Robust Keys from Physical Unclonable Functions

Merrielle Spain, Benjamin Fuller, Kyle Ingols, and Robert Cunningham
MIT Lincoln Laboratory
{merrielle.spain, bfuller, kwi, rkc}@ll.mit.edu

***Abstract*—Weak physical unclonable functions (PUFs) can instantiate read-proof hardware tokens (Tuyls et al. 2006, CHES) where benign variation, such as changing temperature, yields a consistent key, but invasive attempts to learn the key destroy it. Previous approaches evaluate security by measuring how much an invasive attack changes the derived key (Pappu et al. 2002, Science). If some attack insufficiently changes the derived key, an expert must redesign the hardware.**

An unexplored alternative uses software to enhance token response to known physical attacks. Our approach draws on machine learning. We propose a variant of linear discriminant analysis (LDA), called PUF LDA, which reduces noise levels in PUF instances while enhancing changes from known attacks.

We compare PUF LDA with standard techniques using an optical coating PUF and the following feature types: raw pixels, fast Fourier transform, short-time Fourier transform, and wavelets. We measure the true positive rate for valid detection at a 0% false positive rate (no mistakes on samples taken after an attack). PUF LDA improves the true positive rate from 50% on average (with a large variance across PUFs) to near 100%.

While a well-designed physical process is irreplaceable, PUF LDA enables system designers to improve the PUF reliability-security tradeoff by incorporating attacks without redesigning the hardware token.

I. Introduction

A physical unclonable function (PUF) is a physical structure that harnesses manufacturing randomness to generate unpredictable outputs [1]. Various hardware components can support PUFs: ring oscillators [2], [3], [4], cross-coupled latches or flip-flops [5], capacitive particles in a coating material [6], and beads in an optical card [1]. PUFs were originally designed to identify hardware using challenge-response pairs [1].

Tuyls et al. created a PUF by applying a coating doped with dielectric particles to a custom integrated circuit with a top layer of capacitance sensors [6]. Instead of providing many challenge-response pairs, Tuyls et al. derived a single cryptographic key from the capacitance pattern. Attempts to breach the coating altered the particle configuration, destroying the key. This system provides read-proof key storage, and is known as a *weak* PUF [7], [8], or physically obfuscated key [9].

A strong PUF has many input/output pairs; an adversary may sample some outputs, but security derives from their inability to sample exhaustively. A weak PUF has a single output (or few outputs); security requires the output to remain secret from the adversary. We focus on improving weak PUFs. The informal properties of a weak PUF are (adapted from [10]):

- *Robustness:* the outputs are stable
- *Unpredictability:* the outputs of different devices are independent
- *Unsamplability:* the adversary cannot model, recreate, or read the PUF (this augments unclonability with a response to attacks)

This work is sponsored by Assistant Secretary of Defense for Research & Engineering under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

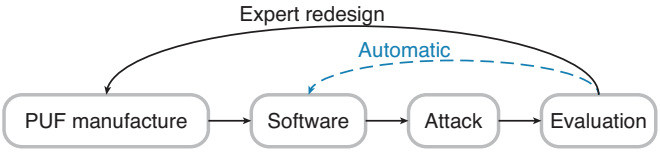

Fig. 1. Previously, improving the reliability-security tradeoff of a PUF required expert redesign of the physical token. Our approach automatically integrates attack response information into software (dashed path). This approach complements token redesign.

If a known attack can read the output of a weak PUF, then the PUF lacks unsamplability and requires redesign (Figure 1). Traditionally, domain experts redesign the hardware in a costly and lengthy process. Instead, we incorporate known attacks at the software level. Redesigning the software offers several advantages—affordability, automation, and deployability—but the physical token must provide unpredictability and unsamplability. Our approach complements thoughtful hardware design.

Our software leverages machine learning to improve robustness and unsamplability (against known attacks) while minimally affecting unpredictability. However, standard techniques fail to achieve these goals simultaneously. For instance, a classification decision fails to provide enough entropy for a strong key. Our approach instead draws on dimensionality reduction, which can keep sufficient entropy for key derivation.

Standard dimensionality reduction techniques, such as linear discriminant analysis (LDA) [11], solve the wrong problem (Section II-A); their objective functions include irrelevant variation from damaged samples (samples taken after an attack), and assume that the means of damaged samples represent them well. Section II-B develops a new dimensionality reduction technique, PUF LDA, that departs from the notion of classes to treat damaged samples differently than multiple valid (unattacked) PUFs. We evaluate PUF LDA with optical coating PUFs, using image-relevant feature types. Section II-C describes our PUF fabrication, Section II-D details our data collection, Section III evaluates our approach, and Section IV concludes.

