SD-PUF: Spliced Digital Physical Unclonable Function

Jin Miao, Meng Li, Subhendu Roy, *Member, IEEE*, Yuzhe Ma, and Bei Yu, *Member, IEEE*

Abstract*—Digital circuit physical unclonable function (PUF) has been attracting attentions for the merits of resilience to the environmental and operational variations that analog PUFs suffer from. Existing state-of-the-art digital circuit PUFs, however, are either hybrid of analog-digital circuits which are still under the shadow of vulnerability, or impractical for real-world applications. In this paper, we propose a novel highly nonlinear and secure digital PUF (D-PUF) and the spliced version SD-PUF. The fingerprints are extracted from intentionally induced very large-scale integration interconnect randomness during lithography process, as well as a post-silicon shuffling process. Strongly skewed CMOS latches are used to ensure the immunity against environmental and operational variations. Crucially, a highly nonlinear logic network is proposed to effectively spread and augment any subtle interconnect randomness, which also enables strong resilience against machine learning attacks. On top of it, the expandable architecture of the proposed logic network empowers a novel post-silicon shuffle-splice mechanism, where multiple randomly selected D-PUFs are spliced to be one SD-PUF, pushing the statistical security to a much higher level, while significantly reducing the mask cost per PUF device. It also decouples the trustworthy demands enforced to the foundries or other third party manufacturers. Our proposed PUFs demonstrate close to ideal performance in terms of statistical metrics, including 0 intra-Hamming distance. Various state-of-the-art machine learning models show prediction accuracies almost no better than random guesses when attacking to the proposed PUFs. We also mathematically prove the probability of existence of identical SD-PUF pair is significantly lower than that of D-PUF pair, e.g., such probability of an SD-PUF spliced by 30 D-PUFs is 2.3×10^{-22}, which is 19 order magnitude lower than that of D-PUF. Benefited from the proposed shuffle-splice mechanism, the mask cost per SD-PUF is also reduced by $300\times$ than that of D-PUF.

***Index Terms*—Digital physical unclonable function (D-PUF), hardware security, learning attacks, PUF.**

I. INTRODUCTION

THE demand on highly secure and reliable authentication solutions has been significantly increasing with the era of Internet of Things. The pervasive embedded computing devices need to first assure safety of sensitive information and life-critical actions in the physical world. Silicon physical unclonable function (PUF) is an innovative low cost hardware security primitive [1]–[5] that derives authentication fingerprints from manufacturing process variations in integrated circuits.

PUFs are classified into strong PUF and weak PUF, depending on the number of unique challenge-response pairs (CRPs) the PUF can produce. A strong PUF has sufficiently large CRP space, hence it is impossible to enumerate or predict any CRP within a limited time-frame. By contrast, a weak PUF is often used for secure key generation due to the limited CRP space. We focus on strong PUF in this paper.

A number of PUF architectures have been proposed in literature, and today the most popular PUFs are often modeled with memory [6]–[8], delay [9]–[12], etc. All those PUF architectures utilize *transistors*' intrinsic randomness under process variations, which are by nature analog attributes. Just as any analog circuit systems, analog PUFs are commonly vulnerable to environmental and operational variations. Therefore, analog PUF architectures are often equipped with a fuzzy extractor [13] or an error correction system [14]. Those auxiliary circuits often require hardware cost and power consumptions several order magnitude higher than the PUF circuit itself. In addition, some analog PUF models, e.g., delay-based PUF, have been demonstrated insecure from side channel and machine learning attacks [15], [16].

There is a strong demand to derive digital circuit PUFs in order to tackle the reliability limitations in analog PUFs as well as to provide low latency, high throughput PUF solutions. In recent years, there have been some literatures proposing prototype digital circuit PUFs [17], [18]. Xu and Potkonjak [17] proposed PUF based on FPGA that combines FPGA fabric with a standard analog PUF. The analog PUF is used at initialization stage and can be discarded afterwards. However, this is not a completely digital circuit PUF, hence it faces the same limitations as analog PUF does, particularly in the initialization stage. Later, Xu and Potkonjak [18] proposed the first conceptual level fully digital circuit PUF by using defective digital IC chips. It is based on the observation that a small circuit fault can drastically impact the overall functionality of a digital logic, hence such circuit faults, which were modeled as stuck-at-faults and bridge-faults, can be used as the fingerprint for PUF. However, those fault models are too simplified to catch the physical impacts induced by the fault. Depending on specific contexts, some circuit faults may lead to serious physical and logical chain-effect, and ultimately break down the entire circuit system. This contradicts to the original purpose of using digital circuit PUFs. For example, the wired-AND model in CMOS circuits may result in a direct current

Manuscript received March 13, 2017; revised June 20, 2017; accepted August 2, 2017. Date of publication August 15, 2017; date of current version April 19, 2018. This work was supported by the Research Grants Council of Hong Kong SAR under Project CUHK24209017. This paper was recommended by Associate Editor C. H. Chang. *(Corresponding author: Jin Miao.)*

J. Miao and S. Roy are with Cadence Design Systems, San Jose, CA 95134 USA (e-mail: jinmiao@utexas.edu).

M. Li is with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78712 USA.

Y. Ma and B. Yu are with the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2017.2740296

0278-0070 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

path from supply voltage to ground, and put the CMOS gate to an uncertain operating region. This may cause large power waste and even unstable outputs in CMOS gates. Therefore, it is necessary to reconsider the feasibility by *direct* use of a defective IC chip as PUF. Nevertheless, the concept of utilizing faulty circuits is still valuable and inspires our development of PUFs.

In addition, one critical question is yet to be addressed—what would be the optimal logical circuit to use for the maximal security? In [18], an array multiplier was taken to demonstrate the defective IC PUF concept, however, neither linearity analysis nor learning-based reverse engineering was conducted. In fact, unlike analog PUFs where the nonlinearity is derived by transistor attributes, many digital logic circuits intrinsically can be *linearly separable* [19], and relying on arbitrary logic circuits may lead to insecure PUFs vulnerable to even linear model machine learning attacks. Therefore, a nonlinear logic architecture is highly desired to realize a secure PUF.

In this paper, we first propose a single chip-based, highly nonlinear and secure digital PUF (D-PUF) that overcomes the reliability drawbacks in analog PUFs, as well as the practicality and security issues in existing literatures. The proposed D-PUF[1] take advantages of *Boolean* type interconnect randomness induced by lithography variations, and crucially, the digitalization is realized by using a strongly skewed latch to ensure the Boolean status for all internal signals. The interconnect randomness is ultimately spread and cross-coupled by a novel highly nonlinear logic network architecture. The proposed D-PUF demonstrates close to ideal statistical performance and strong resilience to machine learning attacks. Later, on top of the D-PUF, a post-silicon shuffle-splice mechanism is introduced to shuffle and connect multiple D-PUFs to form a single spliced D-PUF (SD-PUF). We prove formally that such SD-PUF significantly strengthens the statistical securities over D-PUF, while also dramatically reducing the average mask cost for each PUF device. Such post-silicon procedure also alleviates the security restrictions in foundries, where the shuffle-splice procedure can be conducted in-house.

Our major contributions are summarized as follows.

1) Quantitatively justify the feasibility of utilizing the interconnect randomness induced by lithography variations.
2) Propose to use strongly skewed latches to ensure a complete and reliable digital circuit PUF architecture.
3) Propose a novel highly nonlinear logic network architecture that can effectively spread and augment any interconnect randomness, as well as achieve strong resilience to machine learning attacks.
4) Propose a novel shuffle-splice post-silicon mechanism to significantly strengthen the securities of PUFs, while reducing mask cost.
5) Mathematical bounds are derived and proved to ensure the uniqueness of SD-PUF.

The rest of this paper is organized as follows. In Section II, we discuss the source of Boolean type randomness during lithography process. In Section III, we propose our solution to make the interconnect randomness compatible to CMOS technology. In Section IV, we propose the single chip-based D-PUF architecture. In Section V, we analyze the properties of D-PUF. In Section VI, we propose the post-silicon shuffle-splice mechanism for SD-PUF, and prove the mathematical bounds to ensure the uniqueness of SD-PUF. In Section VII, we show silicon cost and detailed lithography results as well as evaluate D-PUFs and SD-PUFs with various statistical metrics and machine learning attacks, followed by the conclusion in Section VIII.

[1]The preliminary version has been presented at the International Conference on Computer-Aided Design (ICCAD) in 2016 [20].

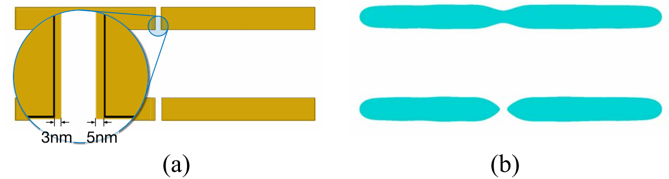


Fig. 1. Interconnect under lithography variations. (a) Mask stripe-pairs with split distance of 20 nm (top) and 28 nm (bottom). The top mask is zoomed in. (b) Lithography simulation outputs (shapes on wafer).

II. Boolean Randomness by Lithography

Identifying a feasible Boolean randomness source is half the battle to make a D-PUF. Conventional analog PUFs rely on transistor's intrinsic randomness, including delay, current, resistance, capacitance, etc. Xu and Potkonjak [18] first proposed to take advantage of the randomness from very large-scale integration (VLSI) interconnect, namely the metal wires. Such interconnect randomness, by itself, is a Boolean type variable, i.e., in either connected or disconnected status. The feasibility of utilizing the interconnect randomness, however, was not justified. It was also unclear whether or how such randomness can be controlled during design and manufacturing stages. In this section, we will analyze how process variations affect the VLSI interconnect, and will quantitatively demonstrate the feasibility of utilizing such randomness.

As VLSI technology node scales down to nanometer regime, one of the major interconnect geometrical variations comes from lithography. The lithography variations can be categorized into "systematic" and "local" variations.[2] The systematic variation, including dose (light density) and focus variations in lithography system, refers to a systematic offset applied to a group of adjacent layout patterns, and is often considered as inter-die variation. The local variation, by contrast, including mask errors and line edge roughness, refers to localized or intra-die randomness for each individual layout pattern. Considering typical PUF circuits to be small in size, the local variation is more of importance to generate unique fingerprints and should dominate the systematic variations.

There have been literatures utilizing lithography variations for PUF designs [21]–[23]. Kumar and Burleson [21] proposed to utilize local pitch variations for PUF design. Rather than Boolean type randomness, the lithography variations were used to generate transistor analog fingerprints, leading to an analog PUF. Sreedhar and Kundu [22] and Forte and Srivastava [23] claimed to use focus and dose variations as a ubiquitous approach applied to general PUF designs. As aforementioned, dose and focus are systematic variations mostly impacting inter-die variations, the overall performance improvements can be limited.

[2]Local variations are also referred as random variations.

