# An Accurate Probabilistic Reliability Model for Silicon PUFs

Roel Maes

Intrinsic-ID, Eindhoven, the Netherlands
roel.maes@intrinsic-id.com

**Abstract.** The power of an accurate model for describing a physical process or designing a physical system is beyond doubt. The currently used reliability model for physically unclonable functions (PUFs) assumes an equally likely error for every evaluation of every PUF response bit. This limits an accurate description since experiments show that certain responses are more error-prone than others, but this *fixed error rate model* only captures *average case* behavior. We introduce a new PUF reliability model taking this observed heterogeneous nature of PUF cells into account. An extensive experimental validation demonstrates the new predicted distributions describe the empirically observed data statistics almost perfectly, even considering sensitivity to operational temperature. This allows studying PUF reliability behavior in full detail, including average and *worst case* probabilities, and is an invaluable tool for designing more efficient and better adapted PUFs and PUF-based systems.

## 1 Introduction

After a decade of ongoing scientific research and sustained technical development, silicon PUF technology [1,2] is steadily finding its way into electronic products [3,4]. To meet the high reliability and security constraints imposed by such applications, bare silicon PUFs don't operate on their own but are embedded in a system. The fundamental physical security of such a system originates from the PUF implementation, but considerable post-processing is involved to meet the overall requirements and facilitate the intended application, e.g. key storage. Constructing a PUF system is an intricate design exercise since it requires balancing typically opposing goals between reliability, security and efficiency.

The starting point of a PUF system design is evidently the probabilistic behavior of the PUF itself, both regarding reliability (*error behavior*) and security (*unpredictability behavior*). The more insight one has in these details, the better one is able to fine tune design choices, and the more confidence one has in the obtained results. To consistently deal with a PUF's probabilistic behavior, an accurate model which closely fits empirical

statistics is of great importance. Such a model should be sufficiently generic to confidently extrapolate predictions to unobserved points and allow working with a variety of PUF constructions. It will proof an indispensable tool for analyzing the design space of a PUF system and converging on an optimized solution. The main focus of this work is the development and analysis of a more accurate and generic *reliability model* for silicon PUFs than the one in use today, and a demonstration of its advantages.

*Related Work.* The commonly used PUF reliability model, e.g. in [2,5,6,7,8,9] and many others, is that of a *fixed error rate*, i.e. each evaluation of each response bit is assumed equally likely to be wrong. Many details are lost by reducing the reliability behavior to a single average-case parameter. A first extension of this model, e.g. as used in [10,11,12], is the binary differentiation between *stable* and *unstable* PUF response bits. This idea is generalized in [13] which demonstrates that PUF cell reliabilities are continuously distributed, from very unreliable to almost perfectly stable.

*Contributions.* In this work, we start from and greatly expand on the model as proposed in [13], to describe PUF reliability behavior in a much more accurate and detailed manner as has been done up to now. The basic model from [13] is modified to more realistically describe error-behavior, and extended to take environmental dependencies like temperature into account. This new model is extensively validated on reliability data from measurements of PUFs implemented in 65nm CMOS. The fit between predicted distributions and empirical statistics is strikingly accurate at all measured temperatures from $-40°$C to $+85°$C. Moreover, the model proves to be very generic by being extremely accurate for different types of memory-based PUF types, like the SRAM PUF [2], the buskeeper PUF [14] and the D Flip-flop PUF [15], as well as for the delay-based arbiter PUF [16]. We also demonstrate the gained insight offered by such an accurate model, by analyzing the implications for key generation. This clearly shows the limitations of the old fixed error rate model, and the added value of designing a PUF system using the new model.

*Overview.* Sect. 2 introduces the newly proposed model, motivates the assumed relations, and derives the hypothesized distribution functions. The model's accuracy is consequently validated in Sect. 3 by fitting it on empirical statistics from actual silicon PUF measurements. The gained insights of the new model and their consequences for PUFs and PUF-based applications are discussed in Sect. 4. Finally, we identify the potential for future work based on these findings and conclude in Sect. 5.

## 2 Model Description

### 2.1 Notation and Preliminaries

Without loss of generality we consider silicon PUFs with single-bit responses. For the sake of clarity, the presented model is introduced in terms of memory-based PUFs, where each bit is produced by an individual (memory) *PUF cell*.[1] However, as demonstrated, the applicability of the model is certainly not limited to memory-based PUFs, but is also particularly accurate in describing the reliability behavior of delay-based silicon PUFs.

*Variable Notation.* Most of the model's variables are random in nature. We distinguish between random values sampled *once* for a particular PUF cell $i$ (upon creation) and remain fixed for the cell's entire lifetime, which are denoted with subscript indexing ($m_i$), and others which are resampled every time the cell is evaluated, which are denoted with superscript indexing ($n_i^{(j)}$ for evalutation $j$ of cell $i$). Random variables in general are denoted as capital literals, e.g. $M$ is the random variable which is sampled to a value $m_i$ for cell $i$, according to the distribution of $M$.

*Distribution Functions.* The distribution of a random variable $X$ is characterized by its probability density function ($\mathbf{pdf}_X(x)$) and/or its cumulative distribution function ($\mathbf{cdf}_X(x)$). For discrete random variables, the probability density function degenerates to a probability mass function ($\mathbf{pmf}_X(x)$). Two basic distributions used in this work are the (standard) normal distribution ($\mathbf{pdf}_X(x) = \varphi(x)$ and $\mathbf{cdf}_X(x) = \Phi(x)$) and the binomial distribution ($\mathbf{pmf}_X(x) = f_{\text{bino}}(x; n, p)$ and $\mathbf{cdf}_X(x) = F_{\text{bino}}(x; n, p)$). We refer to App. A for details on these distributions.

### 2.2 The "Old" Model: PUF Response with Fixed Error Rate

We first briefly discuss the probabilistic model which is thus far used in the majority of related literature (e.g. in [2,5,6,7,8,9]) for assessing the reliability of PUFs and their applications.

*Rationale.* The foundation of the old model is the assumption that all cells of a PUF are *homogeneous*, i.e. every cell in the PUF is equally likely to produce an error at any time. This means the reliability behavior of the PUF as a whole is described by a single fixed parameter: the *(bit)*

[1] We refer to the literature on memory-based PUFs and silicon PUFs in general for more details on their operation and implementation. See e.g. [17] for an overview.

*error rate* ($p_e$). This is the probability that *any* evaluation of *any* cell differs from its enrolled response, and is assumed equal to the average-case behavior averaged over many cells.