978-1-4799-4112-4/14/$31.00

II. Methods

This section describes techniques to incorporate attacks on PUFs into design. Our technique maps samples to a space suitable for key derivation. We aim to pull valid samples close together, and to push damaged samples far away.

We apply machine learning, but the adversarial setting excludes standard techniques. Retaining enough entropy to derive a strong key excludes applying classifiers; instead, we apply dimensionality reduction.

A. Dimensionality Reduction

We aim to derive a consistent key under benign variation, and a random key after an attack. A fuzzy extractor generates the correct key for samples within a certain distance from the template (initial reading) and different keys for everything else [12]. A fuzzy extractor reduces the task to mapping benign variation to close values and "adversarial" variation far away. Instances from different PUFs must produce a high entropy distribution in the resultant space. The original space has copious entropy, but a poor reliability-security tradeoff; a small noise tolerance rejects valid readings, while a large noise tolerance accepts damaged samples. We aim to create an algorithm that improves the reliability-security tradeoff by simultaneously increasing the distance between valid and damaged samples (unsamplability), reducing the distance between repeated readings of the same PUF (robustness), and retaining important features (unpredictability).

Manually designing an algorithm to distinguish attacks from normal variation requires modeling how attacks alter output. Machine learning infers structure directly from data, avoiding impractical PUF modeling. Supervised machine learning generates a model from a human annotated training set, and evaluates the model with a separate test set.

Principal Components Analysis (PCA) is a common dimensionality reduction approach for unsupervised data [13], [14]. It projects onto the subspace that maximizes variance. However, PCA removes entropy between PUFs as readily as it removes noise (Section II-B).

With class labels on a training set, one can apply Linear Discriminant Analysis (LDA) [11], [14]. LDA measures the directional spread, the variance between classes over the variance within classes. LDA projects data onto a few, greatest-spread, linearly independent vectors, maximizing class separation. Specifically, this minimizes within class variance and maximizes between class variance.

The within class variance, S_W, for N samples denoted x_i, with class mean μ_k for class k of K classes, is defined

$$S_W = \frac{1}{N-K} \sum_{k=1}^{K} \sum_{i \in C_k} (x_i - \mu_k)(x_i - \mu_k)^\mathsf{T} \quad (1)$$

The between class variance, S_B, with grand mean μ, is defined

$$S_B = \frac{1}{K} \sum_{k=1}^{K} (\mu_k - \mu)(\mu_k - \mu)^\mathsf{T} \quad (2)$$

LDA composes the projection matrix from the eigenvectors w, with the largest eigenvalues in $S_B w = \lambda S_W w$.

Unfortunately, an ideal way to directly apply LDA to our setting eludes us. One approach creates a separate class for each PUF and another for its damaged counterpart. Another treats all PUFs as a single class and all damaged PUFs as another class. Both approaches have significant shortcomings.

Multiclass LDA treats K valid PUFs as K classes and their damaged counterparts as another K, resulting in twice as many classes as PUFs. This separates valid PUFs from their damaged counterparts and each other. However, this assumes arbitrary damage appears similar and is well represented by its mean—an unfounded assumption. Also, the number of classes upper bounds the rank of S_B, so a high entropy output requires many training PUFs.

Two-class LDA places all damaged PUF samples in one class and valid PUF samples in another. Two-class LDA suffers from the problems of multiclass LDA and crowds different PUFs instances together, destroying unpredictability.

B. PUF LDA

Example data illustrates problems with standard techniques (Figure 2). Figure 2a shows a subspace where the damaged samples settle far from, but to both sides of, the valid samples, resulting in similar means. LDA undervalues the useful dimension because the means overlap. Projecting onto the LDA-selected subspace mixes the damaged and valid samples (Figure 2b). PCA finds the dimensions of greatest variance. However, these dimensions may not separate valid and damaged samples. Figure 2c shows an example where PCA fails and LDA succeeds.

To avoid these problems, we treat damaged samples as something other than a class. Our approach repels damaged samples from their corresponding valid mean, instead of repelling class means from each other. It builds the within class matrix from the valid samples alone. Both changes remove the unrepresentative damaged mean. The first change increases the rank of the between class matrix, while the second change stops forcing the damaged samples together.

