowledge of the system and access to only moderately sophisticated equipment. They often try to take advantage of existing vulnerabilities in the system, rather than creating one.
Knowledgeable insiders	Experienced attackers, usually dissatisfied or unhappy employees or contracts, with significant specialized technical education and experience and a decent knowledge of the product or system. They often possess sophisticated tools and equipment for analysis.
Funded organizations	Specialized attackers, with high technical skills and funding resources. They are capable of extensive analysis of the system and sophisticated attacks, and possess advanced analysis tools.

Table 4.2: Threat classification [ADDS91].

4.2.2 Threat Events Analysis

Often referred as *attacks*, threat events deliberately initiated consist of a set of actions that attempt to help adversaries in reaching their malicious goal, here key disclosure. To identify these actions, practitioners can take advantage of existing graphic representations of security threats. We further proceed by using attack trees for key exposure threat modeling.

Threat Modeling Using Attack Trees

Attack trees have been introduced by Schneier in [Sys99] as a methodical way of describing the security of systems. They define an attack suite through a node hierarchy, having goal as a root node. When multiple goals exist, one tree for each of them should be built. Child nodes refine an abstract goal into a more concrete set of sub-goals [MO05]. The non-refined nodes, leafs in the tree, are basic attacks. The set of leaf nodes, in which only the occurrence of all the leaf nodes could reach the attack goal is further referred as *attack scenario* [RDZ11].

Initially, goal refinement was performed in a conjunctive and/or disjunctive way. Conjunctive refinement links child nodes through an *AND* connector, meaning that all the sub-goals need to be satisfied. Disjunctive refinement connects child nodes through an *OR* operation, requiring at least one of the sub-goals to be achieved by the adversary. Over time, more advanced node connectors have been proposed in literature. One of them which is of interest for the key exposure risk analysis is the threshold based connector. It determines the minimum number of accomplished child nodes for success of the parent node, e.g., K-out-of-N (K/N) [JWP11].

Figure 4.4 illustrates the attack tree concept through an example widely encountered in literature. The goal is opening the safe and can be achieved by picking the lock, learning the combination, cutting the safe, or installing the safe improperly such that it can be easily

opened. Each of these attack ways becomes a sub-goal and when applicable, are further refined. The reader may have noticed that all sub-goals except for the "eavesdrop" one are refined in a disjunctive manner.

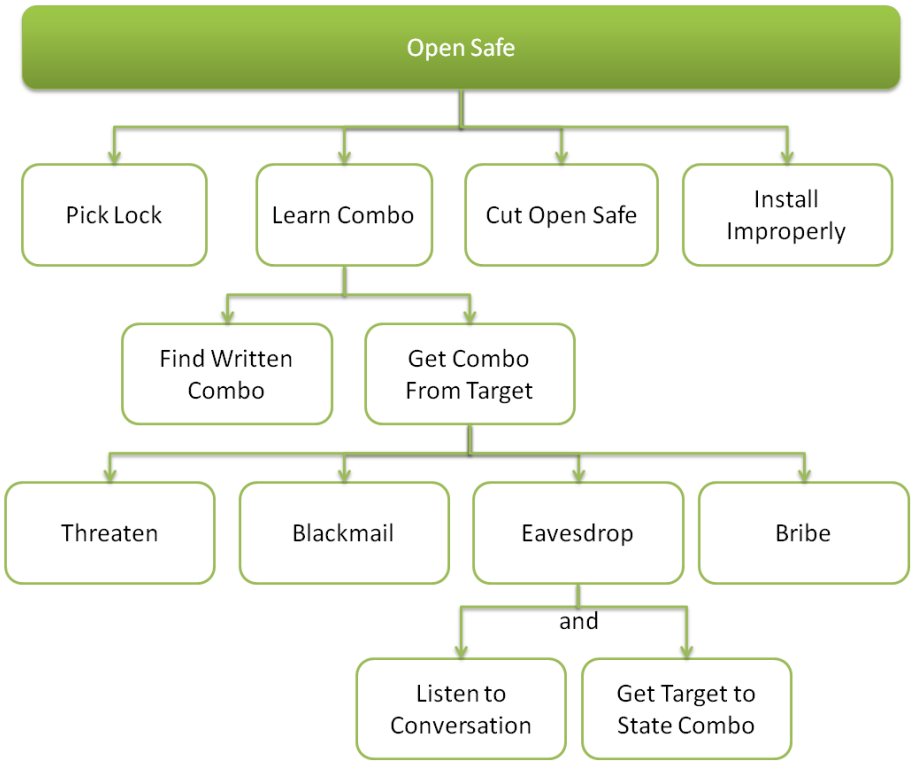

Figure 4.4: Example attack tree [Sys99].

For large attack trees, the graphical representation may become cumbersome and therefore textual representations might be preferred. The graph in Figure 4.4 is textually represented as:

```
GOAL: Open Safe
OR  1. Pick lock;
    2. Learn combo;
    OR  2.1 Find written combo;
        2.2 Get combo from target;
            OR  2.2.1 Threaten;
                2.2.2 Blackmail;
                2.2.3 Eavesdrop;
                AND 2.2.3.1 Listen to conversation;
                    2.2.3.2 Get target to state combo;
                2.2.4 Bribe;
    3. Cut open safe;
    4. Install improperly;
```

Note that the tree is not complete, but is only meant to enhance the reader's understanding of attack trees. Trying all the possible combinations is only another type of attack the tree can be expanded with.

Attack trees not only bring the advantage of allowing for better identification of the attack landscape, but support the evaluation of attack scenarios against several criteria as well. Three nodes can be augmented with boolean attributes, e.g. "possible/impossible", "easy/difficult", "intrusive/non-intrusive", "legal/ilegal", etc. Continuous values, e.g. attack cost, likelihood, can be also applied. Values for attributes are associated starting with leaf nodes. For non-leaf nodes, attribute values are derived based on the values of the children.

Key Disclosure Attack Trees

Adversaries might take advantage of security flaws at any of the key management stages. Common sense predicts that attacks targeting keys that are currently or have been already replaced are less expected to occur. Therefore, we further focus on the compromise of key generation process, key operational and backup storage, key distribution process and key usage, as illustrated in Table 4.3. Note that when key secret sharing is enforced, key exposure attacks may target different key management stages for disclosing different secret shares. For example, assuming a (2,2) secret sharing scheme, adversaries may target the distribution process to obtain one of the secret shares and obtain the other secret share during its usage.

Type	Threat event
Key disclosure	Compromise key during key generation process.
	Compromise key operational or backup storage.
	Compromise key during key distribution process.
	Compromise key during usage.

Table 4.3: General threat events.

Having these said, attack trees modeling key disclosure attacks can be built around the following rules:

1. **The attack goal is key disclosure**, which becomes the root of the tree;
2. **The root node is refined as a set of sub-goals**. Assuming no secret sharing is enforced, the sub-goals consist in compromising relevant key management processes (see Table 4.3), in order to extract the cryptographic key. The key disclosure attack tree containing the goal and first-level sub-goals is textually represented below:

```
GOAL: Key disclosure
OR  1. Compromise the key generation process;
    2. Compromise the key storage;
```

```
3. Compromise the key distribution process;
5. Compromise the key usage;
```

When key secret sharing is enforced, in addition to compromising those key management processes which handle the unshared key, the sub-goals include compromising the minimum number of key shares required as well.

For example, assuming:

- (2,2)-threshold key secret sharing enforced and
- Key generation and backup storage being the only management processes performed on the unshared key,

the attack tree goal and sub-goals become:

```
GOAL: Key disclosure
OR  1. Compromise the key generation process;
    2. Compromise key shares;
        AND 2.1 Compromise key share S1;
            OR  2.1.1 Compromise the key share distribution process;
                2.1.2 Compromise the key share operational storage;
                2.1.3 Compromise the key share usage;
            2.2 Compromise key share S2;
            OR  2.2.1 Compromise the key share distribution process;
                2.2.2 Compromise the key share operational storage;
                2.2.3 Compromise the key share usage;
    3. Compromise the key backup storage;
```

3. **Each sub-goal is further refined** depending on the key management practices within the organization and the threat source considered. For example, assuming a hard-copy of a cryptographic key being stored in a safe located inside a company, the compromise of the key backup storage can be refined as suggested below:

```
GOAL: Key disclosure
OR  1. Compromise the key generation process;
    2. Compromise the key operational storage;
    3. Compromise the key distribution process;
    4. Compromise the key backup storage;
        AND 4.1 Gain unauthorized physical access to safe;
            AND 4.1.1 Unlock door with key;
                OR  4.1.1.1 Steal key;
```

```
                    4.1.1.2 Social engineering;
                4.1.2 Pick lock;
                4.1.3 Break window;
                4.1.4 Follow authorized individual into building;
            4.2 Open safe;
            OR  4.2.1 Pick lock;
                4.2.2 Learn combo;
                OR  4.2.2.1 Find written combo;
                    4.2.2.2 Get combo from target;
                    OR  .1 Threaten;
                        .2 Blackmail;
                        .3 Eavesdrop;
                        AND .1 Listen to conversation;
                            .2 Get target to state combo;
                        .4 Bribe;
                4.2.3 Cut open safe;
                4.2.4 Install improperly;
    5. Compromise the key usage;
```

The refinement of the "Gain unauthorized physical access to safe" sub-goal was adopted from Byres et al. [BFM04].

At the end of this step practitioners are expected to have built the key exposure attack tree that reflects the company's key management environment.

4.3 Vulnerability Identification

Vulnerabilities are weaknesses in an information system, system security procedures, internal controls, or implementation that could be exploited by a threat source [NIS11b]. They can be encountered at each step in the key management process. Several on-line databases are available to practitioners for identifying system vulnerabilities, e.g., the National Vulnerability Database [NVD], Common Weaknesses Enumeration [CWE], Vulnerability Notes Database [CER], SANS top 20 annual security risks [SAN] and Secunia Advisories [Sec]. Their benefits are irrefragable, though inexperienced practitioners may feel overpowered browsing them without initial guidance. In addition, their focus falls on software security, without alerting about unrecommended key management security practices.

To support practitioners in determining vulnerabilities in the key management environment, Table 4.4 enumerates a set of weak points per key management process attackers can take advantage of. Chapter 5 illustrates with examples each of the items in the list. Note that we do not claim the completeness of the list above. However, we strongly believe

that most critical key management vulnerabilities have been identified, making the risk assessment effective.

Key management process	Examples of vulnerabilities
Key generation	Insecure cryptographic module for key generation. Insecure use of cryptographic module for key generation. Insecure key generation scheme. Insecure use of key generation scheme.
Key storage	Insecure storage modules/media. Insecure use of storage module/media.
Key distribution	Insecure key transport module/media. Insecure use of key transport module/media. Insecure transport channel. Insecure key entry mechanism. Insecure use of key entry mechanism. Insecure key secret sharing scheme. Insecure use of key secret sharing scheme.
Key usage	Insecure cryptographic mechanism. Insecure cryptographic module. Insecure use of cryptographic module.

Table 4.4: Key management vulnerabilities.

Exploiting the vulnerabilities is conditioned by the attack vector required, i.e. how close to the target adversaries need to be to perform the attack. The Common Vulnerability Scoring System (CVSS) [PM07] proposes three attack vectors:

- **Local**: Exploiting the vulnerability requires either physical access to the target or a local (shell) account on the target;
- **Adjacent Network**: Exploiting the vulnerability requires access to the local network of the target;
- **Network**: The vulnerability is exploitable from remote networks.

Depending on the attack vector, adversaries not granted the needed access level to the target can attempt to exploit vulnerabilities listed in Table 4.5.

We want to stress that undetected vulnerabilities might lead to undetected key exposure attack opportunities for adversaries. Therefore, identifying a comprehensive set of vulnerabilities that characterizes the key management environment in scope is of major importance for the risk assessment.

Type	Examples of vulnerabilities
Access control	Insecure operating environment. Insecure networks. Insufficient employee training and awareness.

Table 4.5: Potential access control vulnerabilities.

4.4 Likelihood Determination

Likelihood of threat event occurrence is a contributing factor to estimating risks. In this section we investigate likelihood determination methods that make use of the attack tree structure discussed in Section 4.2.2.

One way of determining the likelihood of threat event occurrence consists in:

1. Extracting all attack scenarios from the attack tree.
2. Evaluating how likely is that each of the identified attack scenarios will occur.
3. Obtaining the likelihood of threat event occurrence as a function of previously assessed likelihoods.

Unless automated or applied to small attack trees, this approach implies a tremendous work to be performed and is therefore less preferred. For the likelihood determination in this thesis we propose the use of a more modular approach. The approach takes full advantage of the attack tree structure, it is scalable and does not require practitioners to put extreme effort into using it. The approach is built up using the following steps:

1. **Determine the likelihood for each attack in the attack tree.**

   Recall that attacks are leafs in the attack tree. To estimate their likelihood, practitioners should account for capabilities of the threat source initiating the attack, the ease of vulnerability exploitation and the security countermeasures in place. Next, the rating scale in Table 4.6 can be used to assign semi-quantitative values to likelihoods.

Rating	Likelihood definition
High (1)	The threat-source is highly motivated and sufficiently capable, and controls to prevent the attack from being exercised are ineffective.
Medium (0.5)	The threat-source is motivated and capable, but controls are in place that may impede successful exercise of the attack.
Low (0.1)	The threat-source lacks motivation or capability, or controls are in place to prevent, or at least significantly impede, the attack from being exercised.

Table 4.6: Non-adversarial attack likelihood scale [NIS11b]

2. **Aggregate likelihoods over the constructed attack tree.**

   Starting from the leafs and moving toward the root, likelihoods are aggregated depending on the connector used. In this thesis we present a conservative approach, defining conjunctions and disjunctions as shown in Table 4.5 and respectively Table 4.6. Such an aggregation technique follows the principles of three-valued logic, firstly addressed by Lukasiewicz in the early 20th-century. Lukasiewicz states that propositions are not only either true or false but they can have an intermediary truth-value [DH01]. A rough mapping between the likelihood scale in Table 4.6 and the Lukasiewicz' ternary logic can be envisioned by analyzing the statement: "It is likely that an attack will be successfully exercised by a threat source". Evaluating the clause to false indicates a low likelihood. Similarly, if the statement is true, a high likelihood is expected. A medium likelihood can be mapped to the intermediary truth-value introduced by Lukasiewicz.

   We want to stress that we do not expect everyone to agree with our proposal for aggregating likelihoods. Some organizations may find it inappropriate, preferring a more relaxed model. Practitioners are encouraged to opt for techniques that best fit their organization's expectations.

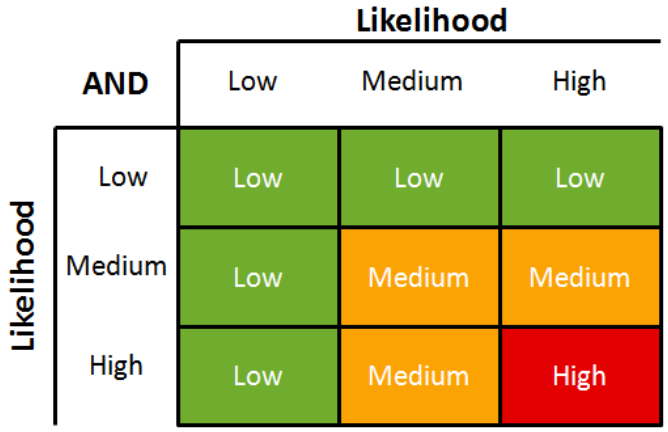

Figure 4.5: P AND Q = $min\{P, Q\}$ (here P,Q denote likelihoods and are Low, Medium, High).