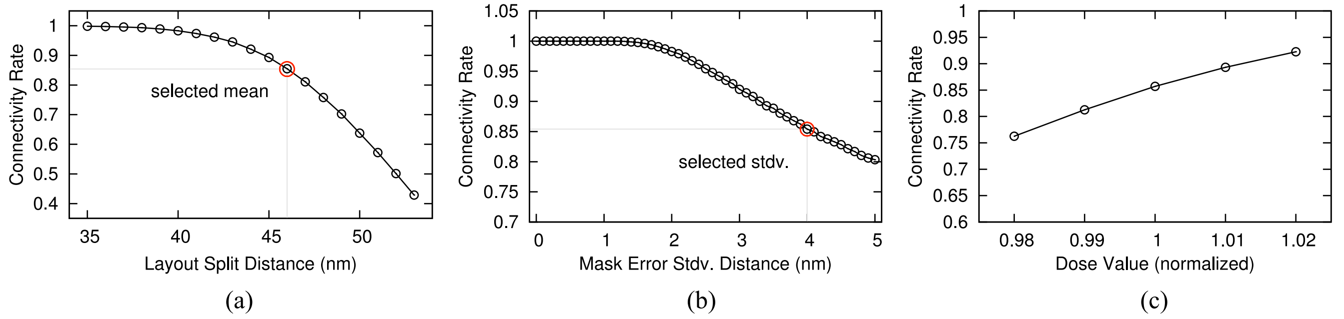


Fig. 2. Interconnect connectivity rate under lithography variations. (a) Layout split distance under mask error stdv of 4 nm. (b) Mask error stdv under split distance of 46 nm. (c) Dose values.

In this paper, we utilize the interconnect randomness from the very local *mask variation* when producing a photograph mask by an electron beam lithography system (the conventional mask manufacturing tool). Mask error enhancement factor (MEEF) [24] is used to quantify how mask variations will be reflected in the final wafer. In advanced technology nodes, even with gridded design rules and resolution enhancement techniques, the MEEF value on line-end can be up to $10\times$ [25], which means a 5-nm variation in mask line-end pattern would cause 50-nm change in the final wafer pattern. In practice, *an electron beam system can easily lead to mask variations well exceed this 5-nm variation threshold* [26].

In this work, the interconnect randomness is realized by intentionally positioning two interconnect layout line-ends close to each other, and due to mask variations, the generated masks will have mismatches. Such mismatch is further magnified by MEEF factor on the wafer, and ultimately leads to uncertain connectivity status. Fig. 1(a) shows two mask stripe-pairs split by a small distance. The bottom stripe-pair is with the original split distance of 28 nm, while the top stripe-pair is with distance of 20 nm due to mask variations. Here we assume 3-nm and 5-nm mismatches for the two line-ends, respectively. We use an industry lithography simulator [27] to simulate these two mask stripe-pairs, and Fig. 1(b) shows the final output images. The 8-nm difference of the split distance in the two stripe-pairs is now converted to two Boolean connectivity statuses: the top stripe-pair is merged and connected, while the bottom remains disconnected.

We now justify how the overall connectivity statistics are impacted by the layout split distance as well as local and systematic lithography variations. Before that, we introduce the concept of *connectivity rate* as the number of eventually connected stripe-pairs over the number of total stripe-pairs. Note that the split distance can be controlled by circuit engineers when designing VLSI layout, and the local variation, specifically the mask error, can be modeled by centered Gaussian distribution[3] [24], [28], where the standard deviation (stdv) is depending on the accuracy and settings of the electron beam system. Both factors can be configured in today's VLSI manufacturing setup. By contrast, the dose variation is a systematic offset applied to specific wafer zones which should be minimized. Such offsets could shift the connectivity status toward a single direction on certain wafer zones, hence degrade the level of interconnect randomness and further the PUF performance. Therefore, the central task in the rest of this section is to justify the feasibility of minimizing the impact from systematic variations by carefully configuring the split distance and mask error.

We first evaluate the mask variations under various split distances ranging from 35 to 53 nm with 1-nm step. For each split distance, we further sweep the mask error stdv ranging from 0 to 5 nm, leading to various mask stripe-pair sets, each with size of 10K. These variously configured 10K stripe-pair sets are later fed into the lithography simulator [27] to get the ultimate stripe shapes on wafer. We then measure the connectivity rate of each 10K stripe set. For example, Fig. 2(a) shows when the mask error stdv is 4 nm, by changing the layout split distance, connectivity rate can vary from about 0.4 to 0.99. And Fig. 2(b) shows when the layout split distance is 46 nm, connectivity rate changes from 1.0 to 0.8 when the mask error varies from 0 to 5 nm.

Further, we evaluate the impact from dose variation that would cause potential systematic offsets to the split distance.[4] In Fig. 2(c), we sweep the normalized dose value from 0.98 to 1.02 (the maximum available range) with split distance of 46 nm and mask error stdv of 4 nm, where the 46 and 4 nm are the selected configurations that can minimize the dose impact. Clearly, the connectivity rate retains in a high value between 0.75 and 0.93. We show in the later section that, higher connectivity rate (but less than 1) generally leads to better security performance in our proposed D-PUF architecture. Therefore, by carefully configuring the layout split distance and electron beam system accuracy, it is feasible to minimize the impact from the systematic offset like dose. In Section VII-A, we will show more comprehensive lithographic simulation results on our proposed D-PUF interconnections.

Overall, we have justified the feasibility of using interconnect geometrical variation for PUF design. However, such interconnect randomness cannot be directly used in the VLSI circuit systems, especially for CMOS circuits, due to serious physical incompatibilities. In the next section, we will propose our solution to make such randomness compatible to digital VLSI systems.

III. Making it CMOS Compatible

In CMOS circuits, any *unexpected* open-circuit and short-circuit may lead to serious circuit failures. Such failures are not only in logical perspective, but can also adversely affect

[3]We show in Section V that, the performance of D-PUF relies on the cumulative connectivity rate regardless of any specific distribution pattern.

[4]In this paper, we ignore the systematic variations from focus, as it can be dealt with in a similar manner as dose [23].

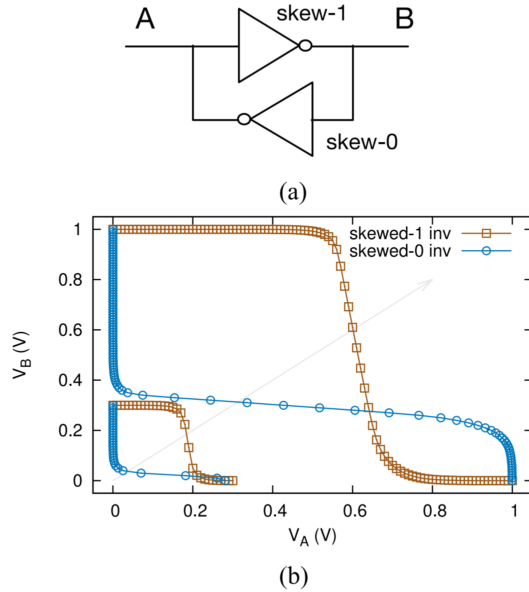


Fig. 3. Handling dangled poly gate by strongly skewed latch. (a) Inverter pair skewed latch structure. (b) VTC relation of a strongly skewed latch.

the physical reliability and power efficiency. Particularly, for the case of open-circuit failure, the transistor poly gate is dangled with floating voltage level. This makes the transistor vulnerable to environmental noise which may even change the transistor operating region. For the case of short-circuit failure, where two CMOS gates' drains are connected, there will be a good chance to create a direct current path from supply voltage to ground, resulting huge power waste as well as unknown transistor operating region. Apparently, both scenarios are opposite to the original wish of D-PUF. In this section, we propose our solution to completely eliminate the aforementioned issues and make the interconnect randomness compatible to CMOS circuits.

The goal is to identify a pure logical structure so that it can work with both open-circuit failures as well as normally connected circuits.[5] A CMOS latch, by connecting two inverters head-to-tail, can be a good candidate. A latch is supposed to either remain a pre-existing state until a new input is applied, or preset an initial state and later never change it. The first feature ensures the compatibility to normal circuit operations, and the latter one makes it possible to work with open-circuit failures. For open-circuit failures, the input to latch is dangled, and the initial state will be automatically set during the power up state [6]. However, for a regular symmetric latch, the initial power up state may be affected by static noises, hence can be inconsistent from time to time. To completely eliminate such uncertainties induced by static noise, in Fig. 3(a), we propose to use a strongly skewed-1 inverter that head-to-tail connects to a strongly skewed-0 inverter. The skewed-0 inverter can further strengthen the skewed-1 inverter, hence together form a strongly skewed-1 latch. Note that, without any loss of generality, we only discuss skewed-1 latch in this paper. Here, the skewed-1 inverter is realized by specifying: 1) pMOS width several times wider than its nMOS counterpart and 2) lower voltage threshold (VT) for pMOS and higher VT for nMOS. The skewed-0 inverter can be derived by the opposite configuration. In Fig. 3(a), if pin A gets disconnected, minor static noise cannot change the power up state of this skewed latch, and the latch will favor more to stay at logic 1. Note that after power up phase, the latch will remain in the logic value until supply voltage is removed. In addition, the skewed-1 inverter has to be designed with larger size than the skewed-0 inverter for the case that when the latch is normally connected to the network without open-circuit failure, the skewed-1 inverter should dominate the skewed-0 inverter. In that case, the skewed latch is reduced to a regular inverter.

In Fig. 3(b), we show the HSPICE simulations with PTM-45-nm model for voltage-transfer curves (VTCs). The skewed-1 inverter has 10× wider width on pMOS than its nMOS counterpart, and the skewed-0 inverter has 4× wider width on nMOS than its pMOS. Besides, the pMOS in skewed-1 inverter uses low VT pMOS and high VT nMOS transistors, and vice versa for skewed-0 inverter. It can be observed that in the power up voltage region, i.e., less than 0.3 V, the noise margin is maximized to favor logic 1, hence practically guarantees a deterministic power up state. When voltage increases to 1.0 V, the latch will remain in logic 1 unless pin-A is applied by a new input.

Note that in memory-based PUF literatures, latch skewness was also discussed and used as the power-up fingerprint [6], [7]. The major difference against memory PUF is that, in memory circuits, all latches are designed to be symmetric, and the skewness comes from the *intrinsic* process variation and is used as the *source* of fingerprint. However, such intrinsic variation does not guarantee all latches to be skewed in memory PUFs, which is the very root cause of the reliability drawbacks for such type of PUFs. By contrast, we intentionally skew *all* latches in order to completely eliminate any possible environmental vulnerability as well as to make the interconnect randomness compatible to CMOS systems. We will further demonstrate by HSPICE in Section VII that, such strongly skewed latches retain consistent power-up state across very wide temperature and voltage ranges. With above preparation, in the next section, we introduce our learning resilient D-PUF architecture.

IV. Proposed D-PUF Architecture

To this point, we have converted the PUF design to be a pure logic design problem. In this section, we shift the focus to identify a nonlinear logic network that can maximize and spread any subtle interconnect randomness, and eventually realize a highly secure D-PUF. We propose to derive a logic network constructed by *regularly repeated* nodes, and we call each node a *unit cell*. In the following, we will first discuss the design for unit cells, and then propose the overall logic network topology, leading to the single chip-based D-PUF. In Section V, we will further analyze in detail the properties of the proposed D-PUF, and in Section VI, we discuss a post-silicon mechanism to further boost the security performance and efficiency by splicing multiple D-PUFs into an SD-PUF.