PUF LDA divides our dataset of K PUFs into $2 \times K$ subsets. The valid samples of PUF k belong to set V_k, whereas damaged samples belong to set D_k. The valid samples of all K PUFs form $\cup_k V_k$, whereas the damaged samples form $\cup_k D_k$. We ignore damaged samples to define within class variance as

$$S_W = \frac{1}{|\cup_k V_k| - K} \sum_{k=1}^{K} \sum_{i \in V_k} (x_i - \mu_{V_k})(x_i - \mu_{V_k})^\mathsf{T} \quad (3)$$

Satisfying unsamplability requires separating damaged and valid instances of the same PUF, and spreading different valid PUFs apart. Thus, we measure the distance from damaged samples, instead of the damaged mean, to the valid mean μ_{V_k},

$$S_{B_D} = \frac{1}{|\cup_k D_k|} \sum_{k=1}^{K} \sum_{i \in D_k} (x_i - \mu_{V_k})(x_i - \mu_{V_k})^\mathsf{T} \quad (4)$$

Recall we want PUFs to change after attack, and different PUFs to appear different. Equation 4 addresses the first goal. We express the second goal by excluding damaged classes

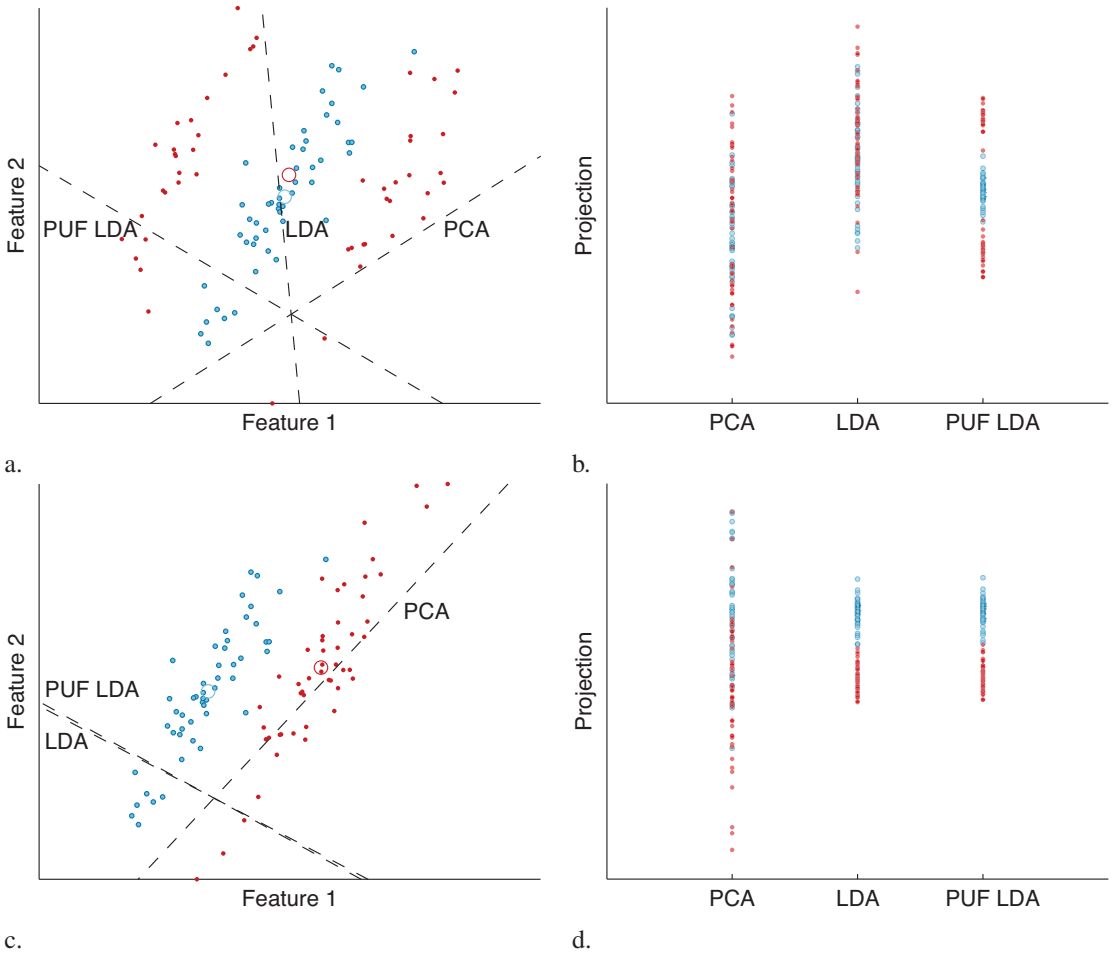

Fig. 2. Dimensionality reduction can incorporate attack response information to find a subspace with a better reliability-security tradeoff. Left: Example data plotted in two dimensions. Outlined blue dots represent valid samples, red dots represent damaged samples, circles represent class means, and dashed lines represent selected subspaces. Right: Data projected onto subspaces for PCA, LDA, and PUF LDA. A successful approach sorts dots by color. Top: Example data where LDA fails to separate damaged samples from valid ones, but PUF LDA succeeds. Bottom: Example data where LDA and PUF LDA both succeed.

