Programmable Hardware-Intrinsic Security Primitives Enabled by Analogue State and Nonlinear Conductance Variations in Integrated Memristors

Hussein Nili[1*], Gina C. Adam[1,3], Brian Hoskins[1], Mirko Prezioso[1], Jeeson Kim[2], M. Reza Mahmoodi[1], Farnood Merrikh Bayat[1], Omid Kavehei[2†], and Dmitri B. Strukov[1]

[1] University of California Santa Barbara, Santa Barbara, CA 93106-9560, U.S.A.
[2] Royal Melbourne Institute of Technology University, Melbourne, Victoria 3000, Australia
[3] National Institute for R&D in Microtechnologies, Bucharest, Romania

[*]hnili@ece.ucsb.edu, [†]omid.kavehei@rmit.edu.au, [§]strukov@ece.ucsb.edu

Hardware-intrinsic security primitives employ instance-specific and process-induced variations in electronic hardware as a source of cryptographic data. Among various emerging technologies, memristors offer unique opportunities in such security applications due to their underlying stochastic operation. Here we show that the analogue tuning and nonlinear conductance variations of memristors can be used to build a basic building block for implementing physically unclonable functions that are resilient, dense, fast, and energy-efficient. Using two vertically integrated 10×10 metal-oxide memristive crossbar circuits, we experimentally demonstrate a security primitive that offers a near ideal 50% average uniformity and diffuseness, as well as a minimum bit error rate of around 1.5% ± 1%. Readjustment of the conductances of the devices allows nearly unique security instances to be implemented with the same crossbar circuit.

The continuing advance of information technology has stimulated an unprecedented expansion of interconnected networks and devices. The significant volume of personal and sensitive information continuously carried over shared and remotely accessible networks poses significant security challenges[1-3], which conventional cryptographic approaches struggle to adequately address. Conventional cryptographic approaches typically rely on "secret keys" stored in nonvolatile memories for data encryption and access authentication, and these are vulnerable to physical and side-channeling attacks, including direct probing and power analysis[4,5]. As a result, security approaches based on physical hardware roots of trust have recently attracted significant attention. Analogous, to a degree, to biometric identifiers, such as retinal and fingerprint imprints, hardware roots of trust are physically embedded with their cryptographic processes through unique, individual structural properties that are virtually unpredictable and practically inimitable[2,4,6-10]. The cryptographic data should be immediately and reliably available upon interrogation and effectively impossible to learn or extrapolate even when challenged by aggressive model-building and machine learning attacks[6].

Physical unclonable functions (PUFs) are a class of hardware security primitives that draw their cryptographic "keys" from fabrication process variations[11-15]. Among the wide variety of proposed PUF implementations utilizing (within-die) spatial variations in electronic devices[3,16-24], those based on crossbar architectures with integrated resistive random access memory (ReRAM or memristors, for short) are particularly promising. This is due to their simple and relatively low-cost fabrication process, small footprint, complimentary metal-oxide-semiconductor (CMOS) circuit integration compatibility[25-28], and process-induced variations in *I-V* characteristics pertinent to the mixed electronic-ionic transport and memory mechanism[25,26,29-33].

The most accessible manifestation of process-induced compositional and structural variations in ReRAM arrays is the spatial (that is, device-to-device) variations of the effective switching thresholds. One example is the voltage at which device conductance is abruptly changed upon application of a ramping bias. A related example is spatial variations in the ON and OFF state conductances in the array upon application of a large voltage or current bias[29-31]. The physical source of these variations is arguably the stochastic nature of ionic switching arising from compositional inhomogeneity of the switching medium, as well as variations in individual device profiles such as electrode imperfections and random variations in surface roughness[25,26,34].

These "entropy" sources were typically the foundation for previously proposed memristor-based security primitives (see Supplementary Table 3 for a comparison of reported primitives). Many of these proposed PUFs require a relatively large number of devices in the crossbar array[8,29,30,35] and extensive peripheral programming and control circuitry[30] to achieve viable operational metrics. Furthermore, a digital mode of operation with devices switched to the extreme ON and OFF states is typically utilized, hence ignoring one of the main advantages of memristive devices: their nonlinear adjustable *I*-*V*s. Indeed, because the device nonlinearity is strongly dependent on the memory state and is correlated with process variations, it can serve as a prominent source of the entropy in memristive arrays[36]. On the other hand, in the digital approach, the crossbar array is effectively reduced to a linear resistive network, which greatly simplifies input-output mapping. The PUF operation in some of the prior proposals also rely on the write operation[37,38], which may not be practical, especially for key generation applications, considering the write endurance limitations of the memristors.

In this Article, we first propose a robust hardware-intrinsic security primitive that takes advantage of variations in the nonlinear *I*-*V* characteristics of memristors; the key novelty of this approach is the analogue tuning of the memristors' conductances to maximize the functional performance of the PUF. We then experimentally demonstrate a fully functional implementation of the security primitives based on integrated memristive circuits. This, we believe, is an important step in the development of a practical PUF network based on the unique features of memristive arrays, and notably extends beyond previous demonstrations, which typically relied on post-processing data measured on individual devices and/or using a very small portion of the challenge-response space.[28,30,31,33,35]

Hardware-intrinsic security primitive

The basic building block for our security hardware is implemented with a two-level stack of monolithically integrated 10×10 memristive arrays (Fig. 1a, b). The fully passive Al_2O_3/TiO_{2-x} memristor crossbars, which have an active device area of ~ 350 × 350 nm^2, were fabricated using *in situ* low-temperature reactive sputtering deposition, ion milling, and a precise planarization step. The middle electrodes are shared between the bottom and top layers (Fig. S1). The fabrication flow ensures a high device yield (>95%) and low <175 °C temperature budget, compatible with CMOS back-end-of-line integration. (The fabrication steps are similar to those described in our earlier report[39] and are explained in more detail in Section 1 of Supplementary Information.) The ON/OFF ratio of currents for the devices in the top and bottom layers is at least two orders of magnitude, on average, when measured at 0.3 V (Fig. S1c). The variations in effective voltage switching thresholds are sufficiently low to permit precise tuning of the devices within the array (Fig. 1c), while still substantial enough to be utilized in the considered application (see below). The device *I*-*V* is nonlinear, especially at higher resistance states (Fig. 1d).

Figure 2 shows how such an effective $M \times N = 20 \times 10$ crossbar circuit with crosspoint device conductances G_{ij} is utilized to implement basic cryptographic functionality. Similar to previous proposals,[36,37] a single-bit binary output *b* is calculated by biasing *m* selected rows with voltage V_B and then comparing the currents running into two groups of *n*/2 selected virtually grounded columns. For simplicity, let us assume that one group always comprises the leftmost columns and the other the rightmost so that

$$b = \begin{cases} 1, & I^+ > I^- \\ 0, & I^+ \le I^- \end{cases} \qquad I^{\pm} = V_{\mathrm{B}} \sum_{j \in S_{\mathrm{C}}^{\pm}} \sum_{i \in S_{\mathrm{R}}} G_{ij}\,(V_{\mathrm{B}}) \tag{1}$$

where S_R is a set of indexes of the selected rows, S_C^+ and S_C^- are sets of indexes of the selected columns in the left and right groups, respectively, and I^+ and I^- are their respective currents. The remaining (unselected) rows and columns in the array are kept floating. With such a scheme, the maximum number of distinct selections is

$$C_{\mathrm{MAX}} = \binom{M}{m} \times \binom{N}{n}. \tag{2}$$

Note that this number can be further substantially increased by considering more complex peripheral circuitry, e.g., by factor $\binom{n}{n/2}$ by taking into account order permutations in the columns, and, as we show later, by a factor N_B, the number of different bias voltages utilized in one selection.