*Limitations.* Though convenient to use, this model's limitations are evident when looking at experimental PUF results. A typical PUF instantiation exhibits *unstable* and *stable* cells, i.e. some cells are more likely to produce an error while other cells are hardly ever wrong. This behavior is not captured by the old model which treats every cell in the same way. However, as shown in Sect. 4, it is wise to take this observation into account when designing PUF-based applications. The main motivation behind the newly introduced model is to accurately capture this cell-specific behavior.

### 2.3 The "New" Model: Cell-Specific Error-Probabilities

In line with the experimental observation that some PUF cells are more error-prone than others, the foundation of the new model lies in the assumed cell *heterogenity*, i.e. every cell in a PUF has an individual error-probability. An early form of this basic idea was introduced in [13] and serves as a starting point for the new model presented here.

**Hidden Variable Model.** The implied approach of [13], which we make explicit, is that of a *hidden variable model*. Basically it is assumed that the *observable variables* of a PUF cell, which describe its observable behavior, are governed by underlying *hidden variables*. By assuming plausible distributions for the hidden variables, the resulting distributions of the observable variables are derived and validated against experimental data.

*The Observable Variables* describe the probabilistic behavior of an evaluation ($j$) of a PUF cell $i$ to a response bit value $r_i^{(j)} \in \{0,1\}$ (a random sampling of $R_i$):

- *The One-Probability* ($p_i$) of a cell $i$ is the probability that it returns '1' upon a random evaluation: $p_i \stackrel{\text{def}}{=} \mathbf{Pr}\left(R_i = 1\right)$. The one-probability is itself a random variable $P$ randomly sampled to a value $p_i \in (0,1)$ for a cell $i$.
- *The Error-Probability* ($p_{e,i}$) of a cell $i$ is the probability that a random evalutation differs from an earlier recorded evaluation of that cell during an *enrollment phase*[2]: $p_{e,i} \stackrel{\text{def}}{=} \mathbf{Pr}\left(R_i \neq r_i^{\text{enroll}}\right)$. The error-probability is itself a random variable $P_e$ randomly sampled to a value $p_{e,i} \in (0,1)$.

[2] In [13], error-probability is defined with respect to a cell's *most-likely* outcome which is not representative for the realistic use of a PUF. Therefore, we consider a

*The Hidden Variables* are abstractions of underlying physical (electrical) processes in a silicon PUF cell circuit. We do not consider low-level physical details explicitly to avoid complex simulations and to maintain a generic model. The used hidden variables are regarded as generic and approximated *lumped* versions of underlying measurable physical quantities:

- *The Process Variable* ($m_i$) quantifies the accumulated effect of process variations on a cell's internals, introduced during manufacturing. This is a random variable ($M$), sampled at a cell's creation time, according to a distribution determined by the manufacturing process.
- *The Noise Variable* ($n_i^{(j)}$) quantifies the accumulated effect of random noise on a cell's internals during evaluation. This is a random variable ($N_i$), resampled for every evaluation of the cell, according to a distribution determined by the cell's susceptibility to noise.

*The Model Relation* is the fundamental connection between hidden and observable variables from which all further conclusions are derived:

$$\boxed{r_i^{(j)} = \begin{cases} 0 \text{ , if } m_i + n_i^{(j)} \leq t \,, \\ 1 \text{ , if } m_i + n_i^{(j)} > t \,. \end{cases}} \tag{1}$$

The implied assumptions of this relation are: *i)* that the hidden variables are *additive*,[3] and *ii)* that the evaluation outcome is the result of a comparison with a constant *threshold parameter* $t$. The relation for the one-probability is directly derived from (1) as: $p_i = \mathbf{Pr}\left(m_i + N_i > t\right) = 1 - \mathbf{cdf}_{N_i}\left(t - m_i\right)$.

**Distributions of the New Model.** Since both hidden variables are considered lumped physical quantities, a normal distribution is a motivated assumption for both: $M \sim \mathcal{N}(\mu_M, \sigma_M^2)$, and $N_i \sim \mathcal{N}(0, \sigma_N^2)$. For ease of notation, the parameters $\lambda_1 = {\sigma_N}/{\sigma_M}$, and $\lambda_2 = {(t - \mu_M)}/{\sigma_M}$ are used. Based on these assumed distributions, the resulting observable variable distributions are derived by employing the model relation as expressed in (1). The one-probability distribution was already derived in [13]:[4]

$$\mathbf{cdf}_P\left(x\right) = \Phi\left(\lambda_1 \Phi^{-1}\left(x\right) + \lambda_2\right) \,. \tag{2}$$

---

random enrollment instead: $r_i^{\text{enroll}}$ is randomly sampled according to the one-probability $p_i$, and can (coincidentally) be an unlikely outcome for the considered cell

[3] This is intuitively justified by considering that the hidden variables are of an *electrical* nature, i.e. voltages or currents. Additivity then follows from Kirchoff's laws.

[4] Since $P$ and $P_e$ represent probabilities, $\mathbf{cdf}_P\left(x\right)$ and $\mathbf{cdf}_{P_e}\left(x\right)$ are only defined for $x \in (0, 1)$.

The detailed derivation of the new error-probability distribution is presented in App. B.1 and results in:[5]

$$\mathbf{cdf}_{P_e}(x) = \lambda_1 \cdot \int_{-\infty}^{\Phi^{-1}(x)} \Phi(-u) \cdot (\varphi(\lambda_1 u + \lambda_2) + \varphi(\lambda_1 u - \lambda_2)) \, \mathrm{d}u \,. \quad (3)$$

### 2.4 Modeling Temperature Dependence

From many PUF experiments (e.g. in [18]) it is clear that the operating conditions of a silicon PUF, such as temperature and voltage, have a noticeable impact on response behavior. At increasingly different conditions this even becomes the primary source of unreliability, much more so than instantaneous random noise. To realistically describe a PUF cell's error-behavior we incorporate these effects in the new model. This is done for *temperature*, which typically has the largest impact on PUF reliability [18].[6]

**Hidden Variable Model: Temperature Extension.** The basic hidden variable model from Sect. 2.3 is extended with a new hidden variable quantifying a cell's sensitivity to temperature: the *temperature dependence* ($d_i$). Since different cells react differently to temperature changes, this is a cell-specific value randomly sampled at manufacturing time. The observable variables are straightforwardly extended to express temperature dependence: $p_i(T) = \mathbf{Pr}(R_i(T) = 1)$ and $p_{e,i}(T; T_{ref}) = \mathbf{Pr}\left(R_i(T) \neq r_i^{\text{enroll}}(T_{ref})\right)$. Note that error-probability depends on two temperatures, at enrollment ($T_{ref}$) and at reconstruction ($T$).

