ARTICLES
https://doi.org/10.1038/s41928-018-0146-5
nature electronics

A provable key destruction scheme based on memristive crossbar arrays

Hao Jiang[1,2], Can Li[1,2], Rui Zhang[1], Peng Yan[1], Peng Lin[1], Yunning Li[1], J. Joshua Yang[1*], Daniel Holcomb[1*] and Qiangfei Xia[1*]

Digital keys are commonly used in today's hardware security systems. However, the provable destruction of these keys after use remains a challenging problem. Most security primitives built using traditional complementary metal–oxide–semiconductor transistors are not well suited to address this issue because of their volatility and unreliability at small scales. Here we show that the unique physical fingerprint of a 128 × 64 hafnium oxide memristor crossbar array integrated with transistors is capable of provable key destruction. The fingerprint is extracted by comparing the conductance of neighbouring memristors, and it can be revealed only if a digital key stored on the same array is erased. On the basis of this provable key destruction technique, we propose a protocol for logic locking/unlocking that can support secure outsourcing of integrated circuit manufacturing. By leveraging the unique properties of memristors, including reconfigurability and variability, our chip demonstrates the integration of security, memory and computing functionalities into the same circuits, and could be used to develop more secure, compact and efficient memristive hardware systems.

Inspired by the 4,000-year-old lock and key scheme[1], digital keys remain fundamental to aspects of security in the era of the Internet of Things[2–6]. Stored keys are frequently used in today's security system to unlock certain functionalities on chips/devices, and to encrypt and decrypt data in a variety of electronic circuits/chips. However, once the user's key-based permissions are revoked or forfeited, the digital key should be erased. Proving that the key has really been erased and whether such erasure is done in the desired chip/device—that is, achieving provable key destruction—is difficult. In the semiconductor industry, logic locking[7,8] is a common technique to mitigate threats including intellectual property (IP) theft, counterfeiting and unauthorized overproduction during the outsourcing of chip fabrication to foundries worldwide. This is currently achieved by adding extra logic gates (key gates), which enable the chip to function correctly only after the designer unlocks these gates with a universal unlocking key. However, the universal unlocking key may be permanently stored in each device, which means that once a device has the key it can forever unlock the logic, voiding the controlling capability of the service provider or chip designer.

Traditional complementary metal–oxide–semiconductor (CMOS)-based hardware security primitives have been extensively studied[9–13]. However, secure key destruction remains a challenge and current CMOS-based security primitives are also vulnerable to side-channel and modelling attacks that exploit information leakage from physical measurements or subtly predictable behaviours respectively[14]. To address these challenges, there is growing interest in developing security primitives based on emerging electronic devices[6], such as memristive devices, which are two-terminal resistance switches. These devices offer great scalability[15,16], CMOS compatibility[17], fast switching speed[18], high endurance[19] and low power consumption[20]. In addition to memory, data storage and unconventional computing applications[21–24], memristors have also been used to build security primitives such as physical unclonable functions[25–31] and true random number generators (TRNGs)[32–36]. Security primitives based on memristor technology take advantage of the intrinsic variations in the switching characteristics of the devices. However, in these implementations, the security module and the memory/computing modules are usually separated, which results in a larger chip area and higher energy consumption when the key is used frequently (because of the energy cost of shuttling information between the modules). Furthermore, provable key destruction with memristor security primitives has not been demonstrated so far, despite the desire for security protocols with erasable physical unclonable function responses[37].

In this Article, we report a provable key destruction scheme based on memristive devices. The security primitive entangles fingerprints and secret keys in a 128 × 64 Ta/HfO_2 memristor crossbar array. The fingerprints are generated by comparing the conductance of pairs of neighbouring cells in the crossbar array when both cells are in the low-resistance states (LRSs). The keys are written to the crossbar array over the embedded memristor fingerprint. The successful extraction of the memristor fingerprints from the crossbar array confirms the key erasure. Furthermore, the intrinsic memristor fingerprints are unique and reliable to each chip/device (as characterized by their interclass and intraclass Hamming distances across switching cycles) and can subsequently be used to attest that the key destruction has occurred on the specific chip/device. To demonstrate provable key destruction for practical problems, we provide a protocol that uses it to implement relockable logic locking/unlocking. Our approach offers a simple, yet efficient, solution to an open challenge in the hardware security community, and its potential to integrate memory, computing and security functionalities in the same circuits could lead to more compact and efficient memristive hardware systems[38].

Fingerprint generation and provable key destruction

Our provable key destruction idea is to store the key in a way that obscures a chip's fingerprint, so that a device can prove destruction of a key by reproducing the fingerprint. The keys that can be used to unlock functionalities on chips/devices are stored in

[1]Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, USA. [2]These authors contributed equally: Hao Jiang and Can Li. *e-mail: jjyang@umass.edu; dholcomb@umass.edu; qxia@umass.edu

