Physical Security Bounds Against Tampering*

Kerstin Lemke, Christof Paar, and Ahmad-Reza Sadeghi

Horst Görtz Institute for IT Security,
Ruhr University Bochum
44780 Bochum, Germany
{lemke, cpaar, sadeghi}@crypto.rub.de

Abstract. We consider the problem of an active adversary physically manipulating computations of a cryptographic device that is implemented in circuitry. Which kind of circuit based security can ever be guaranteed if *all* computations are vulnerable towards fault injection? In this paper, we define physical security parameters against tampering adversaries. Therefore, we present an adversarial model with a strong focus on fault injection techniques based on radiation and particle impact. Physical implementation strategies to counteract tampering attempts are discussed.

Keywords: Fault Analysis, Tamper-Proof Hardware, Physical Security, Implementation Attack, Adversarial Model, Fault Prevention, Error Detection, Fault Detection.

1 Introduction

Active implementation attacks can be classified as fault analysis, physical manipulations and modifications. *Fault analysis* aims to cause an interference with the physical implementation and to enforce an erroneous behavior that can result in a vulnerability of a security service or even a total break. The terms manipulation and modification stem from definitions of physical security, e.g., from ISO-13491-1 [1] and address similar attacks. *Physical manipulation* aims at changing the processing of the physical implementation so that it deviates from the specification. *Physical modification* is an active invasive attack targeting the internal construction of the cryptographic device.

If a cryptographic device is used in an hostile environment special properties for the device are required to ensure a certain level of physical security for the storage and processing of cryptographic keys. For the theoretical perspective we refer to the concepts on *Read-Proof Hardware* and *Tamper-Proof Hardware* as given in [11]. *Read-Proof Hardware* prevents an adversary from reading internal data stored and *Tamper-Proof Hardware* prevents the adversary from changing internal data. Moreover, we use the term of *Tamper-Resistant Hardware* as a

* The work described in this paper has been supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT, the European Network of Excellence in Cryptology.

J. Zhou, M. Yung, and F. Bao (Eds.): ACNS 2006, LNCS 3989, pp. 253–267, 2006.

relaxed term of *Tamper-Proof Hardware*, e.g., the hardware is resistant to tampering to a certain extent. Such bounds are made more precise in this work.

In a tamper-proof implementation, fault injections are not feasible per definition. However, in real life, practical experiments have shown that approaches towards tamper resistance are hard. Many contributions (e.g., [16, 4, 24, 5, 23, 25]) have reported that semiconductor circuits are vulnerable against fault injections. Such findings are related to the development of devices for the use in aerospace and high-energy physics which have to tolerate particle radiation impact during operation [18, 19]. In contrast to applications developed for safety and reliability reasons, security applications have to withstand an active malicious adversary. Prior to the first scientific contribution [9] on fault analysis the FIPS-140 standard already required a cryptographic algorithm test ("known-answer test") [12] to be implemented in cryptographic modules during start-up. Moreover, in an error state, according to [12], the use of cryptographic algorithms shall be inhibited.

We recollect previous fault induction techniques to build an unified adversarial model based on [17] as first step towards bridging the gap between the theoretical framework of [11] and real-world experiences. In our model we cover fault analysis against physical cryptographic devices. We assume that each kind of data memory can be tampered with in a probabilistic sense and that the adversary is able to induce faults at any internal state and computation of the physical device. By doing so, we are able to model the manifold nature of faults as well as to include Differential Fault Analysis ([8, 22]) more adequately in case of physical devices.

As discussed in Section 1.1 the *Algorithmic Tamper-Proof (ATP)* security model [11] does only partly give a framework for existing attacks. In this paper we deal with the problem which kind of implementation based security can be guaranteed in an extended 'real-life' model against tampering. Therefore, we present a physical model with a strong focus on fault injection techniques based on radiation and particle impact. Physical security parameters are outlined and result in implementation strategies to prevent and detect tampering attempts.

1.1 Related Work: ATP Security

The model of *Algorithmic Tamper-Proof (ATP) Security* was introduced in [11]. It assumes that devices are built by using two different components; one being tamper-proof but readable, and the other being read-proof yet tamperable. Only data that is considered to be universally known (i.e., public data) is tamper-proof beyond the reach of the tampering adversary. Other data is subject to tampering, i.e., fault induction. ATP Security defines a powerful tampering adversary who is able to initiate three commands: $\mathsf{Run}(\cdot)$, i.e., the cryptographic computation, $\mathsf{Apply}(\cdot)$, i.e., the fault injection, and $\mathsf{Setup}(\cdot)$. The adversary knows all construction details: especially, the adversary knows each bit-position in the device's memory. It is concluded in [11] that a component is needed which is both read-proof and tamper-proof to achieve general Algorithmic Tamper-Proof (ATP) Security.

The main limitation of [11] is caused by the fact that the command $\mathsf{Run}(\cdot)$ itsself is assumed to be not vulnerable to fault injection. In practice, there is no reason that the adversary does not attack $\mathsf{Run}(\cdot)$ itself. Actually, standard scenarios of Differential Fault Analysis (DFA) apply faults *during* the cryptographic computation [8, 22]. Such a setting becomes especially important in case of tampering at memory-constrained devices as, e.g., a modification prior to $\mathsf{Run}(\cdot)$ can hardly affect *only* the last round of DES. In [11], tamper-proofing a signature (or decryption) scheme is part of the command $\mathsf{Run}(\cdot)$ which first checks whether the storage has integrity using a verification algorithm. If so, the signature (or decryption) algorithm is computed yielding an output as result. Otherwise, self-destruction of the device is invoked. In case the verification algorithm is subject to fault injection, too, the tamper-proofing solution of the ATP model does not hold anymore.