The exemplary PUF network based on the discussed circuit is implemented by tuning the conductances of the crosspoint devices to specific pre-calculated values using the write-verify algorithm.[40] The goal of the tuning procedure is to enhance the contribution of the devices' *I-V* variations to the response of the network while at the same time improving its reliability and randomness. This is achieved by selecting a specific distribution of device conductances and having a proper balance between two types of currents measured at the output - currents via selected devices and sneak path currents passing through the floating portion of the array. In particular, the target conductances for a particular PUF instance are found by randomly generating *C* exclusive selections (i.e., *C* different S_R, S_C^+, and S_C^- sets) and their corresponding desired values for the outputs I^+ and I^- and then minimizing the function

$$\sum_{k=1}^{C} \left[\left(V_{\mathrm{B}} \sum_{j \in (S_{\mathrm{C}}^{+})_k} \sum_{i \in (S_{\mathrm{R}})_k} G_{ij} - I_k^{+} \right)^2 + \left(V_{\mathrm{B}} \sum_{j \in (S_{\mathrm{C}}^{-})_k} \sum_{i \in (S_{\mathrm{R}})_k} G_{ij} - I_k^{-} \right)^2 \right] \tag{3}$$

with a natural constraint that all conductances are nonnegative values. Importantly, I^+ and I^- are Gaussian distributed, and the absolute difference $|I^+ - I^-|$ for each selection is forced to be larger than a certain value (for more details on this algorithm, see Section 2 of Supplementary Information). The described procedure for configuring the crossbar circuit results in a narrow distribution of device conductances in which the PUF uniformity (UF) and diffusiveness (DF) are improved by eliminating biases in the output currents. At the same time, the reliability (BER) of the PUF, in particular its tolerance to the memristors' current fluctuations due to intrinsic noise and potential drift of conductive states[29], is strengthened by enforcing the current readout margins. Furthermore, the PUF uniqueness (UQ) is facilitated by the random nature of the algorithm used to select the conductance distribution. (See the formal definitions of these security metrics in Section 3 of Supplementary Information.)

Different target weight distributions can be precomputed beforehand for a specific memristor technology. Precise tuning of the weights is not required; therefore, implementation of specific unique PUF instances using the proposed algorithm can be relatively fast and incur minimal circuit overhead. Moreover, the same hardware can be programmed to implement different PUF instances, which is another unique feature of our approach. A somewhat faster implementation of different unique PUF instances is achieved by using the "rattling" strategy, which we also consider in this paper. In this case, the initial (e.g., tuned) distribution of the weights is changed (rattled) by applying short voltage pulses of random amplitude and polarity.

PUF demo and characterization of its security metrics

Figure 3a-c shows the results of tuning the memristors' conductance to the values determined by the algorithm. As expected, the target and the tuned conductance distributions were Gaussian-shaped (Figs. 3b and 4a), even when using the rattling scheme (Fig. 4c), with fairly uniform averages along the rows and columns of the crossbar array (Fig. S3a).

The security metrics for the PUF are experimentally characterized using a selection scheme with $m = 5$ rows and $n = 2$ columns and three different voltages V_B: 200 mV, 400 mV, and 600 mV. (According to Eq. (2), for this case C_{MAX} = 697,000 for each voltage bias.) Specifically, we evaluate PUF metrics by generating response data for 384,000 exclusive random selections, i.e., slightly more than half of the total available, at each voltage bias and grouping the single-bit outputs in 64-bit response packets so that there is a total of 6,000 64-bit outputs for each voltage bias generated. (In the considered implementation, the diffuseness is naturally improved by grouping more bits together.)

Figure 3d,e shows UF, DF, and BER for the collected data. In particular, the data show that increasing the voltage bias from 200 mV to 600 mV improves UF from already decent 49.5 ± 6.25% to nearly ideal 50.1 ± 6.26%; another PUF randomness metric, DF, is also close to ideal, being ~50 ± 6.25% for all cases. The better PUF metrics at higher voltages are attributed to the stronger nonlinearity in the device *I-V*s.

To accelerate testing, the reliability of the network (i.e., its BER) was measured using the worst case 16,000 challenges (out of 384,000) that resulted in the smallest current differential readout margins. The results show that BER improves substantially at higher biases, from 3.9 ± 1.8% at V_B = 200 mV to 1.22 ± 1.0% for V_B = 600 mV; this is partially attributed to the improved readout margins. The improvement in BER is even more significant, from 16.36 ± 3.1% at 200 mV to 5.93 ± 2.59% at 600 mV, for PUF operation under an elevated ambient temperature of 90 ^{O}C (inset in Fig. 3e). The latter BER value is comparable to that of simulated BER for conventional PUF implementation[41] despite being measured for the worst-case challenges.

The PUF uniqueness was evaluated by implementing different instances on the same crossbar circuit. First, we measured the uniqueness between pairs of PUF instances that were implemented by varying applied voltages V_B without re-tuning the device conductances (Fig. 3f). Not surprisingly, the maximum UQ of 44.8 ± 6.9% is achieved between the PUFs with the smallest applied voltage (200 mV) and the largest one (600 mV). This is quite natural because variations in nonlinear *I-V*s, which are more prominent at higher biases, result in non-monotonic redistribution of sneak path currents (Figs. 3c and S3b). Such a feature is very useful against power side channel attacks and suggests the possibility of using voltage bias as one of the independent inputs of the selection scheme.

In a more general study, we characterized the uniqueness between PUF instances with differently programmed crossbar devices by applying 32,000 exclusive random input challenges. Figure 4a,b shows the results for 5 different PUF instances, each with a unique tuned conductance distribution according to the described algorithm. The UQ was close to the ideal 50% mean for all studied cases (Fig. 4b), with the smallest variance, 1.9%, at the largest bias voltage of 600 mV. In another experiment, we characterized 10 different PUF instances obtained by the "rattling" strategy, which was applied over the initially tuned distribution (Fig. 4c). In particular, in each case the conductance for each device in the crossbar was rattled by a single 10-µs reset pulse whose amplitude was randomly assigned a value between 0.9 V and 1.6 V, voltages that roughly correspond to 40% and 70%, respectively, of the average reset switching threshold for the studied crossbars. (Such conservative pulse amplitudes were chosen to avoid excessive stress, which may lead to permanent failure of the devices.) Once again, the UQ significantly improved when higher voltages were applied – from 24.8 ± 6.3% for 200 mV to near ideal 50.07 ± 2.1% for 600 mV (Fig. 4d). The robustness of the rattling strategy is further highlighted in the implementation of larger two-layer PUF architectures in which the basic building blocks of the network were realized using multiple rattled

configurations of the same array distribution on the same crossbar circuit (Section 6 of Supplementary Information).

Performance, robustness, and potentials for improvement

The demonstrated functionality of the model device is a proof of concept for the exciting potential of memristors in cryptography. In particular, the experimentally measured data for uniformity, diffuseness, and uniqueness are very close to the ideal values, which correspond to random binary vectors (Section 3 of Supplementary Information). Although the bit error rate is not negligible, we believe that there are many reserves for its improvement, e.g., increasing the size of the crossbar, averaging over several measurements (Fig. S9), using more sophisticated mapping to avoid defective (e.g., noisy) memristors, and/or using error-correcting codes. Furthermore, our preliminary results regarding the robustness of the demonstrated hardware against modelling attacks are also very encouraging. The output data appear to be very weakly correlated (see Fig. S6 and its discussion), and this is further supported by the successful passing of the NIST randomness test suite. The results of our initial attempt at predicting challenge-to-response function using machine learning techniques, which are becoming mainstream tools for attacking security primitives, [8,41,42] also show very strong resilience to modelling attacks (Section 7A of Supplementary Information). Finally, although we have not measured speed and power consumption directly, in part due to the limitations of the experimental setup, crude estimates show that these metrics can be significantly better for the proposed hardware (Section 5 of Supplementary Information) than those of state-of-the-art implementations based on CMOS circuits at similar feature sizes (Table S3).

One drawback of the demonstrated circuit is the small total number of challenge-response pairs (C_{MAX}). This problem can be readily corrected by increasing the effective crossbar circuit dimensions. For example, scaling up the crossbar to $M = N = 100$ should be relatively straightforward for the considered PUF circuits [43,44] given that the requirements for the memristors, especially the requirement for device-to-device variation, are more relaxed than those for digital or analogue computing applications. (Note that the three-dimensional structure of the crossbar is not essential but is beneficial for PUF robustness because of the smaller voltage drops on the crossbar lines.) For such a larger crossbar circuit, $C_{MAX} > 10^{40}$, e.g., when using $m = 20$ and $n = 20$. Furthermore, to increase throughput, multiple bits can be generated simultaneously using the single block by performing several comparisons in parallel.