*The Temperature Model Relation* extends the additive threshold relation of the new model as given by (1) with a temperature dependent term. This relation assumes a linear dependence on the (absolute) temperature with a cell-dependent sensitivity quantified by $d_i$:

$$\boxed{r_i^{(j)}(T) = \begin{cases} 0 \text{ , if } m_i + n_i^{(j)} + d_i \cdot T \leq t \,, \\ 1 \text{ , if } m_i + n_i^{(j)} + d_i \cdot T > t \,. \end{cases}} \quad (4)$$

**Distribution of the Temperature Model.** For the temperature dependence variable we also assume a normal distribution: $D \sim \mathcal{N}(0, \sigma_D^2)$. A

[5] This and following integral expressions are evaluated using numerical methods.

[6] Other conditions can be equivalently modelled but are omitted due to lack of space.

third model parameter is introduced as $\theta = {}^{\sigma_N}/_{\sigma_D}$. Following the temperature model relation expressed by (4), the distribution of the temperature-dependent error-probabilities becomes:

$$\mathbf{cdf}_{P_e(T;T_{ref})}(x) = \frac{\lambda_1\theta}{|\Delta T|} \cdot \int_{-\infty}^{\Phi^{-1}(x)} \int_{-\infty}^{+\infty} \Big[\Phi(-u)\,\varphi\left(\theta\tfrac{v-u}{|\Delta T|}\right) + \Phi(u)\,\varphi\left(\theta\tfrac{v+u}{|\Delta T|}\right)\Big] \cdot \varphi(\lambda_1 u + \lambda_2)\,\mathrm{d}u\,\mathrm{d}v\,. \tag{5}$$

The complete derivation is given in App. B.2. We introduced $\Delta T = T - T_{ref}$, and (5) is only defined for $\Delta T \neq 0$. In case $T = T_{ref}$, the limiting case of (5) for $\Delta T \to 0$ reverts to (3).

## 3 Experimental Validation

We assess the validity of the assumptions made in Sect. 2 by fitting the predicted error-probability distribution to empirically observed statistics. For this purpose we use the extensive experimental PUF data set originating from the UNIQUE project [19], of which the initial analysis was presented in [18,20]. This data set was acquired from 192 ASICs manufactured in 65nm CMOS, each implementing six silicon PUF types. We applied our model in particular to the SRAM, D flip-flop, buskeeper and arbiter PUFs.

### 3.1 From Error-Probability to Error-Count

The error-probability of a particular PUF cell can be estimated by counting the number of errors in a number of cell evaluations and dividing it by that number. However, since the majority of cells typically has an error-probability very close to 0, this estimate is rather inaccurate when the number of evaluations is limited. E.g., based on 100 measurements of cell $i$ which are all error-free, it is impossible to differentiate between $p_{e,i} = 10^{-3}$ or $p_{e,i} = 10^{-6}$ or even smaller. This inaccuracy hampers an accurate fit of the model, especially in the distribution tails (close to 0 and 1) which happen to be the most interesting parts. To overcome this problem we introduce a variable closely related to the error-probability but directly observable in experimental data without estimation accuracy problems: the *error-count* $s_{e,i}^{(n)}$ is the number of evaluations in $n$ measurements of cell $i$ which differ from an enrollment response bit for that cell. By consequence, the value of $s_{e,i}^{(n)}$ is also a random value sampled (at a given temperature $T$), according to the discrete distribution characterized by:

$$\mathbf{pmf}_{S_e^{(n)}(T;T_{ref})}(x) = \int_0^1 f_{\text{bino}}(x; n, u) \cdot \mathbf{pdf}_{P_e(T;T_{ref})}(u)\,\mathrm{d}u\,. \tag{6}$$

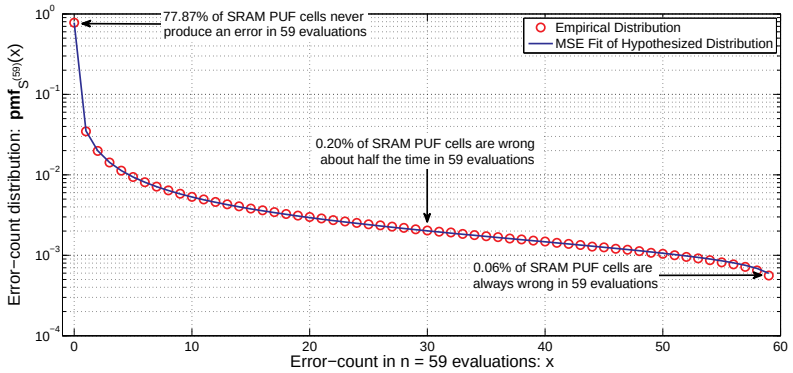


**Fig. 1.** Fit of $\mathbf{pmf}_{S_e^{(59)}}(x)$ on empirical SRAM PUF data at 25°C.

In this section, we focus on fitting this distribution to the error statistics of the experimental PUF data. The expression for $\mathbf{pdf}_{P_e(T;T_{ref})}(u)$ is obtained by differentiating (3) (if $T = T_{ref}$) or (5) (if $T \neq T_{ref}$) and is listed for completeness in App. B.2.

### 3.2 Fitting the Error-Count Distribution

**Fitting $(\lambda_1, \lambda_2)$ at $T_{ref} = 25°C$.** The first experimental data set we use for fitting the parameters $(\lambda_1, \lambda_2)$ consists of 60 evaluations of 65,536 cells from 768 identical but distinct SRAM PUF instantiations at a fixed temperature of $T_{ref} = 25°C$.[7] This totals to $768 \times 65{,}536 = 50{,}331{,}648$ distinct but identically implemented SRAM PUF cells all evaluated 60 times. We randomly pick one enrollment response and 59 reconstruction evaluations from which we calculate the error-count $s_{e,i}^{(59)}$ for each PUF cell $i$ with respect to its enrollment value. From these 50,331,648 randomly sampled error-count values the empirical distribution of $S_e^{(59)}$ is calculated. If the model from Sect. 2.3 is accurate, then the hypothesized distribution of $S_e^{(59)}$ as characterized by (6) should closely fit the empirical histogram. We perform a non-linear optimization over $(\lambda_1, \lambda_2)$ using the Levenberg-Marquardt algorithm to minimize the mean squared error (MSE) between the empirical and hypothesized probability mass functions. The result is shown in Fig. 1 and shows that the function from (6) yields a strikingly accurate fit. The closest fit was found for $\mathbf{(\lambda_1 = 0.1213\,,\ \lambda_2 = 0.0210)}$ with an MSE of merely $4.467 \cdot 10^{-9}$.

