SCRIPT: Bayesian Reliability Filtering and Entropy Accounting for Physically Unclonable Functions

Secure Chip using Random Intrinsic Physics in Transistors

Ronny Abdullah

Abstract

Physically Unclonable Functions (PUFs) turn uncontrollable semiconductor manufacturing variation into hardware fingerprints, but environmental noise flips a fraction of PUF bits across power cycles and makes them unsafe to use directly as cryptographic keys. The standard remedy of discarding unreliable bits is well established, so this work asks the two questions that actually determine whether such filtering is trustworthy. First, can we filter on *credible* reliability rather than a noisy point estimate, so that we do not retain a bit we are merely lucky to have seen behave? Second, what does filtering cost in *security* once inter-chip correlation is taken into account? Using a Beta–Binomial posterior per cell, a Monte-Carlo bit-error evaluation, and a min-entropy accounting across a population of correlated simulated devices, I show that Bayesian credible filtering drives the key bit error rate from 14.08 % to 1.30 %, roughly 2.8× lower than a maximum-likelihood point filter at the same threshold, while retaining a smaller but confidently stable set of cells. I then show that the headline security number is governed not by reliability but by fabrication correlation: independent devices realize the full min-entropy of the retained key (≈ 169 bits over 169 cells), whereas a modest process correlation of $\rho = 0.6$ collapses effective min-entropy to ≈ 91 bits. The practical takeaway is that reliability filtering and entropy validation are separate obligations, and that reporting only a post-filter error rate can badly overstate a PUF key's security.

1 Introduction

Modern hardware, including IoT endpoints, secure elements, and medical implants, must be able to prove its identity without storing a secret key in nonvolatile memory, which is expensive and vulnerable to physical extraction. Physically Unclonable Functions address this by deriving a key from uncontrollable manufacturing randomness: even two nominally identical chips are physically distinct at the transistor level, so each produces a different, device-intrinsic response [1, 2].

The difficulty is that a PUF is a *noisy* secret. Threshold voltages, temperature, and supply variation cause a subset of cells to evaluate inconsistently across power-ups. If every cell is used directly, the regenerated key differs from the enrollment key on each boot, and authentication fails or requires heavy error-correction machinery. The long-standing engineering response is to identify and discard unstable cells before key derivation [3, 4]. That method is not novel, and this report does not claim it to be.

What this report contributes is a more careful treatment of the two questions that decide whether such filtering can be trusted in practice:

1. **Confidence-aware filtering.** A point estimate of a cell's bias from a handful of power-ups is itself uncertain. Filtering on that point estimate retains cells that merely *appeared* stable

in a short observation window. I instead maintain a full Beta–Binomial posterior per cell and retain a cell only when its entire 95 % credible interval certifies stability.

2. **Entropy cost of filtering.** Discarding cells and biasing toward "confident" ones reduces entropy, and inter-chip correlation from shared fabrication drift reduces it further. A post-filter bit error rate says nothing about this. I quantify the retained key's effective min-entropy across a population of correlated devices, which is the quantity that actually bounds key security.

2 Cell Model

Each PUF cell i is modeled as a Bernoulli source with an unknown intrinsic bias p_i, the probability that the cell evaluates to 1 on a given power-up:

$$X_{i,t} \sim \text{Bernoulli}(p_i), \qquad t = 1, \ldots, T.$$

Over T independent power-ups the number of ones is Binomial,

$$k_i = \sum_{t=1}^{T} X_{i,t} \sim \text{Binomial}(T, p_i).$$

A cell's *golden* (enrollment) key bit is its majority direction,

$$K_i^\star = \mathbb{1}[\, p_i > 0.5 \,].$$

Cells with p_i near 0 or 1 are stable and safe to use; cells with p_i near 0.5 behave like fair coins and are the source of key instability.

To reflect real SRAM- and arbiter-style PUF statistics, intrinsic biases are drawn from a U-shaped ("bathtub") population, $p_i \sim \text{Beta}(0.35, 0.35)$, which concentrates most cells near the rails with a minority tail of unstable cells near 0.5. All results below use $N = 512$ cells and $T = 25$ enrollment power-ups.