from Equation 2, with valid grand mean μ_V yielding

$$S_{B_V} = \frac{1}{K}\sum_{k=1}^{K}(\mu_{V_k} - \mu_V)(\mu_{V_k} - \mu_V)^\mathsf{T} \tag{5}$$

A deployed system would balance these two goals. However, in these experiments we emphasize S_{B_D} alone, because we fabricated only two PUFs identically so there was natural variation between PUFs.

Returning to our example, PUF LDA selects the useful subspace in Figure 2a, with the projection separating damaged and valid samples in Figure 2b. A fuzzy extractor needs the damaged samples away from the valid samples, whereas LDA aims to push the damaged samples to one side of the valid samples. Figure 2c, the case suited to LDA, shows that PUF LDA chooses a similar subspace, separating damaged and valid samples in Figure 2d.

The main challenge in adapting dimensionality reduction to PUFs is that samples from different attacks do not form a class represented by a mean. We designed PUF LDA to address this problem. PUF LDA has two objectives S_{B_V}: separate the means of different PUFs from each other, and S_{B_D}: separate damaged samples from their valid mean.

C. PUF Fabrication

We fabricated a weak PUF to assess PUF LDA. We manufactured an optical coating PUF with polymer waveguides of two refractive indices. Manufacturing variances roughen the polymer interfaces, creating unique light patterns at an image sensor. These manufacturing variances differentiate boards and provide unpredictability. Attacking the waveguide should alter the light pattern at the image sensor.

The coating must protect light emitters, an image sensor, and a processor to prevent an adversary from interfering with PUF evaluation and key derivation. We aim to improve robustness without reducing unpredictability and unsamplability.

For testing, we manufactured five boards. Each board contains three LEDs and a single image sensor. We treat each LED of a board as a separate PUF instance, yielding 13 valid PUFs. While too few samples to ensure these devices act as PUFs, it enables evaluation of PUF LDA.

D. Data Collection

We captured images at different temperatures to model environmental variation. The temperature cycled between $-10°$C

and 50°C over three hours. We captured 10,000 images for each of the three LEDs. We attacked seven of the PUFs by pressing a tungsten probe into the coating and retracting it immediately. We collected 10,000 images under temperature cycling after each attack.

To measure generalization error, we divided each collection of 10,000 images into three parts: 50% for a training set for PUF LDA, 20% for a test set for machine learning, and 30% for a test set for the full system including key derivation. The median of each PUF's valid training images forms its template.

Inputting relevant features to a dimensionality reduction algorithm harnesses domain knowledge. A helpful basis can reduce dimensionality with little signal loss—maximizing robustness without harming unpredictability or unsamplability.

The fast Fourier Transform (FFT) converts a real-valued image into a complex-valued image in the frequency domain. The short-time Fourier Transform (STFT) introduces location information by breaking the image into rectangular pieces and performing many smaller FFTs. Wavelet transforms recursively decompose an image to extract both location and scale information from an image [15].

Prior to feature extraction, we downsampled the image to 81,920 pixels. The FFT magnitude and phase have 40,960 dimensions each, STFT combines phase and magnitude information to have 81,920 dimensions, and the wavelet transform yields 84,660 dimensions.

We calculate PCA and PUF LDA projections only once, with the training set. We restrict the projection input to 2,500 features and set the projection output to 60 dimensions. We randomly select 2,500 features for PUF LDA input to constrain the problem and prevent overfitting. The projection output has 60 dimensions, to leave sufficient entropy for key derivation.