[7] The 768 SRAM PUFs are implemented on 192 ASICs, with 4 instances per chip.

**Table 1.** Fit results of $\mathbf{pmf}_{S_e^{(n)}}(x)$ on empirical data of different PUF types at 25°C.

| PUF Type | Silicon PUF | MSE of fit | $\lambda_1$ | $\lambda_2$ |
|---|---|---|---|---|
| Memory-based | SRAM PUF | $4.467 \cdot 10^{-9}$ | 0.1213 | 0.0210 |
| Memory-based | Buskeeper PUF | $5.760 \cdot 10^{-10}$ | 0.0929 | 0.0340 |
| Memory-based | D Flip-flop PUF | $1.150 \cdot 10^{-9}$ | 0.0812 | 0.0381 |
| Delay-based | Arbiter PUF | $1.843 \cdot 10^{-9}$ | 0.0676 | 0.0461 |

To demonstrate the generic nature of the proposed model we also apply it to other silicon PUF types. We considered the experimental data of 60 evaluations of 8,192 cells from 384 instantiations, for each of the buskeeper, the D flip-flop and the arbiter PUF.[8] All fitting results are summarized in Table 1 and show that the best fit for each of these alternative PUF types is at least as accurate as that for the SRAM PUF. Remarkably, the model succeeds in accurately predicting the reliability distributions for both memory-based as well as delay-based PUFs.

**Fitting $\boldsymbol{\theta}$ for the SRAM PUF at $\boldsymbol{T = [-40°\text{C}, \ldots, +85°\text{C}]}$.** To validate the temperature dependence of the model as presented in Sect. 2.4, we use an experimental data set obtained from 65,536 cells from a limited set of 20 identical but distinct SRAM PUF instantiations, evaluated 100 times at thirteen temperatures between −40°C and 85°C. This gives a total set of $20 \times 65{,}536 = 1{,}310{,}720$ cells, for each of which we calculate the error count $s_{e,i}^{(100)}(T; T_{ref})$ at every measured temperature with respect to a randomly selected enrollment response at $T_{ref} = 25°\text{C}$. The accuracy of the temperature model is tested by fitting the hypothesized distribution of $S_e^{(100)}(T; 25°\text{C})$, as characterized by (6), to the empirical distribution of these 1,310,720 samples at every measured $T \neq T_{ref}$. We use the estimated parameter values for $(\lambda_1, \lambda_2)$ from the previous experiment, and perform an optimization over the remaining parameter $\theta$ to minimize the average MSE between the empirical and hypothesized probability mass functions over all $T$. The results are shown in Fig. 2 and demonstrate an accurate fit at every considered temperature. A minimal average MSE of $1.643 \cdot 10^{-6}$ over all temperatures is obtained for $\boldsymbol{\theta = 45.0}$, with the largest deviation at the extreme temperature of −40°C (MSE of $5.208 \cdot 10^{-6}$). Given the single parameter linear temperature dependence assumed by the model, as given by (4), the fitted distributions are remarkably accurate.

[8] For the arbiter PUF, a "cell" refers to an evaluation with a random challenge.

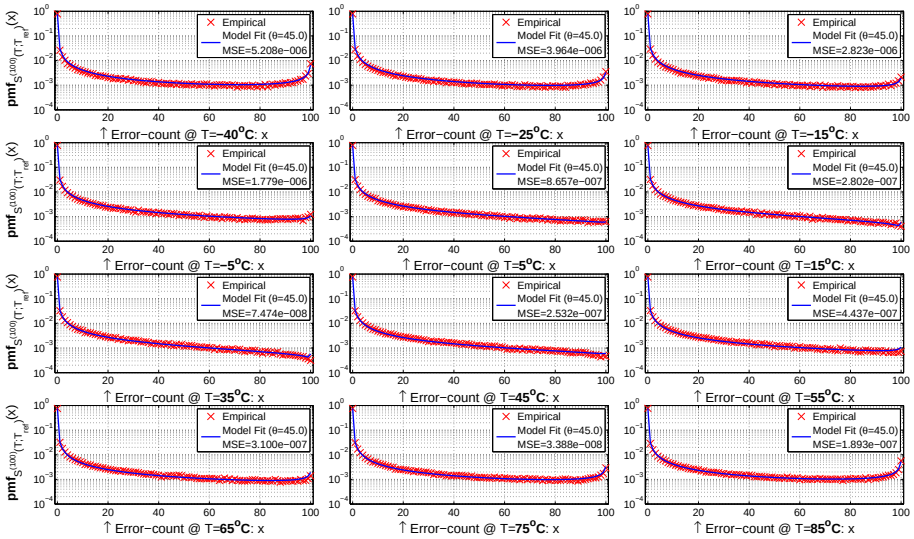


**Fig. 2.** Fit of $\mathbf{pmf}_{S_e^{(100)}(T;T_{ref}=25°\mathrm{C})}(x)$ on empirical SRAM PUF data for different $T$.

## 4 Interpretation and Discussion

We are now able to quantify the consequences of the heterogenity of individual PUF cells. We first interpret the reliability distribution directly in Sect. 4.1 and study the effect on PUF-based key generation in Sect. 4.2.

### 4.1 Interpretation of the New Model Distributions

We consider the experimentally studied SRAM PUF from Sect. 3, with fitted model parameters: $(\lambda_1 = 0.1213, \lambda_2 = 0.0210, \theta = 45.0)$. The error-probability distribution is analysed at the worst-case temperature $T = -40°\mathrm{C}$ with respect to enrollment at $T_{ref} = 25°\mathrm{C}$. The cumulative distribution function is plotted in Fig. 3. From this graph the heterogene nature of the individual PUF cells is immediately clear. A remarkable observation is that about 34% of the SRAM PUF cells have an error-probability $\leq 10^{-15}$, i.e. in any practical setting they are always correct. On the other hand, about 7% of the cells produce an error in more than 50% of their evaluations, and about 1% of the cells in more than 99%.[9] Another remarkable observation is the discrepancy between the *mean*

[9] Cells with very high ($> 50\%$) error-probabilities are caused by a cell coincidentally assuming an unlikely value during enrollment, or alternatively because a cell's prefered value changes over the temperature shift between enrollment and reconstruction.