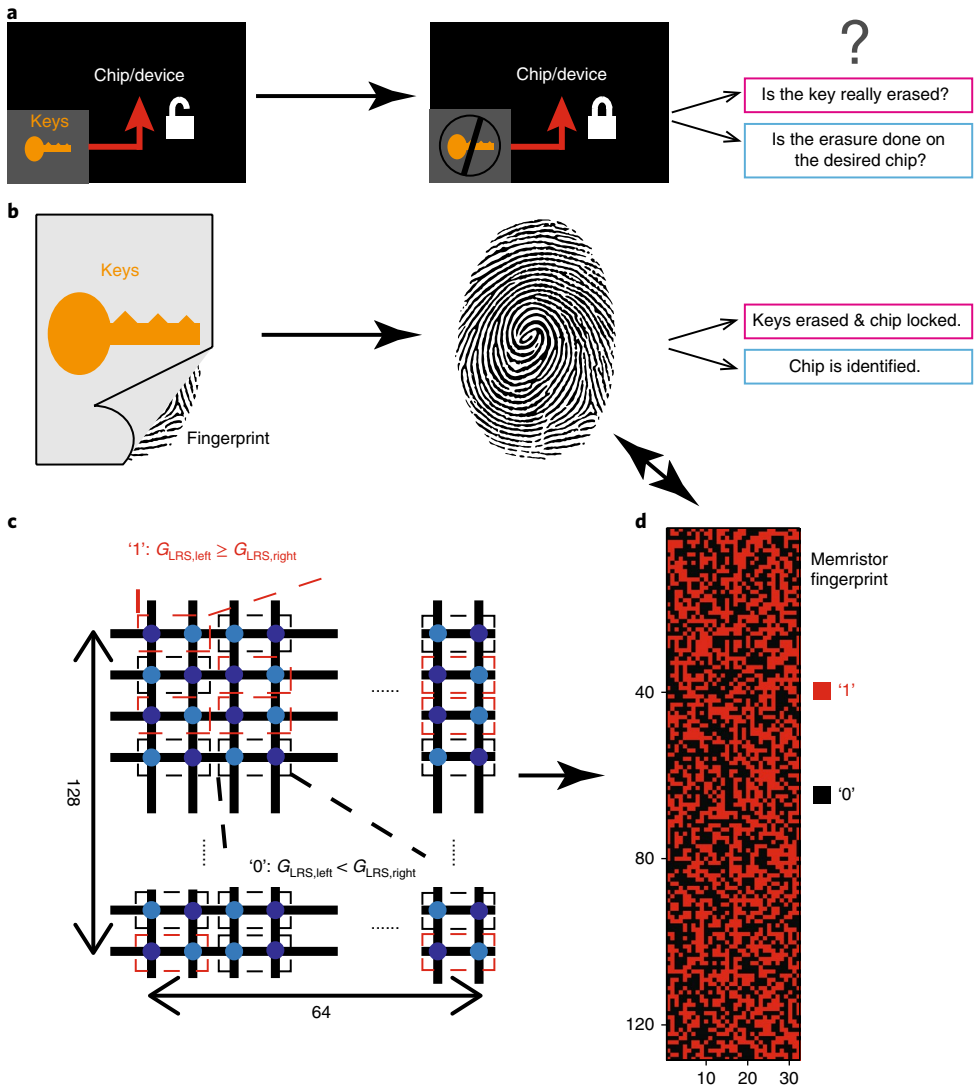

Fig. 1 | The principle of provable key destruction. a, A stored key is used to unlock functionalities on chips/devices, while key destruction brings security concerns as listed. **b**, Our approach for provable key destruction with memristor crossbar arrays. The digital keys are stored over the embedded memristor fingerprint of a chip/device. The successful extraction of the persistent memristor fingerprint confirms that the keys have been erased and attests the identity of the chip that erased the key, since the fingerprint can be extracted only after key erasure and is reliable and unique to each chip/device. **c**, Schematic of fingerprint extraction by comparing LRS conductance between two neighbouring memristor cells (differential pairs). Each cell is included in only one differential pair. After all cells are programmed to LRS (V_{SET}: 2.5 V, 500 µs; V_G: 1.1 V), the fingerprint bit from a differential pair is read as a '1' if the conductance of the left cell ($G_{LRS,left}$) is greater than or equal to that of the right cell ($G_{LRS,right}$), and otherwise is read as a '0' if $G_{LRS,left} < G_{LRS,right}$. **d**, A typical 128 × 32 fingerprint generated through the above-described conductance comparison approach in a 128 × 64 memristor crossbar array.

memristor crossbar arrays over the embedded memristor fingerprints (Fig. 1a,b). Only after key erasure, the memristor fingerprints can be extracted (Fig. 1b). The unique fingerprint can not only be used to lock a chip but can be used to identify the chip.

In practice, all memristors in the crossbar array are programmed to LRS using electrical pulses (see Methods). The conductance values of two memristors (a differential pair) in the neighbouring columns are read and compared (Fig. 1c) to generate a digitized fingerprint bit. The fingerprint bit is a '1' if the conductance of the left cell (G_{left}) is greater than or equal to that of the right one (G_{right}), or a '0' if G_{left} is lower. Some differential pairs are able to produce reliable '0' or '1' bits in all trials while others are not. The statistical analysis on different types of differential pairs can be found in Supplementary Fig. 1 and Supplementary Note 1.

To ensure the unpredictability of fingerprint bits, each memristor cell is used in only one differential pair. As a result, a 128 × 32 fingerprint can be generated from the 128 × 64 physical crossbar array (Fig. 1d). For simplicity, the data processing (comparison of conductance) is performed through an off-chip system, which can be integrated with the memristor crossbar array in the future. It should also be noted that there may be alternative ways to compare the conductance within each differential pair, or other approaches to generate the fingerprint, but the memristor fingerprint should be unique to each chip.

Uniqueness and reliability of the fingerprints

We monolithically integrated Ta/HfO_2/Pt memristors[39] with a foundry-made transistor array into a 128 × 64 one-transistor one-resistance-switch (1T1R) crossbar array (Fig. 2a,b and Supplementary Fig. 2a). The detailed integration process can be found in ref. [22]. We connected the crossbar array through a probe card to a custom-made measurement system[40] that is capable of

