Upper Bounds for the Selection of the Cryptographic Key Lifetimes: Bounding the Risk of Key Exposure in the Presence of Faults

Alfonso De Gregorio[†] [‡]

C&A S.r.l.
V.le Fulvio Testi, 126
Cinisello Balsamo (MI) 20092, Italy
alfonso.degregorio@trustedevidence.com

Abstract. With physical attacks threatening the security of current cryptographic schemes, no security policy can be developed without taking into account the physical nature of computation.
In this article we first introduce the notion of *Cryptographic Key Failure Tolerance*, then we offer a framework for the determination of upper bounds to the key lifetimes for any cryptographic scheme used in the presence of faults, given a desired (negligible) error-bound to the risk of key exposure. Finally we emphasize the importance of choosing keys and designing schemes with good values of failure tolerance, and recommend minimal values for this metric. In fact, in *standard environmental conditions*, cryptographic keys that are especially susceptible to erroneous computations (e.g., RSA keys used with CRT-based implementations) are exposed with a probability greater than a standard error-bound (e.g., 2^{-40}) after operational times shorter than one year, if the failure-rate of the cryptographic infrastructure is greater than 1×10^{-16} *failures/hours.*

Key words: Key Lifetimes, Fault-Attacks, Dependability, Security Policies, Cryptographic Key Failure Tolerance, Reliability Modeling, Side-Channels

1 Introduction

The manifestation of faults at the user interface of a cryptographic module may jeopardize security by enabling an opponent to expose the secret key material [5–12]. In fact, by failing to take into account the physical nature of computation, the current mathematical models of cryptography are unable to protect against physical attacks that exploit in a clever way the peculiarities inherent the physical execution of any algorithm [1, 18].

[†] This work was completed during the summer 2003 while the author was a visitor in the Katholieke Universiteit Leuven, Dept. Elect. Eng.-ESAT/SCD-COSIC, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium

[‡] A preliminary version of this paper appeared as a COSIC Technical Report

Consequently, one should not rely on the services delivered by today's cryptographic modules, if specific dependability guarantees are not satisfied. However, the possession of dependability attributes should be interpreted in a relative, probabilistic, sense [24, 23]. Due to the unavoidable occurrence of transient faults or the presence of dormant faults, there will be always a non-zero probability that the system will fail, sooner or later.

In order to keep the risk of key exposure below a desired boundary ϵ, the use of error detection techniques in fault-tolerant cryptographic modules is necessary but not sufficient [3, 4, 19–21, 25]. In fact, for standard error-bounds (2^{-40}, or also lower values), with typical fault rates, in *standard environmental conditions*, and using fault-tolerant systems with high levels of coverage, the probability of a key exposure may exceed the desired error bound within very short mission times, depending on the number of incorrect cryptographic values necessary to perform the fault attack against a specific cryptographic scheme. For instance, as will be shown in Sect. 3, cryptographic modules, that implement cryptographic schemes especially susceptible to erroneous computations (e.g., RSA based on the residue number system [9, 13]), will expose the key material with a probability greater than ϵ by exceeding the required reliability goal after operational times so short, that the number of scenarios where these schemes finds application in the presence of faults results to be remarkably limited. Trying to increase further the coverage of fault-tolerant systems is not the most viable solution, since it would raise the costs of cryptography modules, by requiring a larger number of hours during the design and assessment phases.

Hence, it is of primary importance to choose key lifetimes so that the key material will no longer be used after that the effective reliability of the system falls below the required goal, necessary to guarantee the desired negligible risk of key exposure.

In this article we first introduce the notion of *Cryptographic Key Failure Tolerance*, then we offer a framework that enables to limit the risk of key exposure to a desired error-bound in the presence of faults, by modeling the reliability of typical cryptographic infrastructures and relating their failure rates, the failure tolerance of the cryptographic keys and the mission duration for the required reliability goals, to the lifetime of keys. Using this framework, we provide guidelines either for the determination of upper bounds to key lifetimes for any cryptographic scheme implemented in generic cryptographic devices, or for the selection of cryptographic infrastructures that can provide the required levels of reliability, whereas specific lifetimes and schemes are desired. As long as the mathematical models of cryptography are not extended to the physical setting, reliability and security will remain strictly related. Consequently, security policies will have to be developed by carefully taking into account the peculiarities inherent in the physical execution of any algorithm.