Except for the AND and OR operators discussed above, K/N connectors are also practical for building key exposure attack tree, as discussed in Section 4.2.2. We write $K/N : P = \{P_1, P_2, ...P_N\}$ to be read as "at least K out of N P_i values should be true". Boolean algebra allows us to express K/N connectors using AND and OR operators, as stated by equations 4.1, 4.2 and 4.3. Note that to increase the readability of the equations we use the "$\wedge$" symbol to represent the AND connector and "$\vee$" for the OR operator.

$$K/N : P = \{P_1, P_2, ...P_N\} \quad \equiv \quad S_1 \vee S_2 \vee ... \vee S_{\binom{N}{K}}, \tag{4.1}$$

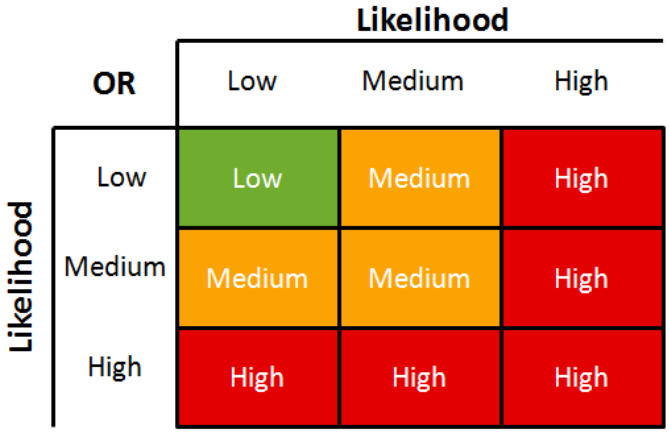

Figure 4.6: P OR Q = $max\{P, Q\}$ (here P,Q denote likelihoods and are Low, Medium, High).

where:

$$S_i = P_{i_1} \wedge P_{i_2} \wedge ... \wedge P_{i_K}, P_{i_j} \in P, \tag{4.2}$$

and

$$(P_{i_1}, P_{i_2}, ..., P_{i_K}) \neq (P_{j_1}, P_{j_2}, ..., P_{j_K}), \forall i \neq j. \tag{4.3}$$

After performing the two steps, each node in the attack tree should have associated a Low, Medium or High likelihood value. The likelihood of each threat event identified in Section 4.2.2 is to be further used to determine the generated risk.

To exemplify the likelihood determination process we consider the hard-copy key backup example introduced in Section 4.2.2. In addition, we assume that: employees are generally aware of the need to protect assets, but are poorly trained with respect to social engineering attacks; the safe storing the hard-copy of the key is difficult to break. Given this information, attack tree nodes could be assigned likelihoods as illustrated in Table 4.7.

4.5 Impact Analysis

The level of impact is the magnitude of harm that a threat event can cause to an organization [NIS11b]. It is a determining factor in estimating risks, along with the likelihood of threat event occurrence. In [BR11], NIST identifies three differentiator factors for impact evaluation:

- **The sensitivity of the information protected with the key.** It is determined by the consequences of its disclosure, including reputation damage, legal fines and penalties. Intuitively, the more information is protected with the cryptographic key, the higher the consequences are expected to be.

Attack tree nodes	Likelihood
GOAL: Key disclosure	
OR 1. Compromise key generation process;	
2. Compromise key operational storage;	
3. Compromise key distribution process;	
4. Compromise key backup storage;	High
AND 4.1 Gain unauthorized physical access to safe;	High
OR 4.1.1 Unlock door with key;	High
OR 4.1.1.1 Steal key;	Low
4.1.2.1 Social engineering;	High
4.1.2 Pick lock;	Medium
4.1.3 Break window;	Low
4.1.4 Follow authorized individual into building;	High
4.2 Open safe;	Medium
OR 4.2.1 Pick lock;	Medium
4.2.2 Learn combo;	Low
OR 4.2.2.1 Find written combo;	Low
4.2.2.2 Get combo from target;	Medium
OR .1 Threaten;	Low
.2 Blackmail;	Low
.3 Eavesdrop;	Medium
AND .1 Listen to conversation;	Medium
.2 Get target to state combo;	High
.4 Bribe;	Low
4.2.3 Cut open safe;	Low
4.2.4 Install improperly;	Low
5. Compromise key key usage;	

Table 4.7: An example of likelihood determination. Note that we only assign likelihoods for the "Compromise key backup storage" sub-tree for which the key environment was assumed.

- **The criticality of the processes protected with the key.** The operational damage suffered by the organization increases with the criticality of compromised processes.

- **The recovery cost.** It is expected that any asset compromised by an adversary in order to obtain the key has a negligible value compared to the value of the key itself. Therefore, the recovery cost is given by the key replacement cost.

Based on these factors, a three-level impact estimation scale is proposed (see Table 4.8) and further adopted for assessing the key exposure impact in this thesis.

Rating	Impact definition
High (100)	Exercise of the vulnerability may result in the highly costly loss of major tangible assets or resources; may significantly violate, harm, or impede an organizations mission, reputation, or interest; or may result in human death or serious injury.
Medium (50)	Exercise of the vulnerability may result in the costly loss of tangible assets or resources; may violate, harm, or impede an organizations mission, reputation, or interest; or may result in human injury.
Low (10)	Exercise of the vulnerability may result in the loss of some tangible assets or resources or may noticeably affect an organizations mission, reputation, or interest.

Table 4.8: Impact Scale [GSF02].

4.6 Risk Estimation

An information security risk is the potential that a given threat will exploit vulnerabilities of an asset or group of assets and thereby cause harm to the organization [Sta08]. Risks (including key exposure risks of interest in this thesis) are assessed as a combination of threat event likelihood and impact:

$$Risk = Likelihood \times Impact, \tag{4.4}$$

where both factors are assumed to have been previously determined. In [GSF02] NIST takes advantage of the semi-quantitative likelihood and impact values to compute the corresponding risk. The result is then mapped to its qualitative correspondent as illustrated in Figure 4.7. A conservative character is induced by the mapping function, which associates half of the risk range to high risks, 40% to medium and only 10% to low. We expect some organizations to be driven by a more relaxed risk model, in which case they could opt for a different risk scale.

We want to stress that the risk values obtained are not guaranteed to provide very accurate estimations. However, they provide practitioners with valuable insight into identifying those weak key management processes which require further in-depth analysis.

At the end of this step practitioners are expected to possess all the information required to fill in Table 4.9.

4.7 Response Identification

To address the identified key exposure risks, four types of risk response strategies are available for consideration: **risk reduction**, **risk acceptance**, **risk transfer** and **risk avoidance**. In this thesis we only address risk reduction controls. In particular, we focus on key renewal periods as a preferred risk response. We investigate the key renewal interval

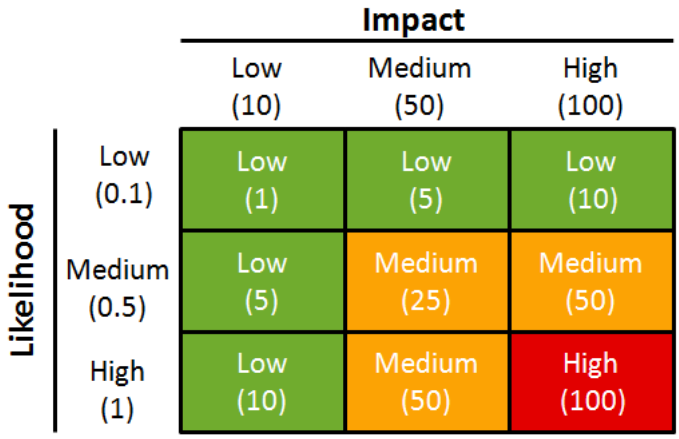

Figure 4.7: Risk Scale: High (51-100); Medium (11-50); Low (1-10) [GSF02].

Threat source	Threat event	Likelihood	Impact	Risk
	Compromise the key generation process.			
	Compromise the key storage.			
	Compromise the key distribution process.			
	Compromise the key usage.			

Table 4.9: (Threat event, Likelihood, Impact, Risk)-tuples.

behavior as a function of key exposure risks caused by changing the key and respectively using it, without numerical estimations to be provided.

Key Renewal Period Estimation

The risk definition determines risk reduction efforts to concentrate on diminishing the likelihood of threat event occurrence and/or the impact caused. In an attempt to reduce the likelihood component, companies are advised to implement security countermeasures that patch vulnerabilities or prevent their exploitation. Yet, potential high costs or business constrains may represent decision factors for not considering strengthening the key environment a practical solution. Due to reasons listed in Section 1.1, changing keys periodically may come up as a preferable alternative for responding to key exposure risks. Since it limits the amount of information protected with a single key, renewing keys is expected not to only reduce the likelihood, but to attenuate the key exposure impact as well. Note that this statement stands under the assumption that existent databases will not be re-encrypted with the new key. Yet, we want to stress that especially when keys are changed through immature processes, regularly renewing keys adds no value to key protection.

The problem description in Section 1.1 indicates a dependency of the key renewal interval

on:

- **The key exposure risk caused by changing the key**, on one side. Since changing keys generally implies the generation and distribution of new ones, we consider this risk (which we further denote by R_C) being given by the highest risk of compromising any of the key generation or distribution processes.
- **The key exposure risk caused by using the key**, on the other side. Assuming no more need for key distribution, the risk (further denoted by R_U) is given by the highest risk of compromising the key during its usage or compromising the key storage.

As an input for the key renewal interval estimation in this thesis we consider the risk of key change, R_C and respectively key use, R_U, emphasized at the end of the risk assessment process. The analysis we perform occurs in two phases. In the first phase we investigate possible directions for the evolution of the two risks over time. In the second phase, we attempt to estimate the moment in time cryptographic keys should be changed.

Risk Evolution over Time

There are several factors that might influence the evolution of R_U over time. First of all, implementing measures to strengthen the key environment is expected to drop the likelihood of key disclosure attacks to succeed. For example, switching from a FIPS 140-3 level 1 certified module to a FIPS 140-3 level 4 module should make it more difficult for attackers to compromise the cryptographic module. During the time interval when no measures to strengthen the key environment are applied, common sense predicts a tendency of the likelihood factor to increase. Reasons behind this statement lay in the more knowledge about the target adversaries are expected to accumulate and the technology evolution increasing attacker's capabilities. Secondly, using the same key for a longer period increases the window opportunity for compromising the key while in use.

Finally, the impact of key disclosure might change as well. In most of the cases, the amount of data protected with the cryptographic key increases over time, potentially leading to a growth of the impact factor. Yet, this is not always the encountered behavior. When the security life of protected data has already reached its end, the compromise of cryptographic keys causes no impact. An illustrative example is a voting system based on a symmetric key to protect voters' choices. If anonymous voting is assumed, votes opening drops the impact to zero.

A similar discussion is performed for R_C too. Unless controls to strengthen the key change process are applied, common sense predicts a tendency of R_C to increase over time, mainly due to better attack planning. In practice, this increase is not always perceivable. In addition, when key generation is based on insufficient randomness, keys might appear more predictable to adversaries with every generated key, increasing the likelihood of key compromise. The discussion changes when companies implement controls to strengthen the key change process, e.g., better random number generators, more secure key distribution

channels. Such measures are meant to make attacks on the key change process less likely to succeed.

The evolution of key exposure risks over time is of major importance for understanding the role of key renewal periods. A general pattern to describe it independent on the company's behavior does not exist. To address this challenge, companies are advised to periodically assess the key exposure risks.

Moment Estimation for Cryptographic Key Change

The analysis in this section investigates the selection of key renewal periods such that they minimize the risk of key exposure. To estimate the moment in time when cryptographic keys are to be changed and therefore recommend key renewal periods, we consider the order relation between the two risks a differentiator, as explained below:

- $\boldsymbol{R_U \gg R_C}$ (to be read as R_U is considerably larger[1] than R_C).

  When using cryptographic keys poses a higher risk on their exposure than the key change process, selecting key renewal periods should focus on reducing R_U. Ideally, R_U is brought into the vicinity of R_C. In this sense, companies are advised to change keys at relatively small intervals. The expected effect of such a control is the reduction of both the key exposure likelihood and the caused impact.

  Though, some companies might consider extending the key renewal interval in order to reduce key replacement costs. In general, the lower R_U, the more the key renewal periods can be expanded. Special attention should be given to ensure that the expanded interval does not significantly affect key confidentiality.

- $\boldsymbol{R_U \ll R_C}$ (to be read as R_U is considerably smaller[2] than R_C).

  A high assessment of R_C is probably the most undesirable situation a company might encounter. Disclosure of cryptographic keys during their generation, distribution or deployment in the system allows for real-time processing of protected data. When database re-encryption with the compromised key is performed, the impact grows even more.

  Given the higher value of R_C compared to R_U, the reader might be inclined to think of delaying the key change process by selecting large key renewal intervals. Such a solution is also supported by the low risk of key compromise during its usage. In practice, using a key for a long period has significant consequences if undetected key disclosure takes place even before the key being used. Therefore, in that case, shorter key renewal intervals might be suitable. Considering cost reasons, the lower R_C, the larger the key renewal interval can be.

[1] R_U has been assessed at least one level higher than R_C, e.g. R_U is High and R_C is Low.
[2] R_U has been assessed at least one level lower than R_C, e.g. R_U is Medium and R_C is High.

- $\boldsymbol{R_U \approx R_C}$ (to be read as R_U is around equal[3] to R_C).

  When R_U is in the close vicinity of R_C, changing keys at relatively small intervals keeps the risk of key exposure around equal to R_C. In general, for a specific key environment, the lower R_U and R_C, the more the key renewal period might be expanded for cost reasons.

The discussion above provides initial directions for key renewal period estimation. Still, much research remains to be done to refine the model and enhance its applicability. Directions for improvement are discussed in Section 6.2.

4.8 Response Decision

This section highlights the need for careful risk response decision and implementation.

For identified key exposure risks, alternative courses of actions should be evaluated [NIS11a]. Organizations should consider key renewal intervals, controls for strengthening the key management environment as well as risk transfer, acceptance and avoidance as possible responses to key exposure risks. Risk response alternatives should be evaluated against the same criteria and a decision should be made. Evaluation criteria for alternatives might include: effectiveness in responding to risks, legal, financial and technical considerations [NIS11a], etc.

Finally, preferred risk responses should be properly implemented. To provide key management guidance and support confidentiality, integrity and availability of cryptographic keys and metadata bound to them, a security policy stating key management requirements should be established and communicated to all relevant employees and third parties. Key users should be informed with respect to key usage practices as well. Referred by NIST as Cryptographic Key Management System (CKMS) Security Policy, the document should support the goals of the Information Security Policy and the Information Management Policy [BBCS10]. For example, if the Information Management Policy states that data must be available for regulatory checks after a long amount of time, the CKMS Security Policy must consider archiving the key which protects the data and store it in a secure environment. More detailed policies derived from the CKMS Security Policy (e.g. Cryptographic Module Security Policy) need to be also consistent with the key management requirements and support the business goals. The relation between the CKMS Security Policy and other information policies is illustrated in Figure 4.8.