A. Unit Cell

In cryptography, exclusive-OR (XOR) logic is the most popular function due to the simplicity in realization and perfect security nature. Due to the linearly nonseparable attribute, an XOR logic outstands other logics like AND, OR, etc., offering

[5]The short-circuit failure case will be constructively avoided in our D-PUF architecture.

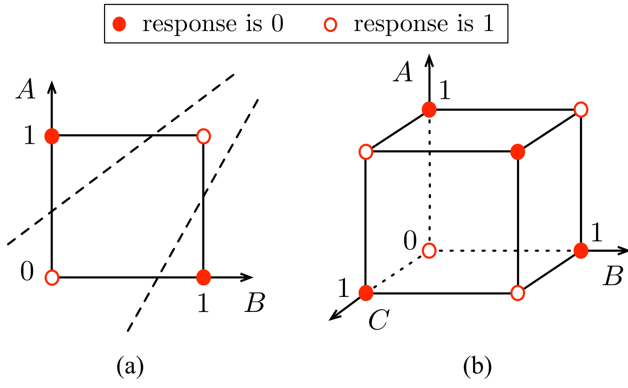


Fig. 4. Linear nonseparable nature for XOR logic. (a) Two-input XOR logic $Y = A \oplus B$ requires at least two lines to separate the 1 and 0 dots. (b) Three-input XOR logic $Y = A \oplus B \oplus C$ requires at least three planes for separation.

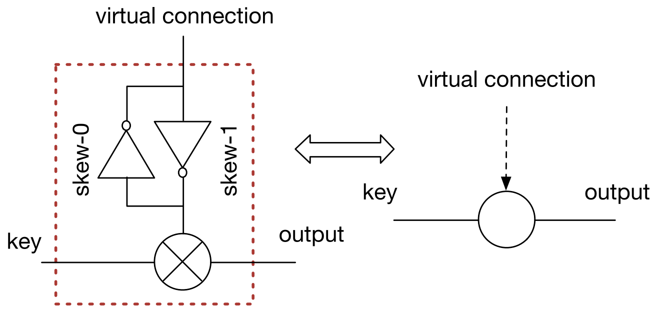


Fig. 5. Unit cell. Left: complete logic structure. Right: simplified symbol.

intrinsic resilience to many learning-based attacks. As shown in Fig. 4, to separate the output of a two-input XOR logic, at least two lines are required, and three planes are required for a three-input XOR logic. Another intriguing feature of XOR logic is the uniform output distribution. In Lemma 1, we show that for a two-input XOR, as long as *one* of the inputs is uniformly distributed, the XOR output is ensured uniformly distributed. This property will be further discussed in Section V once the entire D-PUF topology is introduced. Therefore, XOR logic can be a perfect candidate for our unit cell design.

Lemma 1: $\Pr[y = 1] = \Pr[y = 0] = 0.5$ holds, as long as $\Pr[a = 1] = \Pr[a = 0] = 0.5$, $\forall b \in B$, where a and b are the two inputs of a two-input XOR gate, and y is the output. The symbol $\Pr[y = 1]$ refers to the probability of output y being logic 1, and vice versa.

Proof: For a two-input XOR gate, the probability of output being logic 1 can be written as $\Pr[y = 1] = \Pr[a = 1]\times\Pr[b = 0] + \Pr[a = 0] \times \Pr[b = 1]$. Consider $\Pr[a = 1] = \Pr[a = 0] = 0.5$, we have $\Pr[y = 1] = 0.5 \times \Pr[b = 0] + 0.5 \times \Pr[b = 1] = 0.5 \times (\Pr[b = 0] + \Pr[b = 1]) = 0.5$. Hence $\Pr[y = 0] = 1 - \Pr[y = 1] = 0.5$. ■

Fig. 5 shows the unit cell structure. It is a two-input one-output logic block, constructed by a two-input XOR gate with one of its inputs connected to the strongly skewed-1 latch. The *key* pin is the actual information bit that passes through this XOR gate. The *virtual connection* pin is the source of the randomness. It may or may not connect to the logic network depending on the interconnect randomness status. If this virtual connection pin is connected to a stable logic value, the *output* of the entire unit cell has logic expression of $key \otimes \overline{virtual\ connection}$. If it is dangled, as discussed in Section III, since the skewed-1 latch stays in logic 1 state after power up, the output of the unit cell equals to *key*. In general, the unit cell can be viewed as a random "bit-flip" block, where a 1-bit information (key pin) may or may not get inverted depending on the interconnect randomness. Apparently, in any case, the unit cell output is a legal and stable logic value. For simplicity, we use a bubble symbol to represent the unit cell in Fig. 5, and the dashed arrow for the virtual connection pin.

B. D-PUF Logic Network

Recall that in Fig. 4, higher dimension of XOR inputs require more number of hyper-planes for separation hence indicating higher level of nonlinearity. This hints a cross-coupled XOR logic network. We therefore propose our D-PUF architecture in Fig. 6(a). It is an XOR-based logic network with *N-input* and *N-output*. There are N *rows* and M *columns*, where one *column* refers to one unit cell per *row*. Note that the dashed arrow refers to virtual connection pin of the unit cell, indicating possible disconnection from the network. The unit cell's output pin is sequentially cascaded to next unit cell's key pin, and the virtual connection pin is *possibly* connected to its neighbor row. For 0th and $(N-1)$th rows, i.e., the boundary rows, some unit cells' virtual connection pins are always dangled, and they are marked by letter "Z." The two boundary rows will have undistinguishable impact on the overall PUF performance due to the highly coupled XOR network dependencies.

We write down the logical expression for nonboundary nodes in recursion manner. Boundary node expressions can be easily derived by substituting with *input* or *Z* pins. Here the $k_{i,j}$ refers to *i-row j-column* output, and v refers to the virtual connection status

$$k_{i,j} = \begin{cases} k_{i,j-1} \oplus \left(v \cdot k_{i+1,j-1} + \overline{v}\right), & i \text{ even}, \quad j \text{ even} \\ k_{i,j-1} \oplus \left(v \cdot k_{i-1,j-1} + \overline{v}\right), & i \text{ even}, \quad j \text{ odd} \\ k_{i,j-1} \oplus \left(v \cdot k_{i-1,j} + \overline{v}\right), & i \text{ odd}, \quad j \text{ even} \\ k_{i,j-1} \oplus \left(v \cdot k_{i+1,j} + \overline{v}\right), & i \text{ odd}, \quad j \text{ odd}. \end{cases} \tag{1}$$

For better understanding of the D-PUF network, we can view each row as a magic "signal tunnel." When the 1-bit input information is passing through this tunnel, it may or may not be flipped due to the uncertain status of virtual connections pins. Crucially, since each pair of neighbor rows have bi-directed virtual connections in between, each virtual connection also relies on its precedent as well as upper and lower neighbors' virtual connection statuses, resulting in a highly nonlinear dependency graph. As long as the number of column M is no less than the number of row N, i.e., $M \geq N$, the logic cone of each out_i will have the potential to cover all the inputs in_j. We show an 8-row by 8-column D-PUF example in Fig. 6(b). The logic cone of out_2 is highlighted in red color which covers all inputs. The same coverage is true for every output. In the next section, we will discuss in detail the D-PUF properties.

V. Property Analysis on D-PUF

In this section, we reveal important properties of the D-PUF. First of all, it is intriguing to figure out how does the probability of the overall interconnect status impact the D-PUF performance. In line with the definition used in Section II, we define "connectivity rate" as the number of the nondangled

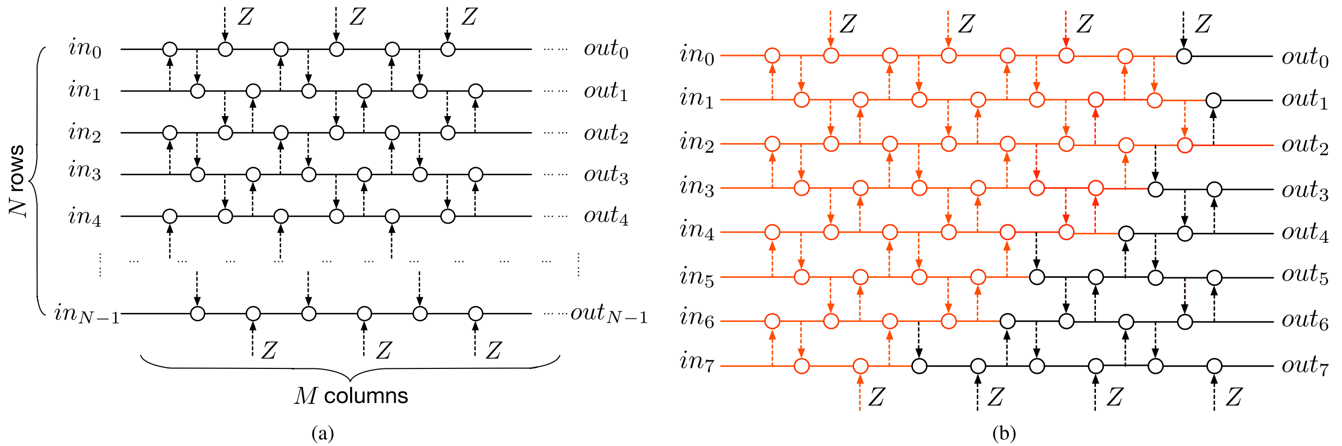


Fig. 6. D-PUF architecture. (a) General architecture with N-row by M-column. Some boundary virtual connection pins are marked by Z indicating dangling status. (b) 8×8 D-PUF. Logic cone of out_2 is highlighted in red color.

virtual connection pins over the total number of virtual connection pins in a D-PUF architecture. For the two corner cases, i.e., the connectivity rate is 0 or 1, this D-PUF logic network is reduced to a deterministic Boolean function, hence no longer a PUF. We therefore need to carefully control the connectivity rate. As we have discussed in Section II, the connectivity rate can be controlled by circuit designers and/or lithography system accuracy.

Consider an N-row by N-column D-PUF, the total number of the virtual connections, is N^2. From PUF design perspective, on one hand, the connectivity rate should not be too high in order to generate a rich space of unique PUFs. Suppose m out of the N^2 virtual connections are connected, i.e., the rate is (m/N^2), there will be m-combinations of N^2, i.e., $[N^2!/(m!(N^2 - m)!)]$ unique PUFs. We know the maximum number of unique combinations occurs when connectivity rate is 0.5. For a 64-row by 64-column D-PUF, this ends up with 1.8×10^{1696} unique PUF chips. However, on the other hand, intuitively, higher connectivity rate means more complex dependencies in the logic network, hence leads to stronger resilience to learning-based attacks.

Property 1: The D-PUF logic network is nonlinear, and higher connectivity rate leads to stronger nonlinearity.

As shown in Fig. 4, since an XOR gate is linearly nonseparable, the cascaded XOR is also linearly nonseparable. High connectivity rate means more unit cells are cross-coupled, hence a higher level of nonlinearity for the dependency graph. The maximum nonlinearity happens when connectivity rate is 1, whereas, such a circuit is no longer a PUF. We will show in Section VII that the D-PUF shows strong resilience to nonlinear machine learning attacks.

Theorem 1: Equation $\Pr[\text{out}_j = 1] = \Pr[\text{out}_j = 0] = 0.5$ holds as long as $\Pr[\text{in}_j = 1] = \Pr[\text{in}_j = 0] = 0.5$, $\forall j \in N$, where N refers to the number of rows in D-PUF.

