nature electronics

REVIEW ARTICLE

https://doi.org/10.1038/s41928-020-0372-5

Physical unclonable functions

Yansong Gao[1,2], Said F. Al-Sarawi[3] and Derek Abbott[4]

A physical unclonable function (PUF) is a device that exploits inherent randomness introduced during manufacturing to give a physical entity a unique 'fingerprint' or trust anchor. These devices are .of potential use in a variety of applications from anti-counterfeiting, identification, authentication and key generation to advanced protocols such as oblivious transfer, key exchange, key renovation and virtual proof of reality. Here we review the development of PUFs, including those that exploit optical, circuit time-delay and volatile/non-volatile memory characteristics. We examine the various applications of PUFs, and consider the security issues that they must confront, highlighting known attacks to date and potential countermeasures. We also consider the key areas for future development such as bit-specific reliability, reconfigurability and public key infrastructure.

Located in the ruins of an Assyrian palace, the earliest known lock dates back 4,000 years[1]. Modern security systems still keep valuables under lock and key, in order to ensure the safety and authenticity of goods, information and identities. Today, electronic systems use digital keys that are typically stored in non-volatile memory (NVM). The digital key is the foundation for building up secure cryptographic protocols that are built on block ciphers, stream ciphers and hash functions. Only the key is kept secret from an attacker, while the other building blocks and protocols are public. Maintaining the security of the key over the lifetime of an embedded entity is, however, non-trivial, especially when the key is digitally stored in NVM without dedicated protection (in practice, such protection leads to an increased cost). In addition, an NVM-stored digital key is assigned — but not inherently bound — to a physical entity, and therefore is vulnerable to being copied.

One alternative approach is to use physical unclonable functions (PUFs), which are clone proof, cost efficient and resistant to various physical attacks. A PUF exploits inherent random variations introduced by manufacturing processes to form secret keys on the fly. The key is internal to the PUF and is not assigned by an outside source; thus, the random variation is analogous to a unique fingerprint (Fig. 1a). This form of randomness is very inexpensive to access. Moreover, random manufacturing variations usually do not affect the digital functionality of integrated circuits (ICs). Even when random variation is measurable, it is still infeasible to create an identical physical 'copy' because full control of micro- and nanoscale fabrication variations is considered impossible[2].

In this Review, we examine different types of PUFs and their various applications. We consider the key challenges in the field — in particular, unreliability, attacks and countermeasures, and tamper resistance — and key areas of future development.

Typical PUFs and primary applications

There are numerous PUF constructions. We focus here on the optical PUF as a representative example of non-silicon PUFs, arbiter PUF (APUF) as an example of time-delay based silicon PUFs, and static random-access memory PUF (SRAM PUF) as an example of intrinsic silicon PUFs. Note that there exist several publicly available PUF datasets: ring-oscillator PUF (ROPUF) datasets[3–5], an APUF dataset[6] and memory-based PUF datasets[4,7].

Optical PUFs. A precursor of the PUF, which was based on exploiting unclonable physical disorder, was the unique object (UNO). The UNO was used, for example, for arms control during the cold war, where a thin coating of light-reflecting particles was sprayed onto the surface of nuclear weapons[8]; as the particles are randomly distributed, the interference pattern after being illuminated is unique and unpredictable as a function of viewing angle. The first well-known PUF was the related optical PUF, which was initially dubbed a physical one-way function[9]. The motivation here was to create a one-way function via physical means rather than number theory. In particular, under light illumination at a given angle, a two-dimensional speckle pattern was created from complex interference within an inhomogeneous transparent plastic token, which was recorded by a charge-coupled device camera (Fig. 1b).

For each PUF, an input query or 'challenge' receives an instance-specific output or 'response', a process known as a challenge–response pair (CRP). With the physical one-way function, the angle is treated as the challenge (input) of the optical PUF, whereas the 2D speckle pattern is treated as the response (output). A CRP can be lodged and then later used to identify the authenticity of an optical PUF. As the 2D speckle pattern is dependent on random inhomogeneities, it is infeasible to forge or counterfeit the optical PUF. Authentication can also be executed remotely, once the CRP is recorded in a secure database only known by the trusted party (server). Thus, the optical PUF has a very large CRP space, which cannot be fully read out within a limited time period (such as days or weeks or even years). During the authentication phase, the server randomly selects a CRP from the database and publicly sends the challenge to the optical PUF holder. The received response is compared with the lodged one, if the Hamming distance between responses is within a preset threshold, and thus the authenticity of the optical PUF integrated entity is established — otherwise authenticity is rejected. To avoid replay-based attacks, each CRP is used only once[10,11].

The optical PUF requires bulky external apparatus to characterize its CRP behaviour, which leads to increased cost. Moreover, its practicality is hindered by its inconvenient compatibility with current complementary metal–oxide–semiconductor (CMOS) fabrication processes.

[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. [2]Data61, CSIRO, Sydney, New South Wales, Australia. [3]Centre for Biomedical Engineering, School of Electrical and Electronic Engineering, The University of Adelaide, Adelaide, South Australia, Australia. [4]School of Electrical and Electronic Engineering, The University of Adelaide, Adelaide, South Australia, Australia. e-mail: yansong.gao@njust.edu.cn; said.alsarawi@adelaide.edu.au; derek.abbott@adelaide.edu.au

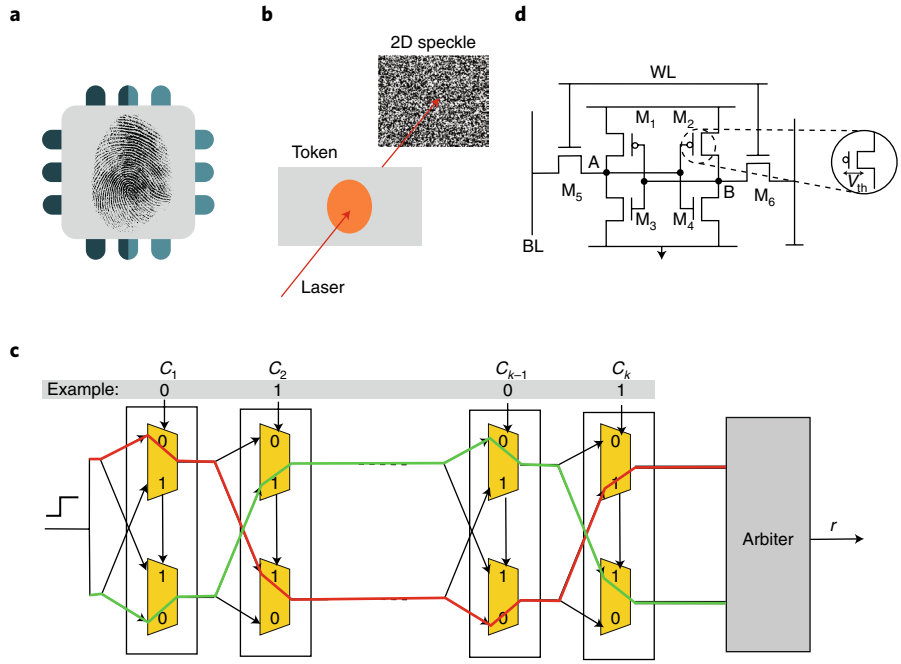

Fig. 1 | PUF basics. a, A PUF exploits intrinsic random variation at the microscale or nanoscale. Such random variation resulting from uncontrollable fabrication processes can be conceptually thought as a unique physical 'fingerprint' of a hardware device. **b**, Optical PUF. Under illumination from a given angle/polarization, complex interference occurs within an inhomogeneous transparent plastic token. Then, a two dimensional (2D) speckle pattern is recorded using a charge-coupled device camera[9]. The angle/polarization is treated as the challenge while the 2D speckle pattern is the response. **c**, APUF. Consisting of multiple stages (1 to *k*), the challenge bits (*C*) select two theoretically identical but practically unequal delay paths at each stage. At the end of the APUF, an arbiter judges whether the top path is faster or not, and reacts with a '1' or '0' response (*r;* ref. [10]). A challenge bit of '0' means two signals pass a given stage in parallel, whilst '1' means two signals cross over. **d**, SRAM PUF. The mismatch of threshold voltage V_{th} of the transistors determines the response[124]. For example, if the V_{th,M_1} is slightly smaller than V_{th,M_2}, at power-up, the transistor M_1 starts conducting before M_2, thus, the logic state at point A = '1'. This in turn prevents M_2 switching on. As a result, the SRAM power-up state prefers to be '1' (point A = '1', point B = '0'), which is the response, while the address of the memory cell is the challenge. WL, word line; BL, bit line. Credit: Fingerprint image in **a**, Andrey Kuzmin / Alamy Stock Vector

Arbiter PUFs. In 2002, a CMOS-compatible APUF[12] was proposed (Fig. 1c). The APUF exploits manufacturing variability that results in random interconnect and transistor gate time delays. This structure is simple, compact, and capable of yielding a very large CRP space. Unlike the optical PUF, the APUF is, however, built upon a linear additive structure and thus vulnerable to various modelling attacks (see Supplementary Section 1) where an attacker uses known CRP observations to create a model that predicts unknown CRP behaviour[13–16]. Although the latest APUF variants[17–19] can resist various modelling attacks, they are unlikely to end the PUF security race. In other words, there may exist other modelling attack methods that could weaken or circumvent the security of the latest proposed APUF variants.