3 Reliability Estimation and Filtering

3.1 Maximum-likelihood point filter (baseline)

The maximum-likelihood estimate of the bias is the observed frequency, $\hat{p}_i = k_i/T$, with reliability score

$$\hat{r}_i = \min(\hat{p}_i,\ 1 - \hat{p}_i).$$

A cell is kept if $\hat{r}_i < \tau$ for threshold $\tau = 0.10$. This is the conventional approach and serves as the baseline.

3.2 Bayesian credible filter

The weakness of the point filter is that $\hat{r}_i$ is itself a random variable: at $T = 25$, a genuinely marginal cell can easily produce an extreme k_i and be wrongly retained. I therefore place a Jeffreys prior $\text{Beta}(\frac{1}{2}, \frac{1}{2})$ on each p_i and form the posterior

$$p_i \mid k_i \ \sim\ \text{Beta}\big(\tfrac{1}{2} + k_i,\ \tfrac{1}{2} + T - k_i\big)\,.$$

Let $[\ell_i, u_i]$ be the central 95 % credible interval. A cell is retained only if that interval *entirely* certifies stability:

$$i \in \mathcal{R} \iff \big(u_i < \tau\big) \ \vee \ \big(\ell_i > 1 - \tau\big).$$

In words: we keep a cell only when we are 95 % confident its bias lies within τ of a rail, not merely when its point estimate happens to. The Bayesian key direction uses the posterior mean, $\hat{K}_i = \mathbb{1}[\,\mathbb{E}[p_i \mid k_i] > 0.5\,]$.

This is strictly more conservative than the point filter and retains fewer cells, but every retained cell carries a certified stability guarantee rather than a lucky observation.

4 Predictive Bit Error Rate

Key reliability is evaluated predictively. For each retained cell I draw fresh future power-ups $X_{i,\text{future}} \sim \text{Bernoulli}(p_i)$ from the *true* bias, never reusing the enrollment observations, and compare against the key direction. The bit error rate over a retained set $\mathcal{S}$ with key K is estimated by Monte Carlo over many future boots:

$$\text{BER}(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \Pr[X_{i,\text{future}} \neq K_i]\,.$$

Evaluating against independently drawn future observations avoids the circular trap of scoring the filter on the same samples that produced it.

5 Entropy Accounting Across Correlated Devices

Reliability is necessary but not sufficient: a perfectly stable key can still be weak if devices resemble one another. Real fabrication induces spatial and lot-level correlation, so sibling chips share systematic drift and their keys are not independent. I model a population of $D = 50$ devices in which each cell's direction is drawn from a shared component with probability ρ and an idiosyncratic component otherwise; $\rho = 0$ yields independent devices and larger ρ yields correlated ones. Each device is independently enrolled and filtered by the Bayesian procedure above.

Inter-chip uniqueness is measured by the pairwise fractional Hamming distance between device keys, restricted to cells reliable in both devices. The ideal value is 0.5 (keys as different as fair coins). I convert the mean Hamming distance μ into an *effective* per-cell min-entropy via the implied cross-device agreement probability $q = \max(\mu, 1 - \mu)$,

$$H_\infty^{\text{cell}} = -\log_2 q, \qquad H_\infty^{\text{key}} = |\mathcal{R}| \cdot H_\infty^{\text{cell}},$$

so that $\mu = 0.5$ recovers one bit per cell and any correlation-driven departure from 0.5 reduces it.

6 Results

	All cells	**MLE point filter**	**Bayesian credible**
Cells retained	512	278	169
Predictive BER	14.08 %	3.60 %	1.30 %

Table 1: Reliability of the regenerated key. The Bayesian credible filter achieves roughly 2.8× lower bit error rate than the point-estimate filter at the same threshold, at the cost of retaining fewer, certified-stable cells.

	Mean Hamming	**Per-cell H_∞**	**Key H_∞ (169 cells)**
Independent ($\rho = 0$)	0.501	0.998	169 bits
Correlated ($\rho = 0.6$)	0.313	0.541	91 bits

Table 2: Effective min-entropy of the filtered key. Reliability is unchanged between these two populations, since both are filtered identically, yet a modest process correlation nearly halves the key's effective min-entropy. The security number is governed by correlation, not by the bit error rate.

Table 1 isolates the value of confidence-aware filtering: moving from a point estimate to a credible-interval criterion at the same threshold τ more than halves the residual error again, because the cells it declines to certify are exactly the marginal cells that a short observation window can misjudge. Table 2 is the more important result. The two device populations are filtered by the identical procedure and would report identical reliability, yet their keys differ sharply in security: independent devices realize essentially the full 169 bits, while $\rho = 0.6$ correlation collapses this to 91 bits. A reliability-only evaluation would have reported both as equally strong.