Reference [11] also discusses restrictions of the model assuming that the adversary is limited, for instance, it is only feasible for the adversary to perform a probabilistic flipping of bits in the device's memory. The type of DFA discussed in [11] requires the strong assumption that the memory type is significantly asymmetric. For this type of DFA, [11] argues that checking for faults can be sufficient for ATP security, even if the device is not equipped with a self-destruct capability. As recently shown, one can even precisely induce faults, e.g., by optical fault induction, as reported in a recent survey on hardware security analysis [25]. Therein, it is demonstrated that any individual bit of SRAM memory could be changed to a definitive state by light injection. Both the targeting state '0' and '1' could be set, just by a lateral adjustment of the light spot.

2 An Overview of Fault Analysis

Fault analysis against cryptographic primitives has become a new research area initiated by [9]. Besides targeting cryptographic primitives there are other applications of fault induction that target generic (non-cryptographic) building blocks.

2.1 Cryptographic Building Blocks

A recent survey on fault analysis against cryptographic primitives can be found in [26].

A Generic Attack: If the memory type used for key storage has the special property that flipping a bit from one state to the other is impossible (e.g., from state '1' to state '0'), all key bits finally accumulate in one state (e.g., state '1') after repetitive fault injections. Assuming the adversary owns cryptograms for each intermediate state, e.g., after each successive induced state transition, the adversary can iterate backwards recursively [7], starting at the known final state, yielding finally the original key value.

Block Ciphers (AES and DES): Fault attacks against block ciphers are differential attacks that require both a correct cryptogram and some faulty ones

for the analysis. In [8] Differential Fault Analysis (DFA) has been introduced against DES. The original attack assumed that faults occur randomly in all rounds of DES and required about 50-200 faults in this model. If precise fault injection is possible, the number can be reduced to about three faults [26]. For AES, some scenarios are presented in [26]. Among them, the most promising one [22] requires two faults for recovering the AES key.

Stream ciphers: In [13] fault analysis techniques are presented targeting the linearity of LFSRs which are typical building blocks of stream ciphers. Another approach has been presented in [6] for the stream cipher RC4. This approach exploits the forced induction of impossible states.

Asymmetric primitives: Fault injection against an RSA-CRT implementation requires only one fault injection with very low requirements on the concrete fault occurrence [9]. Modular exponentiation which is used at RSA as well as ElGamal, Schnorr and DSA signature schemes can be also attacked by fault injection successively [9].

2.2 Non-cryptographic Building Blocks

Here, we use the notion of a *security service* as a general term for any security relevant or security enforcing building block of the cryptographic device.

Modification of Security States: For cryptographic devices, it is necessary to maintain security states by storing attributes, e.g., related to authorizations and privileges achieved. A fault injection against such a security state may end up in a more privileged state.

Modification of a Security Service: Modification of a security service itsself can be invoked by fault injection. By-passing checks of parameter bounds as presented by [3] is one example for this kind of threat.

Denial of Service: Fault injection can result in a permanent mal-function or destruction of circuit components used by a security service. For example, the destruction of a physical random generator might be attractive.

3 Adversary Model

The adversary model presented is an extended version of [17]. By assumption the physical device $\mathcal{D}$ is encapsulated. Especially, it does not offer a logical nor a physical interface to modify the internal memory or the internal construction of $\mathcal{D}$. The set-up for fault analysis based attacks consists of i) the physical device $\mathcal{D}$ under test, ii) a reader device for the data communication interface, and iii) a fault injection set-up. Additionally, iv) a monitoring set-up can be used by the adversary to analyze the fault induction process and its effects, e.g., by measuring side channel leakage. The set-up as well as the information flow is illustrated in Fig. 1 and described in more detail below.

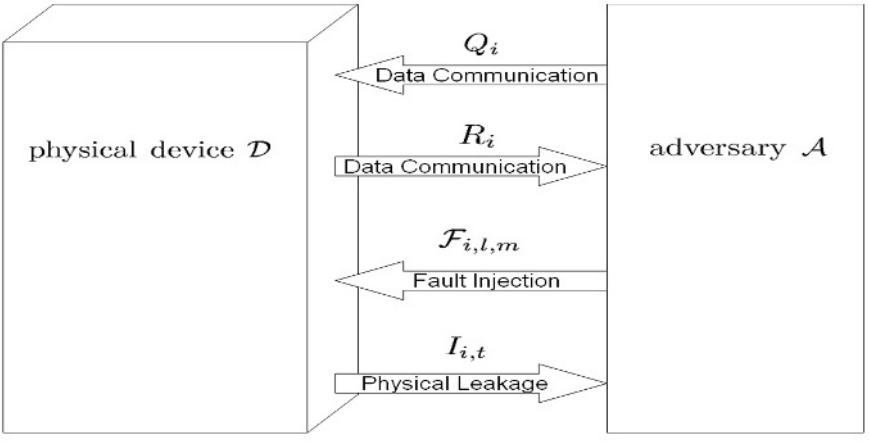

Fig. 1. Information Flow at a Fault Analysis Set-Up

We denote the adversary by $\mathcal{A}$. By assumption $\mathcal{A}$ has physical access to the physical device $\mathcal{D}$ under attack and can run a high number of instances of a security service $\mathcal{S}$. Each instance is initiated by a query Q_i of $\mathcal{A}$ and $\mathcal{D}$ finishes after some computational time at time T_i returning a response R_i where $i \in \{1, \ldots, N\}$. $\mathcal{A}$ applies a probabilistic physical interaction process aiming at disturbing the intended computation of $\mathcal{S}$. $\mathcal{A}$ may be able to monitor the effects caused by physical interaction using auxiliary means, e.g., by observing the instantaneous leakage $I_{i,t}$ of the implementation at a monitoring set-up at time t. If necessary, $\mathcal{A}$ applies cryptanalytical methods for a final analysis step.