Memory-based intrinsic PUFs. The APUF is not an intrinsic PUF as it is not already embedded within an electronic device at manufacture, and thus requires extra design and layout steps. In 2007, the first intrinsic PUF was introduced: an SRAM PUF[20]. The SRAM PUF treats the favoured SRAM cell power-up state as the response and the cell's address as the challenge (Fig. 1d)[21]. Although there are intrinsic PUFs not based on memory[22,23], memory-based intrinsic PUFs including SRAM PUFs, flash memory PUFs and dynamic random-access memory (DRAM) PUFs are more convenient in practice, especially considering the ubiquity of memory in commercial electronic products[24–26]. On one hand, a memory-based PUF is usually limited by the size of CRP space, thus public CRP-based authentication is inapplicable. On the other hand, a memory-based PUF usually offers desired independence among response bits, which enables its primary application in deriving a volatile cryptographic key on demand.

The PUF response is susceptible to thermal noise, varying environmental conditions, and ageing. Thus, raw responses cannot be immediately employed as a cryptographic key. The PUF key generator is usually comprised of two components: a secure sketch module and an entropy accumulator, which are used to turn raw noisy responses into a cryptographic key. These two components together are usually referred to as a fuzzy extractor[27,28]. The secure sketch is responsible for reconciling noisy responses, which has two prevalent constructions: code-offset-based and syndrome-based (Fig. 2)[28]. Regardless of the specific construction, secure sketch is a pair of randomized procedures: the sketching procedure takes a response **r** as an input, then computes the enrolled key and the helper data; and the recovery procedure takes both the helper data and a regenerated response **r**′ to reconstruct the enrolled key. The key from the secure sketch module might not be ideally uniformly distributed: that is, the key does not possess full-bit entropy. To address this, an entropy accumulator is used to compress the key to attain full-bit entropy relying on a random oracle. In practice, this is implemented by a universal hash function.

In general, the sketching and recovery procedures are realized through error correction code (ECC) encoding and decoding, respectively. Most ECC decoding algorithms rely upon hard decision decoding, which treats each response bit with equally likely error probability[10,20,29]. In 2009, a soft-decision decoding scheme based on an accurate PUF reliability model was proposed, where each response has a bit-specific error probability[30]. Actually, for most response bits, their bit-specific error rates are lower than the averaged error rate. As a consequence, soft-decision decoding efficiency is greatly improved. For example, fewer responses are

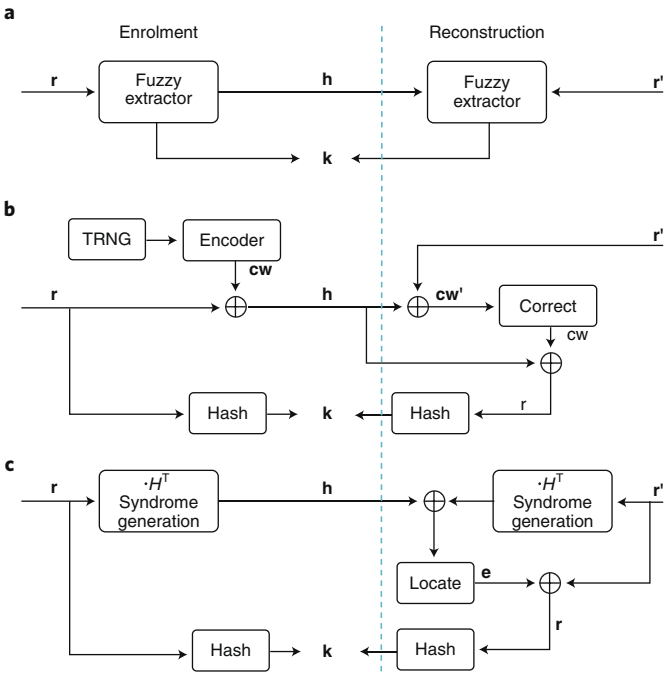

Fig. 2 | A fuzzy extractor and its two dominant constructions. a, A fuzzy extractor that consists of a secure sketch module and an entropy accumulator (realized by hash in practice) turns noisy PUF responses into a cryptographic key with full-bit entropy. It has two phases: a key enrolment phase and a key reconstruction phase. The key enrolment takes the enrolled response **r** as the input, enrols a key **k** and computes the helper data. The helper data and a regenerated response **r**′ are inputs during the key reconstruction phase used to recover the enrolled key **k** on the condition that the Hamming distance between **r** and **r**′ is less than a predefined threshold. There are two dominant secure-sketch constructions: code-offset and syndrome based. **b**, Code-offset based secure-sketch. Random numbers are generated by the indicated TRNG. During sketching (response encoding), a codeword $\mathbf{cw} \in \mathbf{CW}$ is randomly chosen and the computation of helper data $\mathbf{h} = \mathbf{cw} \oplus \mathbf{r}$ is performed, with the enrolled response, **r**. During recovery, one first computes $\mathbf{cw}' = \mathbf{r}' \oplus \mathbf{h} = \mathbf{cw} \oplus \mathbf{e}$, **e** is an error vector considering that the regenerated response **r**′ is different from the enrolled response **r**. Then, the error vector **e** is mapped onto the original codeword **cw** via an error correction algorithm, and the **r** is recovered through $\mathbf{r} = \mathbf{cw} \oplus \mathbf{h}$. Finally, the hash compresses **r** into a cryptographic key **k** with full-bit entropy, consequently restoring the enrolled **k**. **c**, Syndrome based secure-sketch. During sketching, the helper data **h** is computed with $\mathbf{h} = \mathbf{r} \cdot H^{\mathrm{T}}$, where H is a parity-check matrix of the employed linear error correction code **CW**. For recovery, $\mathbf{h}' = \mathbf{r}' \cdot H^{\mathrm{T}}$ is first computed. Through an error location algorithm, the error vector $\mathbf{e} = \mathrm{Locate}(\mathbf{h}' \oplus \mathbf{h})$ is determined. Then, **r** is recovered through $\mathbf{r} = \mathbf{e} \oplus \mathbf{r}'$. Finally, the hash compresses **r** into a cryptographic key **k** with full-bit entropy, consequently restoring the enrolled **k**.

required to derive a fixed-length key[31], which is commonplace in key generation research[28,32–34].

As a security requirement, the derived cryptographic key should contain adequate entropy (128 bits, for example) to meet information-theoretic security. However, there are several sources of entropy leakage when deriving the key from noisy responses via fuzzy extractors. Firstly, leakage is mainly caused by publicly stored helper data, although it can be compensated easily by using more response bits. Secondly, extra entropy losses are induced by response bias and correlation[34,35]. Bias means that the probability of response being '1' or '0' is not ideally 50%, while correlation means that a generated response is not ideally independent of others — spatial correlation is even found in some SRAM PUFs[34,36] due to, for example, word-line dependence. It should be noted that, in general, any set of states that possesses memory persistence can counterintuitively result in correlation under certain conditions[37]. To prevent the extra entropy loss from bias, debiasing the response is a common step conducted before implementing the sketching[35,38]; alternatively, injecting randomness into the helper data can be used to achieve a similar purpose[39]. Thirdly, active helper-data manipulation may also cause serious entropy leakage if the helper-data algorithm that is used in the ECC to reconcile errors is not handled with care (see Supplementary Section 2)[28,40–42].

As ECC decoding is computationally more intensive than encoding, the ECC encoding logic can be embedded in the resource-restricted PUF. The computationally harder decoding logic is performed by a resource-rich server. This arrangement is called a reverse/reusable fuzzy extractor[43–45]. Note that the reverse fuzzy extractor and the fuzzy extractor might not provide identical security guarantees. This is because a reverse fuzzy extractor can result in unanticipated entropy loss under repeated helper-data exposure associated with a given PUF response, unless PUF responses are unbiased[46]. Generally, the extra entropy loss is a result of the leakage of bit-specific reliability information[34]. When the bias is moderate, for example, between 0.42 and 0.58 — a bias requirement that most modern silicon PUFs can meet[47] — employing more response bits is an effective measure to compensate such bias-induced entropy loss[48]. Otherwise, debiasing must be considered. However, debiasing schemes suitable for a fuzzy extractor might not offer reusability — multiple use of the same PUF response — that is suitable for a reverse fuzzy extractor[34]. Therefore, the debiasing method requires careful design[35]. For example, both classic von Neumann debiasing and pair-output von Neumann debiasing with erasures are applicable to the fuzzy extractor, but only the latter offers reusability that is suitable for the reverse fuzzy extractor[35].

Under the conventional information-theoretic setting, entropy needs to be estimated to prove the security of a derived cryptographic key. However, accurate and easy-to-use tight entropy bounds on a derived cryptographic key is still an ongoing area of research due to the complex properties of the PUF, for example, response bias and correlation[34,49]. Instead of arguing security from

an information-theoretic setting, these emerging studies[50–53] use a complexity theoretical argument where no information leakage is observed. This leads to the construction of a computationally-secure fuzzy extractor[51].

PUF taxonomy and applications

The APUF is a key type of time-delay-based PUF but needs careful circuit layout. For example, manual tuning of the delay wires is usually required when the APUF is implemented on field-programmable gate array (FPGA) platforms to eliminate potential severe response bias. The ROPUF is another dominant type of time-delay-based PUF[10] and can be easily implemented on FPGA platforms, for example, without dedicated layout. Variants of ROPUF have been extensively investigated to improve reliability, CRP space and area overhead[54–58]. Similarly, for FPGA platforms without SRAM memories, other mismatch-based PUFs such as latch PUFs[59], flip-flop PUFs[60,61] and butterfly PUFs[62] are proposed to imitate SRAM PUF behaviour. However, all these mismatch-based PUFs are not intrinsic PUFs as they demand additional logic configurations within the FPGA. A detailed description of these and other PUF types can be found in ref. [47]. From the circuit-design perspective, and in terms of native bit error rate, area and power consumption, a public dataset consisting of 54 PUF designs from 2000 to 2018 (ref. [63]) provides trends in the application-specific integrated circuit (ASIC) implementation of PUFs.