Our framework is intended to be used together with the existing guidelines to the selection of cryptographic key sizes [2, 14–17], assuming one agrees with the formulated hypotheses of the prior works or with the explicit assumptions on which our recommendations are based. The existing guidelines should be con-

sidered complementary to the proposed framework, as based on the analysis of the computational effort required to break cryptographic schemes by exaustive search.
The major advantage of this approach, besides its simplicity, is that it allows to keep the risk of key exposure below a desired error-bound using one or more cryptographic modules characterized by different failure rates.

The paper is organized as follows. We describe the model and introduce the notion of *Cryptographic Key Failure Tolerance* in Sect. 2. In Sect. 3 we offer a first framework to model the risk of key exposure in the presence of faults and to derive upper bounds to the lifetime of keys, by incrementally modeling the reliability of the following two cryptographic infrastructures: 1) single systems implementing cryptographic schemes tolerating a generic number of erroneous computations, 2) highly available cryptographic infrastructures characterized by a pool of independent systems providing service concurrently using a generic cryptographic scheme with a common credential. Sect. 4 will be devoted to provide examples of how to use the proposed framework. We discuss the practical consequences of our estimates and emphasize the importance of choosing cryptographic keys and designing cryptographic schemes with good levels of failure tolerance in Sect. 5. Concluding remarks are given in Sect. 6.

2 The Model

The model consist of a cryptographic module containing some cryptographic secret. The interaction with the outside work follows a cryptographic protocol. On some rare occasions, the module is assumed to be affected by faults causing it to output incorrect values [8].

2.1 Key Points

In the presence of faults, the choice of cryptographic key lifetimes depends primarily on the following points:

I. environmental conditions;
II. the failure tolerance of cryptographic keys (defined in Sect. 2.3) - 1^{st} security parameter;
III. desired (negligible) risk of key exposure: the security margin - 2^{nd} security parameter;
IV. failure rates: the rate of occurrence for incorrect values at the cryptographic module user interface - 3^{rd} security parameter;

2.2 Environmental Conditions

We limit our analysis to the black-box scenario characterized by the occurrence and activation of faults *in standard environmental conditions.*

Assumption 1. Our main assumption is that the security of cryptographic modules cannot be compromised by any deliberate or accidental excursions outside their normal operating ranges of environmental conditions. For instance, a cryptographic module has been designed according to today's security standards [22] to operate, or to respond, in a safe way also with widely varying environmental conditions. Or, the computing device can be simply kept in a controlled environment.

All the estimates offered in this paper would be drastically modified if a modification of the environmental conditions can augment the occurrences of failures (i.e., inducing faults).

2.3 Cryptographic Key Failure Tolerance

Definition 1. *Let B be a black-box implementing a cryptographic scheme S and containing a secret key K that is inaccessible to the outside world, and with the set of security parameter(s) P. The Cryptographic Key Failure Tolerance, $CKFT^{m}_{K_{(S,P)}}$, is defined to be the maximum number of incorrect values, occurring according to the fault model m, that B can output through its cryptographic protocol before K gets exposed by a fault-attack.*

Remark 1. In the presence of fault-attacks, the Cryptographic Key Failure Tolerance (CKFT) is a security parameter. As the value assumed by this metric increases, the probability of succeeding in a fault-attack within time T decreases. A quantitative estimate of this probability is provided in Sect. 3.

In Table 1 the failure tolerance of some cryptographic schemes is provided. For example, an AES-128 key can be exposed by 49+1 faulty ciphertexts while considering the fault model assumed in the first Differential Fault Attack presented in [10]. It should be noted how several cryptographic keys may be characterized by a common value of this metric. We denote the set of all cryptographic keys with failure tolerance n under the fault model m, C^{m}_{n}. Obviously, new fault attacks or improvements to already existing attacks can determine new failure tolerance values for a given set of keys. For instance, a beautiful refinement by Lenstra [9] to the first fault-attack on RSA used with Chinese Remainder Theorem (CRT) [6–8] caused the shifting of all RSA private keys used with the CRT from C^{1bit}_{1} to C^{1bit}_{0}, where $1bit$ denotes the fault-model considering faults affecting one bit at a time. So, we will refer in a generic way to the $CKFT$ values.