Key renewal intervals as well as any other key exposure risk mitigating control should be specified as part of the CKMS Security Policy or policies derived from it. Each policy should be reviewed and updated regularly to ensure that it reflects changes in the key management environment. Organizations should ensure that security policies are reliably enforced and the risk to key exposure is minimized.

[3] R_U and R_C are both Low, Medium or High.

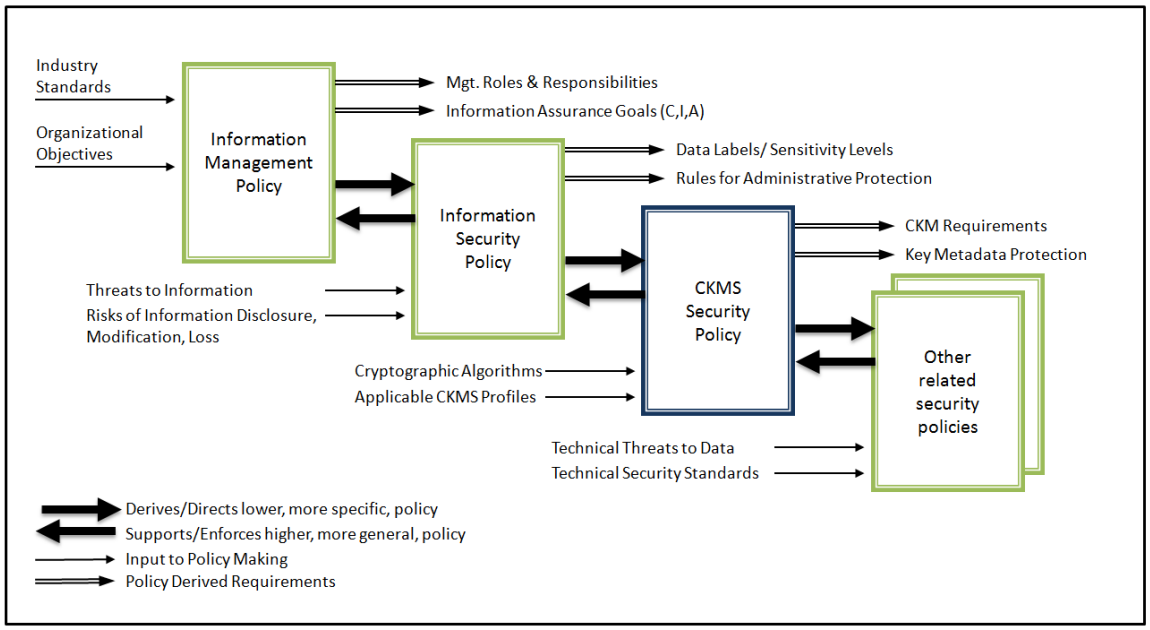

Figure 4.8: Related security policies [BBCS10].

4.9 Discussion

The reader may think of other approaches for estimating reliable key renewal intervals. More static approaches (e.g. the NIST cryptoperiod recommendations provided in [BR11]) will most probably turn out to be simple recommendations independent of the security maturity of the organization in scope. Approaches with a more dynamical character, as for example penetration tests, may allow for a more accurate security vulnerability identification in the key management environment. However, they often become impractical due to reasons listed below:

- **Long-term process**. Successfully performing a key exposure attack should take longer than a suitable key renewal interval. Assuming a mature organization from a security point of view, it turns out that attacking the key management system may take years, becoming impractical.

- **Difficult and expensive**. In order to perform penetration tests for identifying system vulnerability to key exposure, additional hardware and human resources may be required, making the approach expensive. Even when all required resources are available, there may be several other factors that increase the complexity of this approach. Especially for a large organization, getting all the needed approvals for starting the project may require extensive work and persuasion.

- **Additional risks**. Performing a key exposure attack on a system introduces additional risks, which may alter system's availability. In the case of key management

systems, these risk may lead to significant damages, e.g. financial loss due to cryptographic module failure.

- **No guaranteed results**. Even if the results of a dynamic approach indicate a specific key renewal interval, there is no guarantee that by enforcing it your key is secure.

A less time demanding approach could consist in qualitatively assessing several factors that influence the key secrecy and combining them to derive key renewal intervals. Note that a set of influencing factors have been already listed in Chapter 2. However, the main challenge in designing such a model consists in commixing them in a realistic manner that reflects their relevance to key renewal periods. Determining the dependency between these factors and the key renewal interval as well as identifying their individual weights out of a specific attack scenario context is unquestionably ambitious.

Key renewal interval estimations based on key exposure risk assessment come to address most of the mentioned issues. They are relatively easy to understand and apply and map well to the organization's security maturity level, being preferred over the other approaches discussed in this section.

Chapter 5

General Guidance w.r.t. Common Vulnerabilities

Vulnerability identification is probably the most challenging risk assessment step. Unless performed by highly experienced and knowledgeable professionals in the key management and security fields, the vulnerability landscape identified may be far from being comprehensive. This results in unaccounted attack opportunities adversaries may take advantage of. In addition, unrealistic assessment of vulnerability criticality leads to ineffective risk management.

This chapter addresses the identified issues by: giving practitioners insight into spotting vulnerabilities, providing them with general guidance for assessing the difficulty of vulnerability exploitation. We proceed by analyzing the vulnerabilities mentioned in Section 4.3. Note that to avoid redundant discussions, we analyze similar vulnerabilities together. For example, the strength of modules/media for key storage and manual key transport follow a similar discussion and are treated in the same section.

5.1 Insecure Cryptographic Module

Cryptographic modules might become targets for adversaries aiming at compromising cryptographic keys. Preferably secret parameters should be destroyed before unauthorized access to them is gained [Cla88]. The reader may sense here the analogy with a stolen mobile device scenario. To avoid private information getting disclosed, remotely wiping mobile phone data is recommended. While this is generally a human-initiated process, cryptographic modules would have to sense themselves and auto-initiate the wiping mechanism.

The resistance of cryptographic modules to key exposure attacks may vary depending on their implementation, e.g. software implementation on a normal PC, firmware, hardware or hybrid implementation. We further analyze the weaknesses and strengths of each of the mentioned cryptographic module embodiments.

Software Implementations on Regular Devices

While software implementations on normal devices are easier to develop and maintain, most of them are significantly less safe than their firmware and hardware equivalent. The security provided by traditional software implementations on normal PCs/servers is usually upper-bounded by the security level of the memory space protection mechanism and the security level of the operating system on top of which they are running. Reasons behind this claim include [BE02]:

- **Software implementations cannot facilitate their own physical memory.** They make use of external, shared available memory, being more fluid in terms of ease of modification. When memory used by the software module is provided externally, there is no guarantee that other processes will not access or modify it, either accidentally or deliberately [BE02]. Note that cryptographic modules are particularly sensitive to having their own memory space well protected, since they require to store intermediate results during their execution.

- **Software implementations run on top of an operating system.** Therefore, defeating the operating systems leads to cryptographic systems being possible to defeat. Assuming a password-protected encryption key, keylogger software installed in the system may lead to key disclosure. In general, the more services running on top of the operating system, the more likely is that security flaws allowing for key exposure are introduced.

In addition, software solutions residing on untrusted hosts may leak key information through side channel attacks. This type of attacks considers adversaries taking advantage of the physical observable phenomenons caused by the execution of computing tasks in devices [Sta10]. They are classified in literature along the following three axes [YD05]:

- **Control over the computation process:** passive vs. active attacks. While active attacks try to tamper with the device, e.g., by inducing errors in the computation, passive attacks only observe the devices behavior during their processing, without disturbing it [Sta10].

- **Way of accessing the module:** invasive vs. non-invasive attacks. Invasive attacks require depackaging the chip to get direct access to its inside components. At the opposite side, non-invasive attacks only exploit externally available information [Sta10]. Note that for certain attacks, may not even be necessary to be in close proximity of the target [Tir07].

- **Method used in the analysis process:** timing attacks, fault attacks, power analysis attacks, electromagnetic analysis attacks, acoustic attacks, etc.

Even so, classifying any software implementation running on normal PCs/servers as completely insecure tends to be erroneous, due to several techniques developed to make it

difficult or theoretically impossible for adversaries to compromise the cryptographic key. Such techniques include: generic obfuscation transformations[1], white-box cryptography[2], intensive testing and reviewing by security professionals. Software-level solutions to provide resistance of the cryptographic module against side-channel attacks, independent on the hardware they are running on have been proposed as well. Time randomization, permuted execution and masking techniques are only few possible countermeasures. When running on PCs/servers with hardware measures against side channel attacks, (e.g. noise generation, power signal filtering, etc.) the susceptibility to information leakage decreases even more. Even so, predicting all different ways an attacker can gain side-channel information in order to establish the needed countermeasures seems like a challenging goal [Pet10]. Software module implementing leakage resilient cryptography (LRC) was developed to address this issue, by building primitives that provably withstand attacks where the existence of side-channel information is taking for granted. It is unclear, however, whether the attack models considered capture real-world adversaries [SPY+10].

Hardware Implementations

Storing cryptographic keys in hardware on dedicated devices, wrapped with multiple levels of security tends to eliminate the risk of key theft or loss. The logic function is implemented in hardware circuits rather than executed on top of an operating system, drivers, and application software. Therefore, it becomes very difficult to tamper with hardware implementations.

The piece of hardware and its associated software/firmware providing at least the minimum of cryptographic functions is referred to as HSM [Att02]. HSMs come in various forms and shapes, including PCI cards to plugin in PCs, USB tokens, separate boxes that communicate over channels like TCP/IP [Jan08]. They are used in several application areas [Wor08], as for example: document protection, army, PKI environments to generate, store and handle key pairs, key distribution centers.

Despite the fact that it solves several issues encountered in software implementations, hardware modules may still remain vulnerable to side-channel attacks.

[1] Generic obfuscation transformations are meant to harden the analysis of cryptographic primitives by making the code incomprehensible. Intuitively, the obfuscation of a piece of code should not provide an adversary with more information than one can learn from oracle access to that function [HMLS10]. Even so, code obfuscation transformations may not be considered strong enough to sufficiently protect the confidentiality of cryptographic keys [Bre11].

[2] A relatively novel approach, white-box cryptography aims at implementing cryptographic primitives in software such that they remain robust against adversaries controlling the execution environment, e.g. attackers who can intercept CPU calls, have access to the memory or to the binary software implementation and deploy debuggers and emulators [Bre11]. The technique is based on rewriting a key-instantiated implementation such that all key-related leaking information is *hidden*. At the moment of writing this thesis, the technique is still young and requires further research before being adopted in commercial products. However, it is expected that significant progress will be made in the field of white-box cryptography in the near future, allowing for secure software-only solutions running on non-trusted hosts.

Firmware Implementations

Firmware fits between software and hardware. In general, it provides higher security than software implementations, but falls behind its hardware equivalent. Being located in memories embedded to the device itself, it is inaccessible to users and replacing it is often a difficult task. However, firmware may introduce vulnerabilities through API interfaces and extensive modules [ZXK10].

The discussion above suggests a general tendency of cryptographic module strength increasing from software to hardware implementations. However, one should not rely on this tendency to decide how secure modules are. Ideally (although often impractical), vulnerabilities are identified and evaluated by intensively testing cryptographic modules independent of their embodiment type. A widely accepted evaluation standard for cryptographic modules is discussed later in this section.

Note that for key-generating modules, the RNG plays a significant role in establishing the strength of the key. Best practices recommend that cryptographic keys look random, i.e. certain elements of the key space are not more probable than others and that it is not possible for unauthorized entities to gain any knowledge about the keys [Wal93]. Therefore, key generation should rely on using RNGs.

RNGs can be deterministic, also known as pseudo-RNGs or non-deterministic, also known as true or hardware RNGs. Deterministic RNGs generate sequences of numbers based on pre-defined tables or mathematical formulas and an initial seed. The output is not truly random, but looks so to any entity not knowing the seed. On the other hand, non-deterministic RNGs take advantage of the randomness in the physical world (e.g. electrical resistance noise, radioactive decay rates, etc.) to output unpredictable values. However, true RNGs are generally hard to get right, due to physical processes potentially reaching undesired states, e.g., free-running oscillators may lock to some undesired frequency and thus produce some non-random output [SDFF06].

Poor RNGs could lead to guessable symmetric keys. For example, an older version of the Netscape's Secure Socket Layer encryption protocol was cracked in 1995 because a weak RNG was used to create symmetric keys for client-server communication. More specifically, it used a pseudo-RNG seeded with a conglomeration of values (the current time, the process ID number of the Netscape process and its parent's process ID) which could be easily guessed by attackers [Wen01]. Even if not entirely guessable, poor random number generators might affect the entropy of keys and decrease the time required to brute-force them.

To avoid such issues, it is recommended the use of RNGs extensively reviewed by professionals and for which vulnerabilities are not known. A list of NIST-approved pseudo-RNGs is provided in [CE11]. To date, no NIST-approved true RNG exists.

Cryptographic Module Evaluation

FIPS 140 validation is the globally accepted standard for cryptographic module security evaluation. Currently at the third version, the 140 series of FIPS define four levels of security. The higher the level, the more secure cryptographic modules are assumed to be. Since December 2006, the active version of the standard is FIPS 140-2 [FIP01]. The reader can refer to Appendix C for an overview of both criteria and certification levels considered by FIPS 140-2.

According to FIPS 140-3 Development Status[3], FIPS 140-2 is expected to be soon retired in favor of an updated version, FIPS 140-3. Currently released as a third draft for public comments [FIP09], FIPS 140-3 keeps the number of evaluation levels to four while revising the set of criteria to reflect advances in technology. Main changes include [Con]:

- Security Level 4 introduces the multi-factor authentication requirements for operator authentication;
- Security Level 3 and 4 address mitigation assurance requirements for non-invasive attacks;
- Design assurance requirements increase through the levels;
- A software security section organized on four increasing levels has been added to the requirements;
- New self test has been introduced.

FIPS 140-3 evaluation criteria and security requirements are included in Appendix D. Cryptographic modules are assessed against each of these criteria and the assessment result is mapped to a security level as shown in Table D.2. It is an accredited laboratory that tests the compliance of a module with the testing criteria. The up-to-date[4], comprehensive and clearly specified FIPS evaluation criteria explains the increasing number of FIPS-140 certified devices every year (see Figure 5.1).

Despite its popularity, the FIPS evaluation does not guarantee that the certified device is secure. For example, a FIPS 140 Level 2 Certified USB Memory Stick has been cracked at the beginning of 2009 [DS09]. Instead, it ensures that the module stands against a reasonable baseline of security tests performed by qualified professionals [Att02]. This provides practitioners with directions on estimating how likely is that adversaries will succeed in compromising the certified device in order to extract cryptographic keys.

Other drawbacks FIPS 140-3 presents are:

- The certification process may take long: testing can vary between 8 weeks up to a year or more;

[3]The FIPS 140-3 Development status can be found at http://csrc.nist.gov/groups/ST/FIPS140_3/
[4]The FIPS 140 standard is to be re-evaluated at five-year intervals.