Proof: Since $\Pr[\text{in}_j = 1] = \Pr[\text{in}_j = 0] = 0.5$, with Lemma 1, the *output* of the first unit cell U_{j0} in row j has $\Pr[U_{j0} = 1] = \Pr[U_{j0} = 0] = 0.5$, regardless of the virtual connection status on the node U_{j0}. Notion U_{jk} refers to the kth unit cell in row j. By repeatedly applying Lemma 1 to U_{jk} $\forall k \in M$, we have $\Pr[U_{jk} = 1] = \Pr[U_{jk} = 0] = 0.5$. Hence, $\Pr[\text{out}_j = 1] = \Pr[\text{out}_j = 0] = 0.5$. Here M refers to the number of columns. ■

Theorem 1 ensures that, when input follows uniform distribution, output in the same row retains the uniform distribution nature, i.e., equal chance to output 1 and 0. This theorem holds regardless of the virtual connection status.

Property 2: There will be a sufficiently large space of unique D-PUFs even if the connectivity rate is high.

Consider a 64-row by 64-column D-PUF, if 10 virtual connections get disconnected, there will be 3.6×10^{29} unique PUFs, and the connectivity rate is $(4086/4096) = 99.76\%$. When it increases to 20, the unique PUF space size goes up to 6.9×10^{53} and the connectivity rate is still as high as $(4076/4096) = 99.51\%$. Therefore, high connectivity rate does not adverse the uniqueness of the D-PUF and is more preferred for better learning resilience.

Property 3: Increasing the number of columns strengthens the resilience to machine learning attack.

In Fig. 6(a), increasing the column number of unit cells creates more interleaving connections between the neighbor rows, hence higher level of XOR logic dependency can be foreseen for each output. Furthermore, wider columns means more interdependent paths exist in the D-PUF network, hence stronger resilience to learning-based attack. Related discussion will be verified in Section VII.

Property 4: Any subtle change on the virtual connections will be reflected to multiple outputs.

Unlike other logics, like AND, OR, etc., for XOR logic, any input change will be reflected on the output. In addition, due to the cross-coupled and recursive dependencies in D-PUF network [see (1)], such changes will be propagated to multiple outputs. Even slight difference between two D-PUFs can lead to significantly different CRPs characterizations, realizing high uniqueness of fingerprints. This will be verified by the inter-Hamming distance (inter-HD) and "avalanche" effect in Section VII.

VI. SD-PUF: Post-Silicon Boost

So far we have been discussing one chip-based D-PUF, however, if the mask is compromised (e.g., by stolen, or unreliable foundries), it poses a slight but potential risk to duplicate a D-PUF. Although it can be very hard to reproduce

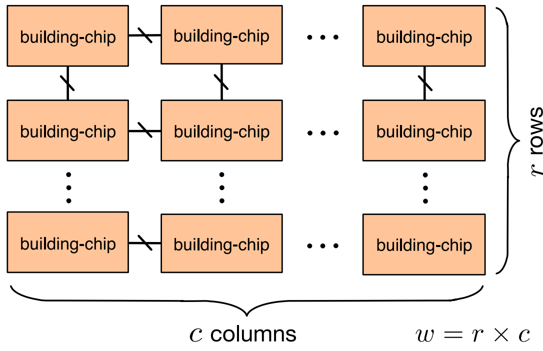


Fig. 7. SD-PUF is made of w building-chips, breaking down to $r \times c$ building-chips, i.e., $w = r \times c$.

an identical lithography process environment, the vulnerability due to mask stolen is still a possible loophole. On top of that, there also exist concerns where excessive numbers of lithography processes on the same mask might increase chances of generating duplicated D-PUFs, which potentially limits the mask reusability and eventually leads to high cost per D-PUF device. In this section, we propose a novel post-silicon process mechanism to eliminate these security concerns, which is proven to significantly boost the PUF statistical performance as well as improving the mask cost efficiency significantly.

A. Shuffle-Splice Mechanism

It can be seen in Fig. 6(a) that a D-PUF is easy to expand or shrink its sizes: by connecting with other D-PUFs or reducing sizes of N, M. By taking advantages of these architectural merits, we propose "shuffle-splice" to resolve the mask security issue. The basic idea is that, instead of making a PUF by one single chip, we combine multiple "building-chips" into an SD-PUF (see Fig. 7). For those building-chips, each individual one is equivalent to a reduced sized D-PUF, and all of them together form a *pool* of building-chips. The splicing mechanism can lead to a significantly large permutation space for SD-PUFs. Crucially, before splicing, the pool of building-chips will be shuffled in a mechanical way, introducing another level of randomness. Note that, those building-chips can be even first I/O packaged to be individual devices, hence the shuffle-splice (board level welding) procedure can be conducted either in-house or in a trustworthy foundry.

Therefore, even if the mask is compromised or even some building-chips get reproduced by attackers, it is still impossible to enumerate all permutations to reproduce a specific SD-PUF. There remains an interesting question yet to be addressed: what is the maximum number of SD-PUFs that can be produced by a single mask but still statistically guaranteed to be mutually unique? Intuitively, this uniqueness is partly regulated by the number of building-chips an SD-PUF consists of, as well as how many building-chips a mask can produce at a time. Those factors regulate the difficulties of enumerating or duplicating an SD-PUF, which will be discussed with details in the next section. Note that among all the four major statistical evaluations on PUF securities, only inter-HD, i.e., the uniqueness of individual PUF chips, will be affected by this shuffle-splice procedure, whereas, the rest metrics like intra-Hamming distance (intra-HD), bit-alias, and uniformity are independent of the splice procedure, as they are more of the result of CMOS latch attributes and the XOR logic network.

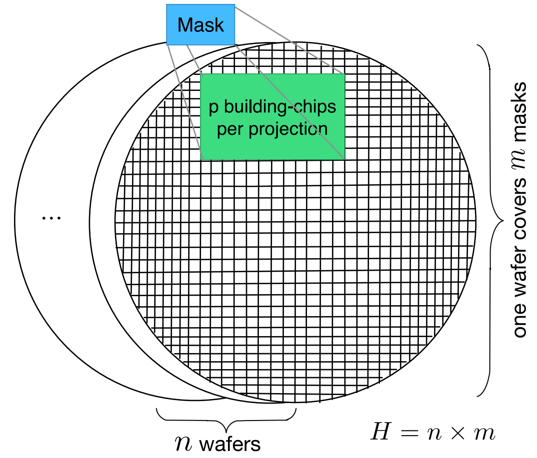


Fig. 8. Each mask is used for H projections, breaking down to m projections per wafer times n wafers, i.e., $H = n \times m$. Here each projection produces p building-chips.

B. Probability Bound on Uniqueness of SD-PUF

Suppose an SD-PUF is made of w building-chips, breaking down to $r \times c$. See Fig. 7, where r and c refer to the numbers of building-chips orthogonal and along the input/output direction, respectively. Further, each individual projection on a given mask can simultaneously produce p building-chips, i.e., one "batch" (see Fig. 8). A total number of up to H batches can be conducted on the same mask while still maintaining a high uniqueness level for each SD-PUF.

We start by categorizing all p building-chips of one batch into two groups: the distinct group and *duplicated* group. For a building-chip if there exists *at least* one other building-chip (from the same batch) sharing the identical virtual connectivities, this building-chip belongs to the duplicated group. The remaining chips belong to the distinct group, where each individual chip possesses unique virtual connectivity among the overall p chips. Since one mask can be used for H batches of projections, including m times per wafer, multiply by n wafers ($H = m \times n$), in the worst scenario, i.e., assuming mask error is the *only* lithography randomness which is practically very unlikely, building-chips produced from the same *geometrical location* of the mask will then share the same virtual connectivities across the H batches. That means, even for a building-chip, say α, which belongs to the distinct group, in the worst case, it may end up with H identical building-chips α.

We denote the ratio of the distinct group size to the total number of building chips in ith batch B_i as γ_i. Therefore, γ_i quantifies the probability of the uniqueness within one mask projection corresponding to B_i (see Fig. 9). We further define B_i^{distinct} as the set of the distinct building chips in B_i, and so $|B_i^{\text{distinct}}| = \gamma_i |B_i| = \gamma_i p$.

Those aforementioned parameters γ_i, p, w, and H are key factors regulating the uniqueness of the SD-PUF. In the following, we will formally derive an upper bound on the probability of any two SD-PUFs sharing the same virtual connectivities. Ideally, this upper bound probability should be as low as possible to ensure a high level security, specifically, inter-HD.

Lemma 2: $\forall b_i^{\text{distinct}} \in B_i^{\text{distinct}}$, there exist at least $(\gamma_j p - 1)$ building-chips in B_j that are distinct from b_i^{distinct}, where $i, j \in [1, H]$.

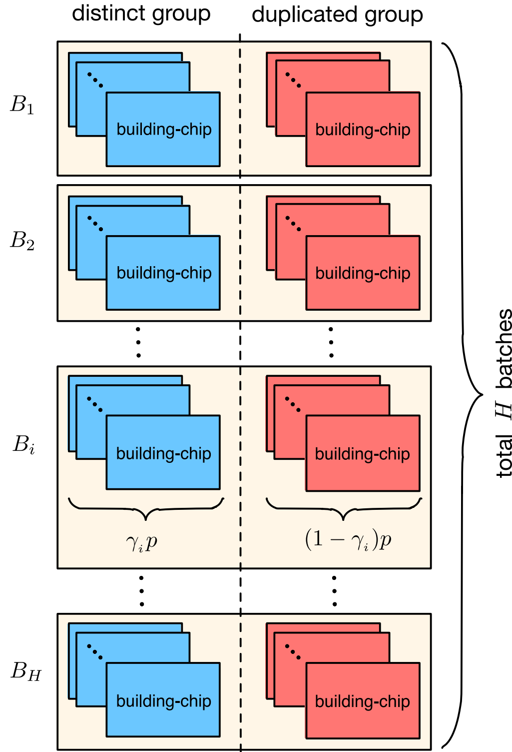


Fig. 9. Total H batches of the projections on the same mask. Building-chips are categories to be "distinct" group (in blue color) and "duplicated" group (in red color). For ith batch, there are $\gamma_i p$ distinct building-chips, and $(1-\gamma_i)p$ duplicated building-chips.

Proof: Let us first consider the trivial case $j = i$. By the definition of "distinct group," b_i^{distinct} is distinct from all other building chips $(p-1)$ in B_i. Therefore, the number of building chips in B_i which are distinct from b_i^{distinct} equals to $(p-1) \geq (\gamma_j p - 1)$ (since $\gamma_j \leq 1$).

For $j \neq i$, there remains two cases. On one hand, if b_i^{distinct} is distinct from all building chips in B_j^{distinct}, the total number of building chips in B_j which are distinct from b_i^{distinct} is no less than $|B_j^{\text{distinct}}| = \gamma_j p \geq (\gamma_j p - 1)$. On the other hand, due to identical geometrical locations, b_i^{distinct} may be identical with one building chip $b_j^{\text{distinct}} \in B_j^{\text{distinct}}$, since b_j^{distinct} is unique in B_j, b_i^{distinct} cannot be identical with other $(p-1)$ building chips in B_j. So the total number of building chips in B_j that are not identical to b_i^{distinct} is $(p-1) \geq (\gamma_j p - 1)$.