2.4 Desired Error-Bound to the Risk of Key Exposure

This is the 2^{nd} security parameter. It can assume every desired value in the interval $(0, 1)$. Typical values are 2^{-40} or lower.

Table 1. The Cryptographic Failure Tolerance of some cryptographic schemes

Crypto Scheme + Sec. Parameter(s)	Fault Model	$CKFT$	Author(s)	Year
Fiat-Shamir Id. Scheme ($t = n$)	∼1bit	$O(n)$	*Boneh, et al.* [8]	1996
RSA (1024 bit)	1bit	$O(n)$	*Boneh, et al.* [8]	1996
Schnorr's Id. Protocol ($p = a, q = n$)	1bit	$n \cdot \log 4n$	*Boneh, et al.* [8]	1996
RSA+CRT	1bit	0	*Lenstra* [9]	1997
AES (n=128)	1bit	49	*Giraud* [10]	2003
AES (n=128)	1byte	249	*Giraud* [10]	2003

2.5 Failure Rates

Throughout the rest of the paper, unless specified differently, we will refer to the failure rate as the rate of occurrence of incorrect values at the user interface of a given cryptographic module, considering the fault model m. This value should not be confused with the generic failure rate of the computing device.
The failure rate is a security parameter. In fact, as will be shown in Sect. 3, as the failure rate increases the mean time to failure (MTTF) decreases, and consequently the probability of succeeding in a fault-attack within time T increases. In Sect. 3, we calculate the failure rates of cryptographic infrastructures composed by multiple independent subsystems[1] providing service concurrently and characterized by different failure rates. Since failure rates of each component are strongly depended on the implementation details of cryptographic modules, we leave them as parameters. Therefore, the estimates are provided for a representative sample of failure rates in the range $[1 \times 10^{-15}, 1 \times 10^{-9}]$, in *failures/hours.*

3 Upper Bounds for the Selection of Cryptographic Key Lifetimes

3.1 Method of Computation

In order to limit the risk of key exposure, it is necessary to limit the lifetime of keys so that the key material will no longer be used when the reliability of the computing system falls below the required goal.

Given the desired error-bound ϵ, and the failure tolerance value that characterize a generic key $CKFT^m_{K_{(S,P)}}$, we first determine the reliability goal $R(t_R)$ necessary to enforce the security margin. Then, by modeling the reliability of specific infrastructures, we determine the final failure rate $\mu^m_{Infr.}$. The resulting value is used to derive reliable life of the infrastructure t_R, or the mission duration for the required reliability goal. This is the upper bound to the lifetime of the key $K_{(S,P)}$.

[1] We consider two subsystems to be independent if electrically isolated from each other, using separate power supplies and located in separate chassis. The subsystems can share common data objects and cryptographic keys.

3.2 Single Cryptographic Modules Implementing a Generic Cryptographic Scheme

Let T be a random variable representing the time of occurrence of incorrect values at the user interface of the computing system. Let $F(T)$ be the distribution of T. Typically, computing systems are assumed to fail according to the exponential distribution. This distribution, being characterized by constant failure rates, is consistent with the *Assumption 1*.

In particular, we use the two-parameter exponential distribution. Its probability density function (pdf) is given by,

$$f(T) = \mu e^{-\mu(T-\gamma)}, \quad f(T) \geq 0, \quad \mu \geq 0, \quad T \geq 0 \text{ or } \gamma \tag{1}$$

The location parameter γ, enables the modeling of those systems that can manifest incorrect values at their user interface only after γ time units (e.g., hours) of operation.

From (1) follows that the two-parameter exponential cumulative density function (cdf) and the exponential reliability function are respectively:

$$Q(T) = 1 - e^{-\mu(T-\gamma)} \tag{2}$$

$$R(T) = 1 - Q(T) = e^{-\mu(T-\gamma)}, \quad 0 \leq R(T) \leq 1 \tag{3}$$

Equations (2) and (3) give respectively the probability of failure, and the reliability of the system.