Moreover, we assume that $\mathcal{A}$ is able to perform multiple fault injections at a fault injection set-up that are bounded by M, where M is a small number. Let L be a small number of spatially separated fault injection set-ups that can be operated in parallel. The distinct fault injections during one invocation of $\mathcal{S}$ are numbered as $\mathcal{F}_{i,l,m}$ with $l \in \{1, \ldots, L\}$ and $m \in \{1, \ldots, M\}$. These fault injections occur at the times $\{t_{i,1,1}, \ldots, t_{i,L,M}\}$ with $t_{i,1,1} \leq \cdots \leq t_{i,L,M} \leq T_i$.

$\mathcal{A}$ is an active adaptive adversary, i.e., both the queries Q_i as well as the parameters of $\mathcal{F}_{i,l,m}$ can be chosen adaptively. We point out that the leakage $I_{i,t}$ is typically not yet available for the configuration of $\mathcal{F}_{i,l,m}$ at the same instantiation of $\mathcal{S}$ unless a more demanding real-time analysis is applied.

For the physical device $\mathcal{D}$ we consider an implementation in circuitry. The target circuit $\mathcal{C}$ that is part of D consists of interconnecting Boolean gates and memory cells [1]. Each spatial position within $\mathcal{C}$ is uniquely represented in three dimensional co-ordinates $\boldsymbol{x} = (x, y, z)$. Processing of $\mathcal{C}$ is modelled by the transition states of the circuit at time t, i.e.,by using four dimensional co-ordinates $(\boldsymbol{x}, t)$. The state of the circuit s_t at time t is given by the contents of the memory

[1] In a refined model one may distinguish different types of memory elements such as flip-flops, RAM, flash and EEPROM.

cells. Faults affecting Boolean gates cause computational faults by introducing glitches or short circuits. Such faults can result in erroneous states stored at memory cells. Faults affecting memory cells cause a direct transition from memory contents s_t to $f(s_t)$ with $s_t \neq f(s_t)$. Fault induction itsself is a probabilistic process with a certain success probability that depends on the circuit $\mathcal{C}$, the underlying physical process $\mathcal{P}$ used for fault injection, and the configuration of the fault analysis set-up $\mathcal{F}_{i,l,m}$.

Summarizing, the information channels are

1. the *Query Channel* modelling $\mathcal{A}$ sending the query Q_i to $\mathcal{D}$,
2. the *Response Channel* modelling $\mathcal{A}$ receiving the response R_i of $\mathcal{D}$,
3. the *Fault Channel* modelling $\mathcal{A}$ applying physical fault injection processes $\mathcal{F}_{i,l,m}$ targeting $\mathcal{D}$, and
4. the *Monitoring Channel* modelling $\mathcal{A}$ receiving physical leakage of $\mathcal{D}$.

Informally speaking (we will give a more precise definition below in case of a digital signature scheme), an adversary $\mathcal{A}$ is *successful*, if the insertion of faults either i) yields access to a security service $\mathcal{S}$ without knowledge of the required secret or ii) yields partial information about the secret.

3.1 Objectives of the Adversary

As introduced in Section 2, manifold attack scenarios for fault analysis have been already proposed. At the core of all these scenarios there is a loop including both an instantiation of the security service $\mathcal{S}$ and a sequence of fault injection processes $\mathcal{F}_{i,l,m}$. A classification into three main categories, namely Simple Fault Analysis (SFA), Successive Simple Fault Analysis (SSFA) and Differential Fault Analysis (DFA), can be found in [17].

For concreteness, we consider a digital signature scheme that is defined as a triple of algorithms (Gen, Sig, Ver) with key generation algorithm Gen, signing algorithm Sig and verifying algorithm Ver. Let (pk, sk) be public and secret key of the signing algorithm Sig that is implemented as security service $\mathcal{S}$ of $\mathcal{D}$ in the circuit $\mathcal{C}$.

In our model, fault injection can both be done *prior* and *during* the computation of a digital signature. Fault injection may modify the computation of $\mathcal{C}$ (resulting in wrong intermediate data of the computation) as well as the actual memory contents of $\mathcal{C}$. It is m_i included in Q_i the chosen message used for signature generation and s_i part of R_i such that $s_i \leftarrow Sig_{sk}(m_i)$. If $Ver_{pk}(m_i, s_i)$= yes, the computation of the signature generation is correct, otherwise it is not.