A more complex approach that might further improve the robustness of PUF primitives against model-building attacks is the implementation of multilayer PUF networks.[45] For example, Figure S5a shows a two-layer implementation[36] in which the first layer, comprising several basic blocks, generates a hidden challenge (a bit vector), which is then applied to the second layer of the network. With N_{L1} primitives in the first layer and with each block biased with unique voltage (out of N_B total), the total number of unique selections increases exponentially with N_{L1} and according to Eq. (2) is larger than 10^{50} even for the considered $M = 20$, $N = 10$, and practical $m = 10$, $n = 4$, $N_B = 8$, and $N_{L1} = 6$. As discussed earlier, the outputs from different selections can be grouped to make the hidden challenge sufficiently long to feed multiple blocks in the next layer and to produce practically large PUF output. The length of the output vector in a basic primitive can be further increased by considering multiple column selections for the same set of selected rows and/or by generating multiple bits based on the applied voltage biases (see the discussion of quaternary PUF in Section 6 of Supplementary Information). Our initial experimental results for the 2-layer architecture, which has so far been implemented using the inferior rattling reconfiguration strategy, show no evident obstacles towards building practically useful multilayer PUF networks (Fig. S5).

Let us stress again that, unlike previous proposals, our approach takes advantage of memristors' *I-V* nonlinearity, its variations from device to device, and the ability to perform analogue tuning of memristors' *I-V*s. The use of *I-V* nonlinearity naturally increases the complexity of the hardware primitive, making modelling and replication of such a system more challenging compared to purely linear systems. This conclusion is partially supported by the results in Fig. S7, which show higher robustness against modelling attacks at larger voltage biases at which the nonlinearity is the strongest. Analogue tuning is

essential for reducing the correlations between different input-output responses and optimizing the readout margins to improve the bit error rate. In principle, purely digital operation could also be utilized, i.e., only setting memristors to the extreme ON and OFF states, although in this case the response is likely to be dominated by a relatively small number of highly conductive devices, which in turn would create unwanted correlations. Finally, even if an adversary can fully characterize the *I-V*s of all devices, the PUF functionality should be impossible to practically reproduce in hardware because of *I-V* nonlinearity and its unique device-to-device variations. This fact is especially valuable for authentication applications. With moderate scaling of the crossbar circuit to enable a large total number of challenge-response pairs and given its very promising speed and power efficiency, the proposed hardware should also be suitable for key generation.

Conclusions

We have reported the design of a basic building block for hardware-intrinsic security primitives based on two-level stacks of monolithically integrated 10×10 ReRAM arrays and successfully verified its functionality by measuring key security metrics. The security primitives exhibit near ideal diffuseness, uniformity, and uniqueness, as well as a low bit error rate and robustness to machine learning attacks that is encouraging for a prototype. Uniquely, our PUFs make use of the nonlinearities and analogue tuning properties of the integrated memristors. In addition to robust functional performance, the approach offers a number of advantages over previous systems, including configurability, low cost due to the high integration density of its passive memristive crossbar circuits, and suitability for monolithical back-end-of-the-line integration with traditional CMOS circuits. The approach also provides a high-speed and low-energy operation. As a result, such hardware should be appropriate for both authentication and key generation applications.

Acknowledgements

This work was supported by AFOSR under the MURI grant FA9550-12-1-0038 and NSF grant CCF-1528502. We thank An Chen and Jeyavijayan Rajendran for useful discussions.

Contributions

H.N., O.K, and D.B.S. conceived the original concept and initiated the work. G.C.A and B.H. fabricated devices. H.N., M.P., and F.M.B. developed the characterization setup and performed measurements. H.N., J.K., and M.R.M. performed simulations and estimated performance, H.N. and D.B.S. wrote the manuscript. All discussed results.

Competing interests

The authors declare no competing financial interests.

Data availability

The data that support the plots within this paper and other findings of this study are available from the corresponding author upon reasonable request.

References

1 Damiani, E., di Vimercati, S. D. C. & Samarati, P. New paradigms for access control in open environments. In *IEEE International Symposium on Signal Processing and Information Technology* 540-545 (2005).

2 Konstantinou, C. *et al.* Cyber-physical systems: A security perspective. In *IEEE European Test Symposiu* 1-8 (2015).

3 Suh, G. E. & Devadas, S. Physical unclonable functions for device authentication and secret key generation. In *ACM Design Automation Conference* 9-14 (2007).

4 Sadeghi, A.-R. & Naccache, D. *Towards Hardware-Intrinsic Security: Foundations and Practice*. (Springer-Verlag New York, Inc., 2010).

5 Kömmerling, O. & Kuhn, M. G. Design principles for tamper-resistant smartcard processors. *Smartcard* **99**, 9-20 (1999).

6 Pappu, R., Recht, B., Taylor, J. & Gershenfeld, N. Physical one-way functions. *Science* **297**, 2026-2030 (2002).

7 Tehranipoor, M. & Wang, C. *Introduction to hardware security and trust*. (Springer Science & Business Media, 2011).
8 Rajendran, J. *et al.* Nano meets security: Exploring nanoelectronic devices for security applications. *Proc. IEEE* **103**, 829-849 (2015).
9 Hu, Z. *et al.* Physically unclonable cryptographic primitives using self-assembled carbon nanotubes. *Nature Nanotechnology* **11**, 559-565 (2016).
10 Delvaux, J. & Verbauwhede, I. Side channel modeling attacks on 65nm arbiter PUFs exploiting CMOS device noise. In *IEEE International Symposium on Hardware-Oriented Security and Trust* 137-142 (2013).
11 Herder, C., Yu, M.-D., Koushanfar, F. & Devadas, S. Physical unclonable functions and applications: A tutorial. *Proc. IEEE* **102**, 1126-1141 (2014).
12 Ruhrmair, U. & Holcomb, D. E. PUFs at a glance. In *Design, Automation and Test in Europe Conference and Exhibition* 1-6 (2014).
13 Yu, M.-D. M., Sowell, R., Singh, A., M'Raïhi, D. & Devadas, S. Performance metrics and empirical results of a PUF cryptographic key generation ASIC. In *IEEE International Symposium on Hardware-Oriented Security and Trust* 108-115 (2012).
14 Roel, M. *Physically unclonable functions: Constructions, properties and applications*, PhD Thesis, University of KU Leuven, (2012).
15 Herder III, C. H. *Towards security without secrets*, PhD Thesis, Massachusetts Institute of Technology, (2016).
16 Kang, H., Hori, Y., Katashita, T., Hagiwara, M. & Iwamura, K. Cryptographie key generation from PUF data using efficient fuzzy extractors. In *IEEE International Conference on Advanced Communication Technology.* 23-26 (2014).
17 Lim, D. *et al*. Extracting secret keys from integrated circuits. *IEEE Transactions on Very Large Scale Integration Systems* **13**, 1200-1205 (2005).
18 Maes, R., Van Herrewege, A. & Verbauwhede, I. PUFKY: a fully functional PUF-based cryptographic key generator. In *International Workshop on Cryptographic Hardware and Embedded Systems.* 302-319 (2012).
19 Rührmair, U. & van Dijk, M. PUFs in security protocols: Attack models and security evaluations. In *IEEE Symposium on Security and Privacy.* 286-300 (2013).
20 van Dijk, M. & Rührmair, U. Physical unclonable functions in cryptographic protocols: security proofs and impossibility results. *IACR Cryptology ePrint Archive* **2012**, 228 (2012).
21 Zhang, L., Kong, Z. H. & Chang, C.-H. PCKGen: A phase change memory based cryptographic key generator. In *IEEE International Symposium on Circuits and Systems* 1444-1447 (2013).
22 Ranasinghe, D. C. & Cole, P. H. Confronting security and privacy threats in modern RFID systems. In *IEEE Fortieth Asilomar Conference on Signals, Systems and Computers* 2058-2064 (2006).
23 Devadas, S. *et al.* Design and implementation of PUF-based" unclonable" RFID ICs for anti-counterfeiting and security applications. In *IEEE International Conference on RFID* 58-64 (2008).
24 Cole, P. H. & Ranasinghe, D. C. *Networked RFID systems and lightweight cryptography*. vol. 10 (Springer-Verlag Berlin Heidelberg, 2008).
25 Waser, R., Dittmann, R., Staikov, G. & Szot, K. Redox-based resistive switching memories - nanoionic mechanisms, prospects, and challenges. *Adv. Mater.* **21**, 2632-2663 (2009).
26 Yang, J. J., Strukov, D. B. & Stewart, D. R. Memristive devices for computing. *Nature Nanotechnology* **8**, 13-24 (2013).
27 Zhang, L., Fong, X., Chang, C.-H., Kong, Z. H. & Roy, K. Feasibility study of emerging non-volatilememory based physical unclonable functions. In *IEEE 6th International Memory Workshop* 1-4 (2014).
28 Liu, R., Wu, H., Pang, Y., Qian, H. & Yu, S. A highly reliable and tamper-resistant RRAM PUF: Design and experimental validation. In *IEEE International Symposium on Hardware Oriented Security and Trust* 13-18 (2016).
29 Chen, A. Comprehensive assessment of RRAM-based PUF for hardware security applications. In *IEEE International Electron Devices Meeting* 10.17.11-10.17.14 (2015).
30 Gao, Y., Ranasinghe, D. C., Al-Sarawi, S. F., Kavehei, O. & Abbott, D. Memristive crypto primitive for building highly secure physical unclonable functions. *Scientific reports* **5** (2015).
31 Rajendran, J., Rose, G. S., Karri, R. & Potkonjak, M. Nano-PPUF: A Memristor-Based Security Primitive. In *IEEE Computer Society Annual Symposium on VLSI.* 84-87 (2012).
32 Chang, S. H. *et al.* Oxide double-layer nanocrossbar for ultrahigh-density bipolar resistive memory. *Adv. Mater.* **23**, 4063-4067 (2011).
33 Gao, L., Chen, P.-Y., Liu, R. & Yu, S. Physical unclonable function exploiting sneak paths in resistive cross-point array. *IEEE Trans. Electron Devices* **63**, 3109-3115 (2016).
34 Moors, M. *et al.* Resistive switching mechanisms on TaO_x and $SrRuO_3$ thin-film surfaces probed by scanning tunneling microscopy. *ACS Nano* **10**, 1481-1492 (2016).
35 Rührmair, U. *et al.* Applications of high-capacity crossbar memories in cryptography. *IEEE Trans. Nanotechnology* **10**, 489-498 (2011).
36 Kim, J. *et al.* A Physical Unclonable Function with Redox-based Nanoionic Resistive Memory. *IEEE Tran. Information Forensics and Security*, doi:10.1109/TIFS.2017.2756562 (2017).
37 Che, W., Plusquellic, J. & Bhunia, S. A non-volatile memory based physically unclonable function without helper data. In *2014 IEEE International Conference on Computer-Aided Design* 148-153 (2014).
38 Rose, G. S. & Meade, C. A. Performance analysis of a memristive crossbar PUF design. In *IEEE Design Automation Conference* 1-6 (2015).