The system is considered to be functioning as long as the key material has not been exposed (i.e., as long as the number of failures is less than or equal to $CKFT^m_{K_{(S,P)}}$) with a probability greater than ϵ. The system can be viewed as a pool of $CKFT^m_{K_{(S,P)}} + 1$ of identical, independent and *non-repairable* subsystems each characterized by a generic failure rate μ, under the fault model m. The components of the pool provide service concurrently. As soon as a failure occurs the number of sub-system decreases by one unit. The system fails when no sub-systems remains in service. Given the desired risk of key exposure ϵ, and $n = CKFT^m_{K_{(S,P)}}$:

$$R(T) = 1 - \prod_{i=1}^{n+1} Q_i(T) \geq 1 - \epsilon \tag{4}$$

The subsystems are identical, hence:

$$R(T) = e^{-\mu(T-\gamma)} \geq 1 - \sqrt[n+1]{\epsilon} \tag{5}$$

Therefore, the key lifetime for $K_{S,P}$, $L(K_{S,P})$, must be:

$$L(K_{S,P}) \leq t_R = \gamma - \frac{\ln(1 - \sqrt[n+1]{\epsilon})}{\mu} \tag{6}$$

Table 2. Upper Bounds to Key Lifetimes for typical failure rates, with a desired error-bound $\epsilon = 2^{-40}$ and $\gamma = 0$. Failure rates are expressed in *failures/hours*; upper bounds to key lifetimes are expressed in *hours*.

C_n^m	μ_0^m	μ_1^m	μ_2^m	μ_3^m	μ_4^m	μ_5^m	μ_6^m
↓	1×10^{-15}	1×10^{-14}	1×10^{-13}	1×10^{-12}	1×10^{-11}	1×10^{-10}	1×10^{-9}
n=0	$\mathbf{9.09 \times 10^{2}}$	$\mathbf{9.09 \times 10^{1}}$	$\mathbf{9.09 \times 10^{0}}$	$\mathbf{9.09 \times 10^{-1}}$	$\mathbf{9.09 \times 10^{-2}}$	$\mathbf{9.09 \times 10^{-3}}$	$\mathbf{9.09 \times 10^{-4}}$
n=1	9.54×10^{8}	9.54×10^{7}	9.54×10^{6}	9.54×10^{5}	$\mathbf{9.54 \times 10^{4}}$	$\mathbf{9.54 \times 10^{3}}$	$\mathbf{9.54 \times 10^{2}}$
n=2	9.69×10^{10}	9.69×10^{9}	9.69×10^{8}	9.69×10^{7}	9.69×10^{6}	9.69×10^{5}	$9.69e \times 10^{4}$
n=3	9.77×10^{11}	9.77×10^{10}	9.77×10^{9}	9.77×10^{8}	9.77×10^{7}	9.77×10^{6}	9.77×10^{5}
n=4	3.91×10^{12}	3.91×10^{11}	3.91×10^{10}	3.91×10^{9}	3.91×10^{8}	3.91×10^{7}	3.91×10^{6}
n=5	9.89×10^{12}	9.89×10^{11}	9.89×10^{10}	9.89×10^{9}	9.89×10^{8}	9.89×10^{7}	9.89×10^{6}
n=6	1.92×10^{13}	1.92×10^{12}	1.92×10^{11}	1.92×10^{10}	1.92×10^{9}	1.92×10^{8}	1.92×10^{7}
n=7	3.17×10^{13}	3.17×10^{12}	3.17×10^{11}	3.17×10^{10}	3.17×10^{9}	3.17×10^{8}	3.17×10^{7}
n=8	4.70×10^{13}	4.70×10^{12}	4.70×10^{11}	4.70×10^{10}	4.70×10^{9}	4.70×10^{8}	4.70×10^{7}
n=9	6.45×10^{13}	6.45×10^{12}	6.45×10^{11}	6.45×10^{10}	6.45×10^{9}	6.45×10^{8}	6.45×10^{7}
n=10	8.38×10^{13}	8.38×10^{12}	8.38×10^{11}	8.38×10^{10}	8.38×10^{9}	8.38×10^{8}	8.38×10^{7}
n=11	1.04×10^{14}	1.04×10^{13}	1.04×10^{12}	1.04×10^{11}	1.04×10^{10}	1.04×10^{9}	1.04×10^{8}

Table 2 provides upper bounds to key lifetimes for a number of representative failure rates affecting systems using keys characterized by a *CKFT* value in the interval $(0, 11)$, $\epsilon = 2^{-40}$, and $\gamma = 0$. Failure rates are expressed in *failures/hours*, whereas upper bounds to key lifetimes are in *hours*.

