

In the format provided by the authors and unedited.

Hardware-intrinsic security primitives enabled by analogue state and nonlinear conductance variations in integrated memristors

Hussein Nili^{1*}, Gina C. Adam ^{1,2}, Brian Hoskins¹, Mirko Prezioso¹, Jeeseon Kim³, M. Reza Mahmoodi¹, Farnood Merrikh Bayat¹, Omid Kavehei ^{3*} and Dmitri B. Strukov^{1*}

¹University of California Santa Barbara, Santa Barbara, CA, USA. ²National Institute for R&D in Microtechnologies, Bucharest, Romania. ³Royal Melbourne Institute of Technology University, Melbourne, Victoria, Australia. *e-mail: hnili@ece.ucsb.edu; omid.kavehei@rmit.edu.au; strukov@ece.ucsb.edu

Supplementary Information

1. Memristive crossbar fabrication and characterization

All fabrication was performed at UCSB’s nanofabrication facility (<https://www.nanotech.ucsb.edu/>). Two-layer monolithically integrated fully passive TiO_{2-x} memristor crossbar circuits with an active device area of $\sim 350 \times 350 \text{ nm}^2$ and with the middle metal lines shared between the top and bottom crossbars were fabricated using *in situ* low-temperature reactive sputtering deposition, DUV lithography, ion milling and a precise planarization step (Figure S1). The stoichiometry of the switching TiO_{2-x} layer was precisely controlled by optimizing the reactive DC sputtering parameters.¹ The Al_2O_3 barrier, the active TiO_{2-x} layer, and the TiN and Pt layers were deposited *in situ* in the sputtering chamber and patterned through Ar ion beam etching (IBE). To provide a lower electrode slope, the incident ion beam and the substrate were partially tilted (with initial and secondary substrate tilt angles of 0 and 40°). The bottom layer was then planarized with fast chemical mechanical polishing (CMP), utilizing an $\sim 750\text{-nm}$ SiO_2 sacrificial layer to achieve global planarization. The middle electrode was then partially exposed in a controlled fashion, and the remaining SiO_2 layer was removed in a CHF_3 atmosphere in an inductively coupled plasma (ICP) chamber. Finally, the top crossbar layer was deposited and patterned using a process similar to that used for the bottom layer. The top electrode was patterned to be a few nanometres wider than the other layers to ensure complete coverage of the exposed middle electrode.

The completed crossbar circuits were wire-bonded and mounted on a custom-printed circuit board controlled by Agilent measurement tools. All of the electrical testing was performed using an Agilent B1500A semiconductor device parameter analyser, an Agilent B1530A waveform generator/fast measurement unit, and a low-leakage Agilent E5250A switch matrix. The distribution of the ON and OFF resistances for all devices is shown in Figure S1c.

2. Algorithm for selecting optimal crosspoint conductances

The algorithm used to find the optimal crosspoint conductances is shown in Figure S2. It involves the following steps:

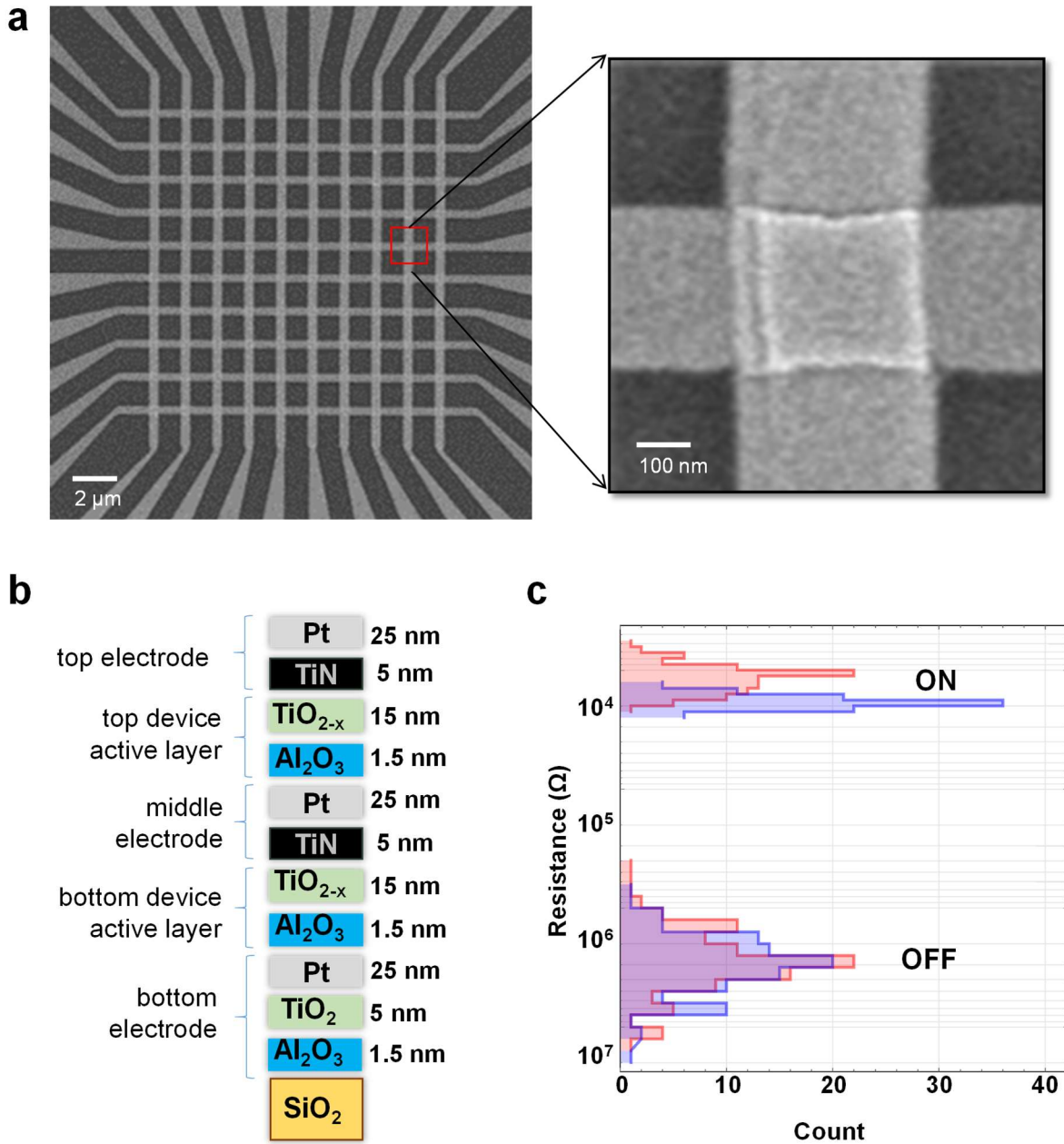
Step 1: The very first step is to generate S random sets of row and column selections; these are denoted S_R and S_C^\pm , respectively. Each selection comprises indexes of the selected 5 rows and 2 columns, and each set of indexes is unique. The typical value of S is 10,000. The values of μ_I , σ_I , and Δ_I , which are used in the next step, are initialized to the empirically found values of 5 μA , 0.5 μA , and 50 nA, respectively.