39 Adam, G. C. *et al.* 3-D memristor crossbars for analog and neuromorphic computing applications. *IEEE Trans. Electron Devices* **64**, 312-318 (2017).

40 Alibart, F., Gao, L., Hoskins, B. D. & Strukov, D. B. High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm. *Nanotechnology* **23**, 075201 (2012).

41 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).

42 Xu, X. & Burleson, W. Hybrid side-channel/machine-learning attacks on PUFs: a new threat? In *IEEE Design, Automation & Test conference in Europe* 349 (2014).

43 Bayat, F. M., Prezioso, M., Chakrabarti, B., Kataeva, I. & Strukov, D. Advancing memristive analog neuromorphic networks: increasing complexity, and coping with imperfect hardware components. *arXiv preprint arXiv:1611.04465* (2016).

44 Kim, K.-H. *et al.* A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications. *Nano lett.* **12**, 389-395 (2011).

45 Holcomb, D. E. & Fu, K. Bitline PUF: building native challenge-response PUF capability into any SRAM. In *International Workshop on Cryptographic Hardware and Embedded Systems* 510-526 (2014).

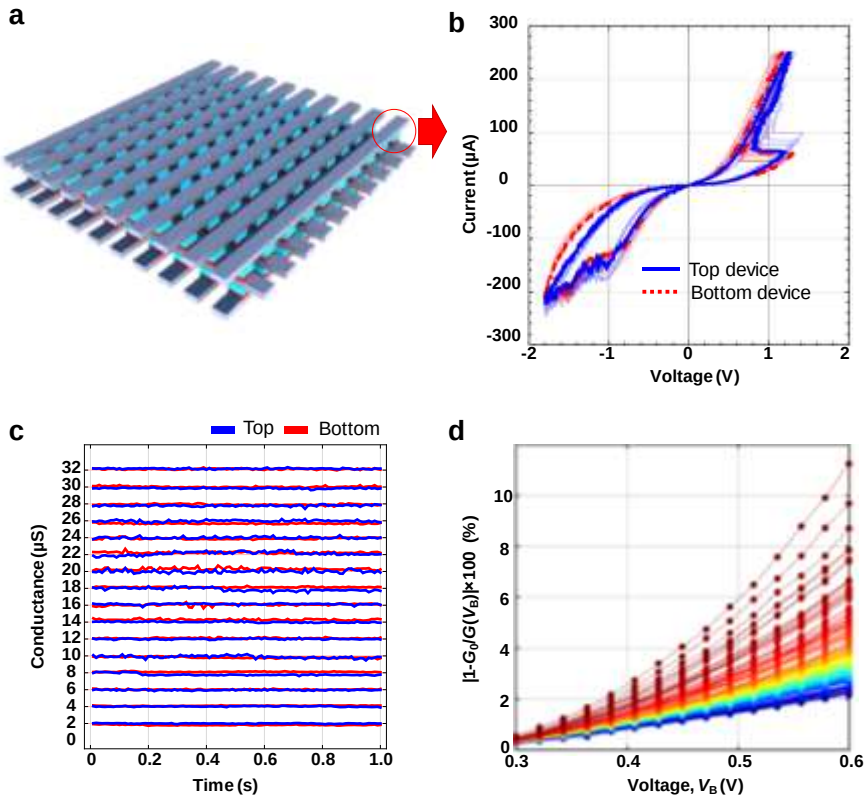


Fig. 1. 3D ReRAM crossbar array. (a) Cartoon of the fabricated circuit. (b) *I*-*V* curves for all 2×10×10 devices; two representative curves are highlighted for comparison. (c) Tuning of the top and bottom devices to 16 different conductive states that are equally spaced from 2 μS to 32 μS. (d) Nonlinearity factor calculated as a ratio of $|1-G_0/G(V_B)|$ for all 200 devices, which were tuned to $G_0 = 4.5 \pm \sim 1$ μS at 200 mV. For convenience, the curves are coloured according to the observed nonlinearity at the highest voltage bias.

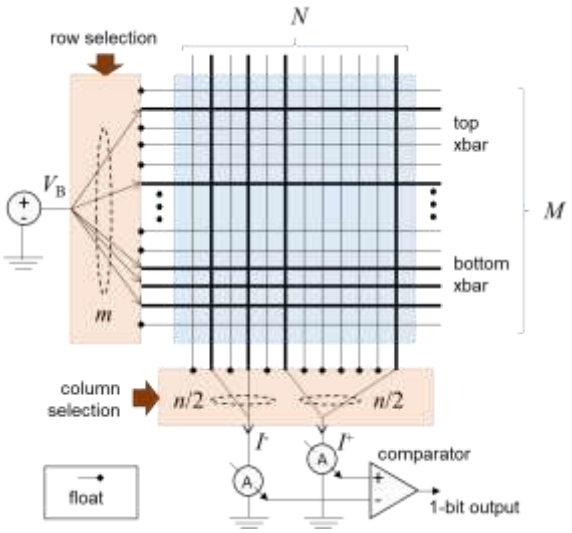


Fig. 2. Memristor-based basic building block for cryptographic hardware. One-bit output is generated by applying a voltage bias to *m* rows (of *M* total) and then comparing the total currents running into the two selected groups comprised of *n* columns (of *N* total). In the simplest implementation, the unselected rows and columns in the array are kept floating.