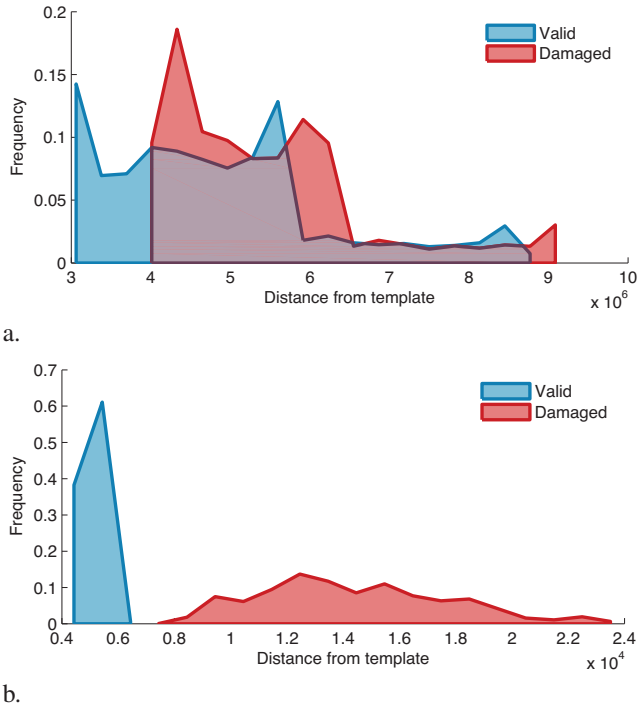

Fig. 3. Histograms of distance to template (initial reading) for valid (blue) and damaged (red) samples. We measure performance as the fraction of valid samples closer to the template than the closest damaged sample (blue left of red). We observe performances for FFT magnitude of (a) 0.3 before and (b) 1.0 after PUF LDA.

III. Results and Discussion

For each PUF, measuring the distance from the template to the damaged and valid images yields two distributions. Daugman's decidability index, a normalized difference of means, and the related sensitivity index d' are inappropriate for the irregularly shaped distributions observed [16]. Techniques that yield a larger gap between damaged and valid distributions perform better. However, distribution pairs with the same overlap but different skewness perform differently. To capture this, we measure the fraction of valid images closer to the template than any damaged image, or equivalently the valid detection true positive rate for a zero false positive rate. Figure 3 shows distributions before and after PUF LDA.

We analyze the performance of feature types under no dimensionality reduction, PCA, and PUF LDA. Figure 4 shows the performance for separating damaged and valid samples under temperature variation, indicating standard deviation with error bars. PUF LDA improves upon raw feature values and PCA. Under PUF LDA, pixels, FFT magnitude, and wavelets all perform well; FFT phase and STFT perform poorly. Efficacy depends on attack location, LED, and feature type.

Figure 5 measures the fraction of images closer to their template than the closest image from another PUF, indicating standard deviation with error bars. PUF LDA keeps separation

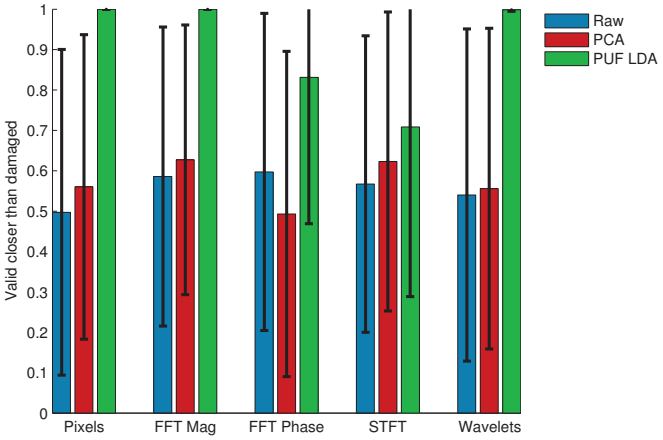

Fig. 4. Detecting attacks with an optical coating weak PUF. Performance under temperature variation for raw (blue), PCA on (red), and PUF LDA on (green) image relevant features: pixel values, FFT magnitude and phase, short-time Fourier Transform, and wavelets. See Figure 3 for an explanation of the performance metric.

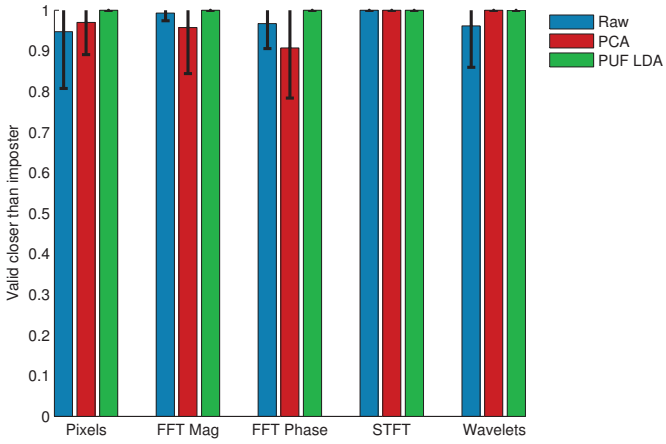

Fig. 5. Dimensionality reduction to improve attack detection maintains unpredictability. We measure the fraction of valid samples closer to their template than the nearest sample from a different PUF.

between valid PUFs, leaving unpredictability and unsamplability intact.