As shown in Fig. 2, $\mathcal{A}$ invokes N instantiations of the signature computation. For each run, $\mathcal{A}$ configures $F_{i,l,m}$, chooses m_i and runs the signature computation $Sig_{sk}(m_i)$. Though configuration of $F_{i,l,m}$ may be done before the signature computation, fault injection of $F_{i,l,m}$ may also be effective during signature computation. $\mathcal{A}$ stores $(m_i, s_i, Ver_{pk}(m_i, s_i))$ for the analysis step. $\mathcal{A}$ is successful with N instantiations of $Sig_{sk}(m_i)$, if $\mathcal{A}$ succeeds in generating a valid signature s for a new message m which was not been used before during the training step. In practice, fault analysis against digital signature schemes may be even stronger: as result, $\mathcal{A}$ then outputs sk.

$H \leftarrow \{\}; I \leftarrow \{\}; State \leftarrow \epsilon;$
$for\ i = 1 \ldots N$
$(State, \mathcal{F}_{i,l,m}, m_i) \leftarrow \mathcal{A}(State, pk, H)$
$I \leftarrow I \cup \{(m_i)\}$
$(s_i) \leftarrow Sig_{sk}(m_i)$
$H \leftarrow H \cup \{(m_i, s_i, Ver_{pk}(m_i, s_i))\}$
$(m, s) \leftarrow \mathcal{A}(pk, H)$
$m \notin I$ and $Ver_{pk}(m, s) = yes$

Fig. 2. Tampering Attack against a Digital Signature Scheme based on adaptive chosen messages

3.2 Physical Means of the Adversary

In this Section we detail on the physical modelling of the circuit $\mathcal{C}$ and the physical interaction process $\mathcal{P}$. Let assume a strong adversary $\mathcal{A}$ who is given a map of $\mathcal{C}$ including a behavioral simulation for each time t. $\mathcal{A}$ is then able to configure the setup $F_{i,l,m}$ for fault injection accordingly to the known circuit layout and processing times.

Interaction Range. According to FIPS 140-2 [2] we introduce the concept of the *cryptographic boundary* that encloses all security relevant and security enforcing parts of an implementation. Additionally, we define a second boundary that we call the *interaction boundary* that is specific for each physical interaction process. If the adversary does not pass the interaction boundary, the physical interaction is not effective at the cryptographic device. The interaction boundary can be an outer boundary of the cryptographic boundary, as, e.g., in case of temperature which affects the entire cryptographic module. Interaction with light is only feasible if a non-transparent encapsulation is partially removed, e.g., the chip is depackaged. Because of the limited range of the interaction, interaction processes using particles with non-zero mass may require the removal of the passivation and other layers which breaches the cryptographic boundary.

The means of $\mathcal{A}$ can be manifold. In our view the main limitations are caused by the technical equipment available. Because of this we distinguish the non-invasive adversary, the semi-invasive adversary, and the invasive adversary that are defined according to earlier work (e.g., [24, 17]) on fault induction.

Let $\mathcal{A}$ choose a physical interaction process $\mathcal{P}$. $\mathcal{A}$ uses *non-invasive* means if the interaction boundary of $\mathcal{P}$ is an outer boundary of the cryptographic boundary. We denote the non-invasive adversary by $\mathcal{A}_{non-inv}$. $\mathcal{A}$ uses *invasive* means if the interaction boundary of $\mathcal{P}$ is an inner boundary of the cryptographic boundary. We denote the invasive adversary by $\mathcal{A}_{inv}$. A *semi-invasive* adversary $\mathcal{A}_{semi-inv}$ uses light or electromagnetic fields as the interaction process and is a special case of $\mathcal{A}_{non-inv}$.

In circuitry, modifications of charges, currents and voltage levels may cause faults of the implementation. Modification of charges can be invoked by injecting charged particles or photons. For example, the underlying physical process for

optical fault induction is the photoelectric effect whereat injected photons are absorbed by the electronic semiconductor that in turn excites electrons from the valence band to the conduction band. Modification of currents can result from manipulating at the electrical circuit or by electromagnetic fields. Modifications of internal voltage levels within the cryptographic boundary are feasible by microprobing or the use of more sophisticated equipment, as focused ion beams. Note that often cumulative effects are needed to induce a fault, e.g., sufficient free carriers have to be generated or driven to load or unload a capacitance of the circuit. In the general case, multiple fault injections can not be considered as stochastically independent single fault injections, especially if their effects overlap in time or space.

Table 1. Physical Means according to the interaction range of an adversary

Adversary	Physical Means
$\mathcal{A}_{non-inv}$	glitches at external interfaces, changes of the environmental conditions
$\mathcal{A}_{semi-inv}$	light, electromagnetic radiation
$\mathcal{A}_{inv}$	active probes, charged particle beams

Spatial Resolution. If a special volume dV of the circuit $\mathcal{C}$ is targeted by the adversary then optimizing success rate requires that the physical interaction process needs to be injected into the cryptographic device with a good resolution in space. The following considerations are most suited for light, electromagnetic fields, and charged particles as interaction process.

We use $F(\boldsymbol{x}, E, t)$ to model the spatial, energetic and temporal density[2] of identical physical particles[3] as a function of a three-dimensional position vector $\boldsymbol{x} = (x, y, z)$, energy E and time t. Before impact on $\mathcal{C}$ the movement of the density is given by the three-dimensional velocity vector $\boldsymbol{v} = (v_x, v_y, v_z)$. For example, $F(\boldsymbol{x}, E, t)$ may describe a mono-energetic[4] light beam of photons that is injected into the circuit for a short amount of time.