Weak and strong PUFs. Optical PUFs and APUFs are generally classified as strong PUFs due to a very large CRP space, whereas an SRAM PUF is categorized as a weak PUF. Beyond being able to realize weak PUF applications, such as key generation, strong PUFs are deployed in advanced cryptographic protocols such as oblivious transfer, bit commitment, and multi-party computation where PUFs are adopted into the universal composition (UC) framework[64–68] and virtual proof of reality (see Supplementary Section 3).

In the UC framework, PUFs are regarded as new, fundamental physical cryptographic primitives of their own, comparable to the bounded storage model, quantum cryptography, or noise-based cryptography[68]. The main threat model difference between the strong PUF and weak PUF is whether the CRP interface can be directly accessed by the attacker: a strong PUF interface is publicly accessible. To date, only the bulky non-silicon optical PUF[69] and the super high information content PUF (SHIC-PUF)[70] are strong PUFs that have not shown vulnerabilities to modelling attacks. Silicon PUFs, and especially variants of APUFs, are potentially affected by powerful modelling attacks. Though there are continuing efforts to improve their resilience to modelling attacks, most variants are either too expensive or too impractical. Therefore, the development of practical, easy-to-integrate and easy-to-use strong PUFs with no vulnerability to modelling attacks remains an ongoing challenge.

Reconfigurable PUFs. Most PUFs exhibit a static CRP behaviour that cannot be refreshed, which limits certain PUF-based applications. For example, considering a reusable PUF token transferred between users, a conventional PUF is insufficient as a new user of the PUF token needs the authentication credentials of the first user. But the key of the first user is sensitive information and must be deleted. Reconfigurable PUFs (RPUF) were conceived in 2004 (ref. [71]) to overcome this issue and became widespread from 2009 (refs. [72–74]). The essential idea is to introduce an update mechanism to refresh the challenge response behaviour. This dynamic reconfigurability without physically changing the overall PUF circuit structure is an appealing feature for applications such as key renewal and revocation, protecting non-volatile storage against invasive attacks[75] and against malicious manufacturers (see Supplementary Section 4). Reconfiguration sources of the RPUF are generally classified into two types: logical and physical. Consequently, there are two RPUF classes: logically reconfigurable PUF (LRPUF) and physically reconfigurable PUF (PRPUF) (see Supplementary Section 4).

For RPUF, there are two key challenges: how can the server remotely and securely update RPUF CRPs, especially PRPUF, based on the enrolled CRPs? And how to avoid a denial-of-service attack performed by a malicious party to intentionally reconfigure the RPUF that desynchronizes the enrolled CRP by the trusted party?

Erasable PUFs. Advanced cryptographic protocols such as key exchange and oblivious transfer enabled by a strong PUF are at risk under the PUF re-use model or post protocol access[68] — the same strong PUF is used to execute consecutive protocol sessions (an example is given in Supplementary Section 5). Erasable PUF[68] addresses this re-use model attack by adding one property to the strong PUF: each CRP can be reconfigured independently. This property distinguishes it from the RPUF. The RPUF does not need to be a strong PUF and each CRP does not have to be refreshed independently. In 2011, the first erasable PUF construction was presented based on the SHIC-PUF that is already a strong PUF[76]. The SHIC-PUF is a high-density nano-crossbar array with a diode at each cross-point. An intentionally slow read speed prevents full readout of SHIC-PUF CRPs. To realize erasability, diode breakdown in its current–voltage characteristic is exploited[76]. However, this erasure is feasible only once given a specific diode or a response. In addition, sneak path currents significantly increase as the rectification behaviour of the diode greatly reduces after breakdown operation, preventing accurate response evaluations[68,76]. Thus, development of an efficient and practical erasable PUF appears to be an important area for future work.

Public PUFs. The security of a PUF relies on the unpredictability of its response for a given challenge based on complex interactions with a physical function. Therefore, any model of the PUF, which mathematically impersonates the physical function of the PUF, must be kept securely. In essence, a PUF can be considered as a storage device that contains secret keys based on static randomness in a hardware instance rather than the easy-to-measure digital form in an NVM. In 2009, a public PUF (PPUF) was proposed[77] — concurrently and independently dubbed the Simulation Possible but Laborious system (SIMPL)[78] — that resembles asymmetric cryptography or public-key cryptography. In this regard, the PUF can be thought of as resembling classical symmetric cryptography. Like the PUF, the PPUF is unique and physically unclonable. By contrast, PPUF hardware is not secret[79], since the PPUF model is assumed to be known to any party including the adversary. Its security relies on response evaluation speed being much faster than the response emulation based on its corresponding model[80–82]. The PPUF can be used in applications such as public-key cryptography, secure location authentication, *k*-anonymity security protocols, and trusted sensing and computing[83]. However, a stable and practical realization of PPUF remains a challenge.

Challenges

There are challenges in integrating PUFs into security systems, especially when the system resources are constrained, mainly caused by unreliability. Moreover, various attacks threaten PUF security, and demand countermeasures.

Unreliability. The unreliability (that is, the bit error rate) of PUFs is a major hurdle for numerous applications. Firstly, to reconcile a noisy response, the power and area overhead resulting from the on-chip ECC logic may be several times larger than the logic overhead of the PUF itself. This tends to be neglected in some applications including IC active metering[84–86], lightweight authentication[87] and even advanced cryptography protocols[64]. Specifically, the ECC logic overhead is usually not properly taken into account when assessing

the overhead for those applications but treated as a default part of the PUF, which may lead to underestimated overhead assessment for the targeted applications[84–86]. Also note that the helper data used for error correction potentially exposes key generation to helper-data manipulation attacks. Simply treating the error correction step as default may already overlook security weaknesses of the underlying application systems (see Supplementary Section 2). Secondly, a noisy response makes it difficult to realize strong PUFs in silicon, for example, variants of the APUF. This is mainly because non-linearity cannot be arbitrarily applied to obfuscate the relationship between the challenge and response to increase resistance to modelling attacks, since adding a strong non-linearity deteriorates the reliability of PUFs, thus, rendering them impractical for use in the field[14]. Thirdly, unreliable responses can be exploited as noisy side-channel information that allows an attacker to carry out modelling attacks on APUF variants[15,16,88].

There are two general approaches to enhance PUF reliability. The commonly used approach is response error correction[27,89,90]. The second approach is error reduction, which can be further classified into two ways: intrinsic and extrinsic error reduction. Intrinsic error reduction exploits circuit design techniques to enhance the response reliability but normally requires custom design considerations[57,91,92] and certain special steps such as hot-carrier injection[93]. Note that the digital PUF (see Supplementary Section 6) falls into this category[91,92,94,95]. The extrinsic method selects reliable responses during the enrolment phase. This includes majority voting through repeated measurements or direct measurements such as in the ROPUF[10] and sense-amplifier PUF cases[96] to determine reliable responses. Recently, machine learning has been exploited to determine highly reliable responses, for example, of the SRAM PUF[97] and the APUF[98].

Attacks and countermeasures. The security race between attackers and defenders never ends for a given security primitive, and the PUF is no exception. First, direct response access to a weak PUF by an adversary must be prevented. Otherwise, the adversary can just copy the readout response into an NVM and can then impersonate the PUF. Second, disabling direct access might not fully prevent characterization of the response through exploiting data remanence (in SRAM PUF attacks, for example) or photonic-emission-based side-channel information (in SRAM PUF and APUF attacks, for example)[99–101]. Third, a focused ion beam can be used to physically edit a different SRAM array exhibiting a similar start-up pattern to the targeted SRAM PUF victim[100], which differs from copying the responses into the NVM for challenge response behaviour impersonation and hence breaks the hardware 'unclonability'.

To a large extent, this is not a truly physical clone at the micro- or nanoscale. Eventually, a defender can examine multiple dimensional behaviours of the PUF to detect 'cloned' hardware, though this may require the hardware to be at hand. Besides the binary response itself, other physical and behavioural traits are worth including, for example, data remanence of SRAM PUFs[99], degree of sensitivity to environmental parameters, and bit-specific reliability.

Taking this multidimensional PUF behaviour into consideration can enhance its security. For example, each SRAM PUF response has its own bit-specific remanence[99]. Therefore, an attacker may be able to use a 'cloned' SRAM PUF to impersonate similar start-up states, but concurrently guaranteeing that the cloned SRAM PUF exhibits an identical data remanence will be virtually impossible. In addition, simultaneous bit-specific reliability behaviour is even harder to physically imitate.

Interconnect meshes can be used to prevent optical attacks including a photonic emission attack performed from an IC's frontside; for backside attacks, through-silicon-via (TSV) technologies can be exploited as a countermeasure[102]. Additionally, optical interaction can be a low-cost alternative for protecting ICs against backside optical emission attacks[102]. Furthermore, intertwining the PUF circuitry within existing functional circuitry can increase the difficulty of an optical attack[100,103].

Power- and timing-side-channel information-aided attacks have been shown to break the security of APUFs[104,105]. In comparison with reliability-side-channel information-based attacks that are readily achieved, these attacks are harder in practice, and can be eliminated by careful circuit-level design. Firstly, power analysis attacks require low noise levels when repeatedly measuring power traces. Therefore, an efficient countermeasure is to inject noise into the PUF device to prevent accurate power trace measurements[105]. Besides injecting noise, dynamic and differential CMOS logic can be utilized to implement the arbiter at the end of APUF structure to balance the power consumption[106] to prevent power-based attacks.

As for timing attacks, the required accurate time measurement circuitry may even not be available on-chip[105], especially for PUF targeted low-end devices. In general, all the above side-channel information, including reliability, photonic emission, and power- and timing-enabled attacks, need repeated measurements to increase the signal-to-noise ratio, hence integrating a true random number generator (TRNG) with the PUF to control challenge generation will efficiently avoid repeated measurements[19,107] to combat these attacks.