The proof completes by combining all above three cases. ■

We define $\overline{\gamma}$ as the mean of set $\{\gamma_i\}$, $i \in [1, H]$, namely, $\overline{\gamma} = (1/H)\sum_i^H \gamma_i$.

Lemma 3: The total number of ways of selecting a pair of distinct building-chips from the building-chip pool is at least $(1/2)H^2\overline{\gamma}p(\overline{\gamma}p - 1)$.

Proof: According to Lemma 2, the lower bound of the total number of building-chips from the building-chip pool of H batches that are not identical with any $b_i^{\text{distinct}} \in B_i^{\text{distinct}}$ is $\sum_j^H(\gamma_j p - 1) = H(\overline{\gamma}p - 1)$. Since $|B_i^{\text{distinct}}| = \gamma_i p$, the total number of distinct building-chip pairs is at least $(1/2)H(\overline{\gamma}p - 1)\sum_i^H(\gamma_i p) = (1/2)H^2\overline{\gamma}p(\overline{\gamma}p - 1)$. The factor $(1/2)$ is to account the counting of the building-chip pairs twice. ■

Theorem 2: Denote $P_{\text{dis}}^{\text{bc}}$ as the probability of any two randomly selected building-chips from the building-chip pool to be distinct. Then $P_{\text{dis}}^{\text{bc}} \geq \overline{\gamma}^2 - (\overline{\gamma}/p)$.

Proof: The total number of ways of selecting any two building-chips out of the entire building-chip pool with size pH is $\binom{pH}{2}$ which is a combination number. Therefore, by using Lemma 3, we have

$$\begin{aligned} P_{\text{dis}}^{\text{bc}} &\geq \frac{1}{2}\frac{pH^2\overline{\gamma}(\overline{\gamma}p-1)}{\binom{pH}{2}} \\ &= \frac{\overline{\gamma}H(\overline{\gamma}p-1)}{pH-1} \\ &= \frac{\overline{\gamma}^2 - \frac{\overline{\gamma}}{p}}{1-\frac{1}{pH}}. \end{aligned} \tag{2}$$

Since $(1/pH) > 0$, (2) reduces to

$$P_{\text{dis}}^{\text{bc}} \geq \overline{\gamma}^2 - \frac{\overline{\gamma}}{p}. \tag{3}$$

■

Corollary 1: Let the probability that any two SD-PUFs are identical be $P_{\text{iden}}^{\text{sdpuf}}$. Then $P_{\text{iden}}^{\text{sdpuf}} \leq (1 - \overline{\gamma}^2 + (\overline{\gamma}/p))^w$.

Proof: The probability that any two randomly selected building-chips are identical is $P_{\text{iden}}^{\text{bc}} = 1 - P_{\text{dis}}^{\text{bc}} \leq 1 - \overline{\gamma}^2 + (\overline{\gamma}/p)$. For any two SD-PUFs to be identical, each corresponding building-chip pair need to be identical. So the probability that two SD-PUFs are identical is given by

$$P_{\text{iden}}^{\text{sdpuf}} = \left(P_{\text{iden}}^{\text{bc}}\right)^w \leq \left(1 - \overline{\gamma}^2 + \frac{\overline{\gamma}}{p}\right)^w. \tag{4}$$

■

Note that $\overline{\gamma}$ is the mean value over all batches, hence it is independent of stdv or any individual batch's γ value, making above derivation a robust probability bound across the lithography process. A typical sized mask can produce building-chips ranging from 10^3 to 10^5 per batch depending on the mask/building-chip size. Hence the above bound can be approximated as $P_{\text{iden}}^{\text{sdpuf}} \leq (1 - \overline{\gamma}^2)^w$. In practical scenarios, γ is extremely close to 1 as per the inter-HD demonstrations in Section VII, where almost ideal inter-HD is observed at high connectivity ratio. Even if we take $\overline{\gamma} = 0.9$, and let $w = 30$, we still have

$$P_{\text{iden}}^{\text{sdpuf}} < 2.3 \times 10^{-22}. \tag{5}$$

This ensures an extremely low probability of existence of identical SD-PUF pair. Let us examine such probability bound for D-PUFs. Note that a building-chip is effectively a D-PUF, therefore, each mask can produce up to $(1-\gamma)p$ duplicated D-PUFs in the worst scenario. To maintain a high level inter-HD, every individual γ_i is required to be extremely close to 1.0, rather than the mean value of $\overline{\gamma}$ with much relaxed requirements. We denote the probability of the existence of identical D-PUF pair be $P_{\text{iden}}^{\text{dpuf}}$. We then have below probability

$$\begin{aligned} P_{\text{iden}}^{\text{dpuf}} &= \frac{\binom{(1-\gamma)p}{2}}{\binom{p}{2}} = \frac{(1-\gamma)p((1-\gamma)p-1)}{p(p-1)} \\ &= 1 - \gamma - \frac{\gamma(1-\gamma)}{1-\frac{1}{p}}. \end{aligned} \tag{6}$$

Substituting $\gamma = 0.9$, and $p = 10^5$, we have $P_{\text{iden}}^{\text{dpuf}} = 9.9 \times 10^{-3}$. Compared to the SD-PUFs, this probability is 10^{19} higher. To achieve the same probability of SD-PUFs, i.e., $P_{\text{iden}}^{\text{sdpuf}} = 2.3 \times 10^{-22}$, γ has to be 0.99999 or higher. Therefore, the shuffle-splice mechanism lifts the strict requirements previously enforced onto γ by taking advantages of the significantly large permutation space.

We take one step further to examine what would be the maximum number of SD-PUFs produced by one single mask so that they are still statistically ensured to be mutually unique. If we denote the total number of SD-PUFs by J, there will be $\binom{J}{2}$ possible pairs of SD-PUFs. With the probability bound from Corollary 1, the expectation of the number of identical pairs of SD-PUFs is $\binom{J}{2} \cdot P_{\text{iden}}^{\text{dpuf}}$. We denote the number of identical pairs as K_{iden}.

Using Markov's inequality, the probability that $K_{\text{iden}} \geq 1$, i.e., there exists at least one pair of SD-PUFs being identical, is given by

$$\Pr[K_{\text{iden}} \geq 1] \leq \frac{\mathrm{E}[K_{\text{iden}}]}{1} = \binom{J}{2}\left(1 - \overline{\gamma}^2 + \frac{\overline{\gamma}}{p}\right)^w. \quad (7)$$

To achieve $\Pr[K_{\text{iden}} \geq 1]$ to be extremely small, e.g., 10^{-7}, and substituting $\gamma = 0.9$, $w = 30$, and $p = 10^5$, we get

$$J < 2.93 \times 10^7. \quad (8)$$

This demonstrates the proposed methodology is very effective and efficient: the same mask can be used to produce sufficiently large number of spliced SD-PUF while statistically ensuring not any pair sharing the same connectivity configurations, even with the very worst case scenario analysis. Compared to D-PUF, where each mask maximally produces 10^5 PUF chips, the SD-PUF lowers the mask cost per PUF chip by 2.93×10^2. We will further verify the SD-PUF security performances in Section VII with simulation results.

C. *Optimal Splicing*

There is still a pending question yet to be answered: for a given number w of building-chips, what would be the optimal values of r, c (see Fig. 7) to maximize the security performance of an SD-PUF? Based on relation $w = r \times c$, for a fixed value of w, increasing the size in 1-D will decrease the other. For one extreme case, where $r = w$, we have the widest bitwidth SD-PUF, however, since the column size of SD-PUF is small, some attractive properties discussed in Section V that are relying on the row couplings will be minimized, hence less resilient to learning-based attacks. On the other hand, if $c = w$, we get the narrowest bitwidth SD-PUF, which obviously limits CRP space, hence also less optimal. Eventually, there should exist a tradeoff between r, c values to achieve optimal security balancing CRP space and resilience to learning attacks. This matter will be further demonstrated in Section VII.

VII. EVALUATIONS OF D-PUF AND SD-PUF

In this section, we first evaluate the silicon cost of a 64×64 D-PUF chip, followed by chip-level interconnection lithography simulations. Afterwards, we focus on evaluating performances of the D-PUFs and SD-PUFs, including statistical performance and resilience to adverse attacks. Evaluations on D-PUFs numerically reveal the intrinsic physical and logical attributes of the CMOS latch structure as well as the proposed XOR network architecture. We study D-PUFs with two configurations: 1) 8-row by 8-column (8×8) and 2) 64-row by 64-column (64×64), where the 8×8 D-PUF is used in order to illustrate a full statistical picture over the entire 256 CRPs. The 64×64 D-PUF, by contrast, represents a more practical PUF implementation, but due to the huge CRP space, we only evaluate a CRP subset. Afterward, we evaluate statistical performances and adverse attacks on SD-PUFs with sizes of 16×16 and 64×64. In the last, we further demonstrate the tradeoff between parameter r, c for optimal learning resilience.

A. *Silicon Cost and Lithography Simulation*

We conduct a custom design flow for a 64×64 D-PUF using NanGate 45-nm open cell library. Fig. 10(a) shows the final layout, where the overall silicon area is 145.92×89.32 μm. For better readability, in Fig. 10, we show only metal-1 (M1) layer for cell placement and metal-4 (M4) layer for the virtual connections. In the zoomed-in view of Fig. 10(b), the orange color metal pieces are the virtual connections, i.e., the dashed line in Fig. 6(a). In this design, we restrict M4 metal layer *dedicated* for all virtual connections. Any other conventional interconnections are not allowed in M4 layer. Therefore, the mask error applied onto M4 layer does not affect functionalities of the M1 layer transistors or other metal layer connections at all. Note that each M4 layer virtual connection consists of a pair of stripes with a small split distance[6] of 46 nm in the middle which is the same value as the selected mean split distance in Fig. 2.

We further verify the actual shape of M4 virtual connections and their connectivity status after getting mapped onto a wafer. On top of that, it is also important to examine if any neighboring M4 virtual connections interfere with each other causing unexpected short-circuit. We adopted the similar mask error settings described in Section II, where Gaussian distribution with mean split distance of 46 nm and stdv of 5 nm is used. The evaluation is conducted through the lithography simulator [27]. However, the original simulator was designed to handle only limited mask size per simulation. To get a more realistic lithography image of the full-size mask, we adopted a window-scanning mechanism in [29] to enhance the simulator. The window-scanning mechanism is to scan the full-size mask by using a squared window with size equal to the maximum mask size that the simulator can handle per simulation. There are two steps involved: 1) scanning and 2) reconstruction. First, for each scan, the simulated image will be cropped to exclude unreliable margin areas due to edging effects, resulting to a smaller but accurate lithography image piece. Then, the final full-size simulation image is reconstructed by combining each cropped image piece based on its geometric location. Note that, the scanning interval is configured to be equal to the cropping margin, so the final reconstructed full-size image is reliable and accurate. Fig. 11 shows a snapshot of the simulated lithography image for M4 layer mask of the 64×64 D-PUF. The mask dimensions match that of Fig. 10. Out of the 20 M4 virtual connections in the snapshot, three of them eventually disconnect, resulting to a local connectivity of 85%.