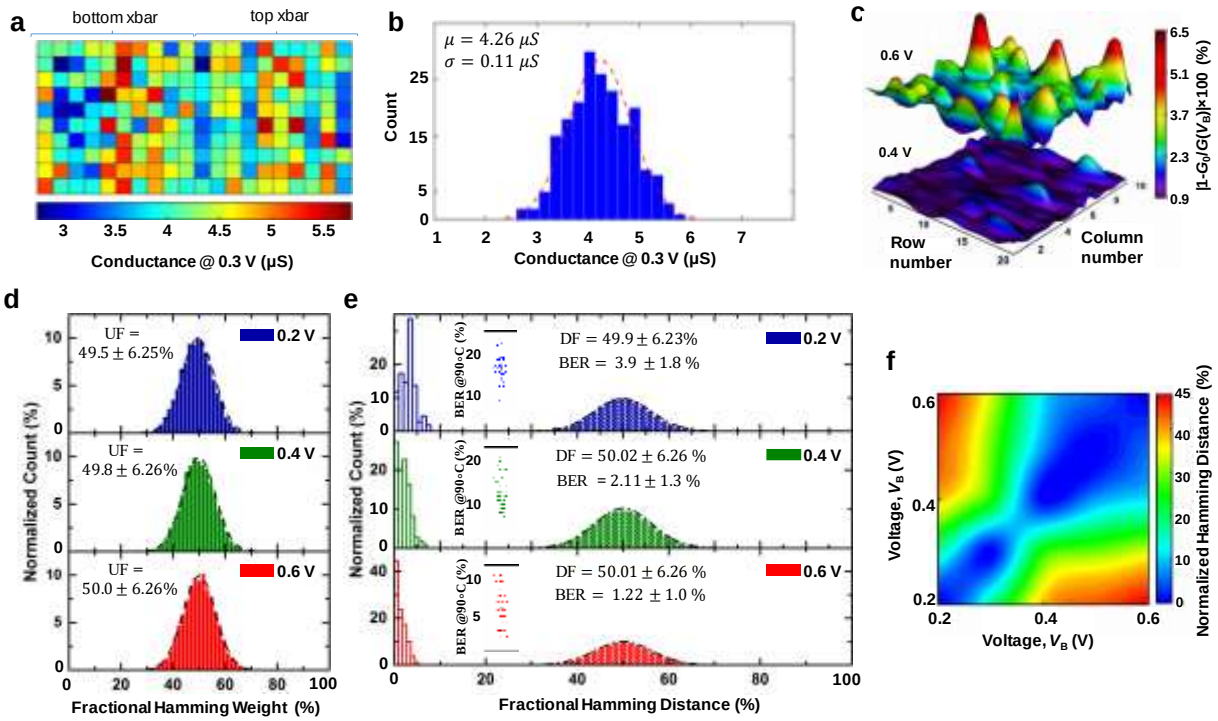


Fig. 3. Experimental results for tuning and security performance. (a) Conductance map (G_0). (b) Corresponding histogram. (c) Nonlinearity factor for two values of V_B for all 200 devices after tuning. In panel b, the dashed line is a guide showing a Gaussian distribution. (d) Uniformity and (e) diffusiveness and bit error rate calculated according to Eqs. S1a, S2a, and S4a. The bit-error rates are calculated by monitoring 16,000 representative challenge-response pairs over a 30-day window in 10-day intervals. To account for ageing and environmental factors, the voltage bias at each measurement was randomly selected from the range $[0.8, 1.2]\times V_B$, which is representative of up to 20% noise on the power supply. The inset shows the bit error rate relative to room temperature for 4,800 challenge-response pairs at 90°C at three different biases. The bars show the 5-95 percentile. The temperature was slowly ramped up to the target value and was kept constant for a period of 30 minutes before measurement was performed over a period of 3 hours. (f) Contour map of the uniqueness between the responses generated using the same challenges at different voltage biases, calculated according to Eq. S3c. (More detailed results for several specific biases are shown in Fig. S3d.)

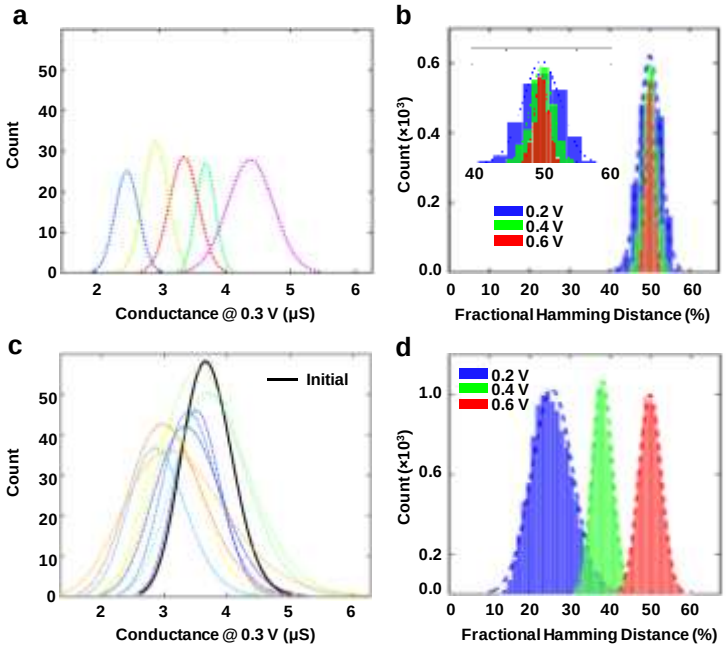


Fig. 4. Experimental results for PUF uniqueness. (a) Conductance distributions after tuning for 5 different PUF instances and (b) the corresponding uniqueness. The measured average and standard deviation are 49.95 ± 2.65%, 49.94 ± 1.75%, and 49.96 ± 0.9% for V_B = 200 mV, 400 mV, and 600 mV voltage biases, respectively. The inset shows a zoom-in view of the data. (c) The conductance distributions after rattling for 10 different PUF instances and (d) their corresponding uniqueness. The measured average and standard deviation are 24.8 ± 6.3%, 38.2 ± 3.3%, and 50.07 ± 2.1% for V_B = 200 mV, 400 mV, and 600 mV voltage biases, respectively. All data for uniqueness are calculated according to Eq. S3b.

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 ~350 × 350 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 ~750-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:

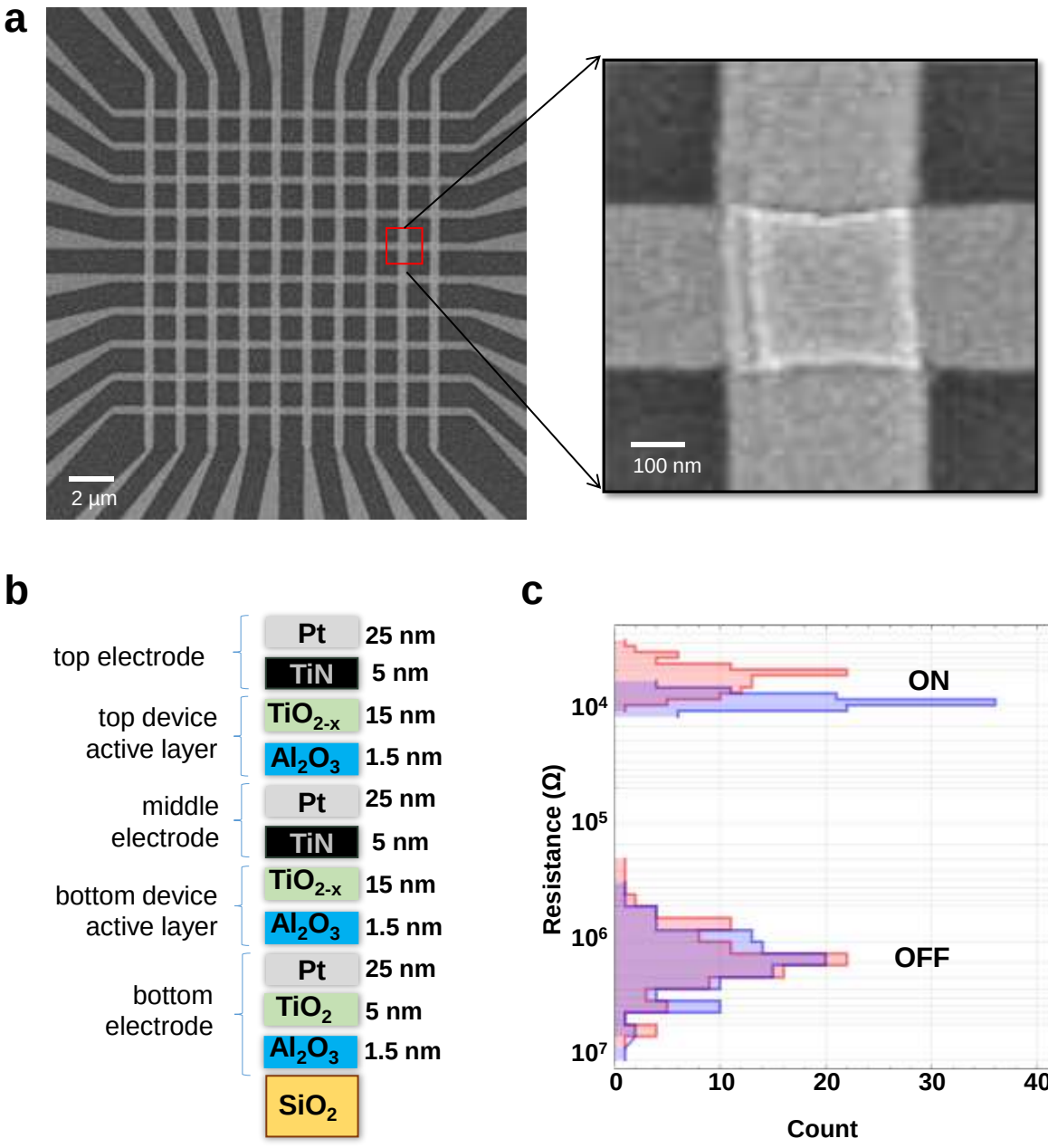
Step 1: The very first step is to generate S random sets of row and column selections; these are denoted S_R and $S_C^{\pm}$, respectively. Each selection comprises indexes of the selected 5 rows and 2 columns, and each set of indexes is unique. The typical value of S is 10,000. The values of μ_I, σ_I, 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 $I + \Delta_I + 4.5\ [\mu A] \times \text{Beta}(2, 25)$, where Beta() is a beta distribution of the first kind with shape parameters $\alpha = 2$ and $\beta = 25$.