High entropy samples present challenges to supervised machine learning, such as PUF LDA. The sparsity of high dimensional data implies noise will correlate with class membership. Since supervised learning selects features based on data, it confuses spurious correlation with signal. This results in poor generalization performance, as an independent test set will lack identical noise. A rule of thumb for (non-regularized) supervised learning suggests three times as many samples as features. As described in Section II-D, we artificially restrict the number of features input to PUF LDA. Alternatively, we could adapt regularized discriminant analysis, which restricts model complexity [17].

In these experiments, the machine learning trained on all damaged PUFs. A thorough validation would involve a test set of different PUFs, instead of just different images. We need more PUFs to fully evaluate attack detection. Similarly, quantifying unpredictability and device uniqueness requires many identically manufactured PUFs.

IV. Conclusion

PUF LDA incorporates PUF attack-response at the software level. By automatically incorporating a physical structure's response to attack, PUF LDA improves the reliability-security tradeoff without redesigning the hardware token. This approach samples, attacks, and then resamples a set of training PUFs. It learns from the data to improve new PUFs, which one would deploy undamaged. Our technique improves robustness and unsamplability against known attacks, while minimally reducing unpredictability. PUF LDA complements an expert redesigning a token. When restricted to a specific physical token, this can cheaply and quickly improve PUF performance. For instance, a software update could incorporate changes for deployed parts.

PUF LDA is applicable to any weak PUF. We restricted our discussion to the single output case, but our approach generalizes to a small number of outputs. We envision these techniques as a standard component of the PUF design process.

Acknowledgements

The authors thank Michael Ericson, Michael Geis, Theodore Lyszczarz, Joshua Kramer, and Michael Vai for their valuable input to the design, implementation, and evaluation of the associated PUF.

References

[1] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.

[2] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Computer and Communication Security Conference*, 2002.

[3] C.-E. Yin and G. Qu, "LISA: Maximizing RO PUF's secret extraction," in *Hardware-Oriented Security and Trust*, June 2010, pp. 100–105.

[4] A. Maiti, J. Casarona, L. McHale, and P. Schaumont, "A large scale characterization of RO-PUF," in *Hardware-Oriented Security and Trust*, June 2010, pp. 94–99.

[5] S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "Extended abstract: The butterfly PUF protecting IP on every FPGA," in *Hardware-Oriented Security and Trust*, 2008, pp. 67–70.

[6] P. Tuyls, G.-J. Schrijen, B. Skoric, J. van Geloven, N. Verhaegh, and R. Wolters, "Read-proof hardware from protective coatings," in *CHES*, 2006, pp. 369–383.

[7] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *Cryptographic Hardware and Embedded Systems-CHES 2007*. Springer, 2007, pp. 63–80.

[8] U. Rührmair, J. Sölter, and F. Sehnke, "On the foundations of physical unclonable functions." *IACR Cryptology ePrint Archive*, vol. 2009, p. 277, 2009.

[9] B. L. Gassend, "Physical random functions," Master's thesis, Massachusetts Institute of Technology, 2003.

[10] F. Armknecht, R. Maes, A.-R. Sadeghi, F.-X. Standaert, and C. Wachsmann, "A formal foundation for the security features of physical functions," in *IEEE Symposium on Security and Privacy (SSP)*. IEEE Computer Society, 2011, pp. 397–412.

[11] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Eugen*, vol. 7, pp. 179–188, 1936.

[12] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.

[13] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.

[14] C. R. Rao, *Linear Statistical Inference and Its Applications*. New York: Wiley, 1973.

[15] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 961–1005, 1990.

[16] J. Daugman, "Probing the uniqueness and randomness of IrisCodes: Results from 200 billion iris pair comparisons," *Proceedings of the IEEE*, vol. 94, no. 11, pp. 1927–1935, 2006.

[17] J. Friedman, "Regularized discriminant analysis," *Journal of the American Statistical Association*, vol. 84, pp. 165–175, 1989.