As side-channel information facilitates attacks on PUFs, a trusted party might consider using the same side-channel information to distinguish a cloned PUF — software or hardware copy — from its victim when the PUF is in hand. Side-channel information has been widely applied to detect hardware Trojans[108,109], while a cloned PUF, to some extent, can be viewed as a malicious Trojan.

PUFs with a very large CRP space, in particular, the APUF and its variants, turn out to be vulnerable to various powerful modelling attacks when not protecting their interface, for example, via encryption[107]. Therefore, to guarantee high security, these PUFs should often be combined with other building blocks, for example, challenge generation bound to a TRNG to prevent repeated response measurements, and/or response error correcting followed by cryptographic hashing to apply strong obfuscation, in order to prevent modelling attacks. In this regard, minimizing the overhead of associated building blocks is of great value and deserves further attention[19,98]. In addition, PUF recombination/recomposition[18,110] through merging different types of PUFs, for example, combining the feedforward-APUF with an XOR-APUF or even combining a ROPUF that has an expanded CRP space with an APUF, are expected to be valuable approaches to substantially withstand modelling attacks[14,111]. The rationale is that a specific machine learning algorithm based modelling attack is most likely only effective on a specific PUF structure/topology[14].

Most importantly, before devising new PUF circuit-level constructions, it is essential to first investigate a systematic approach for designing machine learning resistant strong PUFs by studying the specific properties of underlying PUF building blocks[112]. From an adversarial machine learning perspective[113], the inherent machine learning vulnerability can be exploited to defeat modelling attacks. Specifically, intentionally flipping the response to poison training data (CRP data) to prevent the attacker constructing an accurate model appears promising[114].

Representative attacks on commonly used PUFs as well as corresponding (potential) countermeasures are summarized in Table 1. Note that these attacks and countermeasures can also be applied to other PUF constructions, and are not necessarily limited to the specified PUFs.

Tamper resistance. The PUF key is extracted only when two conditions are simultaneously satisfied: it is queried by a challenge and the power is on. Therefore, the PUF key is never present in an observable digital form during power-off even if the challenge is applied.

Table 1 | Summary of representative attacks on commonly used PUFs and potential countermeasures

Attack	Method	Features leveraged	Target application	Affected PUF	Potential countermeasure	Attack difficulty
Refs. 13,14,147	Machine learning	CRPs	Authentication	APUF variants	Challenge response obfuscation or adversary machine learning[114]	Low
Ref. 104	Machine learning	CRPs; time and power SCs	Authentication	APUF variants	Dynamic and differential CMOS logic[104], prevent repeated measurements[19]	Medium
Refs. 15,16,148	Machine learning	CRPs; reliability or bias SCs	Authentication	APUF variants	CRP lockdown[19], prevent repeated measurements, APUF recomposition/ recombination[18,110]	Low
Refs. 101,149	Machine learning	CRPs; photonic SC	Authentication	APUF variants	Interconnect meshes and optical interaction[102], prevent repeated measurements, 3D integration[141]	High
Ref. 100	Side-channel analysis	Photonic SC	Key generation	SRAM PUF	Circuit layout obfuscation[150]	High
Ref. 99	Side-channel analysis	Data remanence SC	Key generation	SRAM PUF	Constant-time response readout, SRAM power-up state obfuscation[99]	Medium
Ref. 151	Cryptanalysis	CRPs; helper data	Authentication	ROPUF with an expanded CRP space	Challenge response obfuscation	Low
Ref. 150	Side-channel analysis	Electromagnetic SC	Key generation	ROPUF	Circuit layout obfuscation[150]	Medium
Refs. 28,40,42	Helper-data manipulation	Key reconstruction failure observation	Key generation	ROPUF, SRAM PUF	Helper-data integrity check[28], response bias, careful error-correction code selection[42]	Low
Ref. 68	PUF reuse	Static (not erasable) CRPs; CRP access in running consecutive protocol session.	Advanced cryptographic protocols (for example, KE and OT)	Strong PUFs	Response erasability (erasable PUF)[68]	Low

SC, side channel; KE, key exchange; OT, oblivious transfer; 3D, three-dimensional.

An invasive attack that actively manipulates the physical structure will alter the PUF challenge response behaviour or destroy it. This anti-tampering property is of value when a PUF-based physical system is deployed in a (hostile) field, where rogue personnel and external adversaries can gain control over the device to physically manipulate it.

Although PUF tamper resistance is commonly assumed, studies that experimentally evaluate PUF tamper resistance are lacking. One type of anti-tampering protection is achieved by spraying a coating on the IC to realize a so-called coating PUF (see Supplementary Section 7)[115]. Recent studies[116,117] use the coating PUF to build battery-less anti-tamper envelopes for protecting physical cyber systems from physical attacks. Besides coating PUFs, recent work has experimentally demonstrated the tamper resistance of optical PUFs[118], where a physical change alters the CRP behaviour greatly. There are other valuable PUF proposals[103,119–121] for realizing tamper-resistance but these require experimental validation.

Outlook

We consider key areas for future work such as bit-specific reliability, opportunities enabled by nano elements, the flip-side of PUF unreliability in specific applications, and integrating PUFs into public-key infrastructure.

Bit-specific reliability. Although PUF reliability has been studied since its advent, there are still important characteristics that have not been widely investigated. The first is that bit error rate increases when the difference between the operating condition under which the responses were enrolled and the operating condition of the PUF in the field increases. In contrast to conventional single-response enrolment under the nominal operating condition during the PUF provisioning phase, one can lodge multiple reference responses, subject to the same challenge, produced under multiple distinct operating conditions. Thus, multiple reference responses will increase the reliability of the PUF, because one of the reference operating conditions will be closer to the operating condition under which the PUF operates[122]. As a result, using multiple reference responses reduces the error correction overhead of the (reverse) fuzzy extractor.

Second, bit-specific reliability as soft-decision information has been well recognized and applied in key generation. It has not been fully exploited in lightweight authentication given that a resource-rich server has the ability to accurately estimate the bit-specific reliability of any response. In this context, the PUF can directly hash the response, then transmits the hashed value to the server for authentication. The server performs a trial-and-error search over likely error patterns and identifies the processed response. The rationale

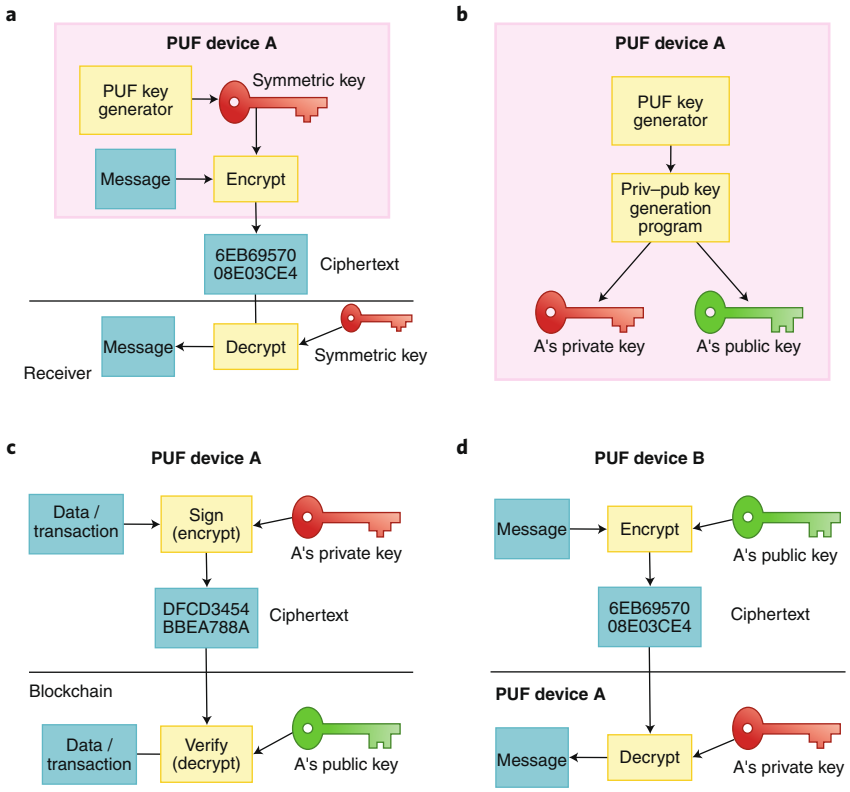

Fig. 3 | Employing a PUF within the public key infrastructure. a, Current work usually uses the PUF key as a shared symmetric key that must be known by the receiver that the PUF is communicating with. Distributing this symmetric key between parties who intend to communicate with the PUF device is inconvenient. **b**, The PUF key is used to produce a private and public key pair (priv–pub key pair), the public key is broadcast on the public key server that any party can access. The PUF device no longer digitally stores the priv–pub key pair, which is instead generated on demand and disabled by default. **c**, Data/transaction that is collected/issued by the PUF device is signed for verification. **d**, Secure message communications among PUF devices, where no other device can open/access the message sent to PUF device A without having A's private key, whilst the message is encrypted by A's public key.

behind this search is that only a small fraction of responses are least reliable[30,31]. So the server can grade the reliability of all response bits and only care about the least reliable bits — the server knows which responses are least reliable. Consequently, the PUF token is oblivious to the noisy nature of the PUF response and it is treated like an NVM stored key.

Most importantly, gaining bit-specific reliability during the PUF provisioning phase is inexpensive in most cases. For instance, one can take advantage of an easy-to-build model of the APUF to accurately predict any response's reliability even in the absence of the APUF[123]. As for ROPUF, the frequency difference of pair-wise ROs is the bit-specific reliability, while the sign of the difference is the response[51]. Overall, careful exploitation of the bit-level reliability is promising for reducing the PUF application overhead while enhancing application security.