Step 2: In this step, S values of I^+ and I^- , i.e., pairs of desired currents for the selections, are randomly generated. For each selection, this is achieved by first randomly choosing (with 0.5 probability) which current of the differential pair will be directly generated and then sampling its value I from a Gaussian distribution with specific μ_I and σ_I . Next, the other value of a current pair is sampled from $I + \Delta_I + 4.5 [\mu\text{A}] \times \text{Beta}(2, 25)$, where $\text{Beta}()$ is a beta distribution of the first kind with shape parameters $\alpha = 2$ and $\beta = 25$.

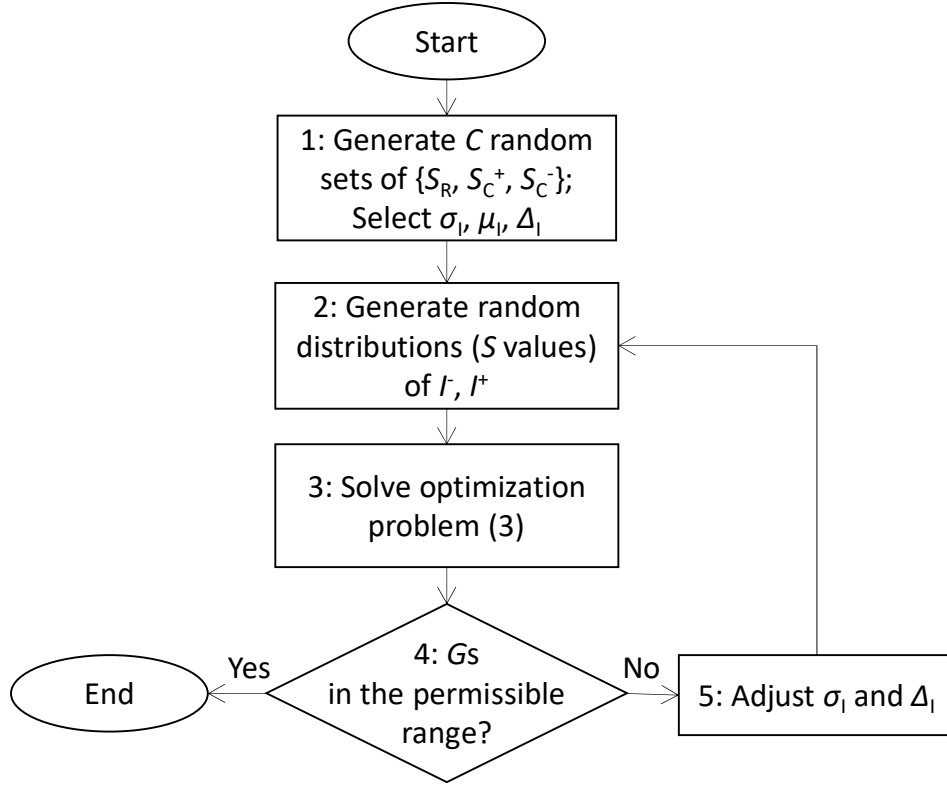
Step 3: The non-negative least squares optimization problem defined by Equation (3) in the main text is solved with the help of Matlab software.

Step 4: The conductances (G s) are checked to determine that they fall within the desired, highly nonlinear range, which is approximately $2.5 \mu\text{S}$ to $4.5 \mu\text{S}$ at 300 mV . If the condition is not satisfied, the algorithm proceeds to step 5.

Step 5: The σ_1 is adjusted manually, after which the steps for generating new distributions of desirable I^+ and I^- and solving the optimization problem are repeated. It should be noted that with optimal μ_1 , σ_1 , and Δ_1 , which are empirically found during fine-tuning of the algorithm, the adjustment step was rather rare in all experiments.



Supplementary Figure 1. (a) Top-view SEM image of the 3D ReRAM crossbar and (b) its device stack material layers and thicknesses. (c) Cumulative histogram for the top (blue) and bottom (red) devices' ON and OFF state resistances measured at 0.3 V .



Supplementary Figure 2. An algorithm for selecting crosspoint device conductances.

3. Security metrics for PUF primitives

The most common operational metrics in security primitives are based on Hamming weight and on the inter- and intra-instance Hamming distance among output vectors. Uniformity (UF) and diffuseness (DF) are used to assess the randomness of a single PUF instance. In particular, UF is a measure of a balance in the PUF response. The uniformity of the K -bit-long binary response (vector B) is simply defined as a normalized Hamming weight

$$UF(B_i) \equiv \frac{1}{K} \sum_{k=1}^K b_{ki}, \quad (S1a)$$

where b_{ki} is a k -th bit of the i -th response B_i . The average uniformity is

$$\langle UF \rangle \equiv \frac{1}{C} \sum_{i=1}^C UF(B_i), \quad (S1b)$$

where C is the total number of challenge-response pairs. The ideal value of UF is 0.5, which represents a perfect balance between the possible responses, i.e., the same number of “0”s and “1”s in the case of a binary response.

Diffuseness (DF) is a measure of the extractable unique information in a given PUF instance.² This metric is used to evaluate the dissimilarity among response vectors corresponding to different challenge vectors from the same PUF instance. The diffuseness between the i -th and the j -th responses is defined as the intra-PUF normalized Hamming distance d

$$DF(B_i, B_j) \equiv \frac{1}{K} d(B_i, B_j). \quad (S2a)$$

The average diffuseness accounting for all possible pairwise comparisons is therefore

$$\langle \text{DF} \rangle \equiv \frac{2}{C(C-1)} \sum_{i=1}^C \sum_{j=i+1}^C d(B_i, B_j). \quad (\text{S2b})$$

Another important metric is uniqueness (UQ), a measure of dissimilarity between response vectors from different PUF instances to the same input challenge. Uniqueness between two response vectors to the same i -th challenge from the l -th and p -th PUF instances is defined as the inter-PUF normalized Hamming distance:

$$\text{UQ}(B_i^p, B_i^l) \equiv \frac{1}{K} d(B_i^p, B_i^l). \quad (\text{S3a})$$

The uniqueness for the i -th challenge averaged over all possible pairwise comparison of PUF instances is

$$\langle \text{UQ}(B_i) \rangle \equiv \frac{2}{P(P-1)} \sum_{p=1}^P \sum_{l=p+1}^P \text{UQ}(B_i^p, B_i^l), \quad (\text{S3b})$$

whereas uniqueness averaged over all responses is

$$\langle \text{UQ} \rangle \equiv \frac{1}{C} \sum_{i=1}^C \langle \text{UQ}(B_i) \rangle, \quad (\text{S3c})$$

where P is the total number of PUF instances. In many security applications, the responses to the same challenge from different PUF instances should be highly dissimilar; thus, the ideal value for UQ is 0.5.

Bit-error-rate (BER) is the measure of PUF reliability and is defined as the normalized intra-trial Hamming distance between responses from the same PUF instance to the same input challenge vectors over different trials. PUF reliability is often evaluated by including additional external factors such as variation in the external temperature or the power supply voltage with time. A typical way of measuring BER is with respect to the initial sample, say at time $t = 0$, i.e.,

$$\text{BER}(B_i) \equiv \frac{1}{T} \sum_{t=1}^T \frac{1}{K} d(B_i(t), B_i(0)), \quad (\text{S4a})$$

where T is the total number of samples. The averaged bit-error-rate over all responses is therefore

$$\langle \text{BER} \rangle \equiv \frac{1}{C} \sum_{i=1}^C \text{BER}(B_i), \quad (\text{S4b})$$

It is useful to note that if the responses are completely uncorrelated random binary vectors of length K , whose bits are generated with 0.5 probability, UF, DF, and UQ follow normal distributions with 0.5 average and $\sqrt{0.25/K}$ standard deviation (i.e., 0.0625 for $K = 64$).