3.3 Highly Available Cryptographic Infrastructures

In this section we extend the modeling of the risk of key exposure to highly available cryptographic infrastructures. In particular, we consider a pool of l heterogeneous independent cryptographic modules (i.e., failing independently), each characterized by its own failure rate μ_l^m, that provides service using a common generic key characterized by a failure tolerance of $CKFT^m_{K_{(S,P)}}$. For example the key material may be stored in a shared secure device, or replicated among the l modules. Moreover we assume the following:

Assumption 2. All cryptographic module present in the pool start to provide service simultaneously (i.e., $\gamma_1 \approx \gamma_2 \approx \ldots \approx \gamma_l$).

Similarly to single cryptographic modules, the infrastructure is considered to be functioning as long as the cryptographic key has not been exposed (i.e., as long as the number of failures is less than or equal to $CKFT^m_{K_{(S,P)}}$) with a probability greater than ϵ. It should be noted that in this scenario the failures of each module should be considered to be cumulative. In fact, by affecting a common resource (i.e., the cryptographic key), each failure affects also the residual service-time of the other components in the pool. For example, assuming that the infrastructure is using a cryptographic key that does not tolerate any failure, it is sufficient a single failure to compromise the service provided by the entire infrastructure. Hence, the pool of cryptographic modules should be modeled as a series of systems.

$$R_{\mathrm{HA}}(T) = \prod_{i=1}^{l} R_i(T) = e^{-\sum_{i=1}^{l} \mu_i^m (T-\gamma)} \tag{7}$$

Equation (7) gives the reliability of the series of cryptographic modules present in the pool. This is equivalent to reliability of a system with failure rate $\mu_{\mathrm{H}A} = \sum_{i=1}^{l} \mu_i^m$. Using (6) is possible to compute the reliable life of the key $K_{(S,P)}$ used by the considered high-availability cryptographic infrastructure:

$$L(K_{S,P}) \leq t_R = \gamma - \frac{\ln(1 - \sqrt[n+1]{\epsilon})}{\sum_{i=1}^{l} \mu_i^m} \tag{8}$$

Remark 2. **Scaling-Out may be a Hazard**
Obviously, the number l of cryptographic modules present in this typical high-availability configuration (i.e., active-active model) affects one of the security parameters, by increasing the exposure of cryptographic credentials. In fact, as the final failure rate increases, the MTTF decreases; hence, decreasing the reliable life of the system. Consequently, the use of cryptographic modules with very low failure rates becomes especially critical when its necessary to design highly available cryptographic infrastructures. In Fig.1 the required reliability goals necessary to limit the risk of key exposure to $\epsilon = 2^{-40}$ are shown for either a single cryptographic module with $\mu_{single}^m = 1 \times 10^{-15}$ *failures/hours*, or a pool of 10 independent and identical cryptographic modules with $\mu_{HA}^m = \sum_{i=1}^{10} \mu_{single}^m$, providing service concurrently using a common cryptographic credential with a CKFT value in the interval $(0, 9)$.

4 Using this Framework

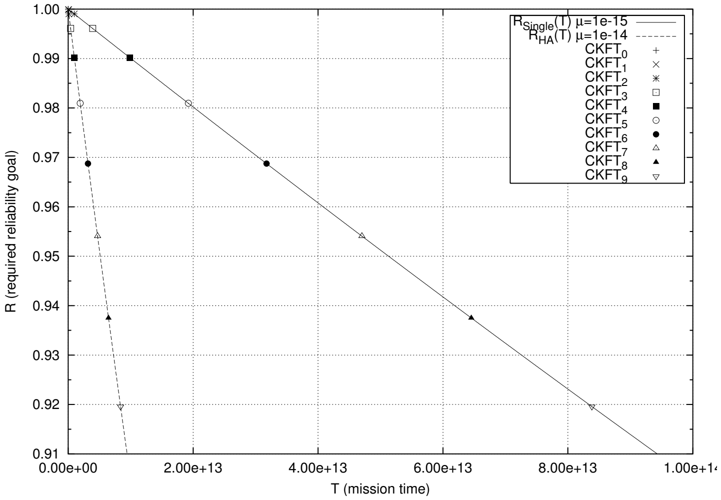
4.1 Estimating Upper Bounds for Cryptographic Key Lifetimes

Suppose one needs to select the lifetime of a cryptographic key that belongs to C_1^m (i.e., has cryptographic failure tolerance 1). Suppose also that is necessary to guarantee a risk of key exposure less than or equal to $\epsilon = 2^{-40}$ using a cryptographic infrastructure with failure rate $\mu_{Infr}^m = 1 \times 10^{-11}$ *failures/hours.* Using (8), or looking at the row of a precomputed table (e.g., Table 2) for the failure tolerance 1, one finds that the key lifetime should not exceed *10 years.* This is only the upper bound. Additional considerations, related to the specific cryptographic scheme and to the application context, may obviously decrease the effective lifetime.