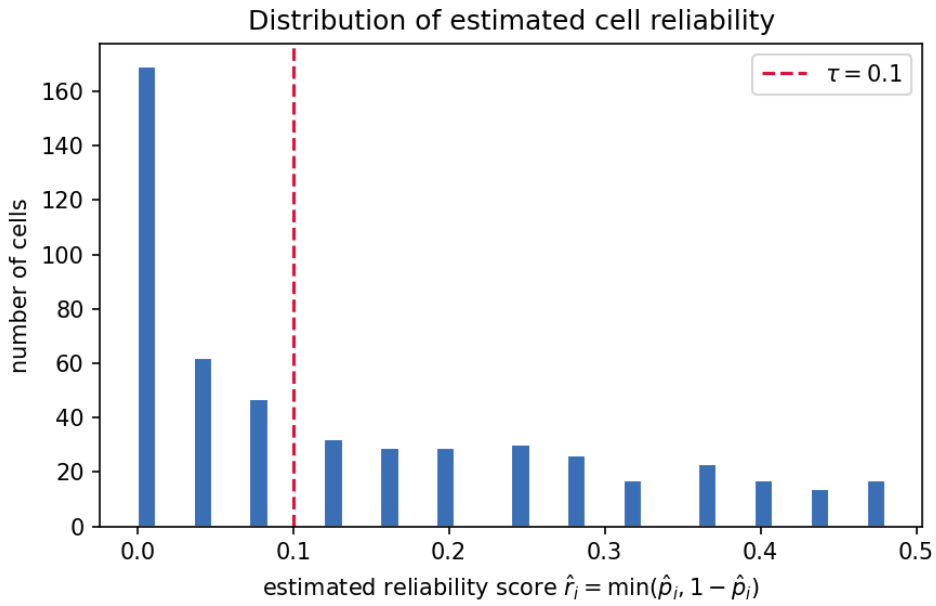

Figure 1: Distribution of estimated per-cell reliability scores $\hat{r}_i = \min(\hat{p}_i, 1-\hat{p}_i)$. Stable cells cluster near zero; unstable cells accumulate toward 0.5. The dashed line marks the threshold $\tau = 0.10$.

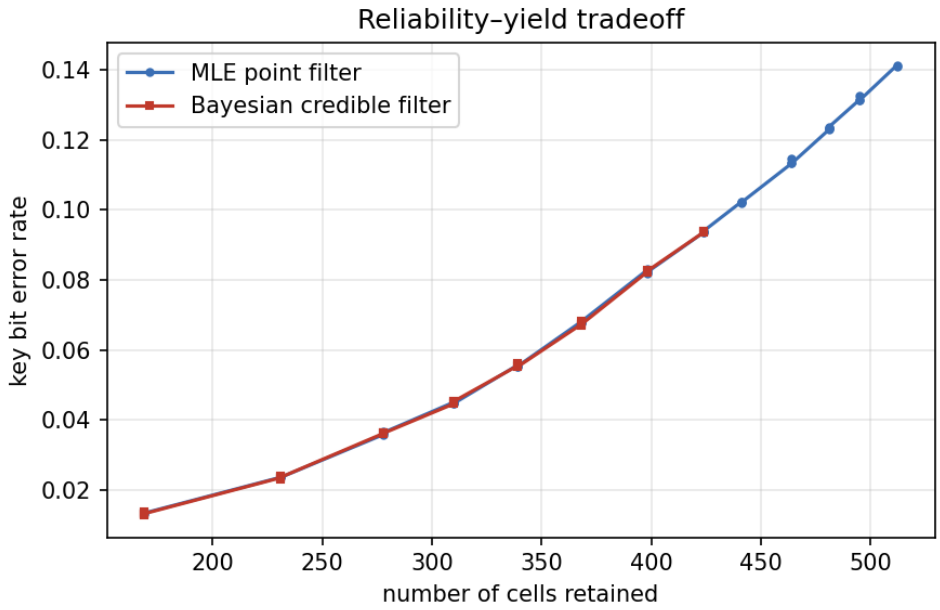

Figure 2: Reliability–yield tradeoff as the threshold is swept. For any target number of retained cells, the Bayesian credible filter attains a lower bit error rate than the maximum-likelihood point filter, because it declines to certify cells whose apparent stability is not statistically supported at $T = 25$.