Without loss of generality the circuit $\mathcal{C}$ is assumed to be in line with the two-dimensional $x - y$ plane (as seen in Fig. 3) at $z = 0$. The z-axis with $z \geq 0$ gives the penetration depth. An interaction process $\mathcal{P}$ of $F(\boldsymbol{x}, E, t)$ with the composition of electronic semiconductor material at position $\boldsymbol{x}$ is described by a differential cross section $\mathrm{d}\sigma(\boldsymbol{x})$, defined as $\mathrm{d}\sigma(\boldsymbol{x}) = \frac{dN(\boldsymbol{x})}{N(\boldsymbol{x})}$, wherein $dN(\boldsymbol{x})$ is the number of interacting particles per time unit dT and $N(\boldsymbol{x})$ is the number of particles that cross the area dA per time unit dT. Assuming that dA lies in a $x - y$ plane on the surface of $\mathcal{C}$ ($z = 0$), $N(\boldsymbol{x})$ is derived by $N(\boldsymbol{x}) = \int_0^{v_z \cdot dT} dz \int_{dA} dx\, dy \int_0^{\infty} dE\, F(\boldsymbol{x}, E, t)$.

Next, we consider the question of success probability to hit a target volume dV of $\mathcal{C}$ that is located at depth z with depth extension dz and spans an area

[2] The number of particles per space unit, per energy unit and per time unit.

[3] Correspondingly, one may consider a movement of a wave.

[4] The energy distribution can be modelled with the δ-function $\delta(E - E_0)$, i.e., all particles have energy E_0.

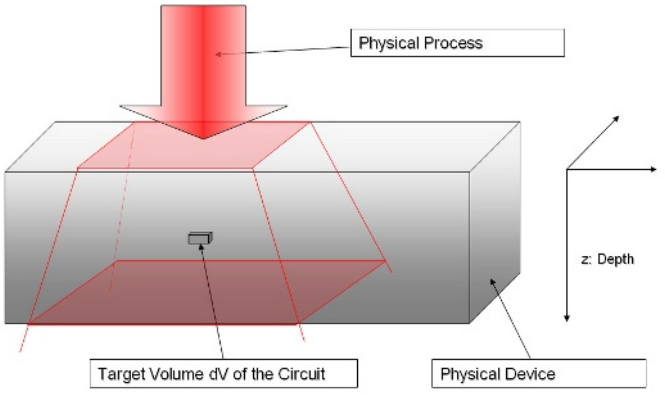

Fig. 3. Impact of the particle beam into the circuit

dA. During transfer through the circuit incident particles are partly absorbed, partly reflected and partly transmitted. Interaction processes with matter cause a decrease and spread of the energetic and spatial distribution of $F(\boldsymbol{x}, E, t)$ with increasing penetration range in $\mathcal{C}$. The interrelationship of $F(\boldsymbol{x}, E, t)$ as a function of penetration depth z is complex and does typically not solely depend on *one* interaction process. We assume that $F(\boldsymbol{x}, E, t)$ can be predicted for $z > 0$, e.g., by using a Monte-Carlo simulation of particles' movement by including the most important interaction processes as well as the circuit layout. The spatial spread of particles due to interactions shall be bounded by $\Delta A(z)$ in each $x-y$ plane within $\mathcal{C}$. Accordingly, the energetic spread shall be bounded by $\Delta E(z)$ in each $x-y$ plane within $\mathcal{C}$. In the general case, also the differential cross section depends on the energy E, so that we consider $\mathrm{d}\sigma(\boldsymbol{x}, E)$ from now on.

Then, $N_{dV} = \int_z^{z+dz} dz' \int_{dA} dx'\, dy' \int_0^{\Delta E(z')} dE'\, F(\boldsymbol{x'}, E', t)\; d\sigma(\boldsymbol{x'}, E')$ is the number of interacting particles in $dV = dz\, dA$. Let ΔV of $\mathcal{C}$ be the overall volume that is affected by the physical interaction process. Accordingly, in the volume ΔV it is $N_{\Delta V} = \int_0^{\Delta z} dz' \int_{\Delta A(z')} dx'\, dy' \int_0^{\Delta E(z')} dE'\, F(\boldsymbol{x'}, E', t)\; d\sigma(\boldsymbol{x'}, E')$ with Δz being the thickness of $\mathcal{C}$. The probability to cause an interaction process within the volume dV that is located between depth z to $z + dz$ with area extension dA given the overall affected volume ΔV with $N_{\Delta V} \neq 0$ is

$$p_V = \frac{N_{dV}}{N_{\Delta V}} = \frac{\int_z^{z+dz} dz' \int_{dA} dx'\, dy' \int_0^{\Delta E(z')} dE'\, F(\boldsymbol{x'}, E', t)\; d\sigma(\boldsymbol{x'}, E')}{\int_0^{\Delta z} dz' \int_{\Delta A(z')} dx'\, dy' \int_0^{\Delta E(z')} dE'\, F(\boldsymbol{x'}, E', t)\; d\sigma(\boldsymbol{x'}, E')} \tag{1}$$

Example 1. Mono-energetic beam with exponential attenuation in homogeneous material: $F(\boldsymbol{x'}, E', t) = F_0\, \delta(E' - E_0)\, e^{-az'}$ with $a = (10\mu m)^{-1}$, $\Delta A(z') = 10\mu m^2$, $\Delta z = 100\mu m$, $dA = 0.02\mu m^2$, $dz = 0.1\mu m$, $z = 20\mu m$ and $\sigma(\boldsymbol{x'}, E') = \sigma_0$. Then, it is $p_V = \frac{N_{dV}}{N_{\Delta V}} = \frac{dA\, e^{-a\, z}(1 - e^{-a\, dz})}{\Delta A\,(1 - e^{-a\, \Delta z})} \Longrightarrow p_V \approx 2.69 \cdot 10^{-6}$.