[6]Since this is a custom design flow, we ignore any DRC violations that would be reported by conventional physical design flow due to the split.

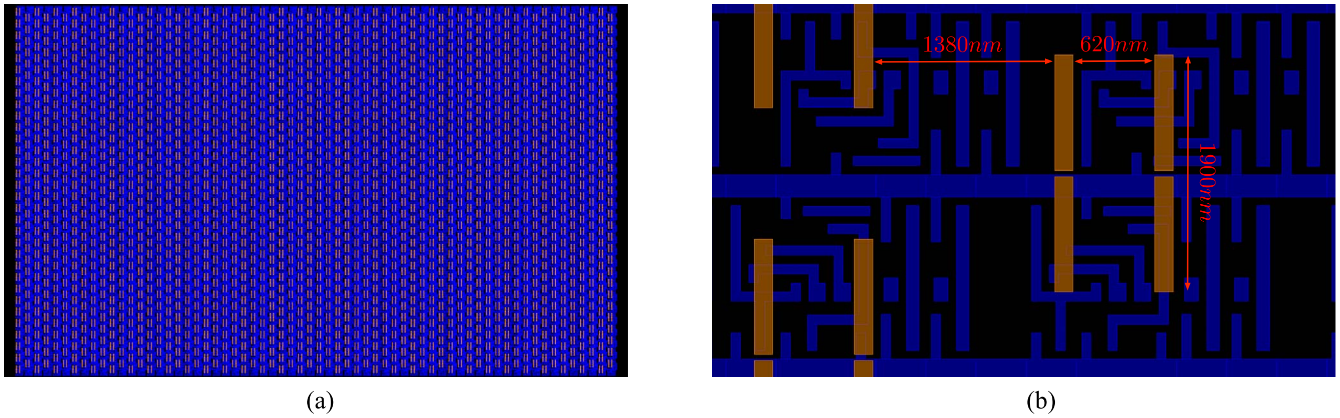


Fig. 10. 64 × 64 D-PUF layout with M1 layer standard cell placement and M4 layer virtual connections. (a) Full chip dimension is 145.92 × 89.32 μm. (b) Zoomed-in view: M1 cell shapes are in blue color and M4 interconnections are in orange color.

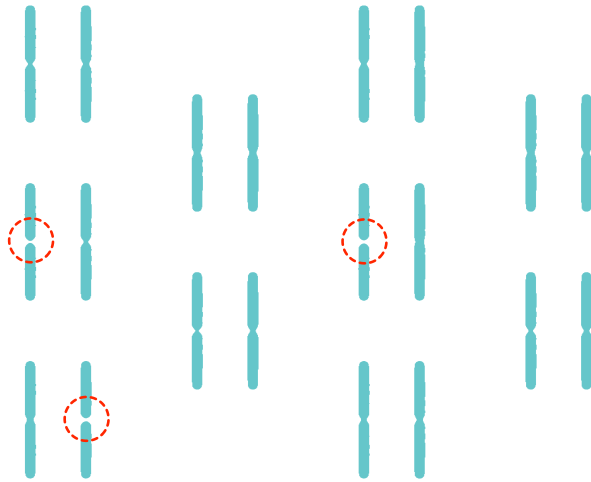


Fig. 11. Snapshot of full chip lithography simulation on M4 layer for Fig. 10. Disconnection status is highlighted by red circles.

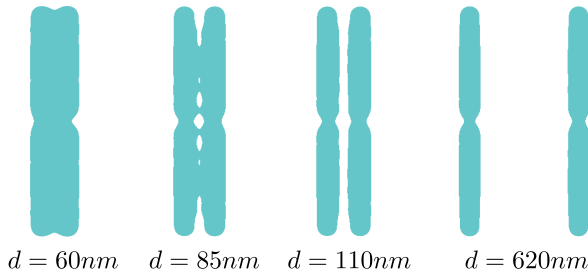


Fig. 12. Horizontal direction lithography interference over various interval distance (measured in mask). There is no more interference when the interval distance is at least 110 nm.

We highlighted those disconnected virtual connections by red circles.

Lastly, we study the potential interferences between any two neighboring M4 interconnections in horizontal direction. Fig. 12 shows four pairs of neighboring M4 interconnections with various interval distance (measured in mask) of 60, 85, 110, and 620 nm, where 620 nm is the minimum *horizontal* distance of any two M4 interconnections shown in Fig. 10. It can be seen that, there exist strong interferences when the interval distance is smaller than 110 nm. In other words, from lithographic perspective, the interval distance of 620 nm in Fig. 10(b) is far beyond the interference threshold, hence further confirms the mask error only impact the intended virtual connections.

B. D-PUF Statistical Evaluations

There are four commonly used metrics for evaluating the statistical performance of a PUF [30]: 1) inter-HD; 2) intra-HD; 3) uniformity; and 4) bit-aliasing. Inter-HD represents the ability of a PUF to uniquely distinguish two chips under the same challenge. Intra-HD captures how reliable of a particular PUF under operational and environmental variations. Uniformity checks how uniform the ratio of 1s and 0s is in the response bits of a PUF. Finally the bit-aliasing captures whether any response bit is biased and showing nearly identical result across different chips.

We first simulate the simple 8 × 8 D-PUF with HSPICE using 45-nm-PTM SPICE model. The skewed latch is designed to have 10× wider pMOS width for skewed-1 inverter, and 4× wider nMOS width for skewed-0 inverter. Besides, pMOS in skewed-1 inverter uses low VT pMOS and high VT nMOS transistors, and vice versa for skewed-0 inverter. We specify the connectivity rate of two cases, 0.2 and 0.9, in order to reveal the impact of connectivity on the performance of D-PUF. When the virtual connection pin of the latch is disconnected, we set high impedance to its input. For intra-HD simulation, we sweep temperatures from −20 °C to 100 °C and voltages from 0.7 to 1.2 V. All 256 unique CRPs are used for the simulation. We show results in Table I. Clearly, the intra-HD is 0 across wide environmental and operational conditions, due to the nature of digitalized system as well as the use of strongly skewed latch. Higher connectivity rate shows better statistical performance which agrees to the earlier discussions in Section V.

We further examine the performance of 64 × 64 case. However, it is impractical to simulate a 64 × 64 D-PUF by HSPICE due to the unacceptable simulation runtime. We therefore developed a behavioral emulator. Note that the only difference between the HSPICE simulation and the behavioral emulation is the emulator cannot catch the intra-HD metric, whereas all the rest statistical metrics can be fully emulated due to the digital nature. Considering the intra-HD for 8 × 8 D-PUF shown to be 0 by SPICE simulation, which is regardless of the network size, the intra-HD can be safely extrapolated to be 0 for 64 × 64 case as well.

TABLE I
STATISTICAL EVALUATION ON 8×8 D-PUF WITH 256 CRPS

Type (Ideal Value)	conn. rate = 0.2		conn. rate = 0.9	
	Mean	Stdv.	Mean	Stdv.
Inter HD (0.5)	0.4188	0.0302	0.4943	0.0061
Intra HD (0.0)	0	0	0	0
Bit Alias (0.5)	0.5000	0.2067	0.5000	0.0730
Uniformity (0.5)	0.5000	0.1768	0.5000	0.1678

TABLE II
STATISTICAL EVALUATION ON 64×64 D-PUF WITH 100K CRPS

Type (Ideal Value)	conn. rate = 0.2		conn. rate = 0.9	
	Mean	Stdv.	Mean	Stdv.
Inter HD (0.5)	0.4999	0.0009	0.5000	0.0009
Intra HD (0.0)	0	0	0	0
Bit Alias (0.5)	0.5000	0.0504	0.5000	0.0499
Uniformity (0.5)	0.5000	0.0625	0.5000	0.0624

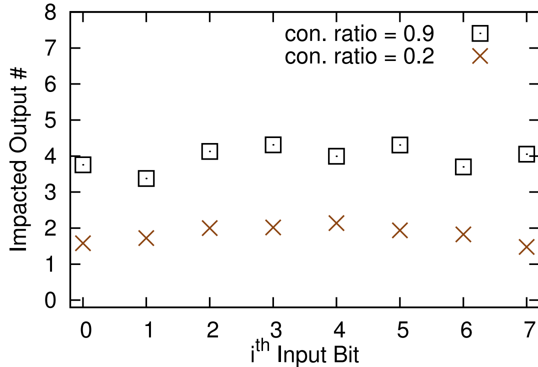


Fig. 13. Avalanche effect of 8×8 D-PUF over each input.

The results are shown in Table II. One hundred thousand randomly selected unique CRPs are tested. For 64×64 D-PUF, even small connectivity rate produces close to ideal inter-HD as well as bit-alias and uniformity, showing a outstanding statistical performance.

In addition, we examine the "avalanche effect" of D-PUF, i.e., Property 4. The avalanche effect checks that how many outputs will be affected (flipped) by changing an individual input bit. This property needs to be examined over the exhaustive CRP space, hence the 8×8 D-PUF is used. We simulate 1000 unique 8×8 D-PUF chips over the complete 256 CRPs. For each input bit flip, we exhaustively collect the number of flipped outputs over all the 2^7 cases. The average number of output flips for each input is plotted in Fig. 13. For connectivity rate of 0.9, about four outputs, i.e., half of the eight outputs, are flipped due to this single input change. Hence the adversary prediction via one bit change at a time is no better than a simple random guess. In other words, the D-PUF provides the theoretically best potential of anti-prediction. In addition, when connectivity rate is 0.2, the number of impacted outputs is reduced to 2. This further supports the conclusion that a higher connectivity rate leads to better avalanche effect.

C. D-PUF Adversary Attacks

Next we evaluate the resilience to various learning-based reverse engineering attacks. We also verify that the connectivity rate and the D-PUF column size will affect the resilience, i.e., Properties 1 and 3. We attack the D-PUF by support vector machine (SVM), where the nonlinear radial bias function kernel is used. Again, we first check the 8-row D-PUF chip family with different column sizes of 8, 16, 32, and 128. The 256 CRPs are divided into training and testing sets. Considering that the training set size may affect the overall prediction accuracy, we sweep the training size from 10% to 90% of the 256 CRPs, and test with the rest CRPs for each case. The ideal prediction error should be exactly 50%, indicating the machine learning prediction is no better than just random guess. Prediction error of 0% means the PUF can be completely predicted. For each D-PUF, we apply the SVM attack onto *each* output bit at a time, and we report only the best prediction error among all output bits. It can be seen in Fig. 14(a) and (b) that, large size of columns helps to increase the prediction error. Besides, D-PUFs with connectivity rate of 0.9 in Fig. 14(b) generally show stronger resilience than that of 0.2 in Fig. 14(a). Combining these two factors together, in Fig. 14(b), we can see the attacks on the D-PUF of 8-row 128-column show constantly bad predictions (about 40% prediction error) even when training set size is 90%.

Further, we apply SVM attack onto 64×64 D-PUF in Fig. 14(c). We randomly sampled 100K CRPs and divide them into various sizes of training and testing sets as well. For connectivity rate of 0.9 and 0.2, the D-PUF constantly shows close to ideal resilience to the attack. Even when connectivity rate is reduced to 0.1, the prediction error is still above 35%, and until the connectivity rate drops to 0.01, we start to observe low prediction errors.