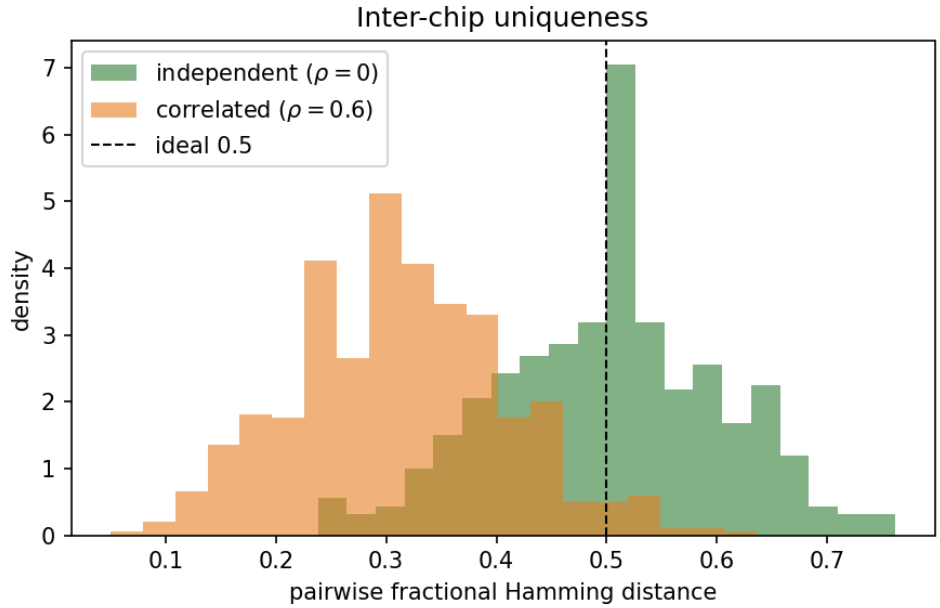

Figure 3: Inter-chip uniqueness across 50 simulated devices. Independent devices center on the ideal fractional Hamming distance of 0.5; a process correlation of $\rho = 0.6$ shifts the distribution left to a mean of 0.313, the direct cause of the min-entropy loss in Table 2.

7 Discussion and Limitations

The central message is that PUF key validation has two separable obligations that are easy to conflate. Reliability filtering, even the improved Bayesian form here, only guarantees that a device reproduces *its own* key. It says nothing about whether that key is *unpredictable across devices*, which is what an attacker exploits. Correlation from shared fabrication is invisible to any single-device reliability metric and yet, as Table 2 shows, can dominate the true security margin.

Several limitations bound these claims. The device population is simulated rather than measured on silicon, and the correlation model is a deliberately simple shared-direction mixture rather than a physically calibrated spatial process; real correlation structure is richer and would need silicon data to characterize. The effective-min-entropy estimate is derived from pairwise Hamming statistics and is an approximation to a full min-entropy bound, not a substitute for the conditioning analysis a production key-derivation function would require. Finally, the Beta–Binomial model treats power-ups as i.i.d. given p_i, whereas real noise is temperature- and voltage-dependent and partially correlated across boots; incorporating those covariates is the natural next step.

8 Conclusion

Reframing a standard PUF reliability filter around two probabilistic questions yields a small but concrete result. Filtering on Beta–Binomial credible intervals rather than maximum-likelihood point estimates lowers the predictive key bit error rate from 14.08 % to 1.30 %, about 2.8× better than the point filter, by certifying stability rather than trusting a short observation. More importantly, a min-entropy accounting across correlated devices shows that this reliability gain is orthogonal to security: identical filtering yields 169 bits of effective min-entropy on independent devices but only 91 bits under modest fabrication correlation. Reliability and entropy are distinct obligations, and a PUF evaluation that reports only a post-filter error rate can substantially overstate the strength of the resulting key.

References

[1] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," *ACM CCS*, 2002.

[2] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, 2002.

[3] R. Maes, P. Tuyls, and I. Verbauwhede, "A soft decision helper data algorithm for SRAM PUFs," *IEEE ISIT*, 2009.

[4] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," *CHES*, 2007.