Spatial and Timing Resolution. So far, we considered spatial resolution. Often additionally timing resolution is required, e.g., the physical interaction process has to be induced at a specific time frame dt of the computation of the implementation, i.e., within the time interval $[t, t + dt]$.

When considering timing resolution in addition to spatial resolution (1) the corresponding probability is

$$p_{VT} = \frac{\int_t^{t+dt} dt' \int_z^{z+dz} dz' \int_{dA} dx'\, dy' \int_0^{\Delta E(z')} dE'\, F(\boldsymbol{x'}, E', t')\, d\sigma(\boldsymbol{x'}, E')}{\int_{-\infty}^{\infty} dt' \int_0^{\Delta z} dz' \int_{\Delta A(z')} dx'\, dy' \int_0^{\Delta E(z')} dE'\, F(\boldsymbol{x'}, E', t')\, d\sigma(\boldsymbol{x'}, E')}. \quad (2)$$

Example 2. Continuing the previous example with

$$F(\boldsymbol{x'}, E', t') = \begin{cases} F_0\, \delta(E' - E_0)\, e^{-az'}, & \text{if } t \le t' \le t + \Delta T \\ 0, & \text{otherwise} \end{cases}$$

with $dt = 10ns$ and $\Delta T = 100ns \Longrightarrow p_{VT} \approx 2.69 \cdot 10^{-7}$.

Immediate Consequences

- If $F(\boldsymbol{x}, E, t')$ does not reach the target area dV it is $p_{VT} = 0$.
- If $F(\boldsymbol{x}, E, t')$ is uniform in space and time and $\frac{dV}{\Delta V} \ll 1$ and $\frac{dt}{\Delta T} \ll 1$ then $p_{VT} \ll 1$ (e.g., in case of thermal radiation). It follows, that for $\mathcal{A}_{non-inv}$ it is $p_{VT} \ll 1$.

Sensitive and non-sensitive volumes of a circuit. We distinguish 'sensitive' and 'non-sensitive' volumes of the circuit $\mathcal{C}$ during computation of $\mathcal{S}$. A *sensitive volume* of the circuit at time t is composed of Boolean gates and memory cells that are used during computation of the security service $\mathcal{S}$ at the time t. The complementary set of volumes in $\mathcal{C}$ at time t is defined as *non-sensitive volume* of the circuit. As a consequence, physical interaction processes in non-sensitive volumes do not lead to a computational fault of $\mathcal{S}$, whereas physical interaction processes in sensitive volumes can have an impact on the computation of $\mathcal{S}$. In a refined version of (2) this fact can be included by neglecting non-sensitive volumes of the circuit at time t.

4 Physical Security Bounds

As already outlined, we assume a strong adversary $\mathcal{A}$ who is given a map of $\mathcal{C}$ including a behavioral simulation that also indicates sensitive and non-sensitive volumes of a circuit $\mathcal{C}$ for each time t. Given these means, $\mathcal{A}$ is able to perform a vulnerability analysis of $\mathcal{C}$ and to identify tampering attack paths of $\mathcal{C}$.

For security notions, metrics are needed to quantify physical properties of $\mathcal{C}$. Defining such quantities for a circuit $\mathcal{C}$ is strongly dependent on the concrete layout and has to consider all feasible attack paths, i.e. the set of all admissible events for $\mathcal{A}$. Suitable metrics of $\mathcal{C}$ could be, but are not limited to (i) the size of target gates, (ii) the attacking time frame for target gates, (iii) the smallest

Euclidean distance between target gates and the cryptographic boundary of $\mathcal{C}$, and (iv) the smallest Euclidean distance between target gates and other sensitive volumes of $\mathcal{C}$.

A circuit $\mathcal{C}$ implementing a security service S is said to be *statistically secure in the average case against an (N, L, M)-limited tampering adversary* if for all physical interaction processes $\mathcal{P}$ there exists a negligible function $\text{negl}(\mathcal{C}, N, L, M)$ such that the success probability of a fault analysis scenario is bounded by $\text{negl}(\mathcal{C}, N, L, M)$. For concreteness, if event E is the fault analysis scenario against a Digital Signature Scheme based on adaptive chosen messages of Fig. 2, then $\Pr(E) \leq \text{negl}(\mathcal{C}, N, L, M)$ for the given circuit $\mathcal{C}$. As previously said, the function $\text{negl}(\mathcal{C}, N, L, M)$ depends on the concrete circuit layout. It is still an open question whether physical quantities can be formally tied to security notions in a realistic physical model for tampering.

4.1 Countermeasure Strategies

We consider generic passive and active physical defense strategies that result from physical means detailed in Section 3.2. Passive defense strategies aim at significantly reducing the success probability for fault injection (fault prevention). Active defense strategies require that $\mathcal{D}$ is capable to detect computational errors resulting from faults (error detection) or the presence of abnormal conditions that may lead to faults (fault detection). In any case, reliable defense strategies have to be part of the construction of $\mathcal{D}$. Combinations of these defense strategies are feasible, especially as most strategies have an impact on different parameters in (2). The decision whether or not the device shall enter a permanent non-responsive mode in case of error or fault detection depends on the concrete impact probability as well as the concrete security service. It is a matter of risk evaluation.