Ultimately, some additional state-of-the-art learning models, including artificial neural network (ANN) and random forest (RF), are also applied to attack the 64×64 D-PUF in Fig. 14(d), where the ANN model is configured with ten hidden layers and sigmoid functions, and the RF model consists of 15 random trees. The prediction error for both models, however, is constantly around 50% across wide range of connectivity rate and training set size. Overall, the proposed D-PUF exhibits extraordinary resilience to learning-based reverse engineering attacks.

D. SD-PUF Evaluations

This section further evaluates the security performance of SD-PUFs including statistical metrics and learning resilience. We first evaluate a 16×16 SD-PUF, which we can exhaustively examine all the CRP spaces. We set $w = 16$, $r = 4$, and $c = 4$, and each building-chip has size of 4×4. Based on Table II, where the intra-HD is almost 0.5, γ can be very close to 1. Let $\gamma = 0.9$, by (7), then the max number of J can be 37K in this example. For the sake of runtime, we sample 1K SD-PUF chips and generate the statistics in Table III. For connectivity rate of 0.2, we see slight suboptimality in inter-HD. When the connectivity rate increases to 0.9, all statistic metrics show close to ideal values.

We further evaluate a more practically sized spliced SD-PUF of 64×64 with 100K CRPs. The building-chip is with size of 8×8, and $w = 64$, $r = c = 8$. Again, if $\gamma = 0.9$, the max J can be up to 1.46×10^{23} without expecting any single pair to have the same connectivity. In Table IV, for both connectivities of 0.2 and 0.9, the spliced SD-PUF achieves close to ideal values on almost all statistic metrics.

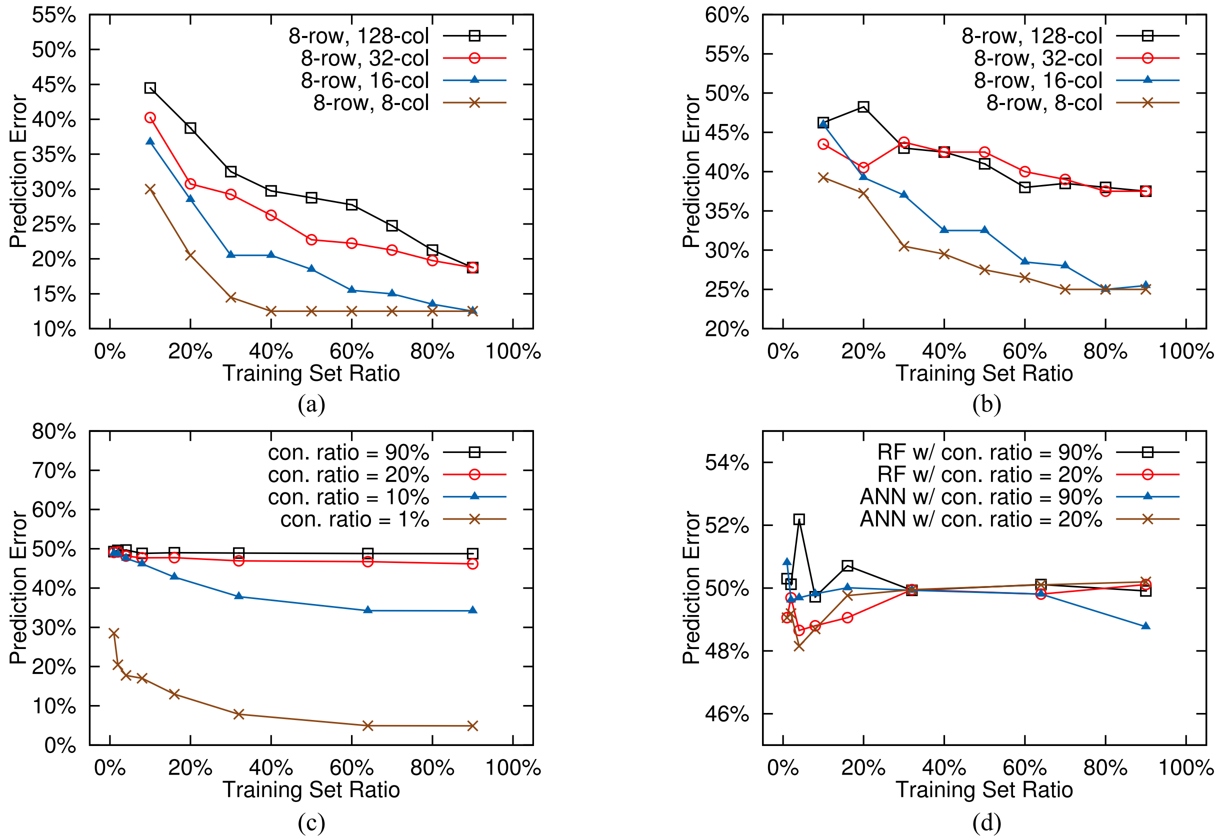


Fig. 14. Various machine learning attacks for D-PUFs over different configurations. (a) SVM attacking on 8-row D-PUFs with connectivity rate of 0.2 over different column sizes and training sizes; (b) SVM attacking on 8-row D-PUFs with connectivity rate of 0.9 over different column sizes and training sizes; (c) SVM attacking on a 64×64 D-PUF over different connectivity rate and training size. (d) Other machine learning attacks on a 64×64 D-PUF, including: 1) ANN with ten hidden layers using sigmoid function and 2) RF with 15 trees in the forest.

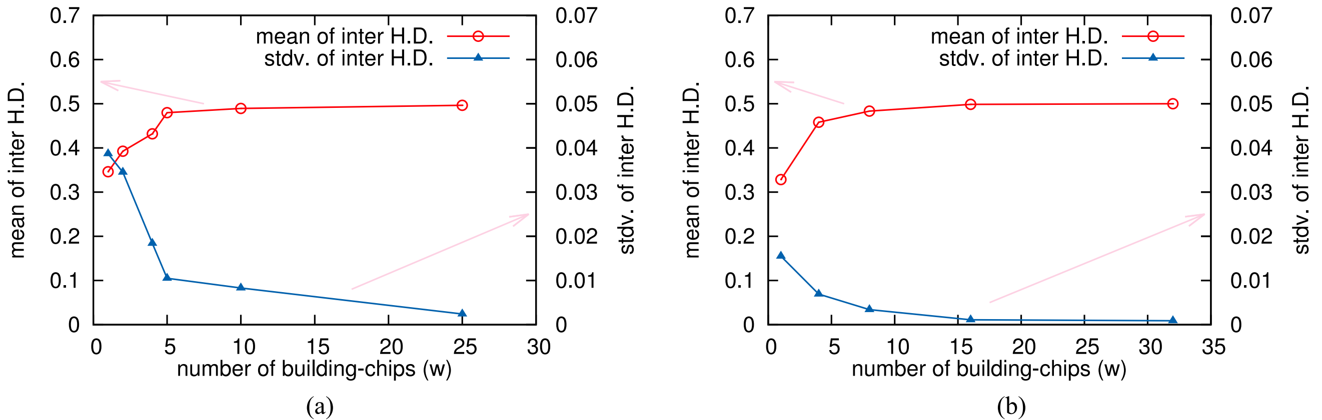


Fig. 15. Various numbers of building-chips used for SD-PUFs and their inter-HDs. Conservatively, we set $\gamma = 0.5$ for each mask. Note that this is only for estimating the impact of the number of building-chips used for splicing. (a) 10×10 SD-PUFs. (b) 64×64 SD-PUFs.

We now demonstrate how the parameter w affect the inter-HD of an SD-PUF. In Fig. 15, we examine SD-PUF of 10×10 and 64×64 with different w values. Being conservative, let $J = 10^3$ and $\gamma = 0.5$, we see the inter-HD is monotonically improving mean and stdv along with the increment of w value. However, it is neither wise to choose an overly large w as this may increase the cost of the SD-PUF. In this experimental setup, $w = 20$ can be sufficiently good.

Lastly, we evaluate the learning resilience of the spliced SD-PUFs. As mentioned in Section VI-C, there exists security tradeoffs between parameters of r, c, hence impacting the resilience to learning attacks. In Fig. 16, we evaluate SD-PUFs of 64×64 with fixed value $w = 16$ but varying r, c values. Each building-chip has size of 4×4. r value is swept as 1, 2, 4, 8, and 16, making the corresponding SD-PUFs to be with sizes of 4×64, 8×32, 16×16, 32×8, and 64×4. Both connectivities of 0.2 and 0.9 are demonstrated. From the curves in Fig. 16, 4×64 SD-PUF shows the worst resilience to learning attacks, due to the smallest CRP space size. 64×4 SD-PUF, although showing better resilience, the error prediction still gets down to 40% for both connectivity configurations. The optimal learning resilience configuration

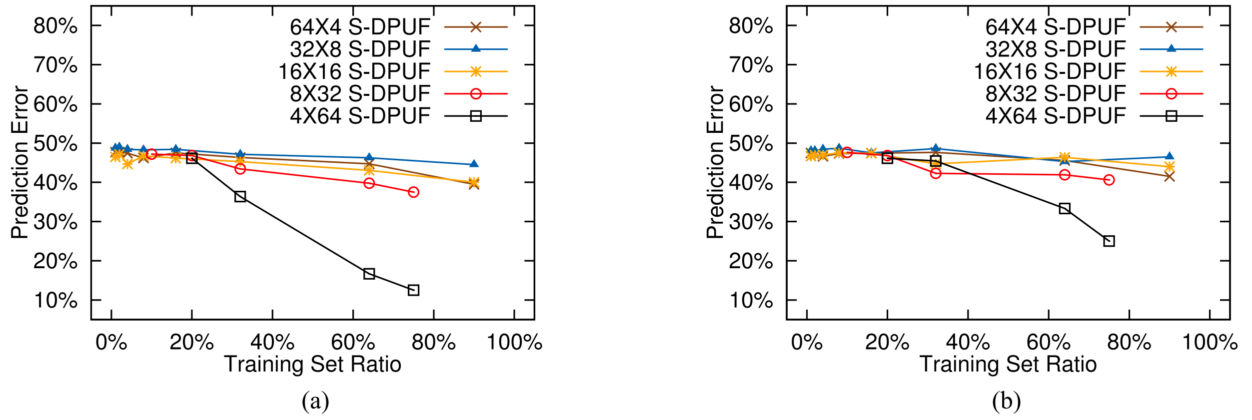


Fig. 16. Machine learning (SVM) attacks on various configurations of 16 pieces of 4×4 building-chips. We sweep the SD-PUFs with sizes of 4×64, 8×32, 16×16, 32×8, and 64×4. (a) Connectivity rate = 0.2. (b) Connectivity rate = 0.9.