Flip-side of unreliability. The flip-side of PUF unreliability is that it can be constructively exploited given some application scenarios. First, noisy responses are a good random source for building a TRNG[24,124–126] that is a critical part of cryptographic primitives and security protocols. Second, PUF unreliability can be utilized to guarantee the veracity of sensed data. The principle is that PUF unreliability is affected by environmental parameters (for example, voltage and temperature); therefore, PUF responses that are sensitive to such environmental parameters are able to discover these sensed parameter values. In other words, the PUF itself functions as the PUF sensor[127,128]. Third, while unreliability induced by voltage and temperature is retrievable, it is hard or expensive to retrieve in the face of device ageing.

By recognizing this fact and employing intrinsic an SRAM PUF, an ageing sensor extends the SRAM PUF functionality to not only detect cloned and overproduced ICs but also recycled ICs with zero area overhead when combating counterfeited ICs[129]. Fourth, when the inherent sensing capability is attached to a strong PUF, virtual proof of reality is enabled[130]. Finally, noisy PUF responses can be exploited to realize a public-key cryptosystem[131].

Public-key infrastructure. Currently, the PUF-derived key is usually presented in a symmetric manner where the verifier who enrols the PUF key and the PUF device (prover) restores the same key (Fig. 3a). This may be cumbersome when a different party who does not enrol the PUF key wants to authenticate and securely communicate with the PUF device, because this symmetric key needs to be securely provided to this party in advance. In this context, the application of PUFs to public key infrastructure becomes attractive. Specifically, during key enrolment, a PUF device generates a key that can be directly served as a private key, otherwise it is used as a random seed to derive a private key. Then, a corresponding public key is generated, building upon the private key, as illustrated in Fig. 3b, and the public key is broadcast on a public-key server.

The advantage of this asymmetric key setting is that the private and public key pair is bound to the PUF device. This facilitates a number of applications. First, any party now can conveniently authenticate this PUF device by sending out a nonce and accepting the PUF authenticity if the private key signature (that is, signed nonce) is correctly verified by using the public key. Second, in a similar way, the data/transaction can be signed when it is sent out, and the receiver can verify which is the claimed PUF device, illustrated in Fig. 3c.

Given such a signature, the PUF device cannot deny the data/transaction issued/signed by itself. This can be integrated into a blockchain system[132], where the recorded data/transaction in the ledger is guaranteed from authorized Internet of Things (IoT) devices based on the signature. Indeed, it would be more valuable that the PUF response, thus the private key, is bound to not only a hardware device but also the user's behaviour[133]. Consequently, both the user and the device are tracked or authenticated.

Last but not least, device-to-device secure communication is naturally enabled, and only the intended PUF device is able to generate a private key to open/decrypt the message encrypted by its public key (Fig. 3d). Within this PUF-based public-key infrastructure, it will be valuable to minimize the implementation overhead of public-key algorithms in order to fit into resource-constrained systems.

PUFs based on nanotechnology. Most PUF designs, nowadays, focus on exploiting process variations intrinsic to CMOS technology as most commercial commodities are based on CMOS technology. Recently, significant progress in emerging nanoelectronic devices has been made[134], especially for nano elements such as phase-change memory (PCM), spin-transfer-torque magnetic random-access memory (STT-MRAM), carbon-nanotube field-effect transistors (CNFETs) and resistive RAM (RRAM). These technologies are emerging for building memories utilizing other storage mechanisms such as memresistance and magnetism, rather than preserving charge on a capacitor as in SRAM, DRAM and flash memory[135]. When these types of memory were introduced, it was expected that one would become a 'universal memory'.

It is now acknowledged that the vision of a universal memory is unrealistic[135]. One common hurdle for all these nano elements is the effect of severe fabrication variations when scaling down to the nano region, which is deleterious to memory-based applications because the read margin deteriorates. However, this inherent randomness is embraced to positive advantage when building PUFs[136,137]. First, taking advantage of resistance variations in both ON (corresponding to logic '1') and OFF (corresponding to logic '0') states — exhibiting a bimodal resistance distribution — highly reliable PUFs[138–140] are realized. Second, these nano elements are relatively simple to fabricate, usually compatible with CMOS fabrication processes with three-dimensional integration capability. All these merits allow multiple layers to be vertically integrated in a stack to mitigate various side-channel attacks by placing the PUF in middle layers[141] that further enhance tamper resistance[121]. Third, a simple nano crossbar architecture facilitates storage of ultra high density information and provides a potentially low-cost PUF realization[70,142,143].

The unique properties of these nano elements provide new opportunities for the realization of special PUF architectures. For example, cycle-to-cycle variations during programming of the RRAM can be exploited for PRPUFs[140,142,144] or even erasable PUFs, and the bidirectional characteristic of RRAM are appropriate for the PPUFs[81]. Last but not least, nano elements (for example, the RRAM) are naturally immune to photon emission attacks, as trap-assisted-tunnelling conduction in RRAM oxide layers is not expected to emit photons[145], whilst a photonic-emission attack threatens silicon PUFs such as SRAM PUFs[100] and APUFs[101]. We refer readers to ref. [136] for a summary of emerging nanotechnology-enabled PUF constructions.

Real-world applications. With the proliferation of IoT devices in applications of autonomous vehicles, smart-homes etc, ensuring security and trust, safeguarding privacy, and combating counterfeits are top priorities for manufacturers and consumers. Various provably secure cryptographic algorithms have been proposed to address the aforementioned concerns, which all rely on the fact that a key must be securely maintained. In this regard, PUFs generate unique and volatile secure keys. Thus, in the industrial sector, PUFs have received significant attention for addressing the security concerns of IoT devices. This is especially the case when the SRAM PUF comes to play, because it is intrinsic, requiring no hardware modification, and thus applicable to all existing electronic commodities with pervasively embedded SRAM memories.

There are already a number of companies, for example, Intrinsic-ID, Verayo, eMemory, QuantumTrace, ICTK and Enthentica, commercializing PUF products, where the PUF has been steadily introduced into the market. The Intrinsic-ID PUF solution built upon an SRAM PUF has already protected 100 million transactions. It has been announced by NXP that the upcoming SmartMX2 security chips will feature an Intrinsic-ID PUF, which is also deployed in Altera's Stratix 10 FPGAs, Microsemi's SmartFusion FPGAs, and Xilinx's Zynq Ultrascale+ FPGAs. The MIT Sanctum processor uses a PUF to generate secret keys in a secure boot process. According to analysis by Gartner[146], there will be 20.4 billion IoT devices by 2020. Consequently, it is expected that PUFs will secure more of those deployed IoT devices.

Received: 10 June 2018; Accepted: 15 January 2020;
Published online: 24 February 2020

References

1. Bonomi, J. *Nineveh and its Palaces* (Bradbury & Evans, 1852).
2. Rührmair, U., Devadas, S. & Koushanfar, F. in *Introduction to Hardware Security and Trust* (eds. Tehranipoor, M. & Wang, C.) Ch. 4 (Springer, 2012).
3. Maiti, A., Casarona, J., McHale, L. and Schaumont, P. A large scale characterization of RO-PUF. In *Int. Symp. Hardware-Oriented Security and Trust (HOST)* 94–99 (IEEE, 2010).
4. *PUF Datasets* (Trust Hub, accessed 7 January 2020); https://www.trust-hub.org/data
5. Hesselbarth, R., Wilde, F., Gu, C. & Hanley, N. *Large Scale RO PUF Analysis over Slice Type, Evaluation Time and Temperature on 28nm Xilinx FPGAs*. (Fraunhofer AISEC, accessed 7 January 2020); https://s3.eu-central-1.amazonaws.com/aisecresearchdata/2018fpga-ro-data/index.html
6. Hussain, S. U. *ArbiterPUF FPGA Programmable Delay Lines* (accessed 7 January 2020); https://doi.org/10.6084/m9.figshare.3188731.v2
7. Su, Y. et al. *Secucode: Intrinsic PUF Entangled Secure Wireless Code Dissemination for Computational RFID Devices* (IEEE, accessed 7 January 2020); https://doi.org/10.21227/H27T0S
8. Graybeal, S. N. & McFate, P. B. Getting out of the STARTing block. *Sci. Am.* **261**, 61–67 (1989).
9. Pappu, R., Recht, B., Taylor, J. & Gershenfeld, N. Physical one-way functions. *Science* **297**, 2026–2030 (2002).
10. Suh, G. E. & Devadas, S. Physical unclonable functions for device authentication and secret key generation. In *Proc. 44th ACM Annual Design Automation Conference* 9–14 (ACM, 2007).
11. Herder, C., Yu, M.-D., Koushanfar, F. & Devadas, S. Physical unclonable functions and applications: a tutorial. *Proc. IEEE* **102**, 1126–1141 (2014).
12. Gassend, B., Clarke, D., Van Dijk, M. & Devadas, S. Silicon physical random functions. In *Proc. ACM Conf. Computer and Communications Security* 148–160 (ACM, 2002).
13. Rührmair, U. et al. Modeling attacks on physical unclonable functions. In *Proc. ACM Conf. Computer and Communications Security* 237–249 (ACM, 2010).
The most efficient modelling attacks on PUFs by only using challenge–response pairs.
14. Rührmair, U. et al. PUF modeling attacks on simulated and silicon data. *IEEE Trans. Inf. Forensics Security* **8**, 1876–1891 (2013).
15. Becker, G. T. The gap between promise and reality: on the insecurity of XOR Arbiter PUFs. In *Proc. Cryptographic Hardware and Embedded Systems* 535–555 (Springer, 2015).
Efficient modelling attacks on PUFs with assistance of response unreliability as side-channel information.
16. Becker, G. T. On the pitfalls of using arbiter-PUFs as building blocks. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **34**, 1295–1307 (2015).
17. Nguyen, P. H. et al. The interpose PUF: secure PUF design against state-of-the-art machine learning attacks. *IACR Trans. Crypt. Hardw. Embed. Syst.* **2019**, 243–290 (2019).
18. Sahoo, D. P., Mukhopadhyay, D., Chakraborty, R. S. & Nguyen, P. H. A multiplexer-based arbiter PUF composition with enhanced reliability and security. *IEEE Trans. Comput.* **67**, 403–417 (2018).