Table 2. Passive and Active Defense Strategies

Strategy	Impact on Parameter	Security Objective
Shrinking	dA, dz and N	fault prevention
Passive Encapsulation	z and $\mathrm{d}\sigma(\boldsymbol{x}, E)$	fault prevention
Timing Modifications	t, dt and N	fault prevention
Error Detection Codes	L and N	error detection
Physical Duplication	L and N	error detection
Repeating Computations	M and N	error detection
Sensors	$\Delta A(z)$ and N	fault detection

Shrinking: Due to the shrinking process, integrated circuits become more and more compact. Shrinking decreases the target volume $dA \cdot dz$. Upcoming chip technology is based on 90 nm structures. For comparison, a focus of a laser beam on the chip surface of 1 μm was reported in [24] at an optical fault injection setup. Due to the limited spatial resolution, multiple faults at neighboring gates are much more likely to occur than single faults at the target resulting in an increase

of N. Note that shrinking may enhance the sensitivity of the circuit so that less free carriers or currents are needed for fault injection.

Passive Encapsulation: Passive encapsulation aims that the interaction process is absorbed or reflected before its effects reach the target area, i.e., $F(\boldsymbol{x}, E, t)$ should not reach the target area dV at depth z resulting in $p_{VT} = 0$ in (2). Such an encapsulation has to be constructed within the cryptographic boundary of the device to prevent it from the reach of $\mathcal{A}_{non-inv}$ and $\mathcal{A}_{semi-inv}$. One approach includes shields that are non-transparent in a broad light spectrum and prevent throughpassing of photons, i.e., aiming at high values of $\mathrm{d}\sigma(\boldsymbol{x}, E)$ within the shield. A simpler design aim is to place security critical parts in center of the chip to prevent both attacks from the front as well as from the back-end side of the chip. If considering different physical interaction processes $\mathcal{P}$, the range of $F(\boldsymbol{x}, E, t)$ in semiconductor materials has to be evaluated, i.e., the average value of the depth to which a particle will penetrate in the course of slowing down to rest. This depth is measured along the initial direction of the particle. For high energy particles these data can be found at [21]. However, against invasive adversaries the effectiveness of passive encapsulation is quite limited.

Timing Modifications: This strategy can be useful if timing is crucial in a concrete fault analysis scenario. The objective is to randomly embed the relevant time interval dt within a larger time interval which leads to an enhancement of N. A possible realization includes delaying and interrupting the processing of $\mathcal{C}$, e.g., according to the value of a randomized internal counter. If the physical leakage of $\mathcal{C}$ can not be analyzed in real-time, an adversary is not able to adapt to the randomized timing. Instead, the source of randomness in the circuit may become an attractive target. Similarly, increasing the clock frequency of the circuits may help to increase N.

Error Detection Codes: Error detection codes of data items are well known for software implementations. For implementation in circuitry, [15] introduced parity based error detection at a substitution-permutation network based block cipher. In [20] error detection techniques based on multiple parity bits and non-linear codes are evaluated. Among them, r-bit non-linear codes are the most promising, but at cost of an area overhead that is nearly comparable to duplication. As result, error detection codes lead to an enhancement of L which in turn typically increases N.

Physical Duplication: The objective is to duplicate critical target volumes of the circuit. In the context of asynchronous circuits, [10] has already proposed this idea to improve tamper resistance. These circuits make use of redundant data encoding, i.e., each bit is encoded onto two wires. Such dual-rail coding offers the opportunity to encode an alarm signal that can be used for error detection by the physical device. For memory cells, a 'dual flip-flop dual-rail' design is proposed. The main idea is that an error state on any gate input is always propagated to the gate output. In case of area duplication, the number of locations for fault injection is typically doubled, i.e., L is enhanced and precise control over the fault injection process is needed to prevent an error detection.

Repeating Computations: Repeating computations of the circuit and comparison of results is another strategy for error detection. However, this method is not reliable if a permanent fault is present in the circuit. In case of transient errors, repeating leads to an enhancement of M.

Sensors: Here, a network of short-distance sensors is spanned at critical parts of the circuit. The mean distance between sensors then gives an upper bound on the area $\Delta A(z)$ at which fault injection may not be detected by the sensors. It is aimed that an adversary has to precisely focus only on the target volume dV which establishes a hard problem for $\mathcal{A}_{non-inv}$ and $\mathcal{A}_{semi-inv}$. Alarm detection may be deployed at an active encapsulation within the cryptographic boundary of the device. Again, this encapsulation should be out of the reach of $\mathcal{A}_{non-inv}$ and $\mathcal{A}_{semi-inv}$. A different approach is given in reference [10]: the authors suggest to include small optical tamper sensors within each standard cell. These sensors consist of one or two transistors and enforce an error state if illuminated.

5 Conclusion

Implementation security is different from algorithmic security: for the assessment of implementation security, properties of the concrete layout and timing of the circuit are needed. In this contribution we initiate an approach towards the evaluation of physical security against tampering adversaries. We consider manipulating computations in circuitry and give a physical model on fault injection based on radiation and particle impact. We assume that fault injection can be both applied prior and during computations of a physical security service which is a realistic assumption that should be also included in provable security models. We hope that this framework is useful to both map concrete impact probabilities of a given circuit as well as to improve the circuits' layout.

