**Micro- and nanoelectronics. Condensed matter physics**

**Микро- и наноэлектроника. Физика конденсированного состояния**

*UDC 004.832.32*
*https://doi.org/10.32362/2500-316X-2026-14-2-80-102*
*EDN LLZOKJ*



REVIEW ARTICLE

# Physically unclonable functions in digital integrated circuits

**Evgenii Ph. Pevtsov [@], Tatyana A. Demenkova, Yuri A. Korotaev [@], Alexander S. Sigov**

*MIREA – Russian Technological University, Moscow, 119454 Russia*
*[@] Corresponding authors, e-mail: pevtsov@mirea.ru, korotaevya@yandex.ru*

• **Submitted:** 16.09.2025 • **Revised:** 23.09.2025 • **Accepted:** 12.02.2026

**Abstract**

**Objectives.** Modules that implement physically unclonable functions (PUFs) within a digital chip facilitate the direct use of challenge–response pairs by device applications that can query and read the PUF without external tools or exposing data outside the chip. A PUF can be implemented using technological processes and components already applied in device fabrication. The first of two reviews on PUFs as elements of hardware security infrastructure, the present paper focuses on the formal description of PUFs and designs based on memory modules and timing analysis.

**Methods.** The following quantitative indicators were applied to formally describe PUFs: computability, uniqueness, feasibility, cloning resistance, and protection against unauthorized access.

**Results.** PUFs are considered as physical devices with unique signatures. A classification into three PUF groups is proposed: delay-based, memory-based, and analog. Typical examples of the first two groups are outlined. While delay-based solutions provide a large challenge–response space, they require symmetry and/or calibration. In contrast, memory-based PUFs are easier to implement in integrated circuits. With suitable post-processing, they can achieve high reproducibility, making them practical for many applications. Approaches to mitigating voltage and temperature variations are described along with examples of strong memory-oriented PUFs and circuit techniques that enhance resistance to attacks.

**Conclusions.** PUF-based security technologies demonstrate significant potential, particularly for the Internet of Things. When combined with post-processing and compensation methods, PUFs constitute a mature and effective tool for hardware security.

**Keywords:** physically unclonable function, PUF, integrated circuits, hardware security, arbiter PUF, memory-based PUF, SRAM, DRAM, Internet of Things

**For citation:** Pevtsov E.Ph., Demenkova T.A., Korotaev Yu.A., Sigov A.S. Physically unclonable functions in digital integrated circuits. *Russian Technological Journal.* 2026;14(2):80–102. https://doi.org/10.32362/2500-316X-2026-14-2-80-102, https://www.elibrary.ru/LLZOKJ

**Financial disclosure:** The authors have no financial or proprietary interest in any material or method mentioned.

The authors declare no conflicts of interest.

ОБЗОРНАЯ СТАТЬЯ

# Физически неклонируемые функции в цифровых интегральных схемах

**Е.Ф. Певцов [@], Т.А. Деменкова, Ю.А. Коротаев [@], А.С. Сигов**

*MIREA – Russian Technological University, Moscow, 119454 Russia*
[@] *Авторы для переписки, e-mail: pevtsov@mirea.ru, korotaevya@yandex.ru*

• **Поступила:** 16.09.2025 • **Доработана:** 23.09.2025 • **Принята к опубликованию:** 12.02.2026

**Резюме**

**Цели.** Преимуществом модулей, реализующих физически неклонируемые функции (ФНФ) и встроенных в цифровой чип, является то, что отклики на запросы могут быть напрямую использованы другими приложениями устройства. Устройство способно запрашивать и считывать ФНФ без привлечения внешних инструментов и вывода запроса и ответа за пределы чипа. ФНФ может быть реализована с использованием технологических операций и компонентов, применяемых при изготовлении самого устройства. Статья является первой из двух обзорных публикаций, посвященных ФНФ как компонентам инфраструктуры аппаратной безопасности. Данная статья фокусируется на формальном описании ФНФ и их конструкциях, основанных на модулях памяти и анализе временных характеристик сигналов.

**Методы.** Использованы методы определения количественных показателей и признаков для формального описания ФНФ: вычислимость, уникальность, возможность реализации, сложность клонирования, защита от несанкционированного доступа.

**Результаты.** Рассмотрены реализации ФНФ как физических устройств, обладающих уникальной сигнатурой. Предложена их классификация: ФНФ на основе временных характеристик сигналов, ФНФ на основе схем памяти и аналоговые ФНФ. Приведены наиболее типичные примеры реализаций первых двух типов. Показано, что решения на основе задержек сигналов обеспечивают широкое пространство пар «запрос – ответ», но требуют симметрии и/или калибровки, тогда как ФНФ на базе памяти проще реализуются в интегральных схемах и при корректной постобработке достигают высокой воспроизводимости, что делает их практичным выбором для многих приложений. Описаны подходы к компенсации влияния вариаций напряжения и температуры. Приведены примеры «сильных» память-ориентированных ФНФ и схемотехнические приемы повышения их стойкости к атакам.