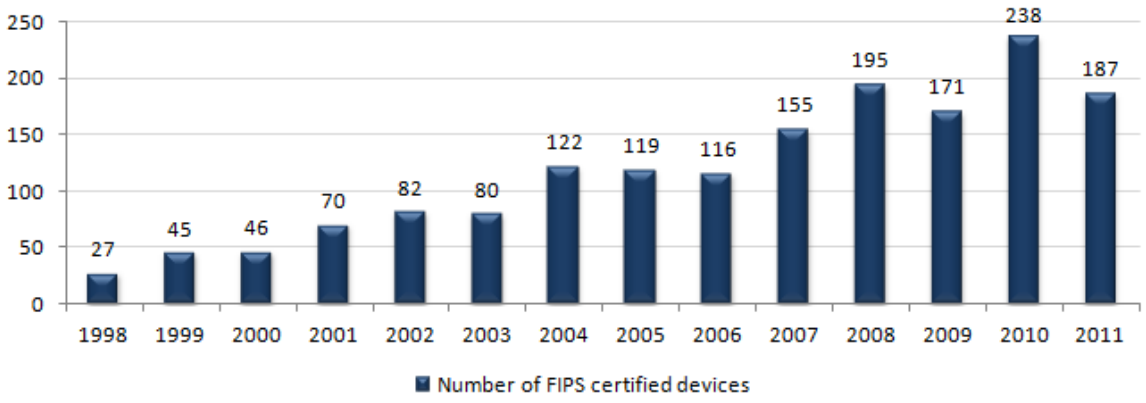

Figure 5.1: Number of FIPS 140 certified devices per year. Data source: [FIP]

- The algorithms that the standard can evaluate are restricted to the US approved ones, such that countries wishing to evaluate with other cryptographic algorithms are not able to use this criteria.
- It considers only four security levels, not offering a fine-grained classification of the evaluated modules.

Nevertheless, these drawbacks do not affect the practical benefits FIPS 140 brings.

Note that ISO/IEC has translated FIPS 140-2 into the ISO/IEC 19790, "Security Requirements for Cryptographic Modules". Examples of changes considered are the inclusion to ISO terms, definition and references as well as the deletion of the EMI/EMC section. Compared to FIPS 140-2, the document is not freely available.

5.2 Insecure Key Storage

Hard-copies (e.g. paper), removable storage media (e.g. USB flash drives, disks, memory cards), end-user and consumer devices (e.g. laptops, mobile phones) tend to be the most uncontrolled storage solutions. They often end up lost, stolen or tossed into trash containers, risking accidental disclosure of contained data to overcurious individuals. Unless logical protection is enforced, e.g. encrypting the content, enforcing strong user authentication, getting access to such media is all adversaries need for extracting contained information, here cryptographic keys or metadata bound to them.

Storing keys on cryptographic tokens, such as smartcards, provide better security [SLSD11]. Cryptographic keys are protected using physical tamper-resistant measures, ensuring that secret information never leaves the physical security perimeter of the hardware [CC09]. Nevertheless, they often remain vulnerable to side channel and cloning attacks. These vulnerabilities are expected to be addressed by FIPS 140-3 level 3 or 4 certified modules.

Special attentions should be paid to key entry devices as well. They should be carefully selected to ensure that sensitive information only reaches the destination device. Entry

mechanisms leaking side channel information give adversaries the possibility to intercept traces and reconstruct the key. In addition, devices retaining key information may become easy targets after the completion of the key entry process.

5.3 Insecure Cryptographic Schemes

Closed proprietary or known weak cryptographic schemes (either key generation, secret sharing schemes, data protection schemes) should be avoided. Building secure schemes is generally hard to get right and unless they have withstood repeated and in depth review, cryptographic schemes are likely to contain security flaws. Suggestive examples of insecure key generation and secret sharing schemes are:

- Reversible key derivation or update functions. Getting knowledge of derived keys allows for determining the key-derivation key. This can be further used to derive new keys or discover previously derived ones.

- Secret sharing schemes that allow parties to gain more knowledge about the secret than outsiders. Assuming $k = n$, a naive, insecure secret sharing scheme is cutting the key K in n equal parts of size $\frac{|K|}{n}$ and distribute them to entities. This reveals at least $1/n$ of the original information to each shareholder. In addition, for small values of n (e.g. $n = 2$), brute-forcing the unknown key part might become practical, leading to complete key disclosure.

To avoid insecure key generation schemes, NIST provides a set of recommendations mainly concentrated in [BA01]. Provably secure secret sharing schemes are widely discussed in literature too, e.g. [Sch99, HKS12].

A more elaborate discussion needs to be performed for data protection cryptographic mechanisms. The security strength of such a mechanism is traditionally described in terms of the amount of work it takes to try all possible keys[5], i.e. the best attack is said to be the exhaustive key search. However, in theory any attack method making fewer than 2^k operations for a k-bit key is considered a theoretical break, even when the effort remains impractically high. These attacks, referred as algorithmic attacks are more than code breaking and target a mathematical function without considering its implementation. Although they should not be underestimated since they succeed in identifying weak ciphers[6] before they are used, algorithmic attacks consider attack models[7] that simplify reality. Therefore, many ciphers pretended to be broken raise no security concerns in practice. Clear reasons behind this statement have been presented by Aumasson in [Aum11] and are summarized below:

[5] There are 2^k possible keys for a k-bit key. On average, the expected number of trials is 2^{k-1}.

[6] A cipher is considered weak as a result of cryptanalysis when a mathematical weakness is proved to exist.

[7] An attack model defines the assumed attacker's capabilities.

- **High complexities**: Compared to brute-force attacks, algorithmic attacks usually only slightly reduce the number of operations needed to be performed, such that the effort remains impractically high.

  For example, at the time of writing this thesis the best attack on AES (and the only one performing better than exhaustive search) was published by Bogdanov et al. in 2011 [BKR11] and has a complexity of $2^{126.1}$ operations for AES-128, $2^{189.7}$ operations for AES-192 and $2^{254.4}$ operations for AES-256. This is very close to the complexity of generic attacks: 2^{128}, 2^{192} and respectively 2^{256}. This attack is the first key-recovery attack on full AES.

- **Building blocks as target**: Several attacks target building blocks of an algorithm and not the entire function. Examples of target building blocks include the compression function of a hashing algorithm or a reduced number of rounds of a block cipher. Although sometimes the attack propagates to the entire function, this is not often the case.

- **Strong attack models:** Attack models often make unrealistic assumptions in real-life, giving the attacker incredibly high powers.

  For example, several theoretical attacks capable of breaking the DES cipher faster than brute-force attacks have been published during the time [BS93, BB93, Mat94]. However, these attacks are not much of a threat to real-life applications since although having a lower time complexity ($\approx 2^{40}$), the best known attack to the moment requires at least 2^{43} known plaintexts encrypted under the same key to be available for the analysis.

- **Memory is not free**: Algorithmic attacks require storing pre-computed data in large tables, but memory requires physical hardware that comes at a price.

  To date, the best known attack on Triple DES using three independent keys requires around 2^{32} plaintexts, 2^{90} DES encryptions and 2^{88} memory [Luc98]. Using such a large memory would considerably slow the attack, making it even less practical than a brute-force one [Aum11].

Based on the discussion above, it becomes reasonable to estimae the strength of the cryptographic mechanism by considering its resistance to brute-force attacks. We further discuss the key length as a major factor in deciding the algorithm's resistance to brute-force attacks.

The Key Length Factor

No more debates are needed for concluding that short keys weakens a cryptographic mechanism. For example, because of its 56-bit key[8], DES is considered an insecure cipher. Since

[8]DES uses a 64-bit key, out of which only 56 bits are randomly chosen.

its release, considerable hardware solutions[9] have been done to break DES. In 1980 a 50 million US dollars machine to extract the key in 2 days has been proposed. An ASIC-design based machine known as *Deep Crack* was built by Electronic Frontier Foundation in 1998, being able to break DES in only 112 hours at a cost of US $130.000. Eight years later, another hardware consisting of commercially available integrated circuits recovered a DES key in 6.4 days on average. Known as COPACOBANA (Cost-Optimized Parallel Code-Breaker), the FPGA cluster provided the performance of more than 3000 PCs and came at a price of only US $10.000. Having a modular design, the special-purpose hardware integrates up to 20 FPGA modules with 6 Spartan 3 FPGAs each [Pel06]. In 2008, a new setup of COPACOBANA (RYVIERA) managed to reach the 292 billion keys per second, breaking a DES key in less than a single day. And the most important thing is that RIVYERA is accessible to everyone providing the most simple to use personal key recovery tools ever designed [Sci08]. Despite its low strength, DES is still in use [OEFC11]. Some have depended on DES for several years already. While individual hackers may not afford to acquire a dedicated hardware for breaking the cryptographic system in a short amount of time, this will surely not represent an impediment for funded organizations.

To address brute-force attacks, NIST currently recommends AES and three-key 3DES as symmetric encryption algorithms and the use of SHA-2 family as hash functions. A three-star rating for symmetric cryptographic algorithms have been also performed by ECRYPT [OEFC11] as illustrated in Figure 5.1, where:

- Three stars assigned to the usage criteria indicates a wide deployment of the scheme, while no star denotes that ECRYPT is not aware of any deployment;
- A three-star rating for the security criteria indicates that ECRYPT is confident in the scheme[10]. No star indicates that the scheme should not be deployed for security reasons.

However, switching over to one of these algorithms might turn out to be demanding. First of all, longer keys require more resources to achieve the same performance as shorter keys [Kri11]. Secondly, even when these resources are attainable, changing the cryptographic mechanism may end up having significant financial consequences, caused by the potential need for temporarily stopping the business activity, the large number of devices which need to be replaced/updated, etc. In this case, organizations might prefer to evaluate the cryptographic mechanism they implement and to select key renewal periods according to the obtained results. A model for evaluating cryptographic mechanisms is further discussed in this section.

Cryptographic Mechanism Evaluation

Assuming an investment of c US dollars for brute-forcing a k-bit key, the model of Lenstra et al. allows to estimate the attack time when the price of computational power is as available

[9] At the cost of one-time investment, a hardware attack is considerably faster than its software-equivalent.
[10] The rating is based on the assumption that parameters are properly selected.

Algorithm	Key/Block size	Known usage	Security
Block ciphers			
DES	56	***	-
3DES	112	***	*
3DES	168	***	**
Kasumi	128	***	**
Blowfish	Various	*	**
AES	All	***	***
Stream ciphers			
RC4	Various	***	**
SNOW 2.0	128/256	**	**
SNOW 3G	128/256	**	**
Hash Functions			
MD5	128	***	-
RIPEMD-128	128	-	*
RIPEMD-160	160	-	**
SHA-1	160	***	*
SHA-2	Various	*	***

Table 5.1: Three-star usage and security rating of symmetric cryptographic algorithms provided by ECRYPT [OEFC11] .

in year y. In this sense, the following two steps should be performed:

1. **Calculate one-day attack cost**.
   The equivalent one-day attack cost c_k is given by:
   $$c_k = \frac{5 * 10^5 * 2^{k-56} * v}{IMY_y} * c_y. \tag{5.1}$$
   The factor $5 * 10^5 * 2^{k-56}$ denotes the number of Mips-Years it takes to brute-force a k-bit length key, where the ratio of the number of cycles required for a single block encryption using the symmetric key system in place and DES is denoted by the variable $v > 0$. By default, $v = 1$. IMY_y and c_y denote the infeasible number of Mips-Years for year y and respectively the hardware cost needed to perform this amount of operations in one day. Both IMY_y and c_y can be easily obtained from Appendix B. We write $[c_k$ US dollars, 1 day, $y]$, meaning that the attack can be performed for a cost c_k within one day in year y.

2. **Apply the cost-performance ratio principle.**
   The cost-performance ratio stays constant in this estimation. Therefore, when an attack can be performed in d days at the cost of c US dollars, fully parallelization (if possible) allows the same attack to be performed in $\frac{d}{p}$ days at a cost of $c \times p$ dollars, for reasonable p values. For example, exhaustive search can be fully parallelizable, by arbitrarily dividing the key search space over any number of processors that may work independently. This proves the correctness of the following equivalence:

$$[c_k \text{ US dollars}, 1 \text{ day}, y] \equiv [c \text{ US dollars}, \tfrac{c_k}{c} \text{ days}, y],$$

where the number of days in the right term represents the time required by the attack. We further denote this time as the brute force attack time for cost c, key length k and year y.

We would like to stress that these are estimates rather than exact numbers.

Quantum Computing

Note that the current research does not focus on quantum computing[11] for the following reasons:

- To date, the use of quantum computing for brute-forcing keys does not represent a threat, since the research is not mature enough yet:

  "A quantum computer would effectively halve the key length, so that a 256-bit key would be only as secure as a 128-bit key today. Pretty serious stuff, but years away from being practical."

  (B. Schneier referring to a 256-bit key length [Sch08])

- Quantum cryptography is not expected to replace classical cryptography soon:

  "I'm always in favor of security research, and I have enjoyed following the developments in quantum cryptography. But as a product, it has no future. It's not that quantum cryptography might be insecure; it's that cryptography is already sufficiently secure."

  (B. Schneier [Sch08])

However, note that opinions regarding the future of quantum computing are divided:

"In the past, people have said, maybe it's 50 years away, it's a dream, maybe it'll happen sometime. … I used to think it was 50. Now I'm thinking like it's 15 or a little more. It's within reach. It's within our lifetime. It's going to happen. [Cha12]"

(Mark Ketchen, IBM's Watson Research Center)

Having these said, we do not completely reject the possibility of a potential quantum computing revolution, but we find it unlikely that quantum computers will be available at normal PC stores in the very close future. However, we recommend practitioners to consider quantum computing when becoming relevant.

[11] Quantum computing makes direct use of quantum mechanical phenomena such as superposition and entanglement in order to process data. It comes as a solution to overcome the current physical limitations of computers, by using atoms instead of transistors. Besides offering substantial computational speedups, it is also expected to preserve the privacy of a computation.

5.4 Insecure Use of Schemes, Modules or Media

Although it is unquestionable that use of insecure schemes, modules or media represents a bad practice, in reality, this is often not the real concern. Instead, insecure use of them is what often leads to security incidents. The following examples illustrate the insecure use of cryptographic schemes, modules or media:

- DH key agreement implementation without prior authentication. Such an implementation makes the scheme susceptible to man-in-the-middle attacks. Use of small subgroup orders also makes brute-force attacks more likely to succeed.
- Key derivation or key update functions fed with weak or already compromised key derivation secrets.
- Insecure cryptographic module configuration, e.g., using modules out of the box. For devices not addressing security by default, their public specifications often represent valuable sources of secret information (e.g. default administrator passwords). This information can further serve malicious purposes and contribute to sensitive data disclosure.
- Lack of secure disposal or re-use of module/media, e.g., no immediate destruction of residues from the printing process that might disclose key components, no removal of potential key traces from the key loading device.
- Removal of property, e.g., equipment or information is taken off-site without proper authorization [Sta07].
- Lack of physical protection of modules/media, e.g. when not printed within blind mailers or sealed immediately, printed key components can be potentially observed by distrustful parties [VIS08].

In most cases, it is the lack of policies, training and awareness of key stakeholders that leads to inappropriate handling of cryptographic schemes, modules or media and generally, to ignorant behavior with respect to security.

5.5 Insecure Key Transport Channel

In cryptography, secure transport channels represent a way of communicating data such that no adversary can overhear or tamper with it. Otherwise, the channel is said to be insecure. Manually transporting keys through public areas gives adversaries the chance to supervise the process and further to spot vulnerabilities whose exploitation results in key compromise. It is recommended that keys manually transported are physically and/or logically protected.