4.2 Selecting Dependable Cryptographic Infrastructures

It is straightforward to use Table 2 (or equation (7)) also to look up the failure rate that is necessary to guarantee the desired negligible risk of key exposure, given a required key lifetime and cryptographic scheme. Suppose one needs

Fig. 1. Reliability goals for CKFT values in the internal (0,9), with Pr. Key Exposure $\epsilon = 2^{-40}$. Mission times are expressed in hours.



to choose a cryptographic infrastructure among a number of alternatives, each characterized by different costs. The entire system must be able to use a cryptographic key with failure tolerance 9 and a lifetime of 4 years, while keeping the risk of key exposure below 2^{-128}. Expressing the failure rate as a function of the reliable life from equation (7) one finds that is sufficient to select an infrastructure characterized by a failure rate not greater than 4×10^{-9} *failures/hours.*

4.3 Scaling-Out Cryptographic Infrastructures

Suppose now that one wants to provide a cryptographic service with an infrastructure characterized at the initial stage by a pool of l cryptographic devices and needs to scale up it, without changing the key material and guaranteeing a risk of key exposure not greater than 2^{-40}.

If the number h of additional sub-system that will be added in the future, and the respective failure rates μ_h^m, are known *a priori*, and there is a required lifetime, it is possible to use Table 2 to look up the column with failure rate $\sum_{i=1}^{l} \mu_i^m + \sum_{j=1}^{h} \mu_h^m$ to find the first level of failure tolerance n_{min}, characterized by an error bound greater than or equal to the desired one. In this scenario, the cryptographic key must be characterized by a level of failure tolerance greater than or equal to n_{min}. It is worth to note that these are conservative estimates, since all the $l + h$ components are assumed to start their operation simultaneously.

If the failure rates of the additional sub-system is not known *a priori* and it is necessary to use a cryptographic scheme with failure tolerance n, it is possible to

Table 3. Effective risk of key exposure for credentials in C_0^m. The estimates are computed for a number of typical lifetimes (in years) and failure rates (*failures/hours*). The exponents are rounded up to the nearest integer.

T	μ_0^m	μ_1^m	μ_2^m	μ_3^m	μ_4^m	μ_5^m	μ_6^m
$\downarrow$	1×10^{-15}	1×10^{-14}	1×10^{-13}	1×10^{-12}	1×10^{-11}	1×10^{-10}	1×10^{-9}
1	2^{-36}	2^{-33}	2^{-30}	2^{-26}	2^{-23}	2^{-20}	2^{-16}
2	2^{-35}	2^{-32}	2^{-29}	2^{-25}	2^{-22}	2^{-19}	2^{-15}
3	2^{-35}	2^{-31}	2^{-28}	2^{-25}	2^{-21}	2^{-18}	2^{-15}
4	2^{-34}	2^{-31}	2^{-28}	2^{-24}	2^{-21}	2^{-18}	2^{-14}
5	2^{-34}	2^{-31}	2^{-27}	2^{-24}	2^{-21}	2^{-17}	2^{-14}
10	2^{-33}	2^{-30}	2^{-26}	2^{-23}	2^{-20}	2^{-16}	2^{-13}
20	2^{-32}	2^{-29}	2^{-25}	2^{-22}	2^{-19}	2^{-15}	2^{-12}

lookup the row n of Table 2, to find the first failure rate μ_{max}^m, characterized by an error bound greater than or equal to the desired lifetime of keys. In this second scenario, the final failure rate of the cryptographic infrastructure $\sum_{i=1}^{l+h} \mu_i^m$ must be less then or equal to μ_{max}^m. Hence, the sum of the failure rates of the additional sub-systems needs to be: $\sum_{j=1}^{h} \mu_j^m \leq \mu_{max}^m - \sum_{i=1}^{l} \mu_i^m$.