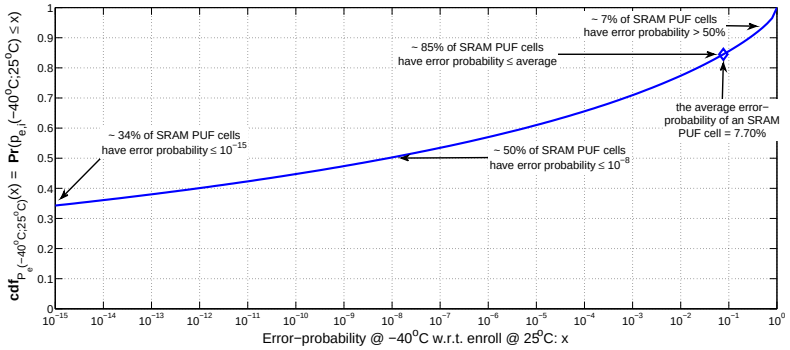


**Fig. 3.** Plot of $\mathbf{cdf}_{P_e\,(T=-40^\circ\mathrm{C};T_{ref}=25^\circ\mathrm{C})}\,(x)$ (Eq.(5)) with interpretation.

error-probability, which is 7.70%, and the *median*, which is only in the order of $10^{-8}$. The large majority of errors in a PUF response is hence caused by a small minority of cells which are wrong very often. This is exactly the kind of behavior which is oblivious in the fixed-error rate model (Sect. 2.2) and motivated us to develop a more accurate model (Sect. 2.3).

### 4.2 Implications for PUF-based Key Generators

Due to their appealing security properties like intrinisic uniqueness and physical unclonability, PUFs provide a strong physical foundation for secure key storage. To turn a PUF response into a secure key, post-processing is required by a *key generator* to boost the reliability and unpredictability to the cryptographically required level. For this purpose, a typical PUF-based key generator deploys a *fuzzy extractor* as introduced by [21], e.g. as implemented by [6,22,9,8]. For the analysis presented here, it suffices to consider a fuzzy extractor as a black box algorithm $\mathrm{FE}(n,t)$ which is able to correct up to $t$ bit errors in an $n$-bit PUF response. We refer to the cited literature for in-depth details about a fuzzy extractor's operation.

**From PUF Cell Error-Probabilities to Key Failure Rate.** A key generation fails when the fuzzy extractor is unable to correct all the PUF response bit errors that simultaneously occur in a single evaluation. The *key failure rate* ($p_{\mathrm{fail}}$) is the probability of this happening: $p_{\mathrm{fail}} = \mathbf{Pr}\,(\#\text{ errors in } n \text{ response bits} > t)$, and should be very small for practical applications (typically $10^{-6}$ or $10^{-9}$). With the fixed error-rate model (Sect. 2.2), as used in all literature on PUF-based key generators up to date, the number of errors in an $n$-bit response is binomially distributed.

This results in a fixed failure rate for every key generator instantiation:

$$\text{(fixed error-rate)} \qquad p_{\text{fail}}(p_e) = 1 - F_{\text{bino}}\left(t; n, p_e\right) . \quad (7)$$

In the more accurate new model with random error-probabilities (Sect. 2.3), the number of errors in an $n$-bit PUF response is no longer binomially distributed, but *Poisson-binomially* distributed [23].[10] The Poisson-binomial cumulative distribution function $F_{\text{PB}}\left(t; \mathbf{p_e^n}\right)$ is evaluated from the list of error-probabilities of $n$ PUF cells: $\mathbf{p_e^n} = (p_{e,1}, p_{e,2}, \ldots, p_{e,n})$. The key failure rate for $\text{FE}(n, t)$ then becomes:

$$\text{(random error-probabilities)} \qquad p_{\text{fail}}(\mathbf{p_e^n}) = 1 - F_{\text{PB}}\left(t; \mathbf{p_e^n}\right) . \quad (8)$$

Since each of the elements of $\mathbf{p_e^n}$ is a randomly sampled variable, the resulting key failure rate will not be a fixed value for every generator, as in the old model, but also a randomly sampled value for each PUF instance.

**The Key Failure Rate Distribution.** We consider a key generator based on the SRAM PUF analysed in Sect. 4.1 (with worst-case reliability at $-40°\text{C}$) and a concatenated fuzzy extractor $\text{FE}(212, 11) \circ \text{FE}(5, 2)$,[11] which extracts a key with 128-bit entropy from 1,060 cells, with $p_{\text{fail}} \leq 10^{-9}$ (on average).

Under the old fixed error-rate model of Sect. 2.2, the constant error rate is set equal to the mean error-probability over all cells: $p_e = 7.70\%$. The achieved average key failure rate is calculated by applying (7) twice:

$$p_{\text{fail}} = 1 - F_{\text{bino}}\left(11; 212, 1 - F_{\text{bino}}\left(2; 5, 0.0770\right)\right) = 1.15 \cdot 10^{-10} .$$

This key generator hence produces a 128-bit key with $p_{\text{fail}} = 1.15 \cdot 10^{-10} \leq 10^{-9}$. However, due the used fixed-error model this only holds for the *average case* key generator. No statements can be made about the distribution of failure rates, e.g. it is unclear which fraction of key generators actually reaches this average, or the required goal of $10^{-9}$. This is a serious limitation which is solved by using the new reliability model.

The random distribution of key failure rates under the new model of Sect. 2.3 is hard to treat analytically since it involves an $n$-dimensional integration over the distribution of $\mathbf{p_e^n}$. However, we are able to efficiently

[10] Some details on this lesser known distribution are given in App. A.

[11] Concatenated fuzzy extractors are typically more efficient than single large fuzzy extractors [6]. The second fuzzy extractor sees the failure rate of the output of the first one as the error probability of its input symbols. For completeness, we mention the error-correcting codes on which the considered fuzzy extractors are based: $\text{FE}(5, 2)$ uses the $(5, 1)$-repetition code and $\text{FE}(212, 11)$ the $(212, 128)$-BCH code.