Insecure electronic channels are susceptible to adversaries listening to the communication, attack referred in literature as *sniffing* or *eavesdropping*. The success likelihood of

eavesdropping attacks is mainly determined by the types of networks data travels through. For example, public networks like Internet allow accessibility to everybody. Therefore, Internet eavesdropping tends to be considered a significant threat when sensitive information is communicated and no protocol having some kind of encryption built-in against network sniffing is used. On the other hand, private LANs are expected to restrict the number of individuals authorized to access the network. Sniffing in private LANs is now more difficult, not only due to the eventual need for bypassing network access control, but mainly due the fact that most networks don't have hubs left[12], but are switched[13]. Further, leased lines contain no router in between, which makes sniffing a hard task to achieve. In reality, whether it is feasible to eavesdrop on leased lines outside a company's building is reduced to questioning whether the party providing the lines is trustful or not. However, assuming an honest provider, sniffing the communication sent over the leased line is expected to be a difficult task. Similar to hub networks, wireless networks send the traffic such that anyone can intercept it by setting the network card in the monitor mode. Sniffing software like AirSnort, airodump-ng or Wireshark is freely available and can be easily used in this sense.

To make eavesdropping ineffective independent on network type, it is highly recommended that cryptographic keys are encrypted using strong mechanisms before being sent through the channel. Note that for wireless networks, WEP encryption rather offers a false sense of security, being easily broken. Unlike WEP, WPA/WPA2 networks negotiates unique encryption keys (known as Pairwise Transient Keys or PTKs) with any device joining the network, such that there is theoretically no risk of one intercepting other user's traffic. Even so, WPA/WPA2 networks are vulnerable to the "Hole 196"[14] attack [Bra11].

5.6 Insecure Operating Environment

The operating environment denotes here the location of a system and the restrictions imposed on physically accessing it. In the current context, a system is a cryptographic module or any electronic media storing a cryptographic key. Intuitively, the stronger the physical access restrictions to a system are, the more difficult it becomes for unauthorized entities to

[12]Fifteen years ago, bus networks, often implemented with hubs, were common [JR05]. In that environment, every network frame was broadcasted to every host connected to the hub. Sniffing in hubs networks becomes extremely easy, by simply plugging a packet monitor into an unused network connection.

[13]In switched networks, the principle of broadcasting is not anymore applicable. Instead, every frame is sent to the switch which then forwards it to the intended recipient. Only the destination will receive the message such that performing the previous attack scenario does not succeed anymore. To overcome this, a large number of methods has been proposed, including ARP spoofing, MAC flooding, MAC duplicating, ICMP redirection, DHCP spoofing and port stealing [Roo06]. Although these techniques are quite simple in theory, obtained results in practice are not always as expected. For example, by using only static ARP entries for each machine in the network, spoofed ARP replies are ignored. Surveillance tools (e.g. arpwatch) can monitor ARP activities and alert the administrator when suspicious changes occurs in the MAC-IP associations, while port security ensures that only allowed devices can connect to a switched port.

[14]In the "Hole 196" attack adversaries pretend to be the gateway. The clients seeing the malicious user as a gateway will respond with their PTKs, which allows the attacker to further intercept and decrypt their communication. Since the malicious user needs to know the group key (GTK) shared by the authorized users in the network, only insiders can exploit this vulnerability.

reach it. Especially for devices encompassing vulnerabilities, the operating environment is a significant factor in estimating the success likelihood of potential attacks. Note that positioning a system in a restricted location should not be seen as a compensation for deploying weak systems. Instead, it is best practice to strengthen both the system and its physical security. Though, there are several cases when business requires cryptographic modules to be located in open areas.

Three different operating environment types are identified by NIST [BR11], without a description of them to be provided:

1. Publicly accessible terminal.
2. Open-office environment.
3. Secure limited access facility.

According to the terminology used, the types enumerated by NIST are only company-related. We further refine this categorization such that it addresses any type of operating environment, independent of whether it is part of a company-restricted area, a public environment or a private location:

1. Public operating environment.
2. Open-office type environment.
3. Secure limited access facility.

A detailed description of each of the aforementioned operating environment types is provided below.

Public Operating Environment

The system operates in a public environment, to which any entity has unrestricted physical access. We identify here two cases:

- **(Unsupervised) publicly accessible terminal**. The system is located such that its terminal is accessible to any individual, without human supervision. There may be cases when surveillance technology (e.g. video cameras, etc.) is in place. If not live monitored, it rather helps for later detection of system misuse than as a countermeasure against intended attacks.

  A typical example of publicly accessible cryptographic modules are automated teller machines (ATMs), located at almost every street corner in major cities. Depending on their deployment, ATMs may provide adversaries with different types of access. The so called built-in ATMs attach to the wall of a business such that tellers access the device from inside the building and customers cannot see the hidden security feature. It also makes it much more difficult for attackers to physically tamper with

the device. On the other hand, freestanding ATMs are not attached to any building facade, anchoring to the ground for safety. Such ATMs are more susceptible to be stolen or tampered with.

Excluding the key stakeholders which we consider honest, any individual interacting with the module needs to follow the same usage rules (e.g. ATMs provide an interface restricting the actions users are allowed to perform). Therefore, differentiating between potential adversaries based on their privileges or roles is not applicable here. Targeting a publicly exposed device saves attackers from the need to enter restricted areas. Moreover, targets vulnerable to non-invasive side channel attacks might allow adversaries to discover the key without rising any suspicion. This reduces the security of unsupervised publicly accessible terminals to the security provided by its embodiment.

- **Public environment with private access only**. The system is operating in a public environment, but accessing it is restricted to authorized entities, e.g. using a smartcard to access the office laptop in a train. Compared to publicly accessible terminals where everyone is entitled to use the system, in this case any potential adversary can only reach the vicinity of the system. Obtaining unauthorized access to it strongly depends on the user's vigilance and device's strength.

  With respect to the possibility of performing side channel attacks, the same discussion as for unsupervised publicly exposed terminals applies.

Open-office Type Operating Environment

Open-office type operating environments refer to the system being located such that a generally large number of individuals have supervised access to it. Typical examples include working offices, banks or supermarkets, where any employee authorized to enter the building can reach the system.

We split this category into the following two subclasses:

- **Supervised public environment**. The system is (conditionally and/or temporally) open to public for usage, but under the supervision of entitled entities, whose responsibility is to ensure that the system is properly operated. When not under supervision, no public access is allowed to the area. The level at which the module is supervised may vary from simply monitoring (recall the ATM example located in a supermarket) to controlling it (e.g. a PoS controlled by a teller in a shop).

- **Open-office environment**. The system is located such that only employees authorized to enter the open-office area can access it. However, it is often presumed that the office is supervised by designated individuals or even informally by other employees, such that tampering with the system becomes an arduous task.

In case of ineffective supervision or unauthorized access to the system, the operating environment should be ranked as public.

Compared to public operating environments, open-office locations ensure that no system attacks can be conducted assuming an honest supervisor. If no public access is allowed to the area hosting the module, there is usually the need for adversaries to bypass access barriers. However, targets susceptible to side channel attacks might still leak cryptographic keys to outsiders.

Secure Limited Access Facility

Secure limited access facilities add an extra strength to the open-office environment category by considerably limiting the number of individuals having physical access to the system. An illustrative example is a locked room within a company. However, this is not the only scenario, e.g. think of an individual who uses a smartcard to access the office laptop while working from home.

We embrace the mentioned scenario types in the following two generic subclasses:

- **Private environment**. The system operates in a private environment, e.g. a home-type environment where only the house residents are entitled to enter the area.

  We do not expect everybody to agree with our decision of considering the private area as a subcategory of secure limited access facility. Based on the fact that an initial unauthorized entity may obtain temporary access to the area, one may notice some resemblance to the open-office type environment. Entering the location is here rather an informal process than a formal one (as in the case of open-office environments), e.g. it is often enough that adversaries make friend with individuals living in the target house in order to get in.

  However, the number of individuals entering the area is often expected to be limited. In addition, it is reasonable to think that people will store cryptographic modules of key storage devices in a working room, where guests are usually not invited. All these motivate our choice for enclosing the private environment as a subcategory of secure operating environment.

- **Secure limited access facility**. The system operates in a delimited, very well protected area inside a company's building, such that only a minimum number of employees or third parties, based on their attributions can pass the access barrier. Improperly enforced access control would require one to degrade the location type to "open-office" or to "public" operating environment, depending on the layers of access required.

  Secure limited access facility are designed to deny physical access to the cryptographic module for both regular employees (e.g. employees whose roles do not require access to the device) and outsiders. While most of the time insiders are required to bypass only one layer of defense to obtain unauthorized access to the restricted area, outsiders often have to enter the open-office environment and then to proceed to the target

location. Bypassing access control multiple times increases the risk of detection and therefore tends to make the attack prone to failure.

Assuming remote side channel attacks are possible, employees or left-alone guests may be able to exploit this vulnerability and discover cryptographic keys. Further, if emissions reach public areas without being altered, the same attack vector may be performed by outsiders too.

We conclude the analysis above by a three-level operating environment categorization listed in Table 5.2. When feasible, companies are advised to place cryptographic devices in secure areas rather than open-office and further, public environments.

Operating environment type	Description
Public environment	Unsupervised public operating environments and public environments with private access only.
Open-office type environment	Open-office and supervised public environments.
Secure areas	Secure limited access facilities and private environments.

Table 5.2: Operating environment classification

5.7 Insecure Networks

Misconfigured networks represent primary entry points for adversaries. To obtain cryptographic keys or gather relevant information in this scope, attackers can take advantage of insecure networks to reach nodes handling cryptographic keys (e.g. cryptographic key generation devices, storage media, etc.) or listen to network communication.

Depending on where they originate from, network attacks can be remote or local. Remote network attacks are launched from public networks and exploit vulnerabilities in the outside defenses of the network, such as firewall, proxy, or router vulnerabilities [CEH10]. Local attacks are launched by adversaries with direct access to the local network, being therefore more dangerous. Moreover, the growth of wireless networks (WLAN) make local attacks a real concern. As radio waves travel through building structures, the WLAN signal can be intercepted outside the building [CEH10]. Therefore, having physical access to the building may no longer represent a condition for the attack to succeed.

Since absolute security cannot be guaranteed, it usually turns out to be a matter of time until a network is compromised. How long such an attack is expected to take depends on how critical the network vulnerabilities contained are. Common network vulnerabilities that adversaries may exploit for compromising the network are discussed below:

- **Inattentive network device administration**. Common examples include:

  - **Default configurations**. Network devices come often ready-to-use [Far], distributed with weak default configurations, e.g., default administrator password, empty access control lists, etc. Unless immediately modified, even amateur attackers can use them to penetrate the network.
  - **Unpatched systems**. Exploitable bugs found in softwares or firmwares installed on network devices are published on security-related websites to alert the community on security vulnerabilities [Red]. Although usually published after being reported to the vendor and patches become available, system administrators do not always keep up with bug reports. Unpatching systems make network devices vulnerable to a variety of reported exploits.
  - **Information disclosure**. Any disclosed information guides adversaries in spotting vulnerabilities in the system. Common examples include technical or excessive details regarding software versions contained in banners or error messages.
  - **Unused services and open ports**. Unneeded services (e.g. Telnet, DHCP) running without administrators being aware of may be abused by adversaries in their attempt to access the network [Red].

- **Unprotected network cables**. Unprotected network cables might allow adversaries to gain unauthorized access to network medium by tapping. The technology for cable tapping has been around for years [Ham68]. According to former government officials, in the mid-1990s the NSA installed a tap on a cable located at hundreds of feet into the ocean.

  It has been shown that intruders can easily tap a fiber line without being detected through the use of a low-cost "Clip-on Coupler"[15] [MG03]. To protect against such attacks, a set of recommendations are provided in [MG03], including continuous, real-time, protocol independent, physical layer monitoring of the fiber network connection or preventing physical access to cables by buying the fiber in concrete.

- **Inherently insecure network devices**. Wireless access points broadcasting outside the building or hubs used to connect multiple network nodes are only few examples of inherently insecure network devices, easing eavesdropping on network traffic. The range of effects of deploying insecure network devices varies depending on the criticality of security flaws contained, possibly culminating with adversaries taking complete control over the network.

- **Insecure network hosts.** Network hosts often become easy targets for viruses or other malicious content being in general operated by regular users. Downloading and running malicious software or visiting infected websites are only few ways users unconsciously support adversaries in preparing attacks. Viruses can then quickly propagate

[15] The Clip-on Coupler is designed for bi-directional and full-duplex signal coupling on fibers. It provides non-intrusive and non-damaging optical coupling solution [GO4].

through the network and weaken its security (e.g. change device configurations) such that attackers can get access to.

When insecure networks are deployed, the confidentiality of the cryptographic key is mainly determined by the strength of the network nodes handling it (see Section 5.1). In some cases, rather than targeting network nodes, adversaries may find intercepting the key while communicated an easier attack vector. Eavesdropping opportunities depending on network types have been discussed in Section 5.5.

5.8 Insufficient Employee Training and Awareness

The **"totality of behavior patterns that come together to ensure protection of information resources of an organization"** [DTH07] defines the **information security culture** of the organization. It is part of the corporate culture, a collective phenomenon that is changing over time and can be influenced by the management of the organization [ST02].

Recent studies [Res11, Inc08] suggest a lack or poor promotion of information security cultures in organizations. Despite the general population becoming increasingly aware of the pervasive threats to information security, there continue to be information security violations that elude company's ability to defend against [Wor07]. It was just in May 2012 when a hacktivist group has managed to break into a company providing billing and customer support tech to several web hosts and leaked 500.000 records containing sensitive data (credit card numbers, usernames, passwords, IP addresses) [New12]. In addition, all the files residing on the target server were removed. The incident exploited the weakest link in the information security chain, humans. Applying social engineering techniques, the attackers fooled customer service representatives into providing admin credentials to target servers.

Social engineering attacks happen frequently. A survey made by Dimensional Research showed that 48% of large companies and 32% of companies of all sizes have experienced 25 or more social engineering attacks since 2009, as illustrated in Figure 5.2. The survey was conducted during July and August 2011 with the participation of 853 IT professionals including IT executives, IT managers and hands-on IT professionals from the United States, United Kingdom, Canada, Australia, New Zeeland and Germany [Res11]. Results pointed new employees, contractors and executive assistants as being the most susceptible to social engineering. The attacks proved to be costly, almost half of participants (48%) claiming a per incident cost of more than 25.000$. The need for addressing incorrect security behavior becomes thus obvious. Asked with respect to the approach companies take in order to increase user awareness of social engineering, only 26% of participants mentioned that ongoing training with employees is actively done.