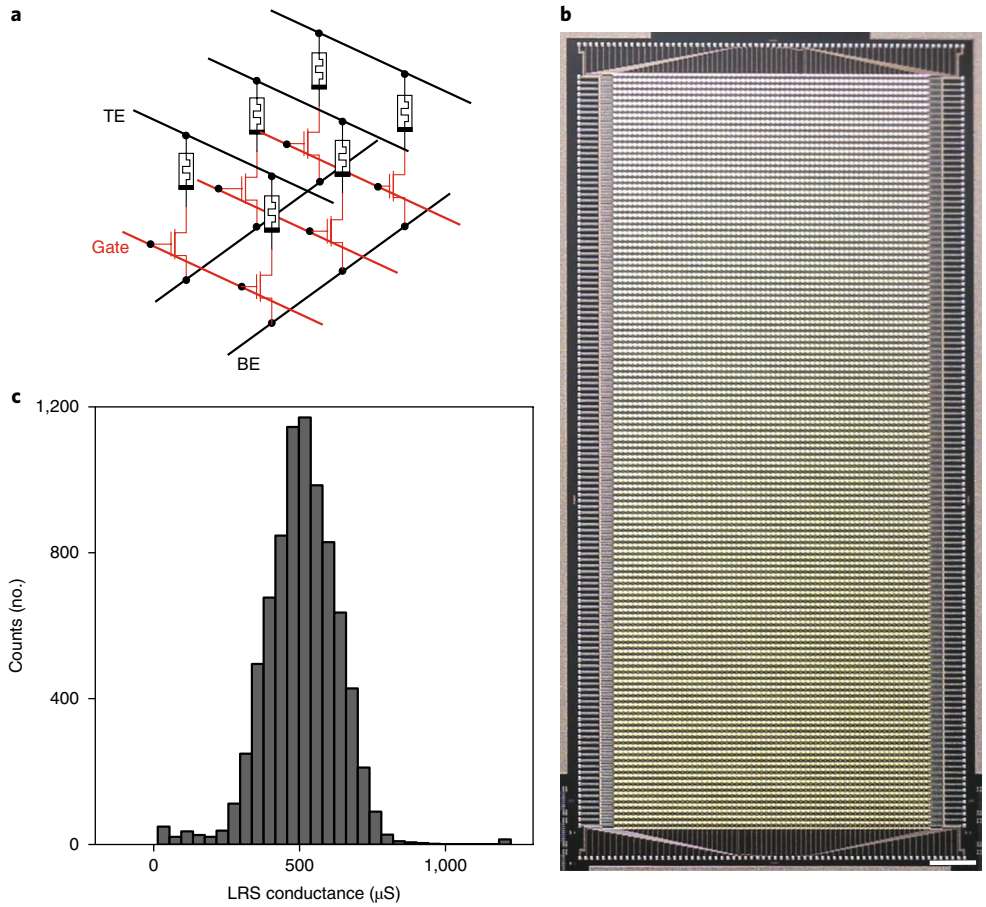

Fig. 2 | 128 × 64 one-transistor one-Ta/HfO2/Pt-memristor (1T1R) array. a, A schematic of the 1T1R configuration in which memristor cells are connected in series with transistors. In the 1T1R crossbar array, rows share BE lines while columns share TE lines and transistor gate lines. **b**, An optical image of a 128 × 64 1T1R crossbar array. Scale bar, 500 μm. **c**, Distribution of LRS conductance from the entire array (V_{SET}: 2.5 V, 500 μs to Ta TEs and V_G: 1.1 V to the transistor) after SET operation.

parallel and precise conductance tuning for all memristors in the array. A typical current–voltage (I–V) switching curve is shown in Supplementary Fig. 2b. With a gate voltage (V_G) of 1.1 V to the transistor and a positive voltage pulse (V_{SET}: 2.5 V, 500 μs) to the top electrode (TE) line (column wire) while the bottom electrode (BE) line (row wire) is grounded, a memristor is SET to its LRS. The LRS conductances of the cells in the array follow a normal distribution, with a mean of 503 μS and a standard deviation of 132 μS (Fig. 2c).

We found that the fingerprint is reliable and unique to each chip. The uniqueness is characterized by the interclass Hamming distance that represents the number of different bits between fingerprints from two different locations, while the reliability is measured by the intraclass Hamming distance that compares the bits extracted from the same locations in different trials. We switched all of the 8,192 devices in the array between the LRS and high-resistance state (HRS) for 200 cycles and extracted the fingerprints after each SET process using the method described in Fig. 1c. Each pair of columns generates a 128-bit fingerprint. Figure 3 shows the distribution of normalized interclass and intraclass Hamming distances of the 128-bit fingerprints collected from a total of 5 arrays from 5 chips made in different batches. The interclass fractional Hamming distance centres at 0.5006 with a standard deviation of 0.0452, suggesting excellent uniqueness of our memristor fingerprint. The mean of the intraclass Hamming distance collected from 100 switching cycles is 0.1382 and the standard deviation is 0.0657, showing the reliability of our memristor array fingerprint. A designer can use a distance threshold for deciding whether two fingerprints come from the same memristor crossbar array, and can decide on that threshold on the basis of relative willingness to accept possible false positive or false negative matches. As the size of the fingerprints increases, it becomes easier to select thresholds that entirely eliminate both false positives and false negatives[9], as shown in Supplementary Fig. 3. The LRS of our Ta/HfO_2 has a linear I–V relationship[39] (the conductance values read at different voltages are the same) and is resistant to temperature effects (Supplementary Fig. 4), suggesting that the memristor fingerprint is resilient to possible environmental variations (for example, voltage and temperature). Our Ta/HfO_2 memristors were integrated on top of a custom-designed transistor array with commercial-fab-made metal wires, the interconnect resistance of which is very low (0.35 Ω per block for rows, 0.32 Ω per block for columns) and hence the previously reported effect of path resistance[41] is not a concern for us. The above results confirm that our memristor fingerprint satisfies all of the requirements and the successful experimental demonstration of the provable key destruction can be expected.

Experimental demonstration of provable key destruction

To demonstrate the provable key destruction, we fabricate the chip, set all memristors to LRS, and extract its embedded physical fingerprint (FP_{chip}, Fig. 4a) that can be stored in a trusted database. We then write a unique random key ($K_{chip,}$ Fig. 4b) to the memristor array with both LRS and HRS values over the fingerprint (so that the fingerprint cannot be reproduced without destroying K_{chip}).