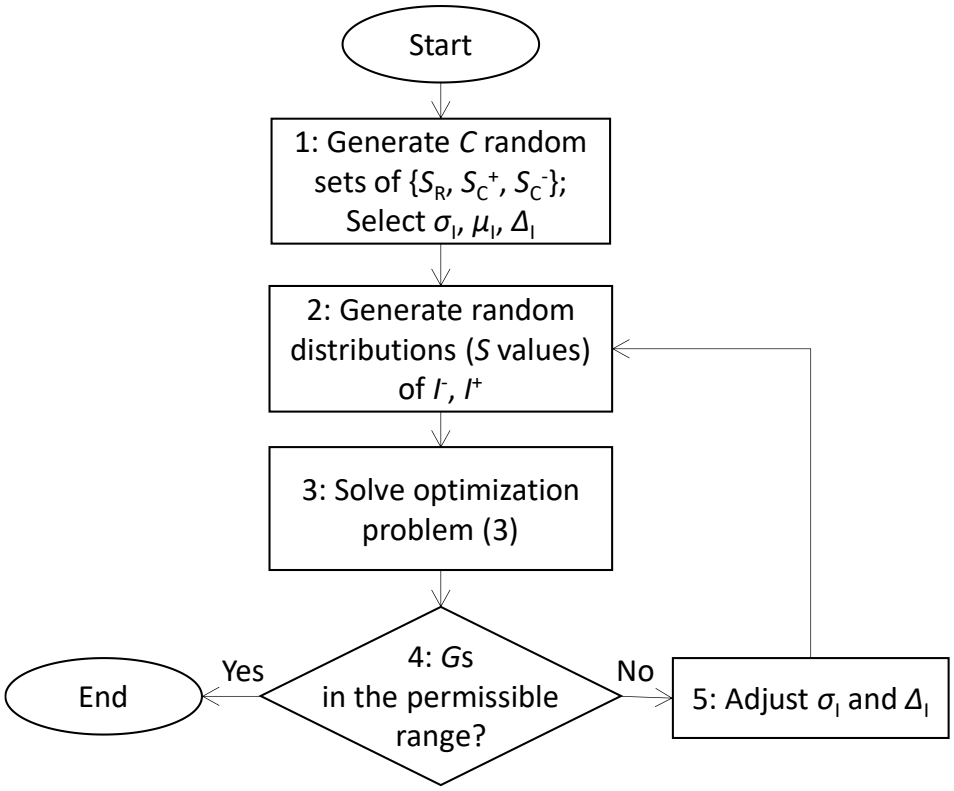
Step 3: The non-negative least squares optimization problem defined by Equation (3) in the main text is solved with the help of Matlab software.

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

Step 5: The σ_I 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_I, σ_I, and Δ_I, 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

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

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

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

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*

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

The average diffuseness accounting for all possible pairwise comparisons is therefore

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

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:

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

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

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

whereas uniqueness averaged over all responses is

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

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.,

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

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

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

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.,

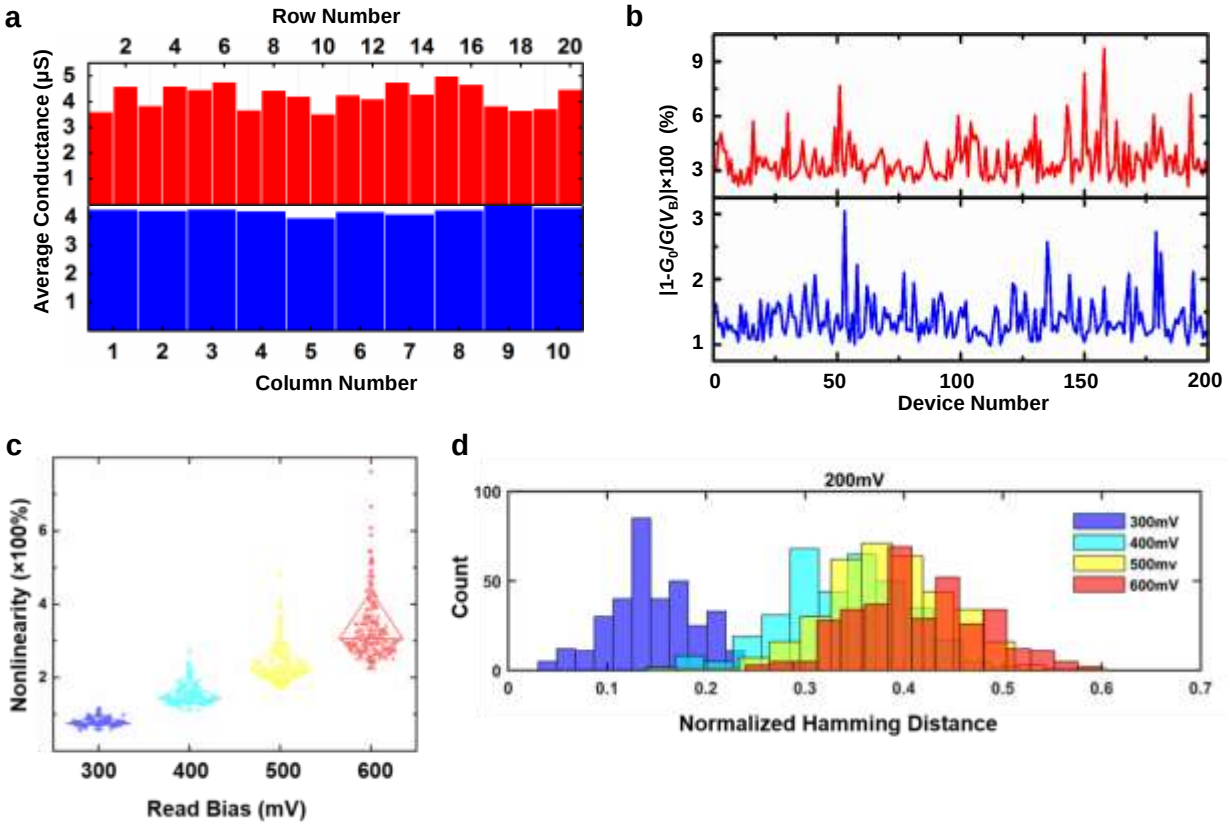
$$\langle \mathrm{DF}(B_i) \rangle \equiv \frac{1}{C} \sum_{j=1}^{C} \mathrm{DF}(B_i, B_j) \tag{S5}$$