The diffuseness is sometimes reported for averaged Hamming distances between a given response and all other responses, i.e.,

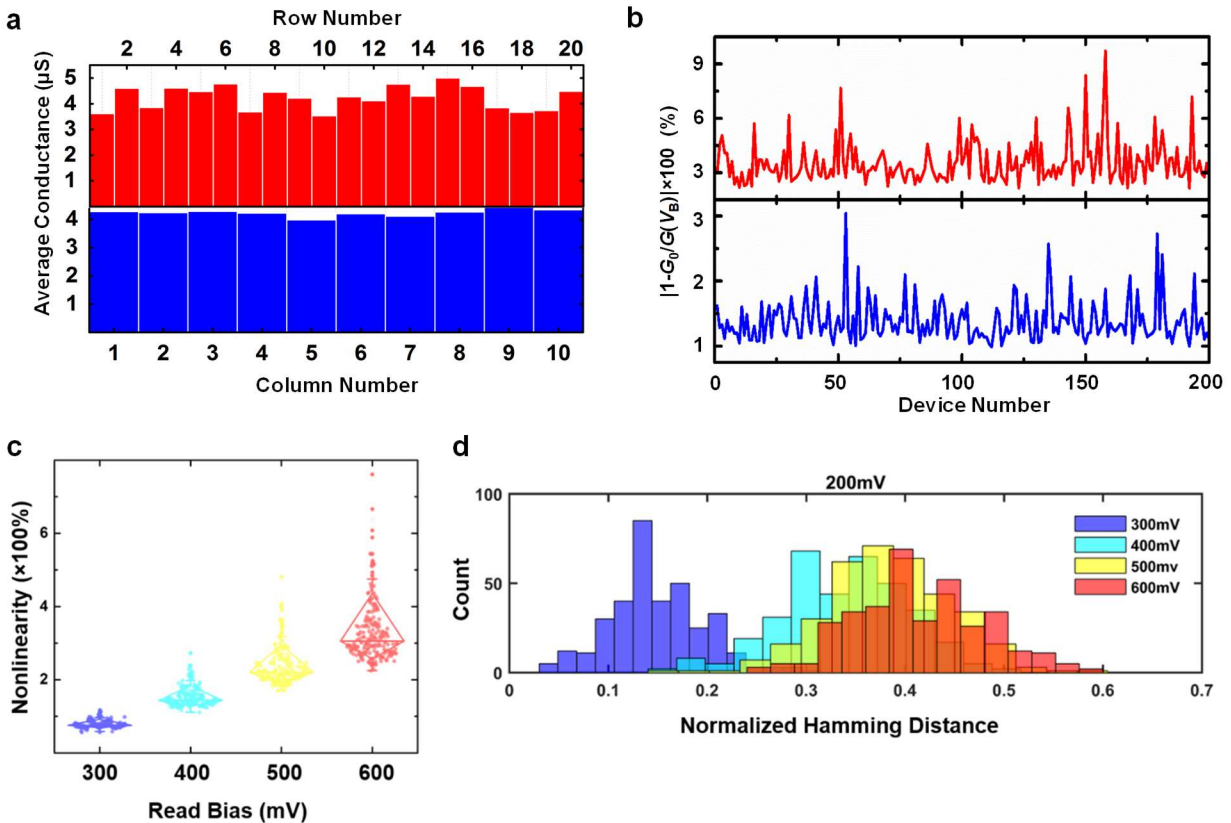
$$\langle \text{DF}(B_i) \rangle \equiv \frac{1}{C} \sum_{j=1}^C \text{DF}(B_i, B_j) \quad (\text{S5})$$

It is easy to show that, for random binary vectors the average value of $\langle \text{DF}(B_i) \rangle$ over all responses is still 0.5, whereas its standard deviation is $\sqrt{0.25/(CK)}$, i.e., the standard deviation is much

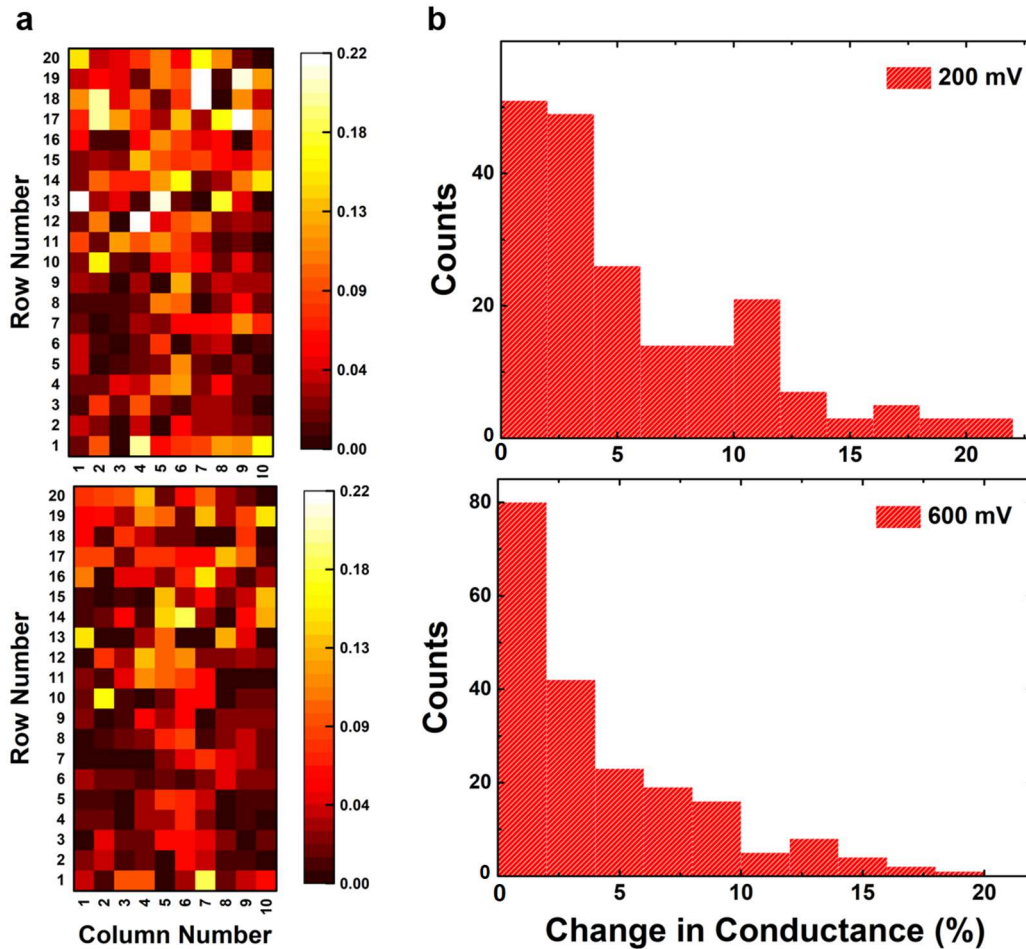
lower than that reported for $DF(B_i, B_j)$. Similarly, the average and standard deviation for $\langle UF(B_i) \rangle$ defined by Eq. S3b are 0.5 and $\sqrt{0.5/(KP(P-1))}$, respectively.