19. Yu, M.-D. et al. A lockdown technique to prevent machine learning on PUFs for lightweight authentication. *IEEE Trans. Multi-Scale Comput. Syst.* **2**, 146–159 (2016).
20. Guajardo, J., Kumar, S. S., Schrijen, G. J. & Tuyls, P. FPGA intrinsic PUFs and their use for IP protection. In *Proc. Cryptographic Hardware and Embedded Systems* 63–80 (Springer, 2007).
21. Holcomb, D. E., Burleson, W. P. & Fu, K. Initial SRAM state as a fingerprint and source of true random numbers for RFID tags. In *Proc. Conf. RFID Security* 2 (2007).
22. Kim, Y. & Lee, Y. Cam PUF: physically unclonable function based on CMOS image sensor fixed pattern noise. In *Proc. 55th Annual Design Automation Conference*. 66 (ACM, 2018).
23. Willers, O., Huth, C., Guajardo, J., Seidel, H. & Deutsch, P. On the feasibility of deriving cryptographic keys from MEMS sensors. *J. Crypt. Eng.* https://doi.org/10.1007/s13389-019-00208-4 (2019).
24. Wang, Y. et al. Flash memory for ubiquitous hardware security functions: true random number generation and device fingerprints. In *Proc. IEEE Symp. Security and Privacy* 33–47 (IEEE, 2012).
25. Tehranipoor, F., Karimian, N., Xiao, K. & Chandy, J. DRAM based intrinsic physical unclonable functions for system level security. In *Proc. Great Lakes Symp. VLSI*. 15–20 (ACM, 2015).
26. Orosa, L. et al. Dataplant: enhancing system security with low-cost in-DRAM value generation primitives. Preprint at https://arxiv.org/abs/1902.07344v2 (2019).
27. Maes, R., Van Herrewege, A. & Verbauwhede, I. PUFKY: a fully functional PUF-based cryptographic key generator. In *Proc. Cryptographic Hardware and Embedded Systems* 302–319 (Springer, 2012).
28. Delvaux, J., Gu, D., Schellekens, D. & Verbauwhede, I. Helper data algorithms for PUF-based key generation: overview and analysis. *IEEE Trans. Comput. -Aided Des. Integr. Circuits Syst.* **34**, 889–902 (2015).
29. Bösch, C., Guajardo, J., Sadeghi, A.-R., Shokrollahi, J. & Tuyls, P. Efficient helper data key extractor on FPGAs. In *Proc. Cryptographic Hardware and Embedded Systems*. 181–197 (Springer, 2008).
Early efficient implementations of fuzzy extractors on FPGA for PUF key generation.
30. Maes, R., Tuyls, P. & Verbauwhede, I. A soft decision helper data algorithm for SRAM PUFs. In *Proc. IEEE Int. Symp. Information Theory* 2101–2105 (IEEE, 2009).
A study of response bit-specific reliability.
31. Maes, R. An accurate probabilistic reliability model for silicon PUFs. In *Proc. Cryptographic Hardware and Embedded Systems* 73–89 (IACR, 2013).
32. Yu, M.-D. & Devadas, S. Secure and robust error correction for physical unclonable functions. *IEEE Des. Test. Comput.* **27**, 48–65 (IEEE, 2010).
33. Hiller, M., Merli, D., Stumpf, F. & Sigl, G. Complementary IBS: application specific error correction for PUFs. In *Proc. IEEE Int. Symp. Hardware-Oriented Security and Trust (HOST)* https://doi.org/10.1109/HST.2012.6224310 (IEEE, 2012).
34. Delvaux, J., Gu, D., Verbauwhede, I., Hiller, M. & Yu, M.-D. Efficient fuzzy extraction of PUF-induced secrets: theory and applications. In *Proc. Cryptographic Hardware and Embedded Systems* 412–431 (Springer, 2016).
35. Maes, R., van der Leest, V., van der Sluis, E. & Willems, F. Secure key generation from biased PUFs: extended version. *J. Crypt. Eng.* **6**, 121–137 (2016).
36. Koeberl, P. et al. Evaluation of a PUF device authentication scheme on a discrete 0.13 µm SRAM. In *Proc. Int. Conf. Trusted Systems* 271–288 (Springer, 2011).
37. Gunn, L. J., Allison, A. & Abbott, D. Allison mixtures: where random digits obey thermodynamic principles. In *Int. J. Mod. Phys.* **33**, 1460360 (2014).
38. Aysu, A., Wang, Y., Schaumont, P. & Orshansky, M. A new maskless debiasing method for lightweight physical unclonable functions. In *Proc. IEEE Int. Symp. Hardware Oriented Security and Trust* 134–139 (IEEE, 2017).
39. Hiller, M. & Önalan, A. G. Hiding secrecy leakage in leaky helper data. In *Proc. Cryptographic Hardware and Embedded Systems* 601–619 (Springer, 2017).
40. Delvaux, J. & Verbauwhede, I. Key-recovery attacks on various RO PUF constructions via helper data manipulation. In *Proc. Conf. Design, Automation & Test in Europe* (IACR, 2014).
41. Delvaux, J. & Verbauwhede, I. Attacking PUF-based pattern matching key generators via helper data manipulation. In *CT-RSA 2014* 106–131 (Springer, 2014).
42. Becker, G. T. Robust fuzzy extractors and helper data manipulation attacks revisited: theory vs practice. *IEEE Trans. Dependable Secur. Comput.* **16**, 783–795 (2019).
This work examines vulnerabilities of various error correction code implementations under helper data manipulation attacks.
43. Boyen, X. Reusable cryptographic fuzzy extractors. In *Proc. 11th ACM Conf. Computer and Communications Security* 82–91 (ACM, 2004).
44. Van Herrewege, A. et al. Reverse fuzzy extractors: enabling lightweight mutual authentication for PUF-enabled RFIDs. In *Proc. Financial Cryptography and Data Security* 374–389 (Springer, 2012).
Work that brought reverse fuzzy extractors to PUFs.
45. Canetti, R., Fuller, B., Paneth, O., Reyzin, L. & Smith, A. Reusable fuzzy extractors for low-entropy distributions. In *Annual Int. Conf. the Theory and Applications of Cryptographic Techniques* 117–146 (Springer, 2016).
46. Kusters, L. et al. Security of helper data schemes for SRAM-PUF in multiple enrollment scenarios. In *Proc. IEEE Int. Symp. Information Theory (ISIT)* 1803–1807 (IEEE, 2017).
47. Roel, M. *Physically Unclonable Functions: Constructions, Properties and Applications*. Ph.D. dissertation, KU Leuven (2012).
Provides details of various PUF structures as well as empirical evaluation results of popular ones.
48. Delvaux, J. *Security Analysis of PUF-Based Key Generation and Entity Authentication*. Ph.D. dissertation, Shanghai Jiao Tong Univ. (2017).
49. Delvaux, J., Gu, D. & Verbauwhede, I. Upper bounds on the min-entropy of RO Sum, arbiter, feed-forward arbiter, and S-ArbRO PUFs. In *Hardware-Oriented Security and Trust (AsianHOST), IEEE Asian* 1–6 (IEEE, 2016).
50. Fuller, B., Meng, X. & Reyzin, L. Computational fuzzy extractors. In *Proc. Conf. Theory and Application of Cryptology and Information Security* 174–193 (Springer, 2013).
51. Herder, C., Ren, L., van Dijk, M., Yu, M.-D. & Devadas, S. Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions. *IEEE Trans. Dependable Secur. Comput.* **14**, 65–82 (2017).
Construction of a fuzzy extractor through complexity theoretical arguments instead of previous information-theoretic arguments.
52. Huth, C., Becker, D., Merchan, J. G., Duplys, P. & Güneysu, T. Securing systems with indispensable entropy: LWE-based lossless computational fuzzy extractor for the internet of things. *IEEE Access* **5**, 11,909–11,926 (2017).
53. Colombier, B., Bossuet, L., Fischer, V. & Hély, D. Key reconciliation protocols for error correction of silicon PUF responses. *IEEE Trans. Inf. Forensics Security* **12**, 1988–2002 (2017).
54. Maiti, A. & Schaumont, P. Improving the quality of a physical unclonable function using configurable ring oscillators. In *Proc. IEEE Int. Conf. Field Programmable Logic and Applications* 703–707 (IEEE, 2009).
55. Maiti, A. & Schaumont, P. Improved ring oscillator PUF: an FPGA friendly secure primitive. *J. Crypt.* **24**, 375–397 (2011).
56. Maiti, A., Kim, I. & Schaumont, P. A robust physical unclonable function with enhanced challenge-response set. *IEEE Trans. Inf. Forensics Security* **7**, 333–345 (2012).
57. Cao, Y., Zhang, L., Chang, C.-H. & Chen, S. A low-power hybrid RO PUF with improved thermal stability for lightweight applications. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **34**, 1143–1147 (2015).
58. Liu, C. Q., Cao, Y. & Chang, C. H. ACRO-PUF: a low-power, reliable and aging-resilient current starved inverter-based ring oscillator physical unclonable function. *IEEE Trans. Circuits Syst. I, Reg. Pap.* **64**, 3138–3149 (2017).
59. Su, Y., Holleman, J. & Otis, B. P. A digital 1.6 pJ/bit chip identification circuit using process variations. *IEEE J. Solid-State Circuits* **43**, 69–77 (2008).
60. Maes, R., Tuyls, P. & Verbauwhede, I. Intrinsic PUFs from flip-flops on reconfigurable devices. In *Proc. 3rd Benelux Workshop on Information and System Security* (2008).
61. van der Leest, V., Schrijen, G.-J., Handschuh, H. & Tuyls, P. Hardware intrinsic security from D flip-flops. In *Proc. 5th ACM Workshop on Scalable Trusted Computing* 53–62 (ACM, 2010).
62. Kumar, S. S., Guajardo, J., Maes, R., Schrijen, G.-J. & Tuyls, P. The butterfly PUF protecting IP on every FPGA. In *Proc. IEEE Int. Symp. Hardware-Oriented Security and Trust* 67–70 (IEEE, 2008).
63. Alioto, M. & Alvarez. A. *Physically Unclonable Function Database* (National Univ. of Singapore, accessed 7 January 2020); http://www.green-ic.org/pufdb
64. Brzuska, C., Fischlin, M., Schröder, H. & Katzenbeisser, S. Physically uncloneable functions in the universal composition framework. In *Annual Cryptology Conference* 51–70 (Springer, 2011).
65. Rührmair, U. & van Dijk, M. Practical security analysis of PUF-based two-player protocols. In *Proc. Cryptographic Hardware and Embedded Systems* 251–267 (Springer, 2012).
66. Rührmair, U. & van Dijk, M. On the practical use of physical unclonable functions in oblivious transfer and bit commitment protocols. *J. Crypt. Eng.* **3**, 17–28 (2013).
67. Damgård, I. & Scafuro, A. Unconditionally secure and universally composable commitments from physical assumptions. In *Proc. Int. Conf. Theory and Application of Cryptology and Information Security* 100–119 (Springer, 2013).
68. Rührmair, U. & Van Dijk, M. PUFs in security protocols: attack models and security evaluations. In *Proc. IEEE Symp. Security and Privacy* 286–300 (IEEE, 2013).