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

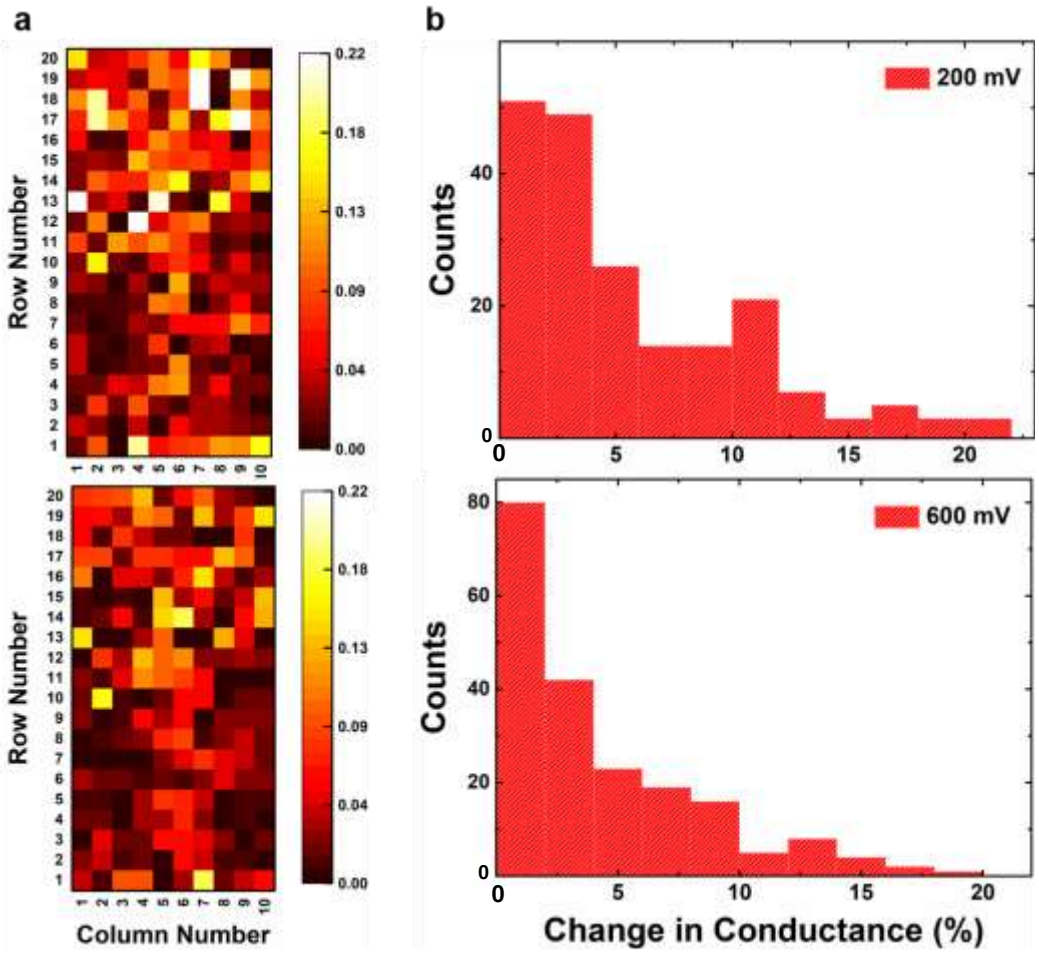
lower than that reported for DF(B_i, B_j). Similarly, the average and standard deviation for < UF(B_i)> defined by Eq. S3b are 0.5 and $\sqrt{0.5/(KP(P-1))}$, respectively.

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 ^{O}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 ^{O}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.

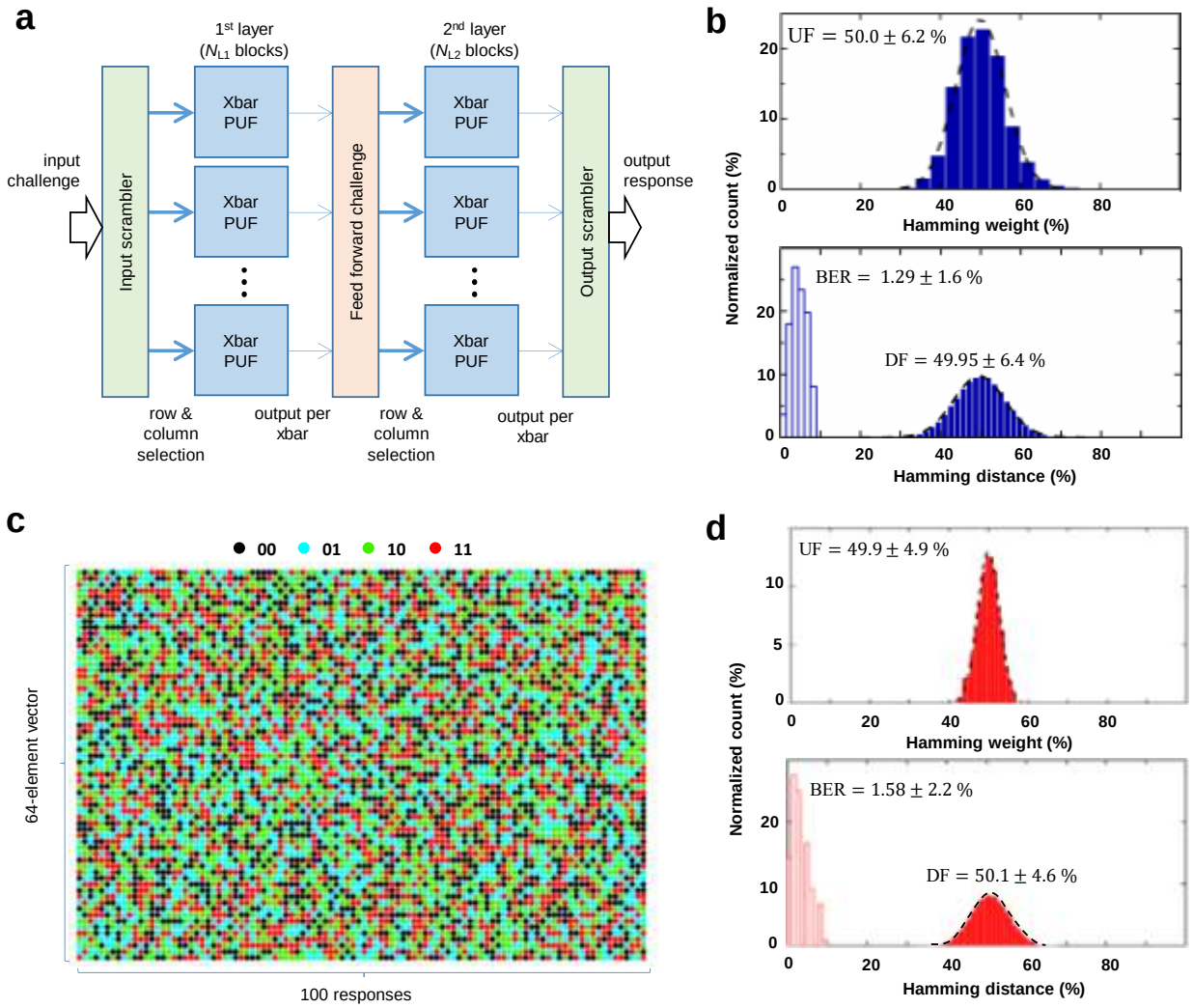
According to our previous work on mixed-signal vector-by-matrix multipliers,[4] the area, maximum settling time and power consumption of a single differential sensing circuit implemented in a 55-nm process are 10 μm^2, 4 ns, and 2.5 μW, respectively, assuming that the maximum and minimum input currents 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 fF/ 1 μm crossbar line capacitance,[5] which results in ~10 fJ per bit. Neglecting the contributions from other circuitry, the total area, latency, and energy consumption for generating one output bit are ~20 μm^2, < 5 ns, and ~20 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.

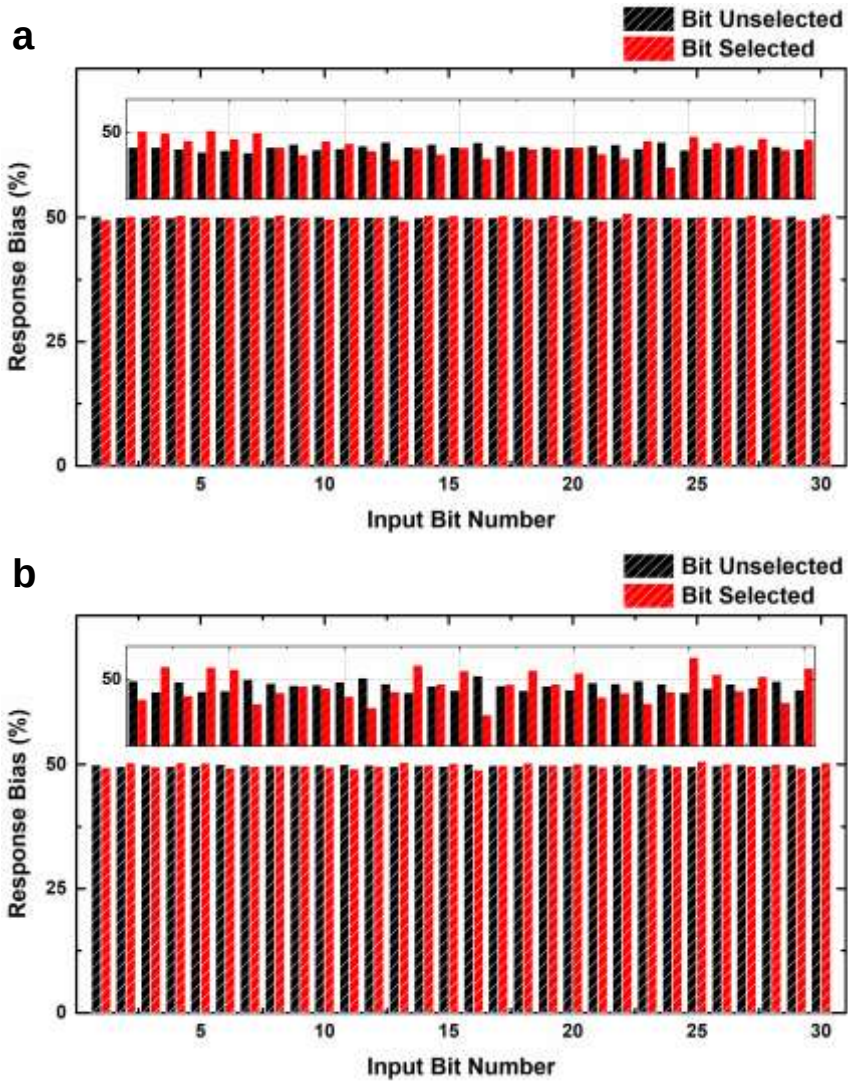
Figure S5b-d shows the experimental results for uniformity and bit error rate for the two considered cases, measured by collecting 500 64-bit and 500 128-bit responses, respectively, for randomly selected mutually exclusive challenges. In the first case, a sequence of 64 selections with each input selection 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 ceiling[$\log_2 C_{MAX}$]= 20 bits.)