5 Practical Consequences of the Presented Estimates

According to equation (8), in order to achieve a reliable life long at least one year, while requiring $\epsilon = 2^{-40}$, cryptographic keys that do not tolerate any erroneous computation (i.e., C_0^m) must be used on a cryptographic infrastructure that fail with a rate lower than $\mu^m = 1 \times 10^{-16}$ *failures/hours*. The required rates decreases further when lower error-bounds are desired.

These are certainly very low rates. Although it is possible to design highly reliable cryptographic modules, the costs necessary during the design and assessment phases and the still low reliable life strongly limits the number of scenarios where keys especially susceptible to erroneous computation may find application. Unfortunately, this is the case of RSA keys used with CRT-based implementations [9, 8]. The same considerations applies for keys in C_1^m at failure rates beyond 9.54×10^{-5} *failures/hours*.

In today's cryptographic applications (e.g., e-commerce and bank secure web servers, smart IC cards) it is common to find RSA keys used with CRT-based implementation characterized by lifetimes long months, or years. These lifetimes are selected without modeling the risk of key exposure in the presence of faults. Table 3 provides estimates of this risk for cryptographic credentials with $CKFT = 0$. The probabilities are furnished for typical lifetimes and failure rates, using (6) and (8). The exponents are rounded up to the nearest integer. The estimates shows hazard rates likely beyond those initially predicted, without considering dependability metrics.

In the next section we emphasize the importance of choosing keys with a good

Table 4. Minimal CKFT required to enable the selection of key lifetimes long up to T_{max} = *200 years*, for a number of ϵ and μ. $\gamma = 0$.

ϵ	μ_0^m	μ_1^m	μ_2^m	μ_3^m	μ_4^m	μ_5^m	μ_6^m
$\downarrow$	1×10^{-15}	1×10^{-14}	1×10^{-13}	1×10^{-12}	1×10^{-11}	1×10^{-10}	1×10^{-9}
2^{-40}	1	1	1	2	2	3	4
2^{-64}	2	2	2	3	4	5	6
2^{-80}	2	3	3	4	5	6	8
2^{-128}	4	4	5	6	8	10	13
2^{-256}	8	9	11	13	16	20	27

CKFT values, by offering estimates of minimal values of this metric necessary to enable the selection of key lifetimes long enough for any real application scenario.

5.1 On the Importance of Good CKFT Values

Let T_{max} a maximum desirable key lifetime (i.e., the maximum lifetime of a key for any real application scenario). From (6) and (8) follows that the minimum value of CKFT required to guarantee a desired ϵ using a cryptographic infrastructure with failure rate μ^m, is given by:

$$CKFT^m_{min} = \lceil \ln_{Q(T_{max}-\gamma)} \epsilon - 1 \rceil \tag{9}$$

In Table 4 we provide the minimal CKFT values for a number of error-bounds and failure rates, and with T_{max} = *200 years*.

6 Conclusions

As long as the mathematical models of cryptography are not extended to the physical setting [1, 18], reliability and security will remain strictly related. Consequently, security policies will have to be developed by carefully taking into account the peculiarities inherent the physical execution of any algorithm. In this paper we have offered a first framework that enables to bound the risk of key exposure in the presence of faults, by modeling the reliability of typical cryptographic infrastructures and relating their failure rates, the failure tolerance of the cryptographic keys, and the desired (negligible) error-bound, to the lifetime of keys.

Acknowledgments. The author would like to thank Bart Preneel for his determinant support and helpful feedbacks, and all the people at COSIC for their great hospitality.