69. Carro-Temboury, M. R., Arppe, R., Vosch, T. & Sørensen, T. J. An optical authentication system based on imaging of excitation-selected lanthanide luminescence. *Sci. Adv.* **4**, e1701384 (2018).
70. Rührmair, U. et al. Applications of high-capacity crossbar memories in cryptography. *IEEE Trans. Nanotechnol.* **10**, 489–498 (2011).
71. Lim, D. *Extracting Secret Keys from Integrated Circuits*. Ph.D. dissertation, MIT (2004).
72. Kursawe, K., Sadeghi, A., Schellekens, D., Skoric, B. & Tuyls, P. Reconfigurable physical unclonable functions-enabling technology for tamper-resistant storage. In *Proc. IEEE Int. Workshop on Hardware-Oriented Security and Trust* 22–29 (IEEE, 2009).
73. Majzoobi, M., Koushanfar, F. & Potkonjak, M. Techniques for design and implementation of secure reconfigurable PUFs. *ACM Trans. Reconfigurable Tech. Syst.* **2**, 5 (2009).
74. Mahmoodi, M., Nili, H., Larimian, S., Guo, X. & Strukov, D. Chipsecure: a reconfigurable analog eflash-based PUF with machine learning attack resiliency in 55 nm CMOS. In *Proc. 56th Annual Design Automation Conference* 8806766 (ACM, 2019).
75. Katzenbeisser, S. et al. Recyclable PUFs: logically reconfigurable PUFs. *J. Crypt. Eng.* **1**, 177–186 (2011).
76. Rührmair, U., Jaeger, C., & Algasinger, M. An attack on PUF-based session key exchange and a hardware-based countermeasure: erasable PUFs. In *Proc. Financial Cryptography and Data Security* 190–204 (Springer, 2011).
77. Beckmann. N & Potkonjak, M. Hardware-based public-key cryptography with public physically unclonable functions. In *Inform. Hiding* **5806**, 206–220 (Springer, 2009).
78. Rührmair, U. *SIMPL Systems: On a Public Key Variant of Physical Unclonable Functions* (IACR, 2009); https://eprint.iacr.org/2009/255
79. Rührmair, U. *Towards Secret-Free Security* (IACR, 2019); https://eprint.iacr.org/2019/388
80. Majzoobi, M. & Koushanfar, F. Time-bounded authentication of FPGAs. *IEEE Trans. Inf. Forensics Security* **6**, 1123–1135 (2011).
81. Rajendran, J., Rose, G. S., Karri, R. & Potkonjak, M. Nano-PPUF: a memristor-based security primitive. In *Proc. IEEE Computer Society Annual Symposium on VLSI* 84–87 (IEEE, 2012).
82. Wendt, J. B. & Potkonjak, M. The bidirectional polyomino partitioned PPUF as a hardware security primitive. In *Proc. IEEE Global Conf. Signal and Information Processing* 257–260 (IEEE, 2013).
83. Potkonjak, M. & Goudar, V. Public physical unclonable functions. *Proc. IEEE* **102**, 1142–1156 (2014).
84. Chakraborty, R. S. & Bhunia, S. HARPOON: an obfuscation-based SoC design methodology for hardware protection. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **28**, 1493–1502 (2009).
85. Koushanfar, F. Provably secure active IC metering techniques for piracy avoidance and digital rights management. *IEEE Trans. Inf. Forensics Security* **7**, 51–63 (2012).
86. Zhang, J., Lin, Y., Lyu, Y. & Qu, G. A PUF-FSM binding scheme for FPGA IP protection and pay-per-device licensing. *IEEE Trans. Inf. Forensics Security* **10**, 1137–1150 (2015).
87. Delvaux, J., Peeters, R., Gu, D. & Verbauwhede, I. A survey on lightweight entity authentication with strong PUFs. *ACM Comput. Surv. (CSUR)* **48**, 26 (2015).
88. Delvaux, J. & Verbauwhede, I. Side channel modeling attacks on 65nm arbiter PUFs exploiting CMOS device noise. In *Proc. IEEE Int. Symp. Hardware-Oriented Security and Trust (HOST)* 137–142 (IEEE, 2013).
Exploitation of the unreliability side-channel to perform modelling attacks on APUFs, which becomes a main attack surface.
89. Hiller, M. et al. Low-area reed decoding in a generalized concatenated code construction for PUFs. In *Proc. IEEE Computer Society Annual Symp. VLSI* 143–148 (IEEE, 2015).
90. Hiller, M. *Key Derivation with Physical Unclonable Functions*. Ph.D. dissertation, Technical Univ. Munich (2016).
91. Xu, T. & Potkonjak, M. Digital PUF using intentional faults. In *Proc. IEEE Int. Symp. Quality Electronic Design* 448–451 (IEEE, 2015).
92. Miao, J., Li, M., Roy, S. & Yu, B. LRR-DPUF: learning resilient and reliable digital physical unclonable function. In *Proc. IEEE Int. Conf. Computer-Aided Design* https://doi.org/10.1145/2966986.2967051 (IEEE, 2016).
93. Bhargava, M. Mai, K. A high reliability PUF using hot carrier injection based response reinforcement. In *Proc. Cryptographic Hardware and Embedded Systems* 90–106 (Springer, 2013).
94. Wang, W.-C., Yona, Y., Diggavi, S. & Gupta, P. LEDPUF: stability-guaranteed physical unclonable functions through locally enhanced defectivity. In *Proc. IEEE Int. Symp. Hardware Oriented Security and Trust* 25–30 (IEEE, 2016).
95. Chuang, K.-H. et al. Physically unclonable function using CMOS breakdown position. In *Proc. IEEE Reliability Physics Symp. (IRPS)* 4C–1 (IEEE, 2017).
96. Bhargava, M. & Mai, K. An efficient reliable PUF-based cryptographic key generator in 65 nm CMOS. In *Proc. Conf. Design, Automation & Test in Europe* 6800284 (IEEE, 2014).
97. Xu, X., Rahmati, A., Holcomb, D. E., Fu, K. & Burleson, W. Reliable physical unclonable functions using data retention voltage of SRAM cells. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **34**, 903–914 (2015).
98. Gao, Y., Ma, H., Al-Sarawi, S. F., Abbott, D. & Ranasinghe, D. C. PUF-FSM: a controlled strong PUF. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **64**, 2532–2543 (2017).
99. Zeitouni, S., Oren, Y., Wachsmann, C., Koeberl, P. & Sadeghi, A.-R. Remanence decay side-channel: the PUF case. *IEEE Trans. Inf. Forensics Security* **11**, 1106–1116 (2016).
100. Helfmeier, C., Boit, C., Nedospasov, D., & Seifert, J.-P. Cloning physically unclonable functions. In *Proc. IEEE Int. Symp. Hardware-Oriented Security and Trust* https://doi.org/10.1109/HST.2013.6581556 (IEEE, 2013).
101. Tajik, S. et al. Photonic side-channel analysis of arbiter PUFs. *J. Crypt.* **30**, 550–571 (2017).
102. Boit, C. et al. From IC debug to hardware security risk: the power of backside access and optical interaction. In *Proc. IEEE Int. Symp. the Physical and Failure Analysis of Integrated Circuits (IPFA)* 365–369 (IEEE, 2016).
103. Sauer, M. et al. Sensitized path PUF: a lightweight embedded physical unclonable function In *Proc. IEEE Conf. Design, Automation & Test in Europe* 680–685 (IEEE, 2017).
104. Rührmair, U. et al. Efficient power and timing side channels for physical unclonable functions. In *Proc. Cryptographic Hardware and Embedded Systems* 476–492 (Springer, 2014).
Side-channel attacks via power and timing information on PUFs.
105. Becker, G. T. & Kumar, R. *Active and Passive Side-Channel Attacks on Delay Based PUF Designs* (IACR, 2014); https://eprint.iacr.org/2014/287
106. Tiri, K., Akmal, M. & Verbauwhede, I. A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards. In *Proc. 28th European Solid-State Circuits Conf.* 403–406 (IEEE, 2002).
107. Delvaux, J. Machine-learning attacks on PolyPUFs, OB-PUFs, RPUFs, LHS-PUFs, and PUF–FSMs. *IEEE Trans. Inf. Foren. Sec.* **14**, 2043–2058 (2019).
108. He, J., Zhao, Y., Guo, X. & Jin, Y. Hardware trojan detection through chip-free electromagnetic side-channel statistical analysis. *IEEE Trans. VLSI Syst.* **25**, 2939–2948 (2017).
109. Rostami, M., Koushanfar, F. & Karri, R. A primer on hardware security: models, methods, and metrics. *Proc. IEEE* **102**, 1283–1295 (2014).
110. Avvaru, S. & Parhi, K. K. Feed-forward XOR PUFs: reliability and attack-resistance analysis. In *Proc. Great Lakes Symp. VLSI*. 287–290 (ACM, 2019).
111. Yu M.-D. & Devadas, S. Recombination of physical unclonable functions. In *GOMACTech-10 Conference* http://hdl.handle.net/1721.1/59817 (United States Dept. of Defence, 2010).
112. Vijayakumar, A., Patil, V. C., Prado, C. B. & Kundu, S. Machine learning resistant strong PUF: possible or a pipe dream? In *Proc. IEEE Int. Symp. Hardware Oriented Security and Trust (HOST)* 19–24 (IEEE, 2016).
113. Biggio, B. & Roli, F. Wild patterns: ten years after the rise of adversarial machine learning. *Pattern Recogn.* **84**, 317–331 (2018).
114. Wang, S.-J., Chen, Y.-S. & Li, K. S.-M. Adversarial attack against modeling attack on PUFs. In *Proc. ACM 56th Annual Design Automation Conf.* 8806766 (ACM, 2019).
115. Tuyls, P. et al. Read-proof hardware from protective coatings. In *Proc. Cryptographic Hardware and Embedded Systems* 369–383 (Springer, 2006).
116. Immler, V., Obermaier, J., König, M., Hiller, M. & Sig, G. B-TREPID: batteryless tamper-resistant envelope with a PUF and integrity detection. In *Proc. IEEE Int. Symp. Hardware Oriented Security and Trust* 49–56 (IEEE, 2018).
117. Obermaier, J., Hiller, M., Immler, V. & Sigl, G. A measurement system for capacitive PUF-based security enclosures. In *Proc. Design Automation Conf.* https://doi.org/10.1109/DAC.2018.8465886 (IEEE, 2018).
118. Anderson, B. R., Gunawidjaja, R. & Eilers, H. Initial tamper tests of novel tamper-indicating optical physical unclonable functions. *Appl. Opt.* **56**, 2863–2872 (2017).
119. Gassend, B., Clarke, D., Van Dijk, M. & Devadas, S. Controlled physical random functions. In *Proc. 18th IEEE Annual Computer Security Applications Conference* 149–160 (IEEE, 2002).
120. Gassend, B. et al. Controlled physical random functions and applications. *ACM Trans. Inform. Syst. Sec.* **10**, 3 (2008).
121. Liu, R., Wu, H., Pang, Y., Qian, H. & Yu, S. A highly reliable and tamper-resistant RRAM PUF: design and experimental validation. In *Proc. IEEE Int. Symp. Hardware Oriented Security and Trust* 13–18 (IEEE, 2016).
122. Gao, Y., Su, Y., Xu, L. & Ranasinghe, D. C. Lightweight (reverse) fuzzy extractor with multiple reference PUF responses. *IEEE Trans. Inf. Foren. Sec.* **14**, 1887–1901 (2019).