4. Supplementary results for PUF characterization

Figure S3a-c shows additional results for the tuning experiment shown in Fig. 3 of the main text. For example, Figure S3c clearly shows that both the median and the standard deviation of the nonlinearity of individual devices increase with increasing bias. Figure S3d shows the distribution of Hamming distances (i.e., the uniqueness) between responses to the same challenges without retuning the weights; the responses were measured at 200 mV and at the specified voltage bias. This figure highlights the value of nonlinearity as an additional source of entropy in the PUF design. (Note that the results shown in Figure S3d are essentially more detailed statistics calculated according to Eq. S3b, though for only a few pairs of voltages, compared to the results shown in Figure 3f of the main text, which represent only the averages of the HD distributions calculated using Eq. S3c.) To evaluate the stability of the conductance distribution, the device conductances were re-measured in a bit-error-rate experiment after a 30-day period of thermal stress at 90 °C.



Supplementary Figure 3. (a) The average conductances (measured at 300 mV) for the devices in a specific row and column after the tuning procedure. (b) Figure 3c data (nonlinearity factor) shown as a linear plot. (c) Box plots of devices’ nonlinearity for all 200 memristors in the crossbar. Here, boxes show the 25-75 percentile area, while the bars signify the 10-90 percentile range. (d) Distributions of intra-bias responses’ uniqueness (UQ) between responses to the same challenges without re-tuning of the weights, measured at 200 mV and the specified voltage bias.



Supplementary Figure 4. (a) Maps and (b) histograms of relative changes in conductance measured at 200 and 600 mV (top and bottom panels, respectively) after a 30-day period following the thermal stress tests at 90 °C.

5. Performance and energy efficiency estimates

The demonstrated resistive crossbar circuit has fairly large feature sizes, much larger than those of recent state-of-the-art CMOS work implementations (Table S3). To conduct a meaningful comparison with prior work, we have estimated the performance and energy efficiency of the proposed security primitive assuming 55-nm lateral dimensions of the memristors. Note that much smaller, ~10-nm metal-oxide memristors based on similar material stacks have been demonstrated to have excellent retention and analogue properties,³ and, in fact, some of the device properties actually improve upon scaling. For example, the dynamic range (ON/OFF current ratio) is typically inversely proportional to the device area for filamentary devices due to the reduction in leakage current. Furthermore, in our comparison, we consider a more practical basic building block with $M = N = 100$ and $m = n = 20$ and assume that 10 response bits are generated in parallel.

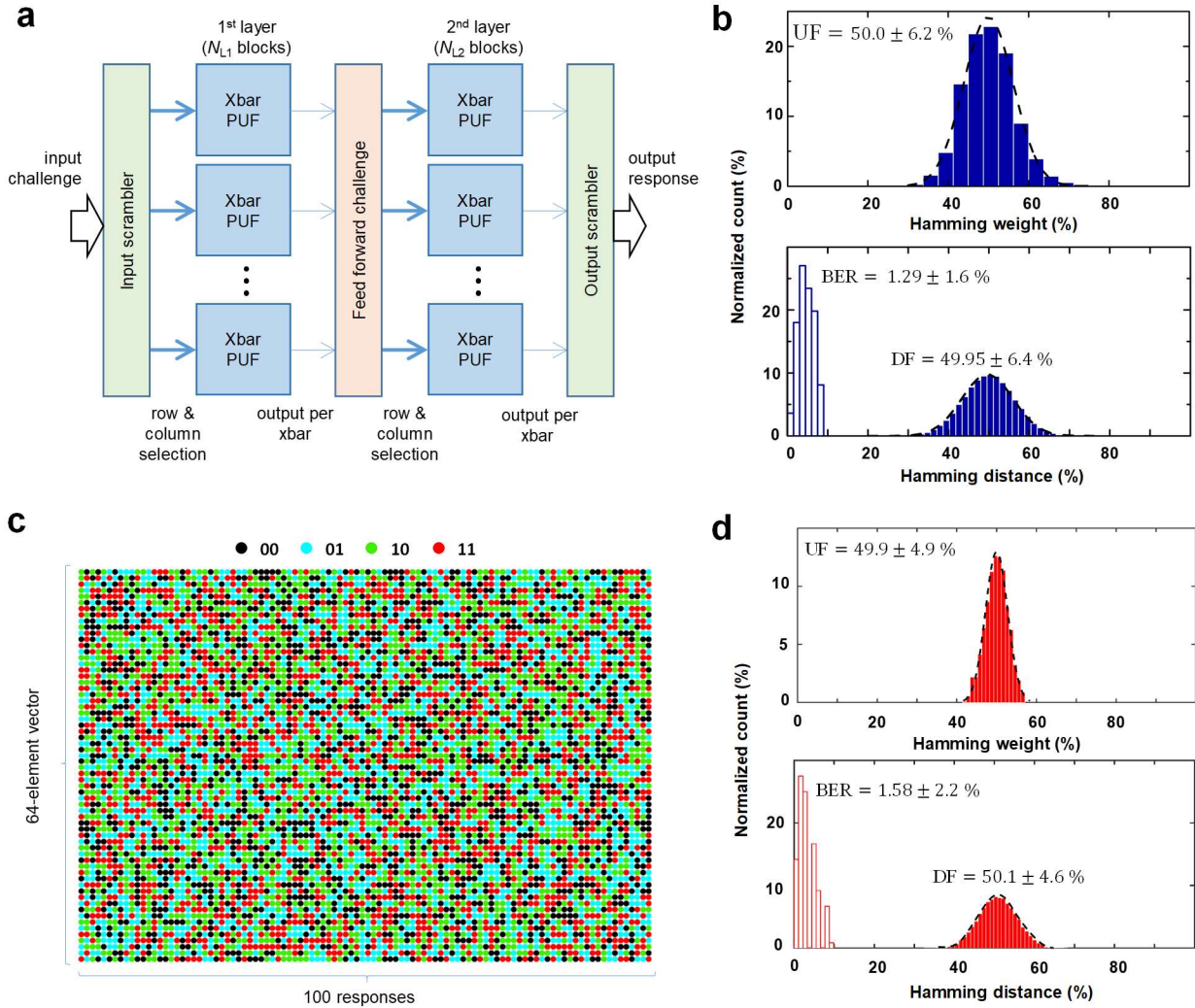
According to our previous work on mixed-signal vector-by-matrix multipliers,⁴ the area, maximum settling time and power consumption of a single differential sensing circuit implemented in a 55-nm process are 10 μm^2 , 4 ns, and 2.5 μW , respectively, assuming that the maximum and minimum input currents are 1 μA / 100 nA. The current assumptions are justified

since the minimum OFF current is reduced by a factor of ~ 40 upon scaling and because half of the read current would be contributed by approximately 20 selected devices and the other half by unselected devices and also given that the device conductances are balanced according to the optimization algorithm. (Additionally, note that the sensing circuit for the mixed-signal vector-by-matrix multiplier, which was implemented in a conveyor-like style, has much stricter requirements for output nonlinearity and driving capabilities; hence, there are some reserves for further optimization.) The dynamic energy for the charging/discharging crossbar circuit is estimated assuming a rather pessimistic $1 \text{ fF} / 1 \text{ } \mu\text{m}$ crossbar line capacitance,⁵ which results in $\sim 10 \text{ fJ}$ per bit. Neglecting the contributions from other circuitry, the total area, latency, and energy consumption for generating one output bit are $\sim 20 \text{ } \mu\text{m}^2$, $< 5 \text{ ns}$, and $\sim 20 \text{ fJ}$, respectively, significantly better than the values achieved by state-of-the-art CMOS implementations, even at more aggressive CMOS nodes (Table S3).