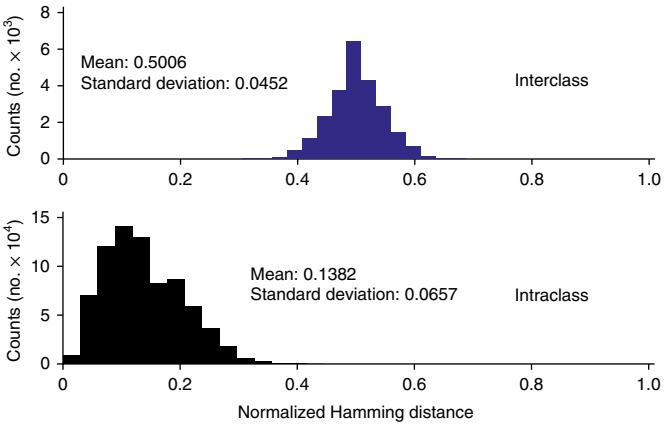

Fig. 3 | Unique and reliable fingerprints in large memristor crossbar arrays. The distribution of interclass and intraclass Hamming distances of 128-bit fingerprints. Two neighbouring columns can generate one 128-bit fingerprint in each switching cycle. The fractional interclass Hamming distance collected from 5 chips has a mean of 0.5006 and standard deviation of 0.0452, suggesting the great uniqueness of the memristor fingerprint. In total, we are able to obtain 154 different 128-bit fingerprints across 5 chips in each switching cycle (31, 32, 28, 32 and 31 on each of the 5 respective chips; some unresponsive columns resulting from poor probe landing were not used). The interclass results of 128-bit fingerprints collected from 5 chips across 2 switching cycles are based on 23,562 counts. The mean of the fractional intraclass Hamming distance is 0.1382 with a standard deviation of 0.0657, confirming the reliability of our memristor fingerprint. The intraclass results of 128-bit fingerprints collected from 5 chips across 100 switching cycles are based on 762,300 counts. Detailed information about all five chips and the statistical results can be found in Supplementary Table 1.

The digital key (K_{chip}) can be generated by any kind of TRNG on the chip such as the one based on a diffusive memristor[36]. The value of K_{chip} is also sent to the trusted party so that it can later be used to unlock functionalities of the specific chip instance that stores K_{chip}. The user at some point may want to destroy K_{chip} in a way that can be verified by the trusted party. Under this circumstance, the user issues an 'erase-key' command that tells the memristor array to switch all devices to LRS to generate a new fingerprint (FP'_{chip}) (Fig. 4c). FP'_{chip} is communicated to the trusted party as proof that K_{chip} has been destroyed. The trusted party compares FP'_{chip} to the known value of FP_{chip} in its database that was previously generated by the same cells. If the Hamming distance between FP'_{chip} and FP_{chip} is within the range of expected distances for same-chip fingerprints, the trusted party confirms that FP'_{chip} is from the specific memristor array that stored the unique value K_{chip}, and therefore that K_{chip} has been irreversibly destroyed, as this is a necessary condition for generating the fingerprint FP'_{chip}. Since the user of the chip does not themselves know the random value of K_{chip} in our protocol, they have no way of ever reproducing the secret key value K_{chip} after it is destroyed. As shown in Fig. 4d, if FP_{chip} and FP'_{chip} are generated from the same chip, the Hamming distances (128 bits) are centred at 20.65 with a standard deviation of 4.88, while if they are from different chips, the Hamming distances have a mean of 64.16 with a standard deviation of 5.88. The two distributions can be further separated by increasing the fingerprint bit size to 256 (Fig. 4e). To the best of our knowledge, there is no existing CMOS implementation of provable key destruction. It is impracticable to use a static random-access memory (SRAM) fingerprint. Although the fingerprint can be obscured by writing random values to SRAM cells in a similar way, SRAM is volatile whereas our application requires the written values to be persistent from enrolment until key destruction. Furthermore, the SRAM fingerprint is susceptible to negative-bias temperature instability ageing, which causes a SRAM cell storing a given value to favour the opposite value in the next generation of fingerprints by powering-up[9]. This creates problems for verifying that the key was destroyed. The reliable memristor as a non-volatile memory requires no external power to store values, which makes the memristor fingerprint less affected by the long-time key storage, in addition to the efficiencies in area and energy.

Logic locking/unlocking with provable key destruction

To demonstrate the feasibility of our provable key destruction for solving practical security problems, we showcase a detailed design of relockable logic locking/unlocking (Fig. 5a), which contains three phases: device enrolment (Fig. 5b); unlocking logic (Fig. 5c); and relocking logic (Fig. 5d). Full details of the protocol that uses provable key destruction for logic locking can be found in Supplementary Note 2. The general concept is briefly described as follows. In the device enrolment phase (Fig. 5b), the embedded memristor fingerprint (FP_{chip}) of each chip is measured, after setting all memristors to LRS, and sent to the IP owner through the chip's asymmetric crypto interface. After that, the chip can generate a random key (K_{chip}) using a TRNG and write it to the memristor array, which is then also sent to the IP owner after asymmetric encryption. Asymmetric cryptography is used here so that both FP_{chip} and K_{chip} are encrypted on the chip with the IP owner's public key (M_{pub}), which can be decrypted only by the IP owner's corresponding private key M_{pri}.