123. Xu, X., Burleson, W. & Holcomb, D. E. Using statistical models to improve the reliability of delay-based PUFs. In *Proc. IEEE Computer Society Annual Symposium on VLSI* 547–552 (IEEE, 2016).
124. Holcomb, D. E., Burleson, W. P. & Fu, K. Power-up SRAM state as an identifying fingerprint and source of true random numbers. *IEEE Trans. Comput.* **58**, 1198–1210 (2009).
125. Keller, C., Gurkaynak, F., Kaeslin, H. & Felber, N. Dynamic memory-based physically unclonable function for the generation of unique identifiers and true random numbers. In *Proc. IEEE Int. Symp. Circuits and Systems* 2740–2743 (IEEE, 2014).
126. Ranasinghe, D. C., Lim, D., Devadas, S., Abbott, D. & Cole, P. H. Random numbers from metastability and thermal noise. *Electron. Lett.* **41**, 891–893 (2005).
127. Gao, Y., Ma, H., Abbott, D. & Al-Sarawi, S. F. PUF sensor: exploiting PUF unreliability for secure wireless sensing. *IEEE Trans. Circuits Syst. I: Reg. Pap.* **64**, 2532–2543 (2017).
128. Rosenfeld, K., Gavas, E., & Karri, R. Sensor physical unclonable functions. In *Proc. IEEE. Int. Symp. Hardware-Oriented Security and Trust (HOST)* 112–117 (IEEE 2010).
129. Guo, Z., Xu, X., Rahman, M. T., Tehranipoor, M. M. & Forte, D. SCARe: an SRAM-based countermeasure against IC recycling. *IEEE Trans. VLSI Syst.* **26**, 744–755 (2018).
130. Rührmair, U. et al. Virtual proofs of reality and their physical implementation. In *Proc. 36th IEEE Symp. Security and Privacy* 70–85 (IEEE, 2015).
131. Herder, C., Fuller, B., van Dijk, M. & Devadas, S. *Public Key Cryptosystems with Noisy Secret Keys* (IACR, 2017); https://eprint.iacr.org/2017/210 (2017).
132. Islam, M. N. & Kundu, S. Enabling IC traceability via blockchain pegged to embedded PUF. *ACM Trans. Des. Autom. Electron. Syst.* **24**, 36 (2019).
133. Scheel, R. A. & Tyagi, A. Characterizing composite user-device touchscreen physical unclonable functions for mobile device authentication. In *Proc. 5th Int. Workshop on Trustworthy Embedded Devices* 3–13 (ACM, 2015).
134. Yu, S. & Chen, P.-Y. Emerging memory technologies: recent trends and prospects. *IEEE Solid-State Circ. Mag.* **8**, 43–56 (2016).
135. Wong, H.-S. P. & Salahuddin, S. Memory leads the way to better computing. *Nat. Nanotechnol.* **10**, 191–194 (2015).
136. 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).
137. Carboni, R. & Ielmini, D. Stochastic memory devices for security and computing. *Adv. Electron. Mat.* **5**, 1900198 (2019).
138. Zhang, L., Kong, Z. H. & Chang, C.-H. PCKGen: a phase change memory based cryptographic key generator. In *Proc. IEEE Int. Symp. Circuits and Systems* 1444–1447 (IEEE, 2013).
139. Che, W., Plusquellic, J. & Bhunia, S. A non-volatile memory based physically unclonable function without helper data. In *Proc. IEEE/ACM Int. Conf. Computer-Aided Design* 148–153 (IEEE, 2014).
140. Pang, Y. et al. A reconfigurable RRAM physically unclonable function utilizing post-process randomness source with $<6\times10^{-6}$ native bit error rate. In *IEEE Int. Solid-State Circuits Conference-(ISSCC)* 402–404 (IEEE, 2019).
141. Xie, Y. et al. Security and vulnerability implications of 3D ICs. *IEEE Trans. Multi-Scale Comput. Syst.* **2**, 108–122 (IEEE, 2016).
142. Gao, Y., Ranasinghe, D. C., Al-Sarawi, S. F., Kavehei, O. & Abbott, D. Memristive crypto primitive for building highly secure physical unclonable functions. *Sci. Rep.* **5**, 12785 (2015).
143. 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).
144. Lee, G. S., Kim, G.-H., Kwak, K., Jeong, D. S. & Ju, H. Enhanced reconfigurable physical unclonable function based on stochastic nature of multilevel cell RRAM. *IEEE Trans. Electron Devices* **66**, 1717–1721 (2019).
145. Karam, R., Liu, R., Chen, P.-Y., Yu, S. & Bhunia, S. Security primitive design with nanoscale devices: a case study with resistive RAM. In *Proc. IEEE International Great Lakes Symposium on VLSI* 299–304 (IEEE, 2016).
146. IoT connected devices to reach 20.4 billion by 2020, says Gartner. *Which-50* https://go.nature.com/386hJ0q (2017).
147. Lim, D. et al. Extracting secret keys from integrated circuits. *IEEE Trans. VLSI Syst.* **13**, 1200–1205 (2005).
148. Delvaux, J. & Verbauwhede, I. Fault injection modeling attacks on 65 nm arbiter and RO sum PUFs via environmental changes. *IEEE Trans. Circuits Syst. I: Reg. Pap.* **61**, 1701–1713 (2014).
149. Ganji, F., Krämer, J., Seifert, J.-P. & Tajik, S. Lattice basis reduction attack against physically unclonable functions. In *Proc. ACM Conf. Computer and Communications Security* 1070–1080 (ACM, 2015).
150. Merli, D. et al. Localized electromagnetic analysis of RO PUFs. In *Int. IEEE Symp. Hardware-Oriented Security and Trust* 19–24 (IEEE, 2013).
151. Nguyen, P. H., Sahoo, D. P., Chakraborty, R. S. & Mukhopadhyay, D. Efficient attacks on robust ring oscillator PUF with enhanced challenge-response set. In *Proc. Design, Automation & Test in Europe Conference & Exhibition* 641–646 (EDA Consortium, 2015).

Acknowledgements

We acknowledge support from NJUST Research Start-Up Funding (AE89991/039) and National Natural Science Foundation of China (61802186).

Author contributions

Y.G., S.F.A.-S. and D.A. conceived the project, carried out the discussions and wrote 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/s41928-020-0372-5.

Correspondence should be addressed to Y.G., S.F.A.-S. or D.A.

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

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.