TABLE III
STATISTICAL EVALUATION ON 16×16 SD-PUF WITH 60K CRPS USING 16 BUILDING-CHIPS OF SIZE 4×4. THOSE 16 BUILDING-CHIPS ARE SPLICED AS 4×4, I.E., $r = c = 4$

Type (Ideal Value)	conn. rate = 0.2		conn. rate = 0.9	
	Mean	Stdv.	Mean	Stdv.
Inter HD (0.5)	0.4529	0.0083	0.5000	0.0004
Intra HD (0.0)	0	0	0	0
Bit Alias (0.5)	0.5016	0.1249	0.4999	0.1251
Uniformity (0.5)	0.5016	0.1550	0.4999	0.0228

TABLE IV
STATISTICAL EVALUATION ON 64×64 SD-PUF WITH 100K CRPS USING 64 BUILDING-CHIPS OF SIZE 8×8. THE BUILDING-CHIPS ARE SPLICED AS 8×8, I.E., $r = c = 8$

Type (Ideal Value)	conn. rate = 0.2		conn. rate = 0.9	
	Mean	Stdv.	Mean	Stdv.
Inter HD (0.5)	0.4998	0.0001	0.5000	0.0001
Intra HD (0.0)	0	0	0	0
Bit Alias (0.5)	0.4999	0.0624	0.5001	0.0625
Uniformity (0.5)	0.4999	0.0183	0.5001	0.0159

appears to be 32×8 which hints a balanced CRP space size and the learning resilience attribute.

VIII. CONCLUSION

In this paper, we propose highly nonlinear and secure digital circuit PUFs including D-PUF and SD-PUF. The randomness of the D-PUF comes from the lithography process variations and is reflected in the form of interconnect randomness. Strongly skewed latches are used to make the interconnect randomness compatible with digital CMOS circuit system ensuring 0 intra-HD. A novel highly nonlinear logic architecture is developed to effectively spread and augment any interconnect randomness throughout the logic network. On top of that, a novel post-silicon shuffling process is applied onto D-PUFs which are later spliced to be SD-PUFs to significantly strengthen the security performance, while reducing mask cost per PUF device. The proposed PUFs have been demonstrated with outstanding statistical performance as well as strong resilience to various state-of-the-art machine learning attacks.

ACKNOWLEDGMENT

The authors would like to thank J. Kuang from Cadence for providing guidance on lithography simulation.

REFERENCES

[1] I. Verbauwhede and R. Maes, "Physically unclonable functions: Manufacturing variability as an unclonable device identifier," in *Proc. ACM Great Lakes Symp. VLSI (GLSVLSI)*, Lausanne, Switzerland, 2011, pp. 455–460.

[2] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proc. IEEE*, vol. 102, no. 8, pp. 1126–1141, Aug. 2014.

[3] M. Rostami, J. B. Wendt, M. Potkonjak, and F. Koushanfar, "Quo vadis, PUF?: Trends and challenges of emerging physical-disorder based security," in *Proc. IEEE/ACM Design Autom. Test Europe (DATE)*, Dresden, Germany, 2014, pp. 1–6.

[4] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Testing techniques for hardware security," in *Proc. IEEE Int. Test Conf. (ITC)*, Santa Clara, CA, USA, 2008, pp. 1–10.

[5] M. Li, J. Miao, K. Zhong, and D. Z. Pan, "Practical public PUF enabled by solving max-flow problem on chip," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, Austin, TX, USA, 2016, pp. 1–6.

[6] D. E. Holcomb, W. P. Burleson, and K. Fu, "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Trans. Comput.*, vol. 58, no. 9, pp. 1198–1210, Sep. 2009.

[7] M. Cortez, A. Dargar, S. Hamdioui, and G.-J. Schrijen, "Modeling SRAM start-up behavior for physical unclonable functions," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst. (DFT)*, Austin, TX, USA, 2012, pp. 1–6.

[8] Y. Zheng, M. S. Hashemian, and S. Bhunia, "RESP: A robust physical unclonable function retrofitted into embedded SRAM array," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, Austin, TX, USA, 2013, pp. 1–9.

[9] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, "Silicon physical random functions," in *Proc. ACM Conf. Comput. Commun. Security (CCS)*, Washington, DC, USA, 2002, pp. 148–160.

[10] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, San Diego, CA, USA, 2007, pp. 9–14.

[11] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "The butterfly PUF protecting IP on every FPGA," in *Proc. IEEE Int. Workshop Hardw. Orient. Security Trust (HOST)*, Anaheim, CA, USA, 2008, pp. 67–70.

[12] C.-E. Yin and G. Qu, "Temperature-aware cooperative ring oscillator PUF," in *Proc. IEEE Int. Workshop Hardw. Orient. Security Trust (HOST)*, San Francisco, CA, USA, 2009, pp. 36–42.

[13] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. EUROCRYPT*, 2004, pp. 523–540.

[14] R. Maes, A. Van Herrewege, and I. Verbauwhede, "PUFKY: A fully functional PUF-based cryptographic key generator," in *Proc. Workshop Cryptograph. Hardw. Embedded Syst. (CHES)*, 2012, pp. 302–319.

[15] U. Rührmair *et al.*, "PUF modeling attacks on simulated and silicon data," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 11, pp. 1876–1891, Nov. 2013.

[16] X. Xu and W. Burleson, "Hybrid side-channel/machine-learning attacks on PUFs: A new threat?" in *Proc. IEEE/ACM Design Autom. Test Europe (DATE)*, Dresden, Germany, 2014, pp. 1–6.

[17] T. Xu and M. Potkonjak, "Robust and flexible FPGA-based digital PUF," in *Proc. IEEE Int. Conf. Field Program. Logic Appl. (FPL)*, Munich, Germany, 2014, pp. 1–6.

[18] T. Xu and M. Potkonjak, "Digital PUF using intentional faults," in *Proc. IEEE Int. Symp. Qual. Electron. Design (ISQED)*, Santa Clara, CA, USA, 2015, pp. 448–451.

[19] T. Hegedüs and N. Megiddo, "On the geometric separability of Boolean functions," *Discrete Appl. Math.*, vol. 66, no. 3, pp. 205–218, 1996.

[20] J. Miao, M. Li, S. Roy, and B. Yu, "LRR-DPUF: Learning resilient and reliable digital physical unclonable function," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Austin, TX, USA, 2016, pp. 1–8.

[21] R. Kumar and W. Burleson, "Litho-aware and low power design of a secure current-based physically unclonable function," in *Proc. IEEE Int. Symp. Low Power Electron. Design (ISLPED)*, Beijing, China, 2013, pp. 402–407.

[22] A. Sreedhar and S. Kundu, "Physically unclonable functions for embeded security based on lithographic variation," in *Proc. IEEE/ACM Design Autom. Test Europe (DATE)*, Grenoble, France, 2011, pp. 1–6.

[23] D. Forte and A. Srivastava, "On improving the uniqueness of silicon-based physically unclonable functions via optical proximity correction," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 2012, pp. 96–105.

[24] A. K. Wong, R. A. Ferguson, and S. M. Mansfield, "The mask error factor in optical lithography," *IEEE Trans. Semicond. Manuf.*, vol. 13, no. 2, pp. 235–242, May 2000.

[25] V. Axelrad, M. Smayling, K. Tsujita, K. Mikami, and H. Yaegashi, "OPC-lite for gridded designs at low k1," in *Proc. SPIE Photomask Technol.*, vol. 9235. 2014, p. 8.

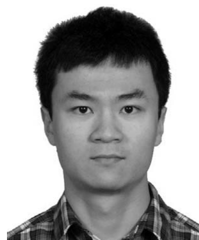
[26] W.-H. Cheng and J. Farnsworth, "Fundamental limit of ebeam lithography," in *Proc. SPIE*, vol. 6607. 2007, Art. no. 660724.

[27] S. Banerjee, Z. Li, and S. R. Nassif, "ICCAD-2013 CAD contest in mask optimization and benchmark suite," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, San Jose, CA, USA, 2013, pp. 271–274.

[28] Y. Granik and N. B. Cobb, "MEEF as a matrix," in *Proc. Photomask*, 2002, pp. 980–991.

[29] Y. Ma, J.-R. Gao, J. Kuang, J. Miao, and B. Yu, "A unified framework for simultaneous layout decomposition and mask optimization," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2017.

[30] A. Maiti, V. Gunreddy, and P. Schaumont, "A systematic method to evaluate and compare the performance of physical unclonable functions," in *Embedded Systems Design With FPGAs*. New York, NY, USA: Springer, 2013, pp. 245–267.



Jin Miao received the B.S. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2010, and the Ph.D. degree in electrical and computer engineering from the University of Texas at Austin, Austin, TX, USA, in 2014.

He is currently a Principle Software Engineer with Cadence Design Systems, San Jose, CA, USA. His current research interests include emerging science and technologies, covering approximate computing, hardware security, and machine learning.

Dr. Miao has been serving as a Reviewer or a TPC Member for a number of journals or conferences, including the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, the Design Automation Conference, the Asia and South Pacific Design Automation Conference, the IEEE International New Circuit and System Conference, etc.

Meng Li received the B.S. degree in microelectronics from Peking University, Beijing, China, in 2013. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Texas at Austin (UT Austin), Austin, TX, USA, under the supervision of Prof. D. Z. Pan.

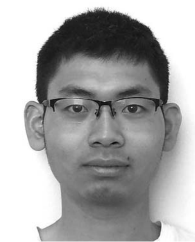
His current research interests include hardware-oriented security, reliability, power grid simulation acceleration, and deep learning.

Mr. Li was a recipient of the Best Paper Award in IEEE International Symposium on Hardware Oriented Security and Trust 2017, and the Graduate Fellowship from UT Austin, in 2013.

Subhendu Roy (S'13–M'16) received the B.E. degree in electronics and telecommunication engineering from Jadavpur University, Kolkata, India, in 2006, the M.Tech. degree in electronic systems from the Indian Institute of Technology Bombay, Mumbai, India, in 2009, and the Ph.D. degree in electrical and computer engineering from the University of Texas at Austin, Austin, TX, USA, in 2015.

He is currently a Principal Software Engineer with Cadence Design Systems, San Jose, CA, USA. He has three years of full-time industry experience in an EDA company, Atrenta, Noida, India (currently acquired by Synopsys), where he was involved in developing tools in the architectural power domain and register-transfer level domain. He was a summer Intern with IBM T. J. Watson Research Center, Yorktown Heights, NY, USA, in 2012, and Mentor Graphics, Fremont, CA, USA, in 2013 and 2014. He holds one patent and has first-authored papers in major EDA conferences/journals, such as the Design Automation Conference, the International Symposium on Physical Design (ISPD), the Asia and South Pacific Design Automation Conference, and the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. His current research interests include design automation for logic synthesis, physical design, and cross-layer optimizations.

Dr. Roy was a recipient of the Best Paper Award at ISPD'14.



Yuzhe Ma received the B.E. degree from the Department of Microelectronics, Sun Yat-sen University, Guangzhou, China, in 2016. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong.

His current research interests include very large-scale integration design for manufacturing, physical design, and machine learning on chips.

Bei Yu (S'11–M'14) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA, in 2014.

He is currently an Assistant Professor with the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong.

Dr. Yu was a recipient of four best paper awards at the 2017 International Symposium on Physical Design, the 2016 SPIE Advanced Lithography Conference, the 2013 International Conference on Computer Aided Design, and the 2012 Asia and South Pacific Design Automation Conference, and three International Conference on Computer-Aided Design contest awards in 2012, 2013, and 2015. He has served in the Editorial Board of *Integration*, the *VLSI Journal* and *IET Cyber-Physical Systems: Theory and Applications*.