6. Multilayer PUF network

Figure S5a shows the general architecture of the proposed 2-level PUF circuit. The challenge specifies all selections that are applied to the PUF input, potentially in several steps (see below) to generate a K -bit output response. In particular, selections are first applied to N_{L1} primitive security blocks in the first layer of the PUF. The output of these blocks is used to generate a feed-forward (hidden) challenge that essentially consists of scrambling the data by passing it via a nonlinear transfer function with the goal of increasing resilience against reverse-engineering of the PUF circuit. The feed-forward challenge then specifies selections to the second layer with N_{L2} blocks, which in turn produces the PUF output. To increase the number of bits in the feed-forward challenge (and the output), its data can be generated in several steps, e.g., by sequentially applying a number of selections, as discussed in the main text. (The scrambling can also be performed at the input and output to further strengthen the PUF’s resilience. Additionally, the PUF circuit may contain dummy blocks that do not contribute to the PUF response and only scramble the network’s power profile.)

As a specific example, let us consider single-bit-output primitive blocks with $M = 20$, $N = 10$, $m = 10$, $n = 4$, and $N_B = 4$ that are used in 2-level PUF with $N_{L1} = N_{L2} = 8$, and $K = 64$. Row and column selections can be specified with bit vectors, so that $M + N + \log_2 N_B = 32$ bit input is sufficient to specify a unique selection for a single block (assuming there are no permutations in the columns). Let us also assume that unique selections are applied to the first-layer blocks and that the selections are the same for the second-layer blocks, i.e., the same feed-forward challenge is applied to all blocks at once. In this case, $K / N_{L2} = 8$ steps are required to generate all 64 output bits, which would require precomputing $(M + N + \log_2 N_B) K / N_{L2}$ bits of feed-forward challenge. Because N_{L1} bits of feed-forward challenge are generated at once, the total number of sequential steps to be performed in the first layer is $(M + N + \log_2 N_B) K / (N_{L1} N_{L2}) = 32$. The effective length of the PUF input, comprising all selections that are applied sequentially, each of which is $(M + N + \log_2 N_B) N_{L1}$ bits long, is therefore $(M + N + \log_2 N_B)^2 K / N_{L2} = 8,192$ bits. (Note that the described example is not intended to be optimal but is rather introduced as a means of presenting the details of the key operations that would be performed in a more complex PUF design. For example, PUF architecture can be optimized by generating multiple bits at once from one block. Evaluating these techniques and understanding the trade-offs between robustness to various attacks and the complexity of the PUF circuit are very important future goals.)



Supplementary Figure 5. More practical memristor PUF architectures. (a) Top-level architecture. In the most general case, the inputs, feed-forward challenge, and outputs can be subject to “scrambling”, i.e., certain nonlinear transfer functions, to improve the robustness and security of the PUF. (b) Measured security metrics for the PUF architecture with $N_{L1} = 10$, $N_{L2} = 1$ and $N_B = 8$ multi-bias selection scheme. (c-d) PUF ($N_{L1} = 10$, $N_{L2} = 1$) with quaternary response. Panel (c) shows an example of one hundred 64-element-long quaternary response keys; (d) shows the experimentally measured results.

Finally, to verify the operation of such an architecture, we have experimentally demonstrated the functionality of a simplified 2-level PUF network. Two slightly different implementations were considered. In both cases, $M = 20$, $N = 10$, $m = 5$, $n = 2$, $N_{L1} = 10$, $N_{L2} = 1$, and a 64×10 -bit feed-forward challenge was used. The locations of the selected rows are binary encoded by pairs of bits in a 10-bit portion of a feed-forward challenge such that the first two bits determine the location of the first selected row among the first four rows of the crossbar, the second pair determines the location of the second selected row among the next four rows of the crossbar, and so on. One column is always selected in the left half of the crossbar, and another column is selected from the right half. The particular locations are calculated by adding the five least significant bits of the 10-bit portion of the hidden challenge for the first column and the five most significant bits for the second one.

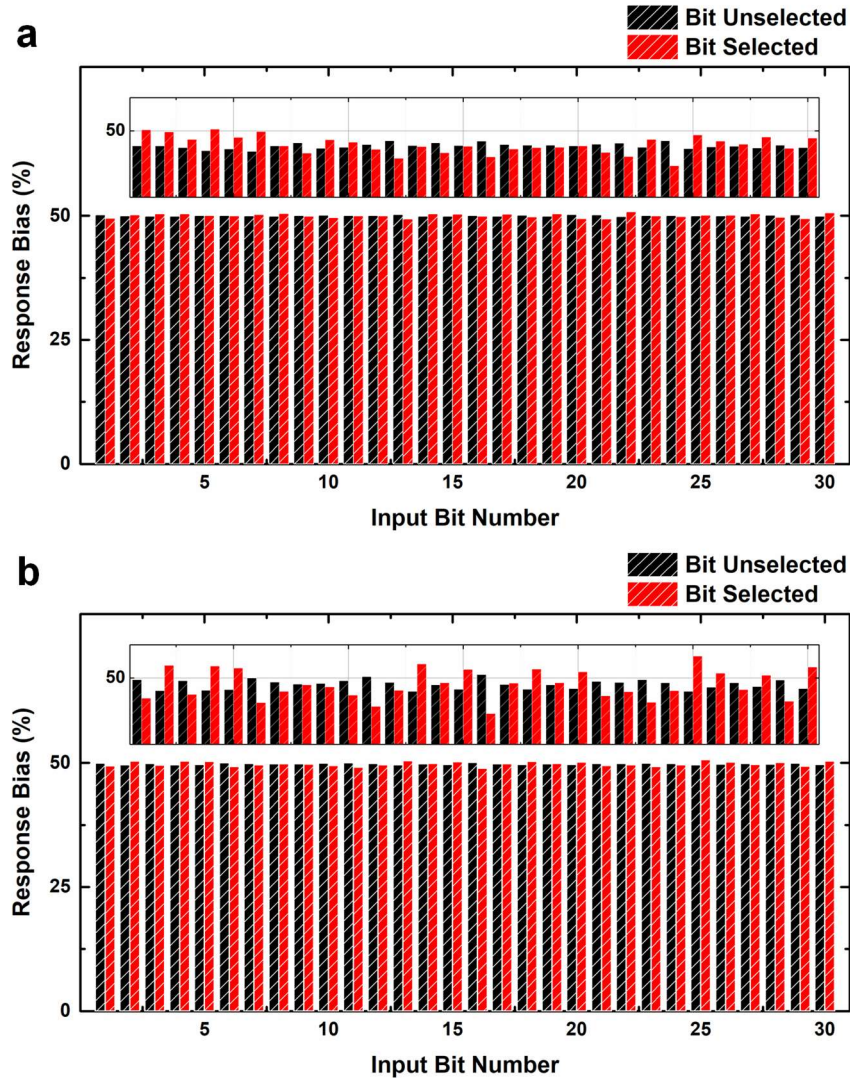
Figure S5b-d shows the experimental results for uniformity and bit error rate for the two considered cases, measured by collecting 500 64-bit and 500 128-bit responses, respectively, for randomly selected mutually exclusive challenges. In the first case, a sequence of 64 selections with each input selection applied simultaneously to all 10 first-layer blocks was used to generate a 64-bit response. Eight different voltages ($N_B = 8$) between 200 mV and 600 mV were used to bias the blocks; in particular, one randomly selected voltage level was used to bias all blocks in the first layer, and another randomly selected voltage level was used to bias the second-layer blocks. The selected voltages were unique for each input challenge. The only difference in the second considered case is that for each 10-bit portion of a hidden challenge, two output bits were generated by the second-layer block by first measuring an output at 200 mV and then at 600 mV.