References

1. ISO 13491-1:1998 Banking – Secure cryptographic devices (retail)– Part 1: Concepts, requirements and evaluation methods.
2. FIPS PUB 140-2, Security Requirements for Cryptographic Modules, 2001.
3. Ross Anderson and Markus Kuhn. Tamper Resistance — A Cautionary Note. In *The Second USENIX Workshop on Electronic Commerce Proocedings*, pages 1–11, 1996.
4. Christian Aumüller, Peter Bier, Wieland Fischer, Peter Hofreiter, and Jean-Pierre Seifert. Fault Attacks on RSA with CRT: Concrete Results and Practical Countermeasures. In Jr. et al. [14], pages 260–275.
5. Hagai Bar-El, Hamid Choukri, David Naccache, Michael Tunstall, and Claire Whelan. The Sorcerer's Apprenctice's Guide to Fault Attacks, available at http://eprint.iacr.org/2004/100. Technical report, 2004.
6. Eli Biham, Louis Granboulan, and Phong Q. Nguyen. Impossible Fault Analysis of RC4 and Differential Fault Analysis of RC4. In Henri Gilbert and Helena Handschuh, editors, *Fast Software Encryption: 12th International Workshop*, volume 3557 of *Lecture Notes in Computer Science*, pages 359–367. Springer, 2005.

7. Eli Biham and Adi Shamir. The Next Stage of Differential Fault Analysis: How to break completely unknown cryptosystems, available at http://jya.com/dfa.htm, 1996.
8. Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In Burton S. Kaliski Jr., editor, *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 513–525. Springer, 1997.
9. Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the Importance of Checking Cryptographic Protocols for Faults (Extended Abstract). In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 37–51. Springer, 1997.
10. Jacques J. A. Fournier, Simon W. Moore, Huiyun Li, Robert D. Mullins, and George S. Taylor. Security evaluation of asynchronous circuits. In Walter et al. [27], pages 137–151.
11. Rosario Gennaro, Anna Lysyanskaya, Tal Malkin, Silvio Micali, and Tal Rabin. Algorithmic Tamper-Proof (ATP) Security: Theoretical Foundations for Security against Hardware Tampering. In Moni Naor, editor, *Theory of Cryptography*, volume 2951 of *Lecture Notes in Computer Science*, pages 258–277. Springer, 2004.
12. William N. Havener, Roberta J. Medlock, Lisa D. Mitchell, and Robert J. Walcott. Derived Test Requirements for FIPS PUB 140-1, Security Requirements for Cryptographic Modules, 1995.
13. Jonathan J. Hoch and Adi Shamir. Fault analysis of stream ciphers. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 240–253. Springer, 2004.
14. Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors. *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*. Springer, 2003.
15. Ramesh Karri, Grigori Kuznetsov, and Michael Goessel. Parity-Based Concurrent Error Detection of Substitution-Permutation Network Block Ciphers. In Walter et al. [27], pages 113–124.
16. Oliver Kömmerling and Markus G. Kuhn. Design Principles for Tamper-Resistant Smartcard Processors. In *Proceedings of the USENIX Workshop on Smartcard Technology (Smartcard '99)*, pages 9–20, 1999.
17. Kerstin Lemke and Christof Paar. An Adversarial Model for Fault Analysis against Low-Cost Cryptographic Devices. In *Workshop on Fault Detection and Tolerance in Cryptography*, pages 82–94, 2005.
18. Regís Leveugle. Early Analysis of Fault Attack Effects for Cryptographic Hardware. In *Workshop on Fault Detection and Tolerance in Cryptography*, 2004.
19. P.-Y. Liardet and Y. Teglia. From Reliability to Safety. In *Workshop on Fault Detection and Tolerance in Cryptography*, 2004.
20. Tal G. Malkin, François-Xavier Standaert, and Moti Yung. A Comparative Cost/Security Analysis of Fault Attack Countermeasures. In *Workshop on Fault Detection and Tolerance in Cryptography*, pages 109–123, 2005.
21. National Institute of Standards and Technology (NIST). Physical Reference Data, available at http://physics.nist.gov/PhysRefData/contents.html.
22. Gilles Piret and Jean-Jacques Quisquater. A Differential Fault Attack Technique against SPN Structures, with Application to the AES and KHAZAD. In Walter et al. [27], pages 77–88.
23. David Samyde and Jean-Jacques Quisquater. Eddy Current for Magnetic Analysis with Active Sensor. In *Proceedings of ESmart 2002*, pages 185–194, 2002.

24. Sergei P. Skorobogatov and Ross J. Anderson. Optical Fault Induction Attacks. In Jr. et al. [14], pages 2–12.
25. Sergei S. Skorobogatov. Semi-invasive attacks — A new approach to hardware security analysis, available at http://www.cl.cam.ac.uk/techreports/ucam-cl-tr-630.pdf. Technical report, 2005.
26. F.-X. Standaert, L. Batina, E. de Mulder, K. Lemke, E. Oswald, and G. Piret. ECRYT D.VAM.4: Electromagnetic Analysis and Fault Attacks: State of the Art. Technical report, 2005.
27. Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors. *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings*, volume 2779 of *Lecture Notes in Computer Science*. Springer, 2003.