At this time, the IP owner knows both the key (K_{chip}) and the memristor fingerprint (FP_{chip}) hiding underneath the key, and the device itself possesses only the key (K_{chip}). In the unlocking logic phase (Fig. 5c), the common key (CK), which can unlock the logic gates, is sent to the chip by the IP owner after being symmetrically encrypted into the input key (IK) with K_{chip}. IK is permanently stored in arbitrary storage on chip and can be decrypted to CK at run time to unlock the logic gates. The use of symmetric cryptography here allows encryption and decryption with the same key (for example, K_{chip} here). In the last relocking logic phase (Fig. 5d), when an 'erase-key' command is given to the memristor array, all cells are switched to LRS and a new fingerprint measurement (FP'_{chip}) is extracted from the same crossbar array where K_{chip} was previously stored. The newly generated FP'_{chip} is sent to the IP owner through the crypto interface and compared against the known FP_{chip}. Verifying that the fingerprints match proves to the IP owner that K_{chip} has been erased and that the logic circuits on chip are now relocked. The trusted computing base (TCB) of the chip is outlined in white in Fig. 5. The TCB comprises the asymmetric encryption circuit and its fixed public key, the memristor array, the TRNG that randomly generates the key (K_{chip}), and the symmetric decryption circuit. All of these

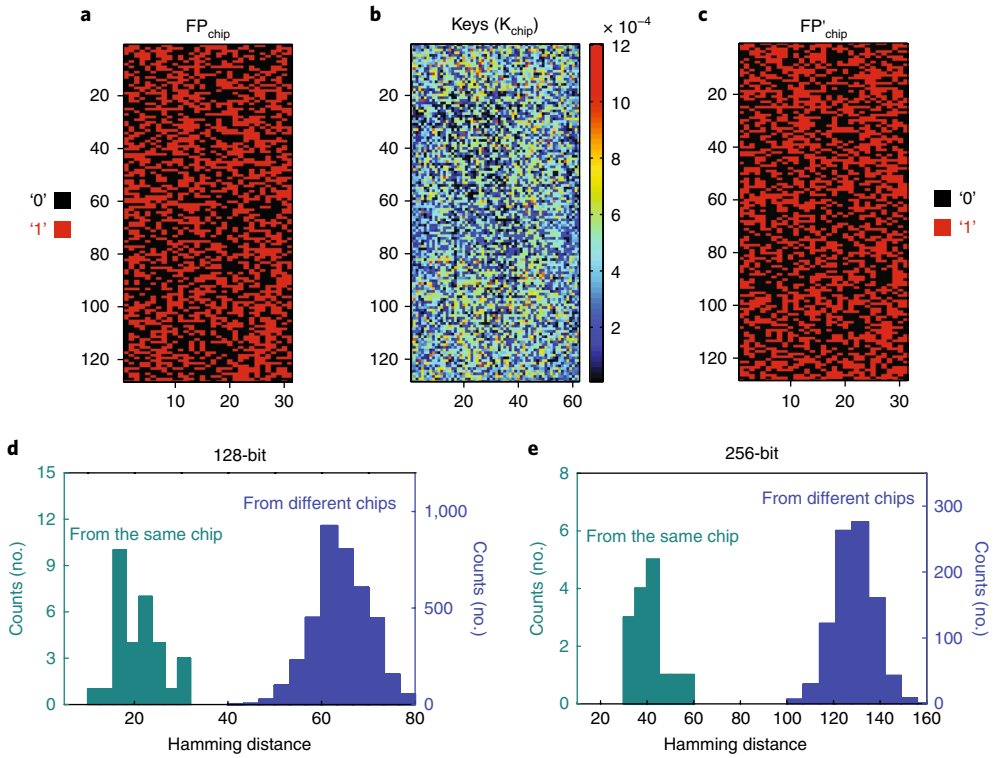

Fig. 4 | Experimental demonstration of provable key destruction in a 128 × 64 memristor crossbar array. a, A known fingerprint (FP_{chip}) is generated at enrolment and stored by a trusted party. **b**, The digital key (K_{chip}) is written into the memristor crossbar arrays. **c**, A second fingerprint (FP'_{chip}) is regenerated from the same memristor array after destroying the key K_{chip}. **d**, Comparison of Hamming distances of 128-bit fingerprints from the same chip or different chips, confirming the feasibility of our proposed scheme. The Hamming distances between FP and FP′ are centred at 20.65 with a standard deviation of 4.88 if they are from the same chip while at 64.16 with a standard deviation of 5.88 if they are from different chips. The distribution of Hamming distances from the same chip between **a** and **c** contains 31 counts in total (two non-responsive columns were removed). The distribution of Hamming distances from different chips here is based on the comparison of 31 128-bit fingerprints from this chip (**b**) with those from another 4 chips. There are, in total, 3,813 counts. **e**, Comparison of Hamming distances of 256-bit fingerprints from the same chip or different chips. Two adjacent 128-bit fingerprints are combined as a 256-bit fingerprint. The Hamming distances between FP and FP′ are centred at 40.93 with a standard deviation of 7.25 if they are from the same chip while at 128.39 with a standard deviation of 8.65 if they are from different chips. The two distributions show better separation when the 256-bit fingerprints are used. A simple distance threshold can be used to determine whether or not two fingerprints are taken from the same chips/devices. Detailed information about the statistical results can be found in Supplementary Table 1.

components are assumed to be implemented correctly as designed, but none of their implementation details is secret. Only the state stored in the memristor array (K_{chip} or FP_{chip}) is secret. Note that IK and the storage that holds it are outside the TCB, as it is useless to an attacker that does not know K_{chip}. It also should be noted that a cryptographic nonce (number or bit string used only once) is used to ensure freshness and prevent an adversary from replaying encrypted fingerprints collected before enrolment. More discussion about the threat model of the proposed design can be found in Supplementary Note 3.

In contrast to most of the previous work where the randomness is harvested from HRS[25,26,28,29], our memristor fingerprints are extracted from LRS. We chose LRS over HRS for a number of reasons. First, HRS usually has a wider cycle-to-cycle distribution that may mask some of the device-to-device variation critical for the proper working of our fingerprints. Second, LRS is usually more robust to switching cycling than HRS (Supplementary Fig. 5), suggesting that our memristor fingerprint based on LRS will remain reliable over a large number of switching cycles. Last but not least, the Ta/HfO_2 device has a better LRS retention property than that of HRS[39]. As a result, the adoption of LRS in our approach can greatly improve the reliability of memristor array fingerprints.