7. Predictability and robustness to machine learning attacks

To investigate the robustness of the demonstrated basic building block with respect to modelling attacks, we have performed a series of additional tests using two sets of data. The first set of data corresponds to one of the tuned distributions discussed in the main text; the second, which is representative of a suboptimally tuned PUF, represents data that we collected at the earlier stages of our project. The two data sets consist of, respectively, 354,000 and 76,800 measured responses to random unique challenges. For simplicity, in all of these tests we have assumed that each challenge is encoded by 30 bits. “1” bit values encode the positions of five selected rows in the upper 20 bits and two selected columns in the lower 10 bits. (Obviously, such a format is sparse, and not all 30-bit numbers correspond to a valid challenge. A dense encoding would require only $\lceil \log_2 C_{MAX} \rceil = 20$ bits.)

A. Correlations

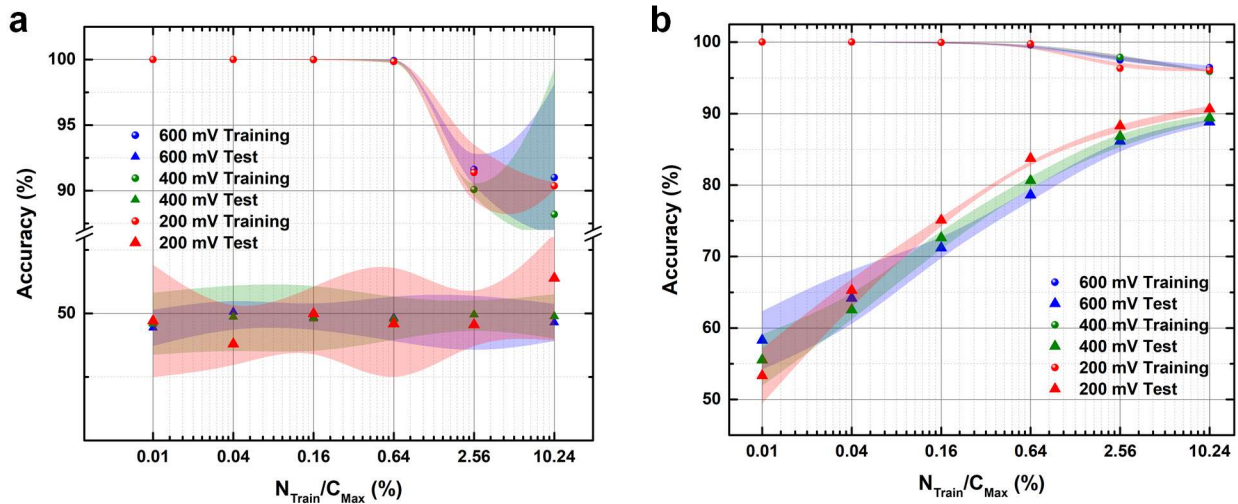
In our first test, we probed for possible bias in the output by checking the uniformity of the response when a particular bit of the challenge bit vector is fixed (Fig. S6a). The uniformity is close to the ideal (50%) for both experiments, though the results are visibly somewhat worse for the second data set (Fig. S6b). These results, however, do not exclude the possibility of more complex correlations involving multiple input bits. Such correlations can be better captured by modelling PUF with binary classifiers based on a feed-forward neural network. Figure S7 shows the preliminary results of such modelling using a multilayer perceptron with 30 inputs, 1 output, and two 250-neuron hidden layers. The network was trained using a random sample of measured input-output data of specified size and then checked against 6,000 (mutually exclusive) randomly selected challenge response pairs. The results show that the output of the near-optimal PUF is difficult to predict even when the training data represent more than 10% of the total number of challenge response pairs. On the other hand, the classification accuracy of the test data for the suboptimal PUF improves significantly when the size of the training set is increased. However, even for the suboptimal PUF, using such a large training set for a more practical PUF network (e.g., with much larger C_{MAX} as discussed earlier in Sections 5 and 6) would be completely unfeasible. Indeed, it is natural to expect that, for a more realistic scenario in which only a very small fraction of the challenge-response pairs is used as the training data, the classification accuracy would be close to the ideal 50% (Fig. S7b). Additionally, note that the results confirm that nonlinearity improves robustness slightly; we expect that the improvement will be more pronounced for more complex PUFs.



Supplementary Figure 6. The distribution of response uniformity when a specific bit of the challenge is fixed to a value of either “1” (selected) or “0” (unselected) for two sets of measured data (at 0.2 V voltage bias), corresponding to (a) near-optimal and (b) suboptimal PUF instances. For example, the first black/red column shows the fraction of the total number of “1” responses with respect to the total number of responses for all measured challenges in which the first bit is set to “0”/ “1”.

B. Output randomness

We further evaluated the randomness of the near-optimal PUF using an NIST statistical test suite⁶ and a long short-term memory (LSTM) neural network model.⁷ In particular, for the first test, the output bits were partitioned into 7000-bit sequences and used to run 15 different NIST benchmarks, each of which was repeated 50 times. (“Universal”, “Random excursions”, and “Random excursions variant” tests were excluded due to insufficient data.) The results, which are shown in Table S1, confirm that the generated responses successfully pass NIST randomness tests, i.e., that the probability value (P-value) exceeds 0.01 and that the uniformity is greater than 0.0001.⁶



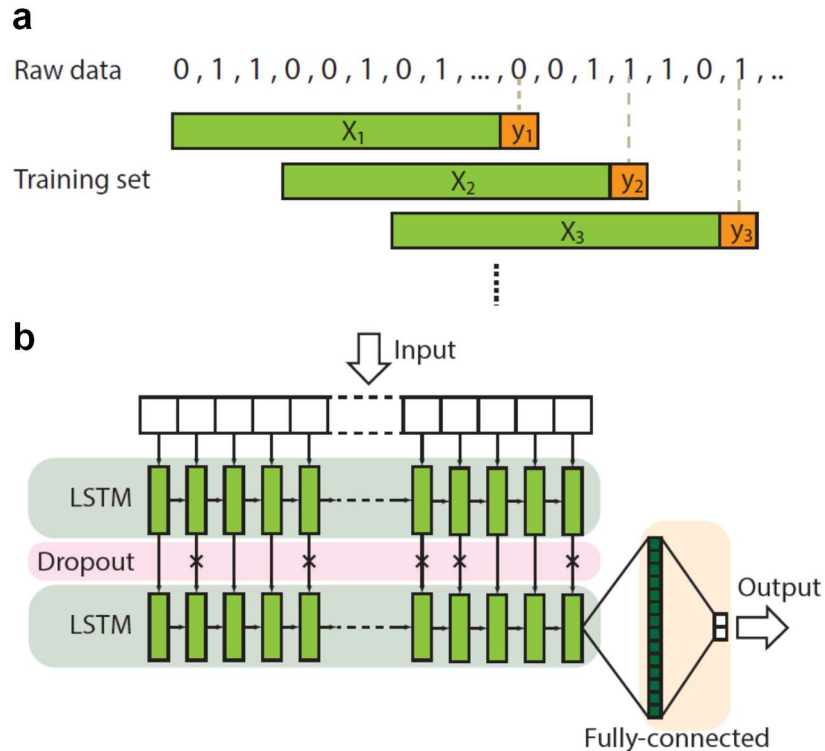
Supplementary Figure 7. Robustness to machine learning attacks for (a) near-optimal and (b) suboptimal PUF simulated utilizing a $30 \times 250 \times 250 \times 1$ multilayer perceptron classifier. The markers denote the average classification accuracy over 10 runs; the thickness of the lines for the test data specifies two standard deviations. All simulation results were obtained with the Matlab module “traingdx” using a hyperbolic tanh activation function in all layers with momentum and adaptive learning rate and the following parameters: 0.01 learning rate, 1.05 / 0.85 ratio to increase/decrease learning rate, 0.9 momentum constant, $1e-10$ minimum performance gradient, $1e-20$ performance goal, 2500 training epochs, 10% validation ratio, and 10 maximum validation failures. For each training run, the network weights in all layers were randomly initialized to values between -1 and 1.