References

1. S. Micali, L. Reyzin. Physically Observable Cryptography. In *Cryptology ePrint Archive: Report 2003/120* http://eprint.iacr.org/2003/120
2. Arjen K. Lenstra and Eric R. Verheul. Selecting Cryptographic Key Sizes. Journal of Cryptology, Vol. 14, No. 4, pp. 255-293, 2001.
3. A. Shamir. Method and Apparatus for protecting public key schemes from timing and fault attacks". U.S. Patent Number 5,991,415, November, 1999; also presented at the rump session of EUROCRYPT'97.
4. Christian Aumller, Peter Bier, Wieland Fischer, Peter Hofreiter, Jean-Pierre Seifert. Fault Attacks on RSA with CRT: Concrete Results and Practical Countermeasures. *Lecture Notes in Computer Science*, Vol. 2523, Springer-Verlag, Berlin, pp. 260-275, 2002.
5. Eli Biham, Adi Shamir. Differential Fault Analysis of Secret Key Cryptosystems. *Lecture Notes in Computer Science*, Vol. 1294, Springer-Verlag, Berlin, pp.513-??, 1997.
6. D. Boneh, R. A. DeMillo, R.J. Lipton. On the Importance of Checking Computations. 1996. *Lecture Notes in Computer Science*, Vol. 1233, Springer-Verlag, Berlin, pp.37-51, 1997
7. D. Boneh, R. A. DeMillo, R.J. Lipton. On the Importance of Checking Cryptographic Protocols for Faults. In *Lecture Notes in Computer Science*, Vol. 1233, Springer-Verlag, Berlin, pp. 37-51, 1997.
8. D. Boneh, R. A. DeMillo, R.J. Lipton. On the Importance of Eliminating Errors in Cryptographic Computations. In *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, Vol. 14, no. 2, Springer-Verlag, Berlin, pp. 101-119, 2001.
9. A.K. Lenstra. Memo on RSA signature generation in the presence of faults. Available from the author: arjen.lenstra@citicorp.com.
10. Christophe Giraud DFA on AES. *Cryptology ePrint Archive, Report 2003/008*, 2003, http://eprint.iacr.org/.
11. Mathieu Ciet and Marc Joye. Elliptic Curve Cryptosystems in the Presence of Permanent and Transient Fault. *Cryptology ePrint Archive, Report 2003/028*, 2003, http://eprint.iacr.org/2003/028/.
12. E. Dottax. Fault Attacks on NESSIE Signature and Identification Schemes.
13. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone. Handbook of Applied Cryptography. CRC Press, ISBN: 0-8493-8523-7, October, 1996.
14. Burt Kaliski, TWIRL and RSA Key Size. RSA Laboratories Technical Notes. Revised May 6, 2003.
15. NESSIE Consortium. Portfolio of Recommended Cryptographic Primitives. February 27, 2003. Available via http://www.cryptonessie.org/.
16. NIST. Special Publication 800-57: Recommendation for Key Management. Part 1: General Guideline. Draft, January 2003. Available via http://csrc.nist.gov/CryptoToolkit/tkkeymgmt.html.
17. M. Blaze, W. Diffie, R. Rivest, B. Schneier, T. Shimomura, 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. Available via http://www.crypto.com/papers/.
18. Yuval Ishai, Amit Sahai, and David Wagner. Private Circuits: Securing Hardware against Probing Attacks. 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, *Lecture Notes in Computer Science*, Vol. 2729, Springer-Verlag, ISBN 3-540-40674-3.

19. G. Bertoni, L. Breveglieri, I. Koren, P. Maistri, and V. Piuri. Error Analysis and Detection Procedures for a Hardware Implementation of the Advanced Encryption Standard. In *IEEE Transactions on Computers*, Vol. 52, No. 4, pp. 493-505, ISSN 0018-9340, April, 2003.
20. G. Bertoni, L. Breveglieri, I. Koren, and V. Piuri. Fault Detection in the Advanced Encryption Standard. In *Proc. Conf. Massively Parallel Computing Systems (MPCS '02)*, pp. 92-97, 2002.
21. G. Bertoni, L. Breveglieri, I. Koren, P. Maistri, and V. Piuri. On the Propagation of Faults and Their Detection in a Hardware Implementation of the Advanced Encryption Standard. In *Proc. Int'l Conf. Application-Specific Systems, Architectures, and Processors (ASAP '02)*, pp. 303-312, 2002.
22. Federal Information Processing Standards Publication 140-2, SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES
23. Kishor S. Trivedi. Probability and Statistics with Reliability, Queueing, and Computer Science Applications - Second Edition. John Wiley, 2001.
24. SQUALE Consortium. Dependability Assessment Criteria. January 1999, http://www.newcastle.research.ec.org/squale/SQUALE4.pdf.
25. R. Karri, W. Kaijie, P. Mishra, and K. Yongkook. Fault-Based Side-Channel Cryptanalysis Tolerant Rijndael Symmetric Block Cipher Architecture. In *Proc. Defect and Fault Tolerance in VLSI Systems (DFN '01)*, pp. 418-426, 2001.