Social engineering is only one of the information security threats exploiting the human character. Inclined to errors, humans make themselves responsible for many weaknesses which cannot be overcome by cryptographic systems on their own. Sharing confidential

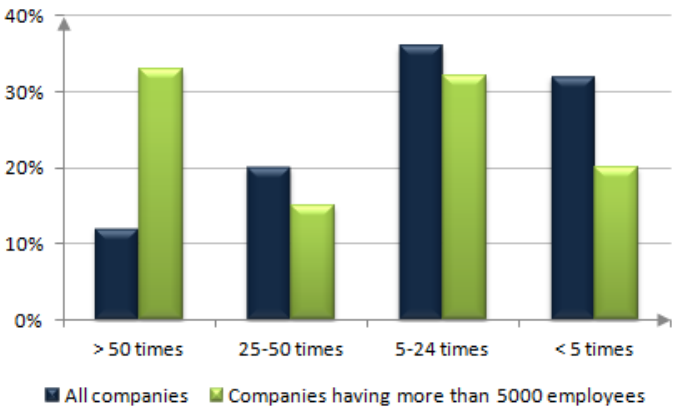

Figure 5.2: Frequency of social engineering attacks [Res11].

information with coworkers who do not have the same authorizations, transferring files to personal computers when working from home, leaving office doors opened and thus allowing for unauthorized access, leaving assets unprotected, etc. contribute as well to profiling a vulnerable human behavior. All these aspects were proved by an in-depth survey conducted by InsightExpress at the initiative of Cisco [Inc08]. Ten countries[16] selected based on the differences in their social and business cultures participated in the survey. A total of 2000 individuals were interviewed, 100 end users and 100 IT professionals from each country.

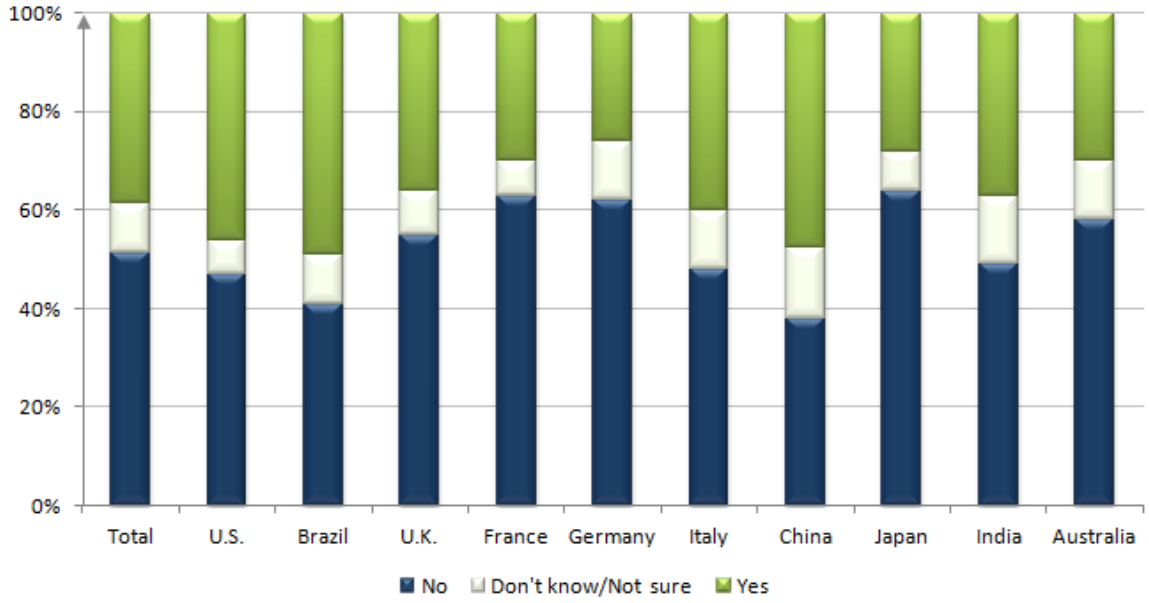

Figure 5.3: Response of participants for encountering unauthorized physical and network access [Inc08].

[16]United States, Unites Kingdom, France, Germany, Italy, Japan, China, India, Australia and Brazil

The results point out that 39% of IT professionals encountered unauthorized physical and network access, as pictured in Figure 5.3. This is mainly motivated by workers allowing for "tailgating" and accessing unauthorized parts of a corporate network or facility. Moreover, approximately one fourth of the employees surveyed admitted sharing sensitive information with family, acquaintances or even strangers, while almost half of the interviewed individuals share work devices with people outside the company, without supervising them [Inc08]. 46% of employees also admitted that transfer files between work and personal computers.

To cope with key exposure attacks, organizations should proceed by understanding the potential unconscious contribution of untrained and unaware employees to such malicious acts. Recommendations are that organizations analyze their actual information security culture and change it such that it fits with the organization's target [STR05]. The stronger the information security culture becomes, the more difficult is expected to be for adversaries to take advantage of the human factor.

5.9 Summary

Table 5.3 summarizes the discussion in this chapter.

Guidance w.r.t. common vulnerabilities	
Cryptographic modules	To estimate how secure cryptographic modules are, companies are advised to evaluate them against widely-recognized criteria. A largely accepted evaluation standard is FIPS 140, discussed in Section 5.1. The embodiment strength tends to increase from **FIPS 140-3 level 1** certified modules, to **FIPS 140-3 level 4** modules. For key-generating modules, recommendations are that only random number generators that have been intensively reviewed by specialized organizations should be used. Practitioners can refer to [CE11] for a list of NIST-approved pseudo-RNGs.
Key storage	The likelihood of compromising key storage solutions when physical access to them is assumed tends to increase from **hard-copies**, **regular devices** (e.g.smartphones, PCs) and **removable storage media** (e.g. USB drives, CDs) to **economic cryptographic tokens** (e.g. smartcards) and further to **highly-secure HSMs** (e.g. FIPS 140-3 level 3 or 4 certified modules).
Insecure cryptographic schemes	Proprietary or cryptographic schemes for which vulnerabilities are publicly known should be avoided. At the time of writing, NIST recommends the use of AES and three-key 3DES as encryption algorithms. For estimating attack costs for brute-forcing keys depending on the computational power increase over the years, the model proposed by Lenstra and Verheul [LV01] can be used (see Section 5.3).

Key transport channel	Unless physically/logically protected, keys manually managed across large distances or electronically transported over unrestricted channels are likely to get disclosed.
Operating environment	Performing key management operations or storing cryptographic keys in **secure areas** makes it less likely for adversaries to physically reach the target. The likelihood tends to increase for **open-office type environments** and further for **public environments**.
Insecure networks	Building attack-resistant networks is definitely a challenging task. To evaluate how likely is that cryptographic keys will leak due to insecure networks, companies are advised to perform network penetration tests.
Employee training and awareness	Surveys have shown that almost one out of two companies encountered unauthorized physical and network access due to poor employee behavior with respect to security. In general, employees not attending periodical security training and awareness programs tend to be susceptible to human mistakes and unwillingly contribute to key disclosure attacks. Such a behavior is expected to be less of a concern when referring to individuals with high experience in the security field.

Table 5.3: Summary of common key exposure vulnerabilities.

Chapter 6

Conclusions and Future Research

In this final chapter we review the research performed in the thesis and discuss directions for further research.

6.1 Conclusions

This thesis sets out to investigate the key renewal period problem and to guide practitioners into determining key renewal intervals. In particular, main research contributions are:

- **A clear definition of the key renewal period problem** was included in the beginning of the thesis. It is meant to support practitioners in understanding the challenges behind key renewal interval estimation. Aiming to protect key confidentiality by limiting key usage intervals, while counting on vulnerable key changing processes is what makes the problem complex. We therefore reduced the key renewal period problem to the need of identifying a point in time when the risks caused by overlong key usage and respectively by key change are balanced.

- **A review of literature recommending key renewal intervals** was provided in Chapter 2. The analysis brings out a lack of mature research in the topic area. Most studies are limited to considering only key resistance to brute-force attacks and recommend key lengths to ensure key confidentiality up to specific years. NIST broadens the key renewal interval analysis by identifying a set of influencing factors. Yet, no framework stating the role each of the factors plays is proposed.

- **A methodology for guiding practitioners into the key renewal interval estimation process** was proposed in Chapter 4. To the author's knowledge, this is the first key renewal interval estimation methodology having organization's key management environment as a differentiator. It is based on risk assessment for estimating key exposure risks caused by key usage and key change. The influencing factors listed

by NIST and some additional ones we identified were used to spot vulnerabilities in the key management environment. A step-by-step description of the risk assessment was included to guide practitioners through the entire process.

Key renewal periods were analyzed as a response to the identified risks. Concrete numerical values for renewal intervals were not proposed, since we consider they should be based on extensive practice in the field. Instead, we focused on investigating the behavior of the interval depending on the assessed key exposure risks.

- **General guidance w.r.t common vulnerabilities** was included in Chapter 5, being mainly addressed to practitioners and readers with limited knowledge in vulnerability identification and likelihood determination. It brings together several concrete examples of how systems may fail to protect key confidentiality as well as best practices to prevent key exposure attacks from happening.

6.2 Future Work

This work is still in progress to better support practitioners in selecting key renewal periods.

First of all, much research remains to be done to refine the methodology discussed in this thesis. For example, a more in-depth discussion on risk change over time as well as a finer-grained analysis depending on the gap between R_U and R_C might prove valuable to key renewal period estimation. Moreover, extending the analysis to provide numerical estimations for key renewal intervals would definitely represent a significant improvement of the methodology. Still, the feasibility of such a task remains to be investigated.

Secondly, we consider extending the guidance in Chapter 5 and create comprehensive databases of potential key management vulnerabilities. This is mainly meant to support inexperienced practitioners who might overlook critical vulnerabilities and therefore generate unrealistic risk assessment results. Another future research direction aims at easing the identification of potential attack scenarios by developing a representative set of key exposure attack trees.

For large organizations, applying the methodology may require more time than they are willing to invest. To enhance the applicability of the methodology, a tool to automate the entire process could prove expedient. We know that several risk assessment tools exist, an inventory of some of them being provided by ENISA in [ENIc]. The degree of applicability of existent tools to the key exposure risk assessment is to be investigated. The selected tool need to be further expanded with key renewal interval controls.

Finally, the validation of the model in practice needs to be done. It would be of interest to apply the methodology on several key environments and to identify the key management processes most companies tend to get wrong. Building a top of most common key management shortcomings encountered at companies would be valuable as well.

List of Figures

List of Tables

Bibliography

[ADDS91] D. G. Abraham, G. M. Dolan, G. P. Double, and J. V. Stevens. Transaction Security System. *IBM Systes Journal*, 30(2):206–229, 1991. [cited at p. 25, 26, 68]

[Att02] J. Attridge. An Overview of Hardware Security Modules, January 2002. [cited at p. 45, 47]

[Aum11] J. Aumasson. Cryptanalysis vs. Reality. BLACK HAT ABU DHABI, 2011. [cited at p. 49, 50]

[BA01] E. Barker and A.Roginsky. Recommendation for cryptographic key generation, NIST 800-133, July 2001. [cited at p. viii, 13, 14, 49]

[BB84] C.H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, page 175, India, 1984. [cited at p. 17]

[BB93] E. Biham and A. Biryukov. An Improvement of Davies' Attack on DES. In *Journal of Cryptology*, pages 461–467. Springer-Verlag, 1993. [cited at p. 50]

[BBCS10] E. Barker, D. Branstad, S. Chokhani, and M. Smid. A Framework for Designing Cryptographic Key Management Systems, NIST SP 800-130, June 2010. [cited at p. vii, viii, 1, 7, 13, 18, 40, 41, 67]

[BCEK07] T. Bradley, A. Chuvakin, A. Elberg, and B.J. Koerner. *PCI Compliance: Understand and Implement Effective PCI Data Security Standard Compliance.* Syngress Publishing, 2007. [cited at p. 9]

[BDR+96] M. Blaze, W. Diffie, R. Rivest, B. Schneier, T. Shimomur, E. Thompson, and M. Wiener. Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security. Report of ad hoc panel of cryptographers and computer scientists, January 1996. http://www.crypto.com/papers/. [cited at p. 10]

[BE02] H. Bar-El. Security Implications of Hardware vs. Software Cryptographic Modules. Technical report, Discretix Technologies Ltd., 2002. [cited at p. 44]

[Ben92] C. H. Bennett. *Physical Review Letters*, 68:3121–3124, 1992. [cited at p. 17]

[BFM04] E.J. Byres, M. Franz, and D. Miller. The Use of Attack Trees in Assessing Vulnerabilities in SCADA Systems. In *Proceedings of the International Infrastructure Survivability Workshop*, Lisbon, Portugal, 2004. [cited at p. 30]

[BKR11] A. Bogdanov, D. Khovratovich, and C. Rechberger. Biclique Cryptanalysis of the full AES. *IACR Cryptology ePrint Archive*, page 449, 2011. [cited at p. 50]

[BR11] E. Barker and A. Roginsky. Recommendation for key management - Part 1: General, NIST SP 800-57, January 2011. [cited at p. viii, 2, 4, 7, 8, 10, 12, 15, 34, 41, 56, 67, 68]

[Bra11] D. Bradbury. Hacking wifi the easy way. *Network Security*, (2):9 – 12, 2011. [cited at p. 55]

[Bre11] W. Brecht. White-Box Cryptography. In Henk C.A. van Tilbog and Sushil Jajodia, editors, *Encyclopedia of Cryptography and Security*, pages 1386–1387. Springer US, 2011. [cited at p. 45]

[BS93] E. Biham and A. Shamir. Differential cryptanalysis of the full 16-round DES. pages 487–496. Springer-Verlag, 1993. [cited at p. 50]

[Car01] D. A. Carts. SANS Institute, A Review of the Diffie-Hellman Algorithm and its Use in Secure Internet Protocols, 2001. http://www.sans.org/reading_room/whitepapers/vpns/review-diffie-hellman-algorithm-secure-internet-protocols_751. [cited at p. 15]

[CC09] C. Cachin and N. Chandran. A Secure Cryptographic Token Interface. In *CSF*, pages 141–153, 2009. [cited at p. 48]

[CE11] J. Campbell and R. J. Easter. Approved Random Number Generators Draft for FIPS PUB 140-2; Security Requirements for Cryptographic Modules, July 2011. [cited at p. 46, 63]

[CEH10] *CEH (Certified Ethical Hacker) V6. 1: Exam 312-50*. Crisp Publications, Incorporated, 2010. [cited at p. 59]

[CER] US-CERT, Vulnerability Notes Database. http://www.kb.cert.org/vuls/. [cited at p. 30]

[Cha12] K. Chang. I.B.M. Researchers Inch Toward Quantum Computer, February 2012. http://www.nytimes.com/2012/02/28/technology/ibm-inch-closer-on-quantum-computer.html. [cited at p. 53]

[Cla88] A. Clark. Physical Protection of Cryptographic Devices. In David Chaum and Wyn Price, editors, *Advances in Cryptology EUROCRYPT 87*, volume 304 of *Lecture Notes in Computer Science*, pages 83–93. Springer Berlin / Heidelberg, 1988. [cited at p. 43]

[Con] Rycombe Consulting. [cited at p. 47]

[CWE] Common Weakness Enumeration. http://cwe.mitre.org/. [cited at p. 30]

[DH76] W. Diffie and M. E. Hellman. New Directions in Cryptography, 1976. [cited at p. 15]

[DH01] D. Dubois and H.Prade. Possibility Theory, Probability Theory and Multiple-Valued Logics: A Clarification. *Annals of Mathematics and Artificial Intelligence*, 32(1-4):35–66, August 2001. [cited at p. 33]

[DS09] M. Deeg and S. Schreiber. Cryptographycally Secure SySS Cracks a USB Flash Drive, December 2009. [cited at p. 47]