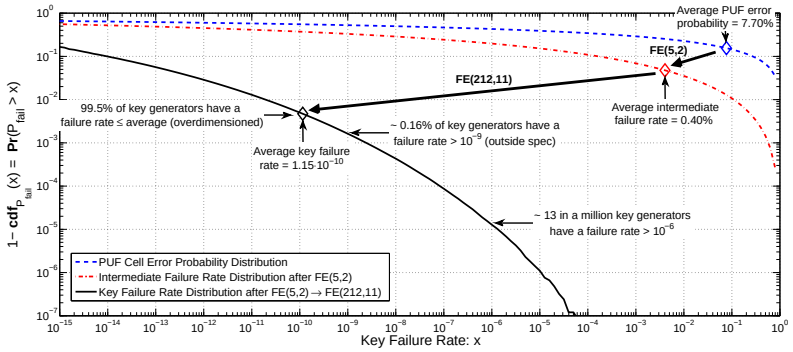


**Fig. 4.** Plot of the failure rate distribution of a PUF-based key generator.

simulate a key generator by randomly picking $n$ error-probabilities according to (5) (using inverse transform sampling) and calculating $p_{\text{fail}}(\mathbf{p_e^n})$ with (8). By repeating this, we get many random samples of $p_{\text{fail}}$ from which its distribution is estimated. We performed a simulation over 50,000,000 key generators, sampling 1,060 random error probabilities for each one, and calculating the resulting $p_{\text{fail}}$ by applying (8) twice. The resulting simulated distribution is shown in Fig. 4, together with the initial PUF cell error-probability distribution and the distribution of intermediate failure rates after FE$(5, 2)$ but before FE$(212, 11)$.

**Interpretation of the Key Failure Rate Distribution.** It is clear that the expected value of the derived key failure rate distribution under the new model is equivalent to the fixed key failure rate predicted under the old fixed error rate model. However, the failure rate distribution as plotted in Fig. 4 provides much more insight, e.g. it indicates not only the average failure rate but also the fraction of key generators actually attaining this average. For the studied example, we see that 99.5% of the generators operate above average, and even up to 99.84% have a failure rate within the specified goal of $p_{\text{fail}} \leq 10^{-9}$. On the other hand, this means that a very small but non-negligible fraction of 0.16% of the generators does not meet the specification. This is potentially important information for an application which is oblivious in the old fixed error rate model!

The small fraction of generators outside spec is not necessarily problematic. A large portion of that 0.16% still has a very small failure rate, only not as small as $10^{-9}$. Only 13 in a million generators have $p_{\text{fail}} > 10^{-6}$, and less than 1 in 10 million generators have $p_{\text{fail}} > 10^{-4}$. Whether this is a problem depends on the envisioned application, such as the num-

ber of devices in the field and the acceptability of a potential failure. In fact, by taking these considerations into account the system specifications might even be relaxed, which will result in a more efficient design. E.g., a PUF-based key generator for a public transport ticketing system, with a huge number of deployed devices but a low criticality of failure, should be approached very differently than that for a life-supporting medical implant, with a relatively small number of devices in the field but an extremely high criticality of failure. The main advantage of the new model proposed in this work is exactly that it allows to study this tradeoff, whereas in the old model one is not aware of it.

## 5 Conclusion and Future Work

We introduced a more realistic new reliability model for silicon PUFs which no longer assumes a single fixed error rate as before but considers randomly distributed cell error-probalities. An hypothetical error-probability distribution was derived based on plausible assumptions, including the effects of environmental factors like temperature. Experimental validations based on a substantial set of silicon PUF measurement data demonstrate a strikingly accurate fit of the predicted distributions on empirical statistics. This is a strong indication of the correctness and generic nature of the newly proposed model. An important implication of the use of this model is the ability to study the full failure distribution of a PUF-based application, whereas the old fixed error rate model only displays *average case* behavior. This introduces a new dimension in the design of PUF systems, allowing more focused specifications and better adapted solutions.

The ability to accurately describe the probabilistic reliability behavior of a silicon PUF spawns various seeds for future research. An obvious continuation of this work is the inclusion of more external parameters and conditions, besides temperature, in the model and the distributions; e.g. supply voltage variation, silicon device aging effects and technology node dependence. A further experimental validation on alternative silicon PUF technologies and under varying conditions will strengthen the applicability of the presented model. The offered possibility to realistically simulate PUF reliability behavior, as demonstrated in Sect. 4.2, could be of great interest in the development of PUF-based applications, e.g. when no real PUF measurements are available. Finally, an interesting parallel research track is the analysis of unpredictability (entropy) of PUF responses using the same methods as presented in this work.

## References


1. Gassend, B., Clarke, D., van Dijk, M., Devadas, S.: Silicon Physical Random Functions. In: ACM Conference on Computer and Communications Security (ACM CCS). (2002) 148–160
2. Guajardo, J., Kumar, S.S., Schrijen, G.J., Tuyls, P.: FPGA intrinsic PUFs and their use for IP protection. In: Workshop on Cryptographic Hardware and Embedded Systems (CHES). (2007) 63–80
3. NXP: PUF - Physical Unclonable Functions: Protecting next-generation Smart Card ICs with SRAM-based PUFs. `http://www.nxp.com/documents/other/75017366.pdf` (February 2013)
4. Microsemi: SmartFusion2 System-on-Chip FPGAs Product Brief. `http://www.actel.com/documents/SmartFusion2_PB.pdf` (February 2013)
5. Suh, G.E., Devadas, S.: Physical unclonable functions for device authentication and secret key generation. In: Design Automation Conference (DAC). (2007) 9–14
6. Bösch, C., Guajardo, J., Sadeghi, A.R., Shokrollahi, J., Tuyls, P.: Efficient Helper Data Key Extractor on FPGAs. In: Workshop on Cryptographic Hardware and Embedded Systems (CHES). (2008) 181–197
7. Maiti, A., Schaumont, P.: Improved Ring Oscillator PUF: An FPGA-friendly Secure Primitive. IACR Journal of Cryptology **24** (2011) 375–397
8. van der Leest, V., Preneel, B., van der Sluis, E.: Soft Decision Error Correction for Compact Memory-Based PUFs Using a Single Enrollment. In: Workshop on Cryptographic Hardware and Embedded Systems (CHES). (2012) 268–282
9. Maes, R., Herrewege, A.V., Verbauwhede, I.: PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator. In: Workshop on Cryptographic Hardware and Embedded Systems (CHES). (2012) 302–319
10. Suzuki, D., Shimizu, K.: The Glitch PUF: A New Delay-PUF Architecture Exploiting Glitch Shapes. In: Workshop on Cryptographic Hardware and Embedded Systems (CHES). (2010) 366–382
11. Bhargava, M., Cakir, C., Mai, K.: Attack resistant sense amplifier based PUFs (SA-PUF) with deterministic and controllable reliability of PUF responses. In: IEEE International Symposium on Hardware-Oriented Security and Trust (HOST). (2010) 106–111
12. Chen, Q., Csaba, G., Lugli, P., Schlichtmann, U., Ruhrmair, U.: The Bistable Ring PUF: A new architecture for strong Physical Unclonable Functions. In: IEEE International Symposium on Hardware-Oriented Security and Trust (HOST). (2011) 134–141
13. Maes, R., Tuyls, P., Verbauwhede, I.: Soft Decision Helper Data Algorithm for SRAM PUFs. In: IEEE Symposium on Information Theory (ISIT). (2009) 2101–2105
14. Simons, P., van der Sluis, E., van der Leest, V.: Buskeeper PUFs, a promising alternative to D Flip-Flop PUFs. In: IEEE International Symposium on Hardware-Oriented Security and Trust (HOST). (2012) 7–12
15. van der Leest, V., Schrijen, G.J., Handschuh, H., Tuyls, P.: Hardware intrinsic security from D flip-flops. In: ACM Workshop on Scalable Trusted Computing (ACM STC). (2010) 53–62
16. Lee, J.W., Lim, D., Gassend, B., Suh, G.E., van Dijk, M., Devadas, S.: A technique to build a secret key in integrated circuits for identification and authentication application. In: Symposium on VLSI Circuits. (2004) 176–159
17. Maes, R., Verbauwhede, I.: Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions. In Sadeghi, A.R., Naccache, D.,