Our Ta/HfO_2 memristor crossbar arrays can integrate computing and security functionalities into the same circuits because of the excellent reconfigurability of the memristors. Since the memristor fingerprints are robust to switching cycles, one can deploy the same crossbar arrays for computing applications and use the fingerprints as hidden security primitives. For example, Supplementary Fig. 6 shows the extracted memristor fingerprints before and after the crossbar array was programmed into a conductance map for discrete cosine transformation (DCT) and inverse DCT. The DCT and inverse DCT conductance matrices can be applied to signal analysis and image compression[22]. From the aspect of security, the access to the memristor crossbar arrays may be allocated to the designer or trusted party only when they are used for both security and computing purposes. Although SRAM was recently demonstrated to run multiple cryptographic functions at the same time[42], our current work represents an implementation of running security and computing functions on the same chip. Such capability will enable more compact and energy-efficient memristive hardware systems.

Conclusions

We have proposed and experimentally demonstrated a provable key destruction scheme with physical fingerprints from large memristor crossbar arrays. The fingerprints are extracted by comparing LRS conductance between neighbouring cells in differential pairs. Attributed to both process variations and intrinsic

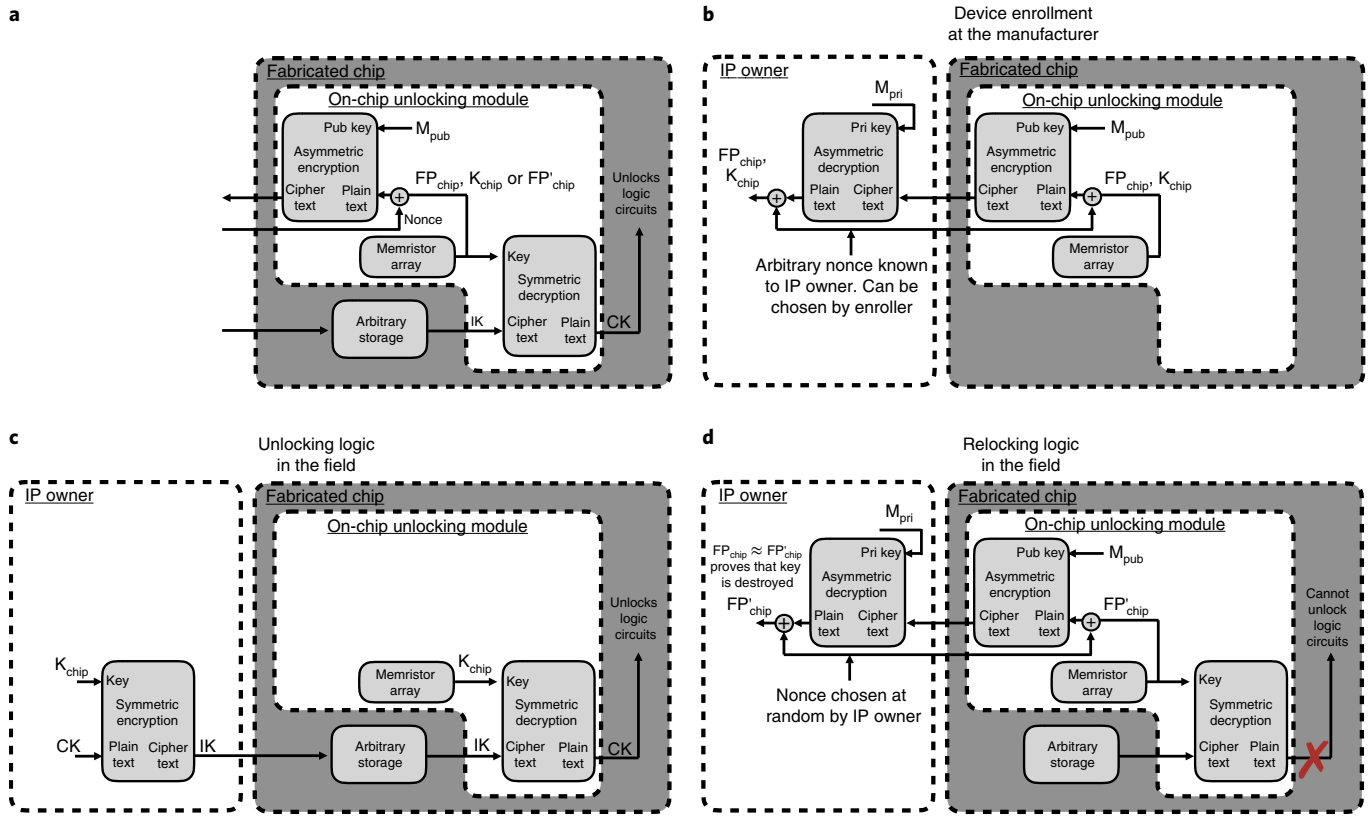

Fig. 5 | Detailed design of relockable logic locking/unlocking with provable key destruction. a, An overall schematic view of the proposed design to use the provable key destruction with memristor crossbar arrays for chip logic locking/unlocking, including three phases, as shown in **b**–**d**. **b**, Device enrolment, where the memristor fingerprint (FP_{chip}) is extracted and afterwards the digital key (K_{chip}) is generated and written over the memristor arrays. Both FP_{chip} and K_{chip} are XORed with a cryptographic nonce known to the IP owner and sent to the IP owner through the chip's asymmetric crypto interface. **c**, Unlocking logic, where the CK is symmetrically encrypted by K_{chip} to generate the IK that is sent to the chip. IK is stored in arbitrary storage on the chip and can be decrypted into CK to unlock logic circuits as long as the chip has K_{chip}. **d**, Relocking logic, where K_{chip} is erased and a second memristor fingerprint (FP'_{chip}) is XORed with a cryptographic nonce, asymmetrically encrypted, and sent to the designer for chip identification. IK now cannot be decrypted into CK and the logic circuits are locked again. The use of a random cryptographic nonce chosen by the IP owner ensures freshness and prevents an adversary from being able to replay encrypted fingerprints collected before enrolment when the cryptographic nonce was not known.