[DTH07] G. Dhillon, G. Tejay, and W. Hong. Identifying Governance Dimensions to Evaluate Information Systems Security in Organizations. In *HICSS*, page 157, 2007. [cited at p. 61]

[ENIa] ENISA Briefing: Quantum Key Distribution. http://www.enisa.europa.eu/activities/risk-management/files/deliverables/briefing-quantum-key-distribution. [cited at p. 16]

[ENIb] ENISA. Being diabetic in 2011, identifying emerging and future risks in remote health monitoring and treatment; annex ii the risk analysis report. http://www.enisa.europa.eu/activities/risk-management/files/deliverables/enisa_being_diabetic_2011_Annex2.pdf/view. [cited at p. 24]

[ENIc] ENISA. Inventory of Risk management/Risk assessment methods and tools. http://rm-inv.enisa.europa.eu/rm_ra_tools.html. [cited at p. 66]

[ENId] ENISA. Risk Treatment. http://www.enisa.europa.eu/activities/risk-management/current-risk/risk-management-inventory/rm-process/risk-treatment. [cited at p. viii]

[ENI11] ENISA. *Study on the use of cryptographic techniques in Europe.* December 2011. [cited at p. 15, 16]

[EYZ10] G. Elahi, E. Yu, and N. Zannone. A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities. *Requir. Eng.*, 15(1):41–62, March 2010. [cited at p. 24, 25]

[Far] F. Farhat. Eavesdropping attack over Wi-Fi. [cited at p. 60]

[FIP] Validated FIPS 140-1 and FIPS 140-2 Cryptographic Modules. http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val1997.htm. [cited at p. 48, 67]

[FIP94] Security requirements for cryptographic modules, FIPS PUB 140-1, June 1994. [cited at p. vii, 15]

[FIP01] Security requirements for cryptographic modules, FIPS PUB 140-2, May 2001. [cited at p. 17, 47, 68, 82]

[FIP09] Security requirements for cryptographic modules FIPS PUB 140-3, Revised DRAFT. September 2009. [cited at p. vii, 47, 68, 84]

[FNI10] FNISA. Mcanismes cryptographiques - Rgles et recommandations, Rev. 1.20, January 2010. http://www.ssi.gouv.fr/site_documents/politiqueproduit/Mecanismes_cryptographique_v1_10_standard.pdf. [cited at p. 10, 12, 67]

[Gir11] D. Giry. Cryptography Key length recommendation, October 2011. [cited at p. 10]

[GO4] GO4FIBER. [cited at p. 60]

[GP00] D. Gottesman and J. Preskill. Secure quantum key exchange using squeezed states, 2000. [cited at p. 17]

[GSF02] A. Goguen G. Stoneburner and A. Feringa. Risk Management Guide for Information Technology Systems Recommendations of the National Institute of Standards and Technology , NIST 800-30, July 2002. [cited at p. vii, viii, 20, 24, 36, 37, 67, 68]

[Ham68] P. D. Hamlin. Patent. Method and Apparatus for tapping a coaxial cable, July 1968. [cited at p. 60]

[Har09] S. Harris. *CISSP Video Mentor*. Que Publishing Company, 1st edition, 2009. [cited at p. 16]

[HKS12] M. Habeeb, D. Kahrobaei, and V. Shpilrain. A Secret Sharing Scheme Based on Group Presentations and the Word Problem. Cryptology ePrint Archive, Report 2012/246, 2012. http://eprint.iacr.org/. [cited at p. 49]

[HMLS10] D. Hofheinz, J. Malone-Lee, and M. Stam. Obfuscation for cryptographic process. 23:121–168, 2010. [cited at p. 45]

[Hog09] G. Hogben. ENISA Briefing: Quantum Key Distribution, November 2009. [cited at p. 17]

[Inc08] Cisco System Inc. Data Leakage Worldwide: Common Risks and Mistaked Employees Make. http://www.cisco.com/en/US/solutions/collateral/ns170/ns896/ns895/white_paper_c11-499060.pdf, 2008. [cited at p. 61, 62, 63, 67]

[ISO04] ISO/IEC. Management of Information and Communication Technology Security - Part 1: Concepts and Models for Information and Communication Technology Security Management. ISO/IEC 13335, 2004. [cited at p. vii, ix]

[Jan08] J. Jansen. An introduction to the use of HSM, 2008. http://www.nic.cz/files/nic/doc/hsm.pdf. [cited at p. 45]

[JR05] J.M. Johansson and S. Riley. *Protect Your Windows Network: From Perimeter to Data (Microsoft Technology)*. Addison-Wesley Professional, 2005. [cited at p. 55]

[JWP11] R. C. Phan J. Wang, J. N. Whitley and D. J. Parish. Uniffied Parametrizable Attack Tree. *International Journal for Information Security Research (IJISR)*, 1, March/June 2011. [cited at p. 26]

[Kel] S. S. Keller. The Key Agreement Schemes Validation System (KASVS). [cited at p. 14]

[Kis11] R. Kissel. Glossary of Key Information Security Terms, NIST IR 7298, Revision 1, february 2011. [cited at p. viii]

[KK11] K. Wall J. Manico F. Donovan K. Kenan, D. Rook. The Open Web Application Security Project, September 2011. [cited at p. 1]

[Kri11] R. Krikken. Cryptography, September 2011. [cited at p. 2, 51]

[Luc98] S. Lucks. Attacking Triple Encryption. In *FSE*, pages 239–253, 1998. [cited at p. 50]

[LV01] A.K. Lenstra and E.R. Verheul. Selecting Cryptographic Key Sizes. *J.Cryptology*, 14(4):255–293, 2001. [cited at p. 10, 11, 12, 63, 67, 80]

[LWW+10] L. Lydersen, C. Wiechers, C. Wittmann, D. Elser, J. Skaar, and V. Makarov. Hacking commercial quantum cryptography systems by tailored bright illumination. *Nature Photonics*, 4(10):686–689, August 2010. [cited at p. 17]

[Mat94] M. Matsui. The First Experimental Cryptanalysis of the Data Encryption Standard. In *CRYPTO*, pages 1–11, 1994. [cited at p. 50]

[MG03] R.J. Bagnali M. Gross. Threats to Fiber-Optic Infrastructure, October 2003. [cited at p. 60]

[MO05] S. Mauw and M. Oostdijk. Foundations of Attack Trees. In *International Conference on Information Security and Cryptology ICISC 2005. LNCS 3935*, pages 186–198. Springer, 2005. [cited at p. 26]

[Net] Ingrian Networks. A Strategic Approach to Enterprise Key Management . http://www.infosecurityproductsguide.com/technology/2007/Enterprise_Key_Management_WP.pdf. [cited at p. 15]

[New12] E Hacking News. Ugnazi hacked whmcs by social engineering attack, May 2012. [cited at p. 61]

[NIS11a] Managing Information Security Risk, NIST SP 800-39, March 2011. http://csrc.nist.gov/publications/nistpubs/800-39/SP800-39-final.pdf. [cited at p. 20, 21, 22, 40, 67]

[NIS11b] NIST. Guide for Conducting Risk Assessments, NIST 800-30 Revision 1, September 2011. [cited at p. viii, 20, 21, 23, 24, 25, 30, 32, 34, 67, 68]

[NVD] National Vulnerability Database. http://nvd.nist.gov/. [cited at p. 30]

[OEFC11] Ecrypt European Network Of Excellence For Cryptology. ECRYPT II Yearly Report on Algorithms and Keysizes (2010-2011) Revision 1.0. *Framework*, (March):116, June 2011. [cited at p. 10, 11, 12, 51, 52, 67, 68]

[Opp01] R. Oppliger. *Internet and Intranet Security, second edition.* Artech House, Inc., Norwood, MA, USA, 2001. [cited at p. 16]

[Pel06] J. Pelz. Cryptanalysis with a cost-optimized FPGA cluster, December 2006. http://www.copacobana.org/paper/IPAM2006_slides.pdf. [cited at p. 51]

[Pet10] M. Petros. Leakage-Resilient Cryptography: A Survey of Recent Advances, 2010. http://cseweb.ucsd.edu/~pmol/Documents/RE.pdf. [cited at p. 45]

[PM02] F. Piper and S. Murphy. *Cryptography: A Very Short Introduction.* Oxford University Press, 2002. [cited at p. 16]

[PM07] S. Romanosky P. Mell, K. Scarfone. A Complete Guide to the Common Vulnerability Scoring System Version 2.0, June 2007. http://www.first.org/cvss/cvss-guide.pdf. [cited at p. vii, 31]

[PPS04] K. G. Paterson, F. Piper, and R. Schack. Quantum cryptography: a practical information security perspective. Cryptology ePrint Archive, Report 2004/156, 2004. http://eprint.iacr.org/. [cited at p. 17]

[RDZ11] D. Ren, S. Du, and H. Zhu. A Novel Attack Tree Based Risk Assessment Approach for Location Privacy Preservation in the VANETs. In *ICC*, pages 1–5, 2011. [cited at p. 26]

[Red] Red Hat Documentation. http://docs.redhat.com/docs/en-US/index.html. [cited at p. 60]

[Res11] Dimensional Research. The Risk of Social Engineering on Information Security. A Survey of IT Professionals., September 2011. [cited at p. viii, 61, 62, 67]

[Roo06] SANS Institute InfoSec Reading Room. Packet Sniffing In a Switched Environment, June/July 2006. [cited at p. 55]

[SAN] SANS top-20 annual security risks. http://www.sans.org. [cited at p. 30]

[Sch99] B. Schoenmakers. A Simple Publicly Verifiable Secret Sharing Scheme and Its Application to Electronic. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 148–164, London, UK, UK, 1999. Springer-Verlag. [cited at p. 49]

[Sch08] B. Schneier. Quantum Cryptography, October 2008. [cited at p. 53]

[Sci08] SciEngines. Breaking DES in less than a single day, 2008. http://www.sciengines.com/company/news-a-events/74-des-in-1-day.html. [cited at p. 51]

[SDFF06] M. Simka, M. Drutarovský, V. Fischer, and J. Fayolle. Model of a true random number generator aimed at cryptographic applications. In *ISCAS*, 2006. [cited at p. 46]

[Sec] Secunia advisories. http://secunia.com/community/advisories/. [cited at p. 30]

[Sha79] A. Shamir. How to share a secret. *Commun. ACM*, 22:612–613, November 1979. [cited at p. 16]

[She11] A. Sheeja. Timestamped Key Management for MissionCritical WAHNETs. *International Journal of Computer Applications in Engineering Sciences (IJCAES)*, 1, March 2011. [cited at p. 13]

[Shi] J. Shikata. Unconditional security. http://www.jsps.go.jp/j-bilat/fos_jf/data/jishi_02/abstract/05.pdf. [cited at p. ix]

[SLSD11] S. Sanjeev, J. Lodhia, R. Srinivasan, and P. Dasgupta. Protecting Cryptographic Keys on Client Platforms Using Virtualization and Raw Disk Image Access. In *SocialCom/PASSAT*, pages 1026–1032, 2011. [cited at p. 48]

[SPY+10] F. Standaert, O. Pereira, Y. Yu, J. Quisquarter, M. Yung, and E. Oswald. Leakage resilient cryptography in practice. In David Sadeghi, Ahmad-Reza an Naccache, editor, *Towards Hardware-Intrinsic Security*, Information Security and Cryptography, pages 99–134. Springerlink Berlin Heidelber, 2010. [cited at p. 45]

[ST02] T. Schlienger and S. Teufel. Information Security Culture: The Socio-Cultural Dimension in Information Security Management. In *SEC*, pages 191–202, 2002. [cited at p. 61]

[Sta07] British Standards. Code of practice for information security management, BS ISO/IEC 27002:2005 BS 7799-1:2005, July 2007. [cited at p. vii, viii, 1, 54]

[Sta08] British Standards. Information technology Security techniques Information security risk management, BS ISO/IEC 27005:2008, 2008. [cited at p. viii, 21, 24, 25, 36, 68, 78]

[Sta10] F. Standaert. Introduction to Side-Channel Attacks Secure Integrated Circuits and Systems. Integrated Circuits and Systems, chapter 2, pages 27–42. Springer US, Boston, MA, 2010. [cited at p. 44]

[STR05] T. Schlienger, S. Teufel, and R. Rues. Information Security Culture Status Quo in Switzerland. In *ISSA*, 2005. [cited at p. 63]

[SWV+09] D. Stucki, N. Walenta, F. Vannel, R.T. Thew, N. Gisin, H. Zbinden, S. Gray, C.R. Towery, and S. Ten. High rate, long-distance quantum key distribution over 250 km of ultra low loss fibres. *New Journal of Physics*, 11(7):075003, 2009. [cited at p. 17]

[Sys99] B. Schneier (Counterpane Systems). Attack Trees, October 1999. http://web.eecs.utk.edu/~dunigan/cns04/attacktrees.pdf. [cited at p. 25, 26, 27, 67]

[THA] THALES. Key Management Q&A. http://keymanagement.typepad.com/blog/key-management-information.html. [cited at p. 13]

[Tir07] K. Tiri. Side-Channel Attack Pitfalls. In *Design Automation Conference. DAC'07. 44th ACM/IEEE*, pages 15–20, June 2007. [cited at p. 44]

[Tzu88] S. Tzu. *The Art of War: Translation by Samuel B. Griffith.* Oxford, U.K., 1988. [cited at p. 24]

[VIS08] VISA. PIN Security Program, Auditor's Guide, January 2008. [cited at p. 16, 54]

[VX10] N. Vavoulas and C. Xenakis. A Quantitative Risk Analysis Approach for Deliberate Threats. In *CRITIS*, pages 13–25, 2010. [cited at p. 25]

[Wal93] F. Walter. Key management. In Bart Preneel, Ren Govaerts, and Joos Vandewalle, editors, *Computer Security and Industrial Cryptography*, volume 741 of *Lecture Notes in Computer Science*, pages 132–147. Springer Berlin / Heidelberg, 1993. 10.1007/3-540-57341-0_58. [cited at p. 46]

[Wen01] M. Wenstrom. *Managing Cisco Network Security.* Cisco Press, 2001. [cited at p. 46]

[Wor07] M. Workman. Gaining Access with Social Engineering: An Empirical Study of the Threat. *Inf. Sys. Sec.*, 16(6):315–331, November 2007. [cited at p. 61]

[Wor08] Atos Worldline. Hardware Security Modules, 2008. http://www.secappdev.org/handouts/2009/hardware%20security%20modules.pdf. [cited at p. 45]

[YD05] Z. YongBin and F. DengGuo. Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing. Cryptology ePrint Archive, Report 2005/388, 2005. http://eprint.iacr.org/. [cited at p. 44]

[ZXK10] D. Zhenlong, L. Xiaoli, and S. Kangkang. Trusted firmware services based on TPM. In *Proceedings of the First international conference on Trusted Systems*, INTRUST'09, pages 227–235, Berlin, Heidelberg, 2010. Springer-Verlag. [cited at p. 46]

Appendix A

Plan-do-act-check Risk Management Cycle

ISMS	Information Security Risk Management Process
Plan	Establishing the context. Risk assessment. Developing risk treatment plan. Risk acceptance.
Do	Implementation of risk treatment plan.
Check	Continual monitoring and reviewing of risks.
Act	Maintain and improve the information security risk management.

Table A.1: Alignment of Information Security Management System (ISMS) and Information Security Risk Management Process [Sta08].