eds.: Towards Hardware-Intrinsic Security. Information Security and Cryptography. Springer (2010) 3–37
18. Katzenbeisser, S., Kocabas, U., Rozic, V., Sadeghi, A.R., Verbauwhede, I., Wachsmann, C.: PUFs: Myth, Fact or Busted? A Security Evaluation of Physically Unclonable Functions (PUFs) Cast in Silicon. In: Workshop on Cryptographic Hardware and Embedded Systems (CHES). (2012) 283–301
19. (EU FP7-ICT 238811): UNIQUE Project - Foundations for Forgery-Resistant Security Hardware. `https://www.unique-project.eu/`
20. Maes, R., Rozic, V., Verbauwhede, I., Koeberl, P., van der Sluis, E., van der Leest, V.: Experimental evaluation of Physically Unclonable Functions in 65 nm CMOS. In: European Solid-State Circuits Conference (ESSCIRC). (2012) 486–489
21. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. SIAM Journal on Computing **38**(1) (March 2008) 97–139
22. Maes, R., Tuyls, P., Verbauwhede, I.: Low-Overhead Implementation of a Soft Decision Helper Data Algorithm for SRAM PUFs. In: Workshop on Cryptographic Hardware and Embedded Systems (CHES). (2009) 332–347
23. Fernandez, M., Williams, S.: Closed-Form Expression for the Poisson-Binomial Probability Density Function. IEEE Transactions on Aerospace and Electronic Systems **46**(2) (april 2010) 803–817

## A Basic Probability Distributions

*The Binomial Distribution* is the discrete distribution of the number of successes in $n$ Bernoulli trials with constant success probability $p$. Its distribution functions are given by:

$$f_{\text{bino}}(x;n,p) = \tbinom{n}{x}p^x(1-p)^{n-x}\,, \text{ and } F_{\text{bino}}(x;n,p) = \textstyle\sum_{i=0}^{\lfloor x\rfloor}\tbinom{n}{i}p^i(1-p)^{n-i}\,.$$

*The Standard Normal Distribution* is the normal distribution with zero mean and unit variance, denoted as: $\mathcal{N}(0,1)$. Any normal distribution can be expressed as a function of the standard normal: if $X \sim \mathcal{N}(\mu,\sigma^2)$, then $\frac{X-\mu}{\sigma} \sim \mathcal{N}(0,1)$. Its distribution functions are given by:

$$\varphi(x) = (2\pi)^{-\frac{1}{2}}e^{-\frac{x^2}{2}}\,, \text{ and } \Phi(x) = \frac{1}{2}\left(1+\operatorname{erf}\left(\tfrac{x}{\sqrt{2}}\right)\right)\,.$$

*The Poisson-Binomial Distribution* is the discrete distribution of the number of successes in $n$ Bernoulli trials when the success probability is no longer constant, but different for every trial. The probability mass function and cumulative distribution function of the Poisson-binomial distribution can be efficiently calculated as shown in [23]:

$$f_{\text{PB}}(x;\mathbf{p_e^n}) = \tfrac{1}{n+1}\sum_{i=0}^{n} C^{-i\cdot x}\prod_{k=1}^{n}\left(p_{e,k}C^i + (1-p_{e,k})\right), \text{ with } C = e^{\frac{j2\pi}{n+1}}\,,$$

$$F_{\text{PB}}(x;\mathbf{p_e^n}) = \tfrac{x+1}{n+1} + \tfrac{1}{n+1}\sum_{i=1}^{n}\tfrac{1-C^{-i\cdot(x+1)}}{1-C^{-i}}\prod_{k=1}^{n}\left(p_{e,k}C^i + (1-p_{e,k})\right)\,.$$

## B Derivation of New Model Distributions[12]

All derived distributions concern random variables representing probabilities. This entails that all derived distribution functions are only defined on $(0,1)$ and make no sense outside this interval. Most of the derived distributions approach infinity for $x \to 0^+$ and $x \to 1^-$, therefore, we only consider the *open* interval $(0,1)$. This implies that, e.g. an error-probability cannot be a hard 0 (absolutely never wrong) or a hard 1 (absolutely always wrong), though it can be arbitrarily close to 0 or 1.

### B.1 Fixed Temperature Model

**The One-Probability Distribution** is derived by considering the definition of its cumulative distribution function:

$$\mathbf{cdf}_P(x) \stackrel{\text{def}}{=} \mathbf{Pr}(P \leq x) = \Phi\left(\lambda_1 \Phi^{-1}(x) + \lambda_2\right),$$
$$\mathbf{pdf}_P(x) \stackrel{\text{def}}{=} \frac{d\,\mathbf{cdf}_P(x)}{dx} = \frac{\lambda_1 \varphi\left(\lambda_1 \Phi^{-1}(x) + \lambda_2\right)}{\varphi(\Phi^{-1}(x))},$$

by substituting the assumed normal distributions for $M$ and $N_i$ and using the short-hand parameters $\lambda_1 = {}^{\sigma_N}/_{\sigma_M}$, and $\lambda_2 = {}^{(t-\mu_M)}/_{\sigma_M}$.