stochasticity in memristive switching, our memristor fingerprints exhibit excellent uniqueness and reliability. We further showcased a detailed protocol that uses provable key destruction for relockable logic locking/unlocking, which allows logic modules to be unlocked and subsequently relocked. The reconfigurability of the memristors enables the same crossbar array to be used for both security and computing/memory applications, saving chip space while increasing power efficiency.

Methods

Ta/HfO_2/Pt memristor fabrication and integration. The Ta/HfO_2/Pt memristor has a 20-nm-thick Pt BE deposited by electron-beam evaporation, a 5 nm HfO_2 switching layer deposited by atomic layer deposition at 250 °C and finally a 50 nm Ta TE. An extra 10 nm Pd layer was deposited on top as the capping layer by d.c. sputtering. Both the BE and TE were patterned by standard photolithography and then lift-off in acetone. The memristor arrays were integrated onto CMOS chips with extremely low wire resistance from a commercial foundry. Details on integration can be found in our recent papers[22,40].

Electrical characterization. The electrical characterizations were carried out using our custom-built multi-board measurement system, details of which can be found in ref. [40]. To switch all memristors to the LRS, a SET voltage pulse (V_{SET}: 2.5 V, 500 µs) is applied to the TE (Ta) with the BE (Pt) grounded and a gate voltage (V_G) of 1.1 V to the transistor. For OFF switching to the HRS, we applied a RESET voltage pulse (V_{RESET}: 1.8 V, 5 µs) to the Pt BE with the Ta TE grounded and a V_G of 5 V. For read operations, 0.2 V and 5 µS pulses were used to avoid affecting the device conductance.

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.

Received: 10 May 2018; Accepted: 7 September 2018;
Published online: 12 October 2018

References

1. Bonomi, J. & Botta, P. E. *Nineveh and its Palaces: the Discoveries of Botta and Layard Applied to the Elucidation of Holy Writ* (Illustrated London Library, London, 1852).
2. Evans, D. *The Internet of Things: How the Next Evolution of the Internet is Changing Everything* (Cisco, 2011).
3. Weber, R. H. Internet of Things – New security and privacy challenges. *Comput. Law Secur. Rev.* **26**, 23–30 (2010).
4. Sahay, S. & Suri, M. Recent trends in hardware security exploiting hybrid CMOS-resistive memory circuits. *Semicond. Sci. Technol.* **32**, 123001 (2017).
5. Van der Leest, V., Maes, R., Schrijen, G. J. & Tuyls, P. Hardware intrinsic security to protect value in the mobile market. In *Proc. Information Security Solutions Europe Conference (ISSE)* (eds Reimer, H., Pohlmann, N. & Schneider, W.) 188–198 (Springer Vieweg, 2014).
6. Gao, Y., Ranasinghe, D. C., Al-sarawi, S. F., Kavehei, O. & Abbott, D. Emerging physical unclonable functions with nanotechnology. *IEEE Access* **4**, 61–80 (2016).
7. Xie. Y. & Srivastava, A. Mitigating SAT attack on logic locking. In *Proc. International Conference on Cryptographic Hardware and Embedded Systems* (eds Gierlichs, B. & Poschmann, A.) 127–146 (Springer, 2016).
8. Roy., J. A., Koushanfar, F. & Markov, I. L. EPIC: ending privacy of integrated circuits. *Computer* **43**, 30–38 (2010).