Appendix B

Key Length Recommendations

Year	Symmetric Key Size	Classical Asymmetric Key Size and SDL Field Size	SDL Key Size	Elliptic Curve Key Size $c = 0$	Elliptic Curve Key Size $c = 18$	Infeasible number of Mips-Years	Lower bound for hardware cost in US$ for a 1 day attack (cf. 4.5)	Corresponding number of years on a 450MHz Pentium II PC
1982	56	417 288	102	105	*85*	$5.00 * 10^5$	$3.98 * 10^7$	$1.11 * 10^3$
1984	58	463 320	105	108	*89*	$1.45 * 10^6$	$4.57 * 10^7$	$3.22 * 10^3$
1986	60	513 352	107	111	*96*	$4.19 * 10^6$	$5.25 * 10^7$	$9.31 * 10^3$
1988	61	566 384	109	114	*101*	$1.21 * 10^7$	$6.04 * 10^7$	$2.69 * 10^4$
1990	63	622 448	112	117	*106*	$3.51 * 10^7$	$6.93 * 10^7$	$7.80 * 10^4$
1991	63	652 448	113	119	*109*	$5.97 * 10^7$	$7.43 * 10^7$	$1.33 * 10^5$
1992	64	682 480	114	120	*112*	$1.02 * 10^8$	$7.96 * 10^7$	$2.26 * 10^5$
1993	65	713 512	116	121	*114*	$1.73 * 10^8$	$8.54 * 10^7$	$3.84 * 10^5$
1994	66	744 544	117	123	*117*	$2.94 * 10^8$	$9.15 * 10^7$	$6.53 * 10^5$
1995	66	777 544	118	124	*121*	$5.00 * 10^8$	$9.81 * 10^7$	$1.11 * 10^6$
1996	67	810 576	120	126	*122*	$8.51 * 10^8$	$1.05 * 10^8$	$1.89 * 10^6$
1997	68	844 608	121	127	*125*	$1.45 * 10^9$	$1.13 * 10^8$	$3.22 * 10^6$
1998	69	879 640	122	129	*129*	$2.46 * 10^9$	$1.21 * 10^8$	$5.48 * 10^6$
1999	70	915 672	123	130	130	$4.19 * 10^9$	$1.29 * 10^8$	$9.31 * 10^6$
2000	70	952 704	125	132	132	$7.13 * 10^9$	$1.39 * 10^8$	$1.58 * 10^7$
2001	71	990 736	126	133	135	$1.21 * 10^{10}$	$1.49 * 10^8$	$2.70 * 10^7$
2002	72	1028 768	127	135	139	$2.06 * 10^{10}$	$1.59 * 10^8$	$4.59 * 10^7$
2003	73	1068 800	129	136	140	$3.51 * 10^{10}$	$1.71 * 10^8$	$7.80 * 10^7$
2004	73	1108 832	130	138	143	$5.98 * 10^{10}$	$1.83 * 10^8$	$1.33 * 10^8$
2005	74	1149 864	131	139	147	$1.02 * 10^{11}$	$1.96 * 10^8$	$2.26 * 10^8$
2006	75	1191 896	133	141	148	$1.73 * 10^{11}$	$2.10 * 10^8$	$3.84 * 10^8$
2007	76	1235 928	134	142	152	$2.94 * 10^{11}$	$2.25 * 10^8$	$6.54 * 10^8$
2008	76	1279 960	135	144	155	$5.01 * 10^{11}$	$2.41 * 10^8$	$1.11 * 10^9$
2009	77	1323 1024	137	145	157	$8.52 * 10^{11}$	$2.59 * 10^8$	$1.89 * 10^9$

2010	78	1369	1056	138	146	160	$1.45 * 10^{12}$	$2.77 * 10^{8}$	$3.22 * 10^{9}$
2011	79	1416	1088	139	148	163	$2.47 * 10^{12}$	$2.97 * 10^{8}$	$5.48 * 10^{9}$
2012	80	1464	1120	141	149	165	$4.19 * 10^{12}$	$3.19 * 10^{8}$	$9.32 * 10^{9}$
2013	80	1513	1184	142	151	168	$7.14 * 10^{12}$	$3.41 * 10^{8}$	$1.59 * 10^{10}$
2014	81	1562	1216	143	152	172	$1.21 * 10^{13}$	$3.66 * 10^{8}$	$2.70 * 10^{10}$
2015	82	1613	1248	145	154	173	$2.07 * 10^{13}$	$3.92 * 10^{8}$	$4.59 * 10^{10}$
2016	83	1664	1312	146	155	177	$3.51 * 10^{13}$	$4.20 * 10^{8}$	$7.81 * 10^{10}$
2017	83	1717	1344	147	157	180	$5.98 * 10^{13}$	$4.51 * 10^{8}$	$1.33 * 10^{11}$
2018	84	1771	1376	149	158	181	$1.02 * 10^{14}$	$4.83 * 10^{8}$	$2.26 * 10^{11}$
2019	85	1825	1440	150	160	185	$1.73 * 10^{14}$	$5.18 * 10^{8}$	$3.85 * 10^{11}$
2020	86	1881	1472	151	161	188	$2.94 * 10^{14}$	$5.55 * 10^{8}$	$6.54 * 10^{11}$
2021	86	1937	1536	153	163	190	$5.01 * 10^{14}$	$5.94 * 10^{8}$	$1.11 * 10^{12}$
2022	87	1995	1568	154	164	193	$8.52 * 10^{14}$	$6.37 * 10^{8}$	$1.89 * 10^{12}$
2023	88	2054	1632	156	166	197	$1.45 * 10^{15}$	$6.83 * 10^{8}$	$3.22 * 10^{12}$
2024	89	2113	1696	157	167	198	$2.47 * 10^{15}$	$7.32 * 10^{8}$	$5.48 * 10^{12}$
2025	89	2174	1728	158	169	202	$4.20 * 10^{15}$	$7.84 * 10^{8}$	$9.33 * 10^{12}$
2026	90	2236	1792	160	170	205	$7.14 * 10^{15}$	$8.41 * 10^{8}$	$1.59 * 10^{13}$
2027	91	2299	1856	161	172	207	$1.21 * 10^{16}$	$9.01 * 10^{8}$	$2.70 * 10^{13}$
2028	92	2362	1888	162	173	210	$2.07 * 10^{16}$	$9.66 * 10^{8}$	$4.59 * 10^{13}$
2029	93	2427	1952	164	175	213	$3.52 * 10^{16}$	$1.04 * 10^{9}$	$7.81 * 10^{13}$
2030	93	2493	2016	165	176	215	$5.98 * 10^{16}$	$1.11 * 10^{9}$	$1.33 * 10^{14}$
2032	95	2629	2144	168	179	222	$1.73 * 10^{17}$	$1.27 * 10^{9}$	$3.85 * 10^{14}$
2034	96	2768	2272	171	182	227	$5.01 * 10^{17}$	$1.46 * 10^{9}$	$1.11 * 10^{15}$
2036	98	2912	2400	173	185	232	$1.45 * 10^{18}$	$1.68 * 10^{9}$	$3.22 * 10^{15}$
2038	99	3061	2528	176	188	239	$4.20 * 10^{18}$	$1.93 * 10^{9}$	$9.33 * 10^{15}$
2040	101	3214	2656	179	191	244	$1.22 * 10^{19}$	$2.22 * 10^{9}$	$2.70 * 10^{16}$
2042	*103*	*3371*	*2784*	*182*	*194*	*248*	$3.52 * 10^{19}$	$2.55 * 10^{9}$	$7.82 * 10^{16}$
2044	*104*	*3533*	*2944*	*185*	*197*	*255*	$1.02 * 10^{20}$	$2.93 * 10^{9}$	$2.26 * 10^{17}$
2046	*106*	*3700*	*3072*	*187*	*200*	*260*	$2.95 * 10^{20}$	$3.36 * 10^{9}$	$6.55 * 10^{17}$
2048	*107*	*3871*	*3232*	*190*	*203*	*265*	$8.53 * 10^{20}$	$3.86 * 10^{9}$	$1.90 * 10^{18}$
2050	*109*	*4047*	*3392*	*193*	*206*	*272*	$2.47 * 10^{21}$	$4.44 * 10^{9}$	$5.49 * 10^{18}$

Figure B.1: Lower bounds for computationally equivalent key sizes [LV01].

Note that recommendations for classical asymmetric, SDL and elliptic curve key sizes are included as well. The information relevant for symmetric ciphers is highlighted by a light green frame.

Appendix C

FIPS 140-2 Certification

	Security Level 1	**Security Level 2**	**Security Level 3**	**Security Level 4**
Cryptographic Module Specification	Specification of cryptographic module, cryptographic boundary, Approved algorithms, and Approved modes of operation. Description of cryptographic module, including all hardware, software, and firmware components. Statement of module security policy			
Cryptographic Module Ports and Interfaces	Required and optional interfaces. Specification of all interfaces and of all input and output data paths.		Data ports for unprotected critical security parameters logically or physically separated from other data ports.	
Roles, Services, and Authentication	Logical separation of required and optional roles and services.	Role-based or identity-based operator authentication.	Identity-based operator authentication.	
Finite State Model	Specification of finite state model. Required states and optional states. State transition diagram and specification of state transitions.			
Physical Security	Production grade equipment.	Locks or tamper evidence.	Tamper detection and response for covers and doors.	Tamper detection and response envelope. EFP or EFT.
Operational Environment	Single operator. Executable code. Approved integrity technique.	Referenced PPs evaluated at EAL2 with specified discretionary access control mechanisms and auditing.	Referenced PPs plus trusted path evaluated at EAL3 plus security policy modeling.	Referenced PPs plus trusted path evaluated at EAL4.
Cryptographic Key Management	Key management mechanisms: random number and key generation, key establishment, key distribution, key entry/output, key storage, and key zeroization.			
	Secret and private keys established using manual methods may be entered or output in plaintext form.		Secret and private keys established using manual methods shall be entered or output encrypted or with split knowledge procedures.	

EMI/EMC	47 CFR FCC Part 15. Subpart B, Class A (Business use). Applicable FCC requirements (for radio).		47 CFR FCC Part 15. Subpart B, Class B (Home use).	
Self-tests	Power-up tests: cryptographic algorithm tests, software/firmware integrity tests, critical functions tests. Conditional tests.			
Design Assurance	Configuration management (CM). Secure installation and generation. Design and policy correspondence. Guidance documents.	CM system. Secure distribution. Functional specification.	High-level language implementation.	Formal model. Detailed explanations (informal proofs). Preconditions and postconditions.
Mitigation of Other Attacks	Specification of mitigation of attacks for which no testable requirements are currently available.			

Table C.1: FIPS 140-2 certification levels [FIP01].

Appendix D

FIPS 140-3 Certification

FIPS 140-3 Evaluation Criteria

C_1: Cryptographic Module Specification	
Module specification, cryptographic boundary, interface specification, security policy. *Approved modes of operation are defined.*	+
Module specification, cryptographic boundary, interface specification, security policy. *Module indication of approved mode of operation.*	++
C_2: Cryptographic Module Interfaces	
Required and Optional Interfaces. Specification of all interfaces and of all input and output data ports.	+
Requirements at L1 remain valid. Data ports for unprotected critical parameters logically or physically separated from other data ports.	++
C_3: Roles, Services and Authentication	
Definition of module's roles and services.	+
Role based or identity-based authentication	++
Identity-based operator authentication	+++
Multi-factor authentication	++++
C_4: Software/Firmware Security	
Approved integrity technique. Defined (Hybrid) Software/Firmware Module Interfaces.	+
Approved digital signature or keyed message authentication code-based integrity test.	++
Approved digital signature based integrity test.	+++
C_5: Operational Environment	
Modifiable operational environment. Control of security sensitive parameters (SSPs). Prevention of outside processes getting access to critical sensitive parameters (CSPs)	+
Modifiable operational environment. Role-based or discretionary access control. Audit mechanism.	++
Operational environment components bound to the firmware module. Controlled loading of additional through the Software/Firmware Load Test.	+++
Operational environment components bound to the firmware module.	++++
C_6: Physical Security	
Production grade equipment.	+

Tamper evidence. Opaque covering or enclosure.	++
Tamper response and zeroization circuitry on removable covers and doors. Protection from probing from module openings. Hard opaque coating or enclosure.	+++
Environmental Failure Protection (EFP) or Environmental Failure Testing (EFT) for temperature and voltage. Tamper detection and zeroization circuitry for multi-chip modules. Fault Injection Mitigation.	++++
C_7: Security Non-invasive Attacks	
Review of documented mitigation techniques against applicable non-invasive attacks.	+
Mitigation against non-invasive attacks with specific test requirements.	++
C_8: SSP Management	
Requirements for random bit generators, SSP generation, SSP establishment, SSP entry and output, SSP storage, and CSP zeroization. Electronically transported CSPs entered or output only encrypted. *Manually transported SSPs may be entered or output in plaintext form.*	+
Requirements for random bit generators, SSP generation, SSP establishment, SSP entry and output, SSP storage, and CSP zeroization. Electronically transported CSPs entered or output only encrypted. *Manually transported SSPs may be entered or output either in encrypted form or using split-knowledge procedures. Trusted Channel required.*	++
C_9: Self-Tests	
Pre-operational and conditional self-tests	+
Requirements at L1 remain valid. Pair-wise consistency for key pairs entered into module. Period self-tests.	++
C_{10}: Configuration Management	
Configuration management system for module, components and documentation.	+
Automated configuration management	++
C_{11}: Design Assurance	
Correspondence between module and Security Policy.	+
Functional specification.	++
Detailed design.	+++
Informal proof of correspondence between pre and post conditions and the functional specification.	++++
C_{12}: Finite State Model	
State transition diagram and specification of state transitions.	+
C_{13}: Development	
Annotated source code, schematics or HDL.	+
Software high-level language. Hardware high-level descriptive language.	++
Formal model.	+++
C_{14}: Vendor Testing	
Functional testing.	+
Low-level testing.	+
C_{15}: Delivery and Operation	
Start-up procedures.	+
Delivery Procedures.	++
Operator authentication using vendor provided authentication information.	+++
C_{16}: Guidance Docs	
Administrator and non-administrator guidance.	+

Table D.1: Evaluation function for FIPS 140-3 criteria set. For detailed explanations we recommend the reader to refer to [FIP09].

FIPS 140-3 Certification Levels

Security Requirements (Criteria)	**Security level**			
	1	**2**	**3**	**4**
C_1: Cryptographic Module Specification	+	+	++	++
C_2: Cryptographic Module Ports and Interfaces	+	+	++	++
C_3: Roles, Services and Authentication	+	++	+++	++++
C_4: Software/Firmware security	+	++	+++	+++
C_5: Operational Environment	+	++	+++	+++
C_6: Physical Security	+	++	+++	++++
C_7: Physical Security Non-invasive Attacks	+	+	++	+++
C_8: Sensitive Security Parameters (SSPs) Management	+	+	++	++
C_9: Self-Tests			+	+
C_{10}: Configuration Management	+	+	++	++
C_{11}: Design Assurance	+	++	+++	++++
C_{12}: Finite State Machine	+	+	+	+
C_{13}: Development	+	++	++	+++
C_{14}: Vendor Testing	+	+	++	++
C_{15}: Delivery and Operation	+	++	++	+++
C_{16}: Guidance Docs	+	+	+	+

Table D.2: FIPS 140-3 certification levels.