**The Error-Probability Distribution** is derived by first considering the conditional probability density function of the error-probability with respect to the one-probability. Note that the error-probability of a cell $i$ is only completely determined at enrollment time, i.e. $p_{e,i} = p_i$ if $r_i^{\text{enroll}} = 0$ and $p_{e,i} = 1 - p_i$ if $r_i^{\text{enroll}} = 1$. The conditional distribution is derived as:

$$\mathbf{pdf}_{P_e|P=p_i}(x) = \begin{cases} p_i & \text{, for } x = 1 - p_i\,, \\ 1 - p_i & \text{, for } x = p_i\,, \\ 0 & \text{, for all other } x\,. \end{cases} = \begin{cases} 1 - x & \text{, for } p_i = 1 - x\,, \\ 1 - x & \text{, for } p_i = x\,, \\ 0 & \text{, for all other } p_i\,. \end{cases}$$

The unconditional probability functions of $P_e$ then follow as:

$$\mathbf{pdf}_{P_e}(x) = \lambda_1(1-x)\frac{\varphi\left(\lambda_1\Phi^{-1}(x)+\lambda_2\right)+\varphi\left(\lambda_1\Phi^{-1}(x)-\lambda_2\right)}{\varphi(\Phi^{-1}(x))},$$
$$\mathbf{cdf}_{P_e}(x) = \lambda_1 \cdot \int_{-\infty}^{\Phi^{-1}(x)} \Phi(-u) \cdot \left(\varphi(\lambda_1 u + \lambda_2) + \varphi(\lambda_1 u - \lambda_2)\right) \mathrm{d}u\,.$$

[12] In order to adhere to the page limit, the substeps in the following derivations are very limited. For a more detailed version of these derivations we refer to the full version of this work to appear on the Cryptology ePrint Archive (`http://eprint.iacr.org/`).

### B.2 Model with Temperature Sensitivity

**Conditional One-Probability Distribution.** The main goal of the temperature extension of the basic model is to describe the evolution of a PUF cell's behavior over changing temperature, i.e. given a reference behavior what will be its behavior when the temperature changes. We first introduce a conditional variant of the one-probability to describe this, and derive the relation of this conditional one-probability to the hidden variables following from the temperature model relation given by (4).

$$p_i(T|T_{ref}) \stackrel{\text{def}}{=} \mathbf{Pr}\left(R_i(T) = 1 | p_i(T_{ref})\right) = \Phi\left(\Phi^{-1}\left(p_i(T_{ref})\right) + \frac{d_i \cdot \Delta T}{\sigma_N}\right),$$

with $\Delta T = T - T_{ref}$ and using the normal distribution assumption for $N_i$. The distribution of the conditional one-probabilities follows from considering the definition of their cumulative distribution function:

$$\mathbf{cdf}_{P(T|T_{ref})}(x) \stackrel{\text{def}}{=} \mathbf{Pr}\left(P(T|T_{ref}) \leq x\right) = \Phi\left(\theta \cdot \frac{\Delta\Phi^{-1}(x)}{|\Delta T|}\right),$$

$$\mathbf{pdf}_{P(T|T_{ref})}(x) = \frac{d\,\mathbf{cdf}_{P(T|T_{ref})}(x)}{dx} = \frac{\theta}{|\Delta T|} \cdot \frac{\varphi\left(\theta \cdot \frac{\Delta\Phi^{-1}(x)}{|\Delta T|}\right)}{\varphi(\Phi^{-1}(x))}.$$

with $\Delta\Phi^{-1}(x) = \Phi^{-1}(x) - \Phi^{-1}\left(p_i(T_{ref})\right)$ and after filling in the normal distribution assumption for $D$ and using the short-hand notation $\theta = \frac{\sigma_N}{\sigma_D}$.

**Error-Probability Distribution.** We first express the conditional distribution of the error-probability conditioned on a known value for the one-probability $p_i(T_{ref})$, and a known value for the conditional one-probability $p_i(T|T_{ref})$:

$$\mathbf{Pr}\left(P_e(T;T_{ref}) = x | P(T|T_{ref}) = y, P(T_{ref}) = p_{i,ref}\right) = \begin{cases} p_{i,ref} & \text{, for } x = 1-y\,, \\ 1 - p_{i,ref} & \text{, for } x = y\,, \\ 0 & \text{, for all other } x\,. \end{cases}$$

We begin with removing the conditioning on $p_i(T|T_{ref})$:

$$\mathbf{Pr}\left(P_e(T;T_{ref}) = x | P(T_{ref}) = p_{i,ref}\right) = (1 - p_{i,ref}) \cdot \mathbf{pdf}_{P(T|T_{ref})}(x) + p_{i,ref} \cdot \mathbf{pdf}_{P(T|T_{ref})}(1-x)\,.$$

The unconditional distribution of $P_e(T;T_{ref})$ then follows as:

$$\begin{aligned} \mathbf{pdf}_{P_e(T;T_{ref})}(x) &= \int_0^1 \Big((1 - p_{i,ref}) \cdot \mathbf{pdf}_{P(T|T_{ref})}(x) + p_{i,ref} \cdot \mathbf{pdf}_{P(T|T_{ref})}(1-x)\Big)\mathbf{pdf}_P\left(p_{i,ref}\right) dp_{i,ref}\,, \\ &= \frac{\lambda_1\theta}{|\Delta T|\varphi(\Phi^{-1}(x))} \cdot \int_{-\infty}^{+\infty} \left[\Phi(-u)\,\varphi\left(\theta\frac{\Phi^{-1}(x)-u}{|\Delta T|}\right) + \Phi(u)\,\varphi\left(\theta\frac{\Phi^{-1}(x)+u}{|\Delta T|}\right)\right] \cdot \varphi(\lambda_1 u + \lambda_2)\,\mathrm{d}u\,. \\ \mathbf{cdf}_{P_e(T;T_{ref})}(x) &= \frac{\lambda_1\theta}{|\Delta T|} \cdot \int_{-\infty}^{\Phi^{-1}(x)} \int_{-\infty}^{+\infty} \left[\Phi(-u)\,\varphi\left(\theta\frac{v-u}{|\Delta T|}\right) + \Phi(u)\,\varphi\left(\theta\frac{v+u}{|\Delta T|}\right)\right] \cdot \varphi(\lambda_1 u + \lambda_2)\,\mathrm{d}u\,\mathrm{d}v\,. \end{aligned}$$

For $\Delta T \to 0^+$ this reverts to the distribution functions for the basic fixed temperature model as derived in App. B.1.