Supplementary Table 1. Results of the NIST randomness test

	200 mV		400 mV		600 mV	
	Pass rate (%)	Uniformity of P-value	Pass rate (%)	Uniformity of P-value	Pass rate (%)	Uniformity of P-value
Frequency	96	0.935716	98	0.040108	98	0.040108
Block frequency	100	0.350485	96	0.011791	96	0.011791
Runs	100	0.971699	100	0.816537	100	0.816537
Longest run	100	0.779188	100	0.350485	100	0.350485
FFT	98	0.350485	100	0.851383	98	0.851383
Non-overlapping template	97.30	All \geq 0.0001	95.95	All \geq 0.0001	100	All \geq 0.0001
Overlapping template	98	0.616305	100	0.013569	96	0.013569
Linear complexity	96	0.816537	96	0.534146	100	0.534146
Serial	100	0.289667	98	0.851383	96	0.851383
Serial	100	0.137282	100	0.616305	96	0.616305
Approximate entropy	100	0.289667	98	0.699313	100	0.699313
Cumulative sums - forward	96	0.494392	96	0.383827	100	0.383827
Cumulative sums - backward	96	0.739918	98	0.534146	100	0.534146

We then evaluated the response predictability for the near-optimal data set using the LSTM architecture proposed by Graves⁷ (Fig. S8), which is a special case of a recurrent neural network that is capable of handling long-range dependencies in general-purpose sequence modelling tasks. The implemented network is based on two LSTM layers and ReLU as an activation function. Features with size of 128 extracted by the two LSTM layers are fed into two fully connected layers with sigmoid and softmax functions, respectively, as activation functions. We employed the model in Keras 2.0.6 with Tensorflow 1.1.0 backend. Three network configurations were used to evaluate the response sequence (Table S2). The measured response data were tested in such a way that N adjacent bits were considered as input, and the immediately following bit was treated as the label (Fig. S8a). The input samples were shifted by $S = 3$ bit positions.

The near-ideal unpredictability of the output sequence for the three training sets and the output dimensions configurations further point to the suitability of the proposed approach for implementing highly secure and resilient architectures. Nevertheless, further investigation of PUF circuits’ vulnerabilities to advanced deep-learning algorithms is important future work.



Supplementary Figure 8. Modelling with long short-term memory neural network. (a) Input data preparation and (b) LSTM architecture. The Python code utilized for LSTM simulations is available at <https://github.com/RMITnano/PUF-LSTM>.

8. Experimental characterizations and test data

All the evaluated experimental datasets have been uploaded to <https://www.ece.ucsb.edu/~strukov/papers/2018/PUFdata/> for public access. Therein, the data are categorized with respect to the corresponding evaluation metrics, along with instructions for extraction and evaluation.

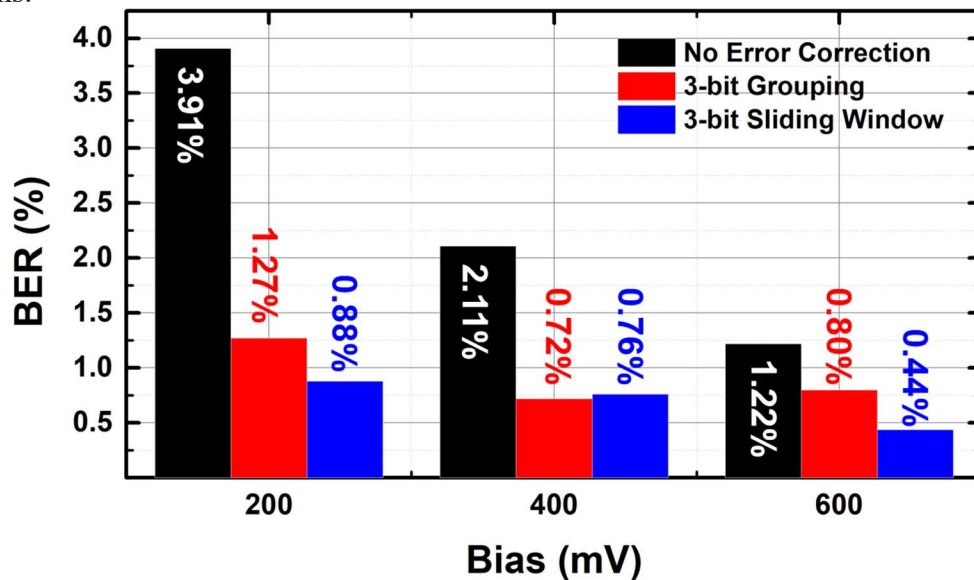
Supplementary Table 2. Machine learning attack results using the LSTM-Dropout-LSTM-Dense-Dense-Softmax network.

Training sequence length	Output dimensions	Predictability (%)
301	LSTM: 128, Dense: 128, 2	50.41
101	LSTM: 128, Dense: 128, 2	50.52
64	LSTM: 256, Dense: 256, 2	50.28

9. Prospects for improving BER

Key generation applications require very repeatable and reliable PUF operation, and hence various BER boosting techniques are typically employed to improve raw BER of PUF's basic building blocks [22]. For example, a three-step approach involving temporal majority voting, burn-in hardening and dark-bit masking was utilized to reduce the BER from 25% to 0.98% in CMOS-based PUFs [23].

The high density, low latency, and high throughput of our approach should allow for a wide range of options for improving BER. For instance, Figure S9 shows the preliminary results for two majority voting approaches. In the first case, the same challenge is applied three times and the output bit is determined by the majority among three bits. This approach would help against occasional errors. In the second approach, which could tolerate completely unreliable challenges, 3 bits are first computed by applying different challenges. A single output bit is then determined by majority voting. The results show that even the most rudimentary error correcting techniques can reduce the BER significantly. We expect that more advanced error correcting codes, which could be applied to larger groups of bits, and other techniques such as masking of bad memory cells and remapping around them, would enable sufficiently low BER for secret key generation applications.

**Supplementary Figure 9.** Comparison between the original and improved BER results for the worst-case 16 kb data (Fig. 3e of the main text) using simple temporal and spatial majority voting techniques.

10. Comparison with prior work

Supplementary Table 3. Comparison of reported PUF primitives based on different technologies.