A. Correlations

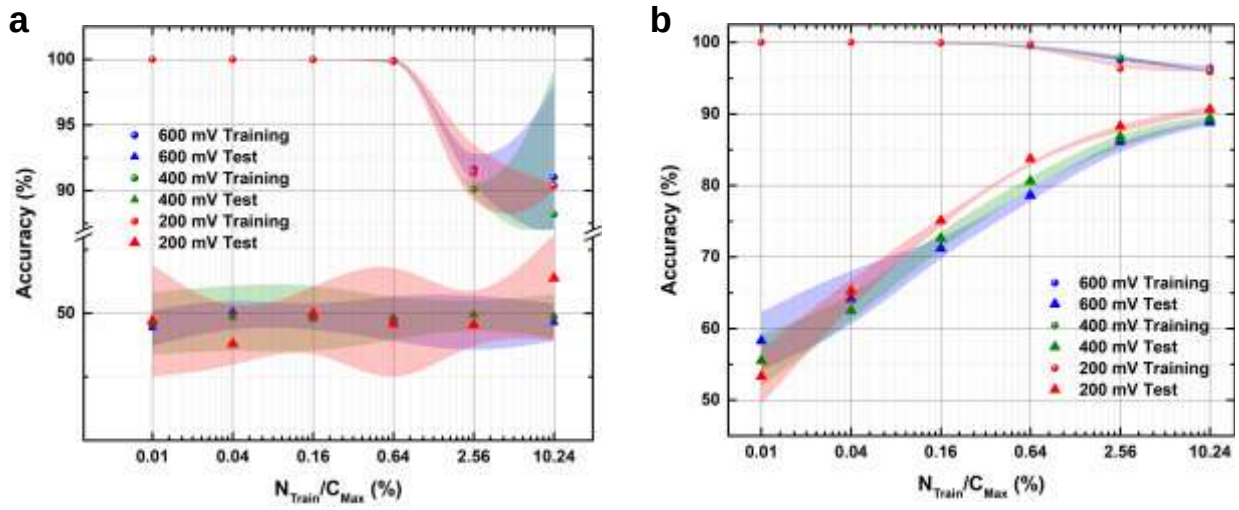
In our first test, we probed for possible bias in the output by checking the uniformity of the response when a particular bit of the challenge bit vector is fixed (Fig. S6a). The uniformity is close to the ideal (50%) for both experiments, though the results are visibly somewhat worse for the second 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 suite[6] 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]



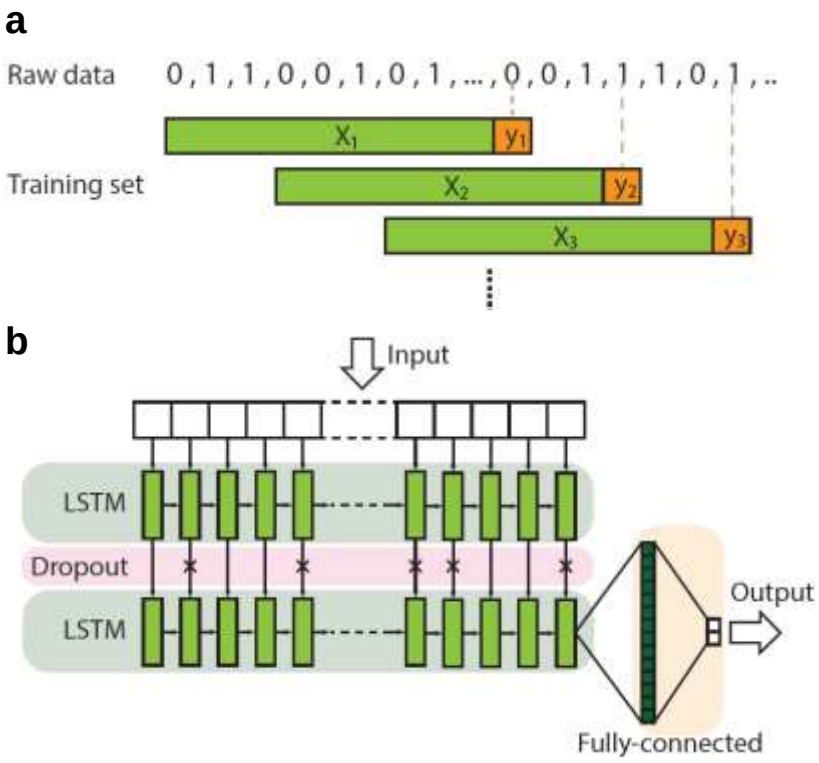
Supplementary Figure 7. Robustness to machine learning attacks for (a) near-optimal and (b) suboptimal PUF simulated utilizing a 30×250×250×1 multilayer perceptron classifier. The markers denote the average classification accuracy over 10 runs; the thickness of the lines for the test data specifies two 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 ≥ 0.0001	95.95	All ≥ 0.0001	100	All ≥ 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 Graves[7] (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 $\underline{N}$ 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.

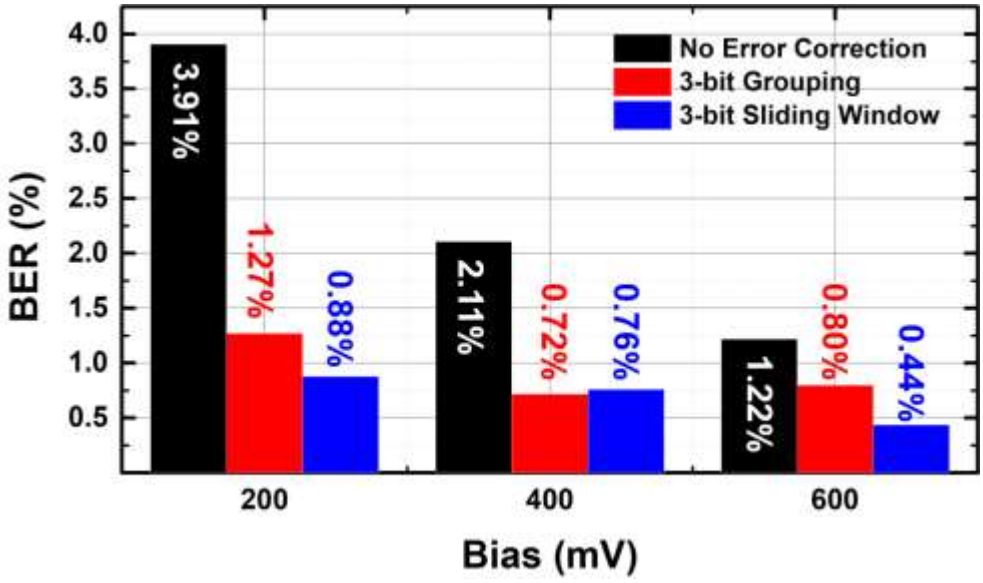
Supplementary Table 2. Machine learning attack results using the LSTM-Dropout-LSTM-Dense-Dense-Softmax network.

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_{ON} 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 Thesie, 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).