9. Holcomb, D. E., Burleson, W. P. & Fu, K. Power-up SRAM state as an identifying fingerprint and source of true random numbers. *IEEE Trans. Computers* **58**, 1198–1210 (2009).
10. Lofstrom, K., Daasch, W. & Taylor D. IC identification circuit using device mismatch. In *Proc. IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* 372–373 (IEEE, 2000).
11. Su, Y., Holleman, J. & Otis, B. A. 1.6 pJ/bit 96% stable chip ID generating circuit using process variations. In *Proc. IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* 406–407 (IEEE, 2007).
12. Xiong, W. et al. Run-time accessible DRAM PUFs in commodity devices. In *Proc. International Conference on Cryptographic Hardware and Embedded Systems* 432–453 (IACR, 2016).
13. Wang, Y. et al. Flash memory for ubiquitous hardware security functions: true random number generation and device fingerprints. In *Proc. IEEE Symposium on Security and Privacy* 33–47 (IEEE, 2012).
14. Xu, X. & Burleson, W. Hybrid side-channel/machine-learning attacks on PUFs: A new threat? *Design, Automation and Test in Europe Conference and Exhibition (DATE)* 24–28 (IEEE, 2014).
15. Pi, S., Lin, P. & Xia, Q. Cross point arrays of 8 nm×8 nm memristive devices fabricated with nanoimprint lithography. *J. Vacuum Sci. Technol. B* **31**, 06FA02 (2013).
16. Pi, S. et al. Memristor crossbars with 4.5 terabits per inch square density and two nanometer dimension. Preprint at https://arxiv.org/abs/1804.09848 (2018).
17. Xia, Q. et al. Memristor−CMOS hybrid integrated circuits for reconfigurable logic. *Nano Lett.* **9**, 3640–3645 (2009).
18. Choi, B. J. et al. High-speed and low-energy nitride memristors. *Adv. Funct. Mater.* **26**, 5290–5296 (2016).
19. Lee, M. J. et al. A fast, high-endurance and scalable non-volatile memory device made from asymmetric Ta_2O_{5-x}/TaO_{2-x} bilayer structures. *Nat. Mater.* **10**, 625–630 (2011).
20. Pickett, M. D. & Williams, R. S. Sub-100 fJ and sub-nanosecond thermally driven threshold switching in niobium oxide crosspoint nanodevices. *Nanotechnology* **23**, 215202 (2012).
21. Yang, J. J., Strukov, D. B. & Stewart, D. R. Memristive devices for computing. *Nat. Nanotech.* **8**, 13–24 (2013).
22. Li, C. et al. Analogue signal and image processing with large memristor crossbars. *Nat. Electron.* **1**, 52–59 (2018).
23. Li, C. et al. Efficient and self-adaptive in-situ learning in multilayer memristor neural networks. *Nat. Commun.* **9**, 2385 (2018).
24. Sheridan, P. M. et al. Sparse coding with memristor networks. *Nat. Nanotech.* **12**, 784–789 (2017).
25. Chen, A. Utilizing the variability of resistive random access memory to implement reconfigurable physical unclonable functions. *IEEE Electron Devices Lett.* **59**, 1172–1182 (2012).
26. Liu, R., Wu, H., Pang, Y., Qian, H. & Yu, S. Experimental characterization of physical unclonable function based on 1kb resistive random access memory arrays. *IEEE Electron Devices Lett.* **36**, 1380–1383 (2015).
27. Gao, L., Chen, P., Liu, R. & Yu, S. Physical unclonable function exploiting sneak paths in resistive cross-point array. *IEEE Trans. Electron Dev.* **63**, 3109–3115 (2016).
28. Liu, R., Wu, H., Pang, Y., Qian, H. & Yu, S. A highly reliable and tamper-resistance RRAM PUF: design and experimental validation. In *Proc. IEEE International Symposium on Hardware Oriented Security and Trust (HOST)* 13–18 (IEEE, 2016).
29. Pang, Y. et al. Optimization of RRAM-based physical unclonable function with a novel differential readout method. *IEEE Electron Dev. Lett.* **38**, 168–171 (2017).
30. Zhang, R. et al. Nanoscale diffusive memristor crossbars as physical unclonable functions. *Nanoscale* **10**, 2721–2726 (2018).
31. Nili, H. et al. Hardware-intrinsic security primitives enabled by analogue state and nonlinear conductance variations in integrated memristors. *Nat. Electron.* **1**, 197–202 (2018).
32. Huang, C. Y., Shen, W. C., Tseng, Y. H., King, Y. C. & Lin, C. J. A contact-resistive random-access-memory-based true random number generator. *IEEE Electron Dev. Lett.* **33**, 1108–1110 (2012).
33. Balatti, S., Ambrogio, S., Wang, Z. & Ielmini, D. True random number generation by variability of resistive switching in oxide-based devices. *IEEE J. Emerg. Select. Top. Circuits Syst.* **5**, 214–221 (2015).
34. Balatti, S. et al. Physical unbiased generation of random numbers with coupled resistive switching devices. *IEEE Trans. Electron Dev.* **63**, 2029–2035 (2016).
35. Wei, Z. et al. True random number generator using current difference based on a fractional stochastic model in 40-nm embedded ReRAM. In *Proc. IEEE Electron Devices Meeting* 4.8.1–4.8.4 (IEEE, 2016).
36. Jiang, H. et al. A novel true random number generator based on a stochastic diffusive memristor. *Nat. Commun.* **8**, 882 (2017).
37. Rührmair, U. & Dijk, M. V. PUFs in security protocols: attack models and security evaluations. In *Proc. IEEE Symposium on Security and Privacy* 286–300 (IEEE, 2013).
38. Zidan, M. A., Strachan, J. P. & Lu, W. D. The future of electronics based on memristive systems. *Nat. Electron.* **1**, 22–29 (2017).
39. Jiang, H. et al. Sub-10 nm Ta channel responsible for superior performance of a HfO_2 memristor. *Sci. Rep.* **6**, 28525 (2016).
40. Hu, M. et al. Memristor-based analog computation and neural network classification with a dot product engine. *Adv. Mater.* **30**, 1705914 (2018).
41. Chen, P. et al. Exploiting resistive cross-point array for compact design of physical unclonable function. In *Proc. IEEE International Symposium on Hardware Oriented Security and Trust (HOST)* 26–31 (IEEE, 2015).
42. Zhang, Y. et al. Recryptor: a reconfigurable cryptographic cortex-M0 processor with in-memory and near-memory computing for IoT security. *IEEE J. Solid-State Circuits* **53**, 995–1005 (2018).

Acknowledgements

This work was supported in part by the US Air Force Research Laboratory (AFRL; grant no. FA8750-15-2-0044), and the National Science Foundation (CNS-1749845). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of AFRL. R.Z. and P.Y. are on leave from Tianjin University and Huazhong University of Science and Technology, and acknowledge the support from the Chinese Scholarship Council (CSC) under grants 201606250162 and 201606160074, respectively. The authors would like to thank X. Xu and S. Pi for helpful discussions.

Author contributions

Q.X., D.H., J.J.Y. and H.J. conceived the idea and designed the experiments. P.Y., C.L. and H.J. built the integrated chips. H.J., C.L. and R.Z. performed electrical measurements. Q.X., D.H., J.J.Y and H.J. analysed the data. Y.L. and P.L. helped with experiments and data analysis. Q.X., D.H. and H.J. wrote the manuscript. All authors discussed the results, and commented on and approved the final version of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41928-018-0146-5.

Reprints and permissions information is available at www.nature.com/reprints.

Correspondence and requests for materials should be addressed to J.J.Y. or D.H. or Q.X.