Reference	[8]	[8]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18-20]	[21]	This work	
Core technology	65nm CMOS SRAM	CMOS arbiter	65nm CMOS ring oscillator	22nm tri-gate CMOS	STT-MRAM	MTJ	90nm NMOS	CNT	ReRAM	ReRAM	CNT	ReRAM	ReRAM	ReRAM (ZnO NW)	ReRAM	
Randomness source	Geometry	Geometry	Geometry	Geometry	Geometry	Geometry	Geometry	Geometry	R_{ON} / R_{OFF} variations	R_{OFF} variations	Geometry and placement	R_{OFF} variations	Write-time variations	Write-time variation	$I-V$ nonlinearity variations	
Type of work	EXP	EXP	EXP	EXP	SIM	EXP	SIM	S&E	SIM	S&E	EXP	S&E	SIM	EXP/SIM	EXP	
Demo complexity	4×64 kb SRAM array	256×64 bit arbiter PUF	4096 ring oscillator+16×3 2-bit counter	250Kbit	-	10×20 array	-	-	-	-	5×5 CNT array	-	64×8 array (largest case)	6 single devices/8×8 array for SIM	2×10×10 3D integrated arrays	
Cell size / area	306F ² / 0.213 mm ²	0.279 mm ²	39000F ² / 0.241 mm ²	-	6.79 μm ² for 64 bits	6.74 μm ² for 64 bits	-	14 nm channel length	-	F = 200 nm	Trench width ~ 30-70 nm	F = 50 μm	-	2.15 μm × 570 nm (L, D)	F = 350 nm	
Programmability	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	
Uniqueness (%)	49.72 ± 0.3	47.13 ± 0.44	49.60 ± 1.11	-	50.0 ± 0.1	47	-	49.67	47	49.95	50 ± 0.39	49.85	50	-	50.0	
Reliability (%)	94.53 ± 0.14	96.96 ± 0.08	98.47 ± 0.39	91.2 (worst case)	~100	97.75 in 800 runs	95	96.5	90	~ 98	~97	98.67	95.1 (best case)	-	~ 97 – 98.9 (worst case)	
Uniformity (%)	-	-	-	-	-	-	-	49.67	47	-	-	47.28	50	50	49.5-50	
Diffuseness (%)	-	-	-	-	-	-	-	-	-	-	50.0 (for binary keys)	49.86	-	-	~ 50.0	
NIST test (or entropy)	Not reported (0.942)	Not reported (0.896)	Not reported (0.946)	Not reported (Full entropy)	Not reported (0.985)	Not reported (0.9997)	Not reported	Not reported	Not reported	Not reported	Passed	Not reported	Not reported (0.996 best case)	Not reported	Passed	
Readout speed	-	-	-	-	> 10 ns	5 ns	250 ps	43 ps	-	-	-	-	-	-	5 ns*	
Energy	1.1 pJ / bit	-	474.8 fJ / bit	192 fJ/bit	-	4 mW at 1 V	37.5 fJ / bit	0.67 fJ / bit (90 nm node)	-	-	-	-	0.26-2.22 mW	-	20 fJ/bit*	
Environmental factors	TR: -40-85 °C, VR: 0.6-1 V	TR: -40-85 °C, VR: ±10%	TR: -40-85 °C, VR: 0.4-0.5 V	-	TR: 70-125 °C, VR: ±10%	TR: 25-75 °C	TR: 55-125 °C, VR: ±20%	TR: 20-80 °C, VR: ±22.5% 7.5% channel length variation	-	TR: 25-75 °C	TR: 25-85 °C,	TR: 0-175 °C, VR: ±10%, +20 nA undetectable range, 90% yield	-	-	-	TR: 25-90 °C, VR: ±20%

SIM: Simulation only; **S&E:** Simulation based on measured device data; **EXP:** Experiment; **TR:** Temperature range; **VR:** Voltage range

* Estimates assuming 55 nm process and 100×100 array with 10 output bits generated in parallel

References

- 1 Hoskins, B. D. & Strukov, D. B. Maximizing stoichiometry control in reactive sputter deposition of TiO₂. *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films* **35**, 020606 (2017).
- 2 Hori, Y., Yoshida, T., Katashita, T. & Satoh, A. in *IEEE International Conference on Reconfigurable Computing and FPGAs* 298-303 (2010).
- 3 Govoreanu, B. *et al.* in *IEEE International Electron Devices Meeting 2013* 10.12. 11-10.12. 14 (2013).
- 4 Mahmoodi, M. R. & Strukov, D. B. An ultra-low-energy current-mode sensing circuit enabling POps/J analog computing. *in preparation* (2017).
- 5 Strukov, D. B. & Likharev, K. K. CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices. *Nanotechnology* **16**, 888 (2005).
- 6 Rukhin, A. *et al.* Statistical test suite for random and pseudorandom number generators for cryptographic applications, NIST special publication. (2010).
- 7 Graves, A. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* (2013).
- 8 Roel, M. *Physically unclonable functions: Constructions, properties and applications*. PhD Thesis, University of KU Leuven (2012).
- 9 Mathew, S., Satpathy, S., Suresh, V. & Krishnamurthy, R. K. in *IEEE Custom Integrated Circuits Conference* 1-4 (2017).
- 10 Zhang, L., Fong, X., Chang, C.-H., Kong, Z. H. & Roy, K. in *IEEE International Symposium on Circuits and Systems* (2014).
- 11 Das, J., Scott, K., Rajaram, S., Burgett, D. & Bhanja, S. MRAM PUF: A novel geometry based magnetic PUF with integrated CMOS. *IEEE Trans. Nanotechnology* **14**, 436-443 (2015).
- 12 Majzoobi, M., Ghiaasi, G., Koushanfar, F. & Nassif, S. R. in *IEEE International Symposium on Circuits and Systems* 2071-2074 (2011).
- 13 Konigsmark, S. C., Hwang, L. K., Chen, D. & Wong, M. D. in *IEEE Asia and South Pacific Design Automation Conference* 73-78 (2014).
- 14 Rajendran, J. *et al.* Nano meets security: Exploring nanoelectronic devices for security applications. *Procs. IEEE* **103**, 829-849 (2015).
- 15 Chen, P. Y. *et al.* in *IEEE International Symposium on Hardware Oriented Security and Trust* 26-31 (2015).
- 16 Hu, Z. *et al.* Physically unclonable cryptographic primitives using self-assembled carbon nanotubes. *Nature Nanotechnology* **11**, 559-565 (2016).
- 17 Kim, J. *et al.* A Physical Unclonable Function with Redox-based Nanoionic Resistive Memory. *IEEE Trans. Information Forensics and Security* (2017).
- 18 Rose, G. S. & Meade, C. A. in *IEEE Design Automation Conference* 1-6 (2015).
- 19 Uddin, M., Majumder, M. B. & Rose, G. S. Robustness Analysis of a Memristive Crossbar PUF Against Modeling Attacks. *IEEE Trans. Nanotechnology* **16**, 396-405 (2017).
- 20 Uddin, M. *et al.* in *IEEE Computer Society Annual Symposium on VLSI* 212-217 (2016).
- 21 Mazady, A., Rahman, M. T., Forte, D. & Anwar, M. Memristor PUF - A Security Primitive: Theory and Experiment. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* **5**, 222-229 (2015).
- 22 Kaiyuan, Y., Blaauw, D. & Sylvester D. Hardware Designs for Security in Ultra-Low- Power IoT Systems: An Overview and Survey. *IEEE Micro* **37**, 72-89 (2017).
- 23 Mathew, S. K., *et al.* in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* 278-279 (2014).