**Выводы.** Технология обеспечения безопасности на основе ФНФ обладает значительным потенциалом, особенно для применения в устройствах интернета вещей. Проведенный анализ показывает, что в сочетании с методами постобработки и компенсации эксплуатационных факторов ФНФ является зрелым инструментом обеспечения аппаратной безопасности.

**Ключевые слова:** физически неклонируемая функция, ФНФ, интегральные схемы, аппаратная безопасность, ФНФ типа «арбитр», ФНФ на основе памяти, SRAM, DRAM, интернет вещей

**Для цитирования:** Певцов Е.Ф., Деменкова Т.А., Коротаев Ю.А., Сигов А.С. Физически неклонируемые функции в цифровых интегральных схемах. *Russian Technological Journal*. 2026;14(2):80–102. https://doi.org/10.32362/2500-316X-2026-14-2-80-102, https://www.elibrary.ru/LLZOKJ

**Прозрачность финансовой деятельности:** Авторы не имеют финансовой заинтересованности в представленных материалах или методах.

Авторы заявляют об отсутствии конфликта интересов.

## INTRODUCTION

The presence of sensitive intellectual data in hardware devices designed to carry out specialized artificial intelligence (AI) tasks makes them an attractive target for cyberattacks. As well as intercepting data for financial gain by compromising the security of these devices, attackers can steal intellectual property in order to reverse engineer them and produce counterfeit versions. In addition to producing counterfeit copies, refurbished or re-manufactured devices may also be sold as new products, resulting in lost revenue for original manufacturers and security concerns due to reduced product lifespan and reliability.

Equipment security can be ensured through the physical implementation of secure circuits. Such circuits are used along with other increasingly complex methods of protection against counterfeiting for device authentication and random-access key generation. Security circuits have a unique signature resembling that of human retinal, fingerprint, and DNA patterns. Due to the random nature of such signatures, they are difficult to predict and clone, thus preventing unauthorized access to data. In this context, the implementation of reliable hardware platforms for secure communication, device authentication, and protection against various software and hardware risks and hacker attacks becomes a priority research direction.

A physically unclonable function (PUF) is a physical object whose functionality cannot be duplicated (cloned) through physical means (e.g., by creating another system based on the same technology). Given set of input data and operating conditions (i.e., challenge), a PUF provides a unique output signal (i.e., response) that acts as a digital fingerprint or unique identifier for a specific instance of the device. This property makes PUFs useful in applications that require high levels of security, such as in cryptographic systems, Internet of Things (IoT) devices, and other applications that prioritize privacy protection.

By definition, a PUF performs a specific operation when provided with certain input data to produce an output that can be evaluated or measured. In the engineering sense, a PUF should be considered a function, i.e., a procedure performed by or affecting a specific (physical) system. Typically, the input data for a PUF is referred to as a challenge, to which a specific response is generated at the output. The combination of the challenge and response is commonly referred to as a challenge-response pair (CRP). The CRP generation process is defined by the relationship between challenges and responses established using a specific PUF implementation.

PUFs have been widely studied in the scientific literature, particularly due to the importance of ensuring the security of IoT devices [1]. The paper summarizes the findings presented in recent publications regarding modern PUFs and their implementations [2-4].

## QUANTITATIVE PUF INDICATORS

A comprehensive explanation of the features that can be utilized to evaluate various PUF implementations is provided in [5–7]. Classification and identification theories are applied within a single PUF type as well as for comparing different types of PUFs.

The effects of a particular PUF design are measured using two types of quantitative indicators:

- Inter-distance. This metric measures the difference in responses obtained from a single challenge for two instances of the same PUF. The corresponding designation for this indicator is proposed in [7] and represents a random variable that describes the distance between the responses of two PUF instances from the same call.
- Intra-distance. A quantitative measure that describes the difference between two estimates of a single PUF instance. It represents the difference in response values when the same challenge is applied twice to the same physical implementation of the PUF. Such an intra-distance metric is a random variable that describes the distance between response values from the same PUF instance using the same challenge.

These indicators, which determine the reproducibility and uniqueness of a PUF, use the same challenge for both metrics. However, the specific value of the quantitative characteristics of one or more instances of a PUF may vary depending on the complexity and number of tests. In other words, the quantitative estimates used to measure these characteristics may vary depending on the nature of the response. In cases where the response is a bit string, the inter-Hamming distance (inter-HD) is used as the criterion.

In order to summarize the inter- and intra-distance characteristics of a specific type of PUF, histograms are often used. Such histograms show the results of running a number of challenges on one PUF device and the results of running a series of different challenges on several PUF devices of the same type. As stated in [6], both histograms can be approximated by a Gaussian distribution in many cases, with respective mean values $\mu_{inter}$ and $\mu_{intra}$ and, if available, respective standard deviations $\sigma_{inter}$ and $\sigma_{intra}$.

From the definition it follows that $\mu_{intra}$ represents the average noise level of the responses to characterize the reproducibility of the measured response compared to other observations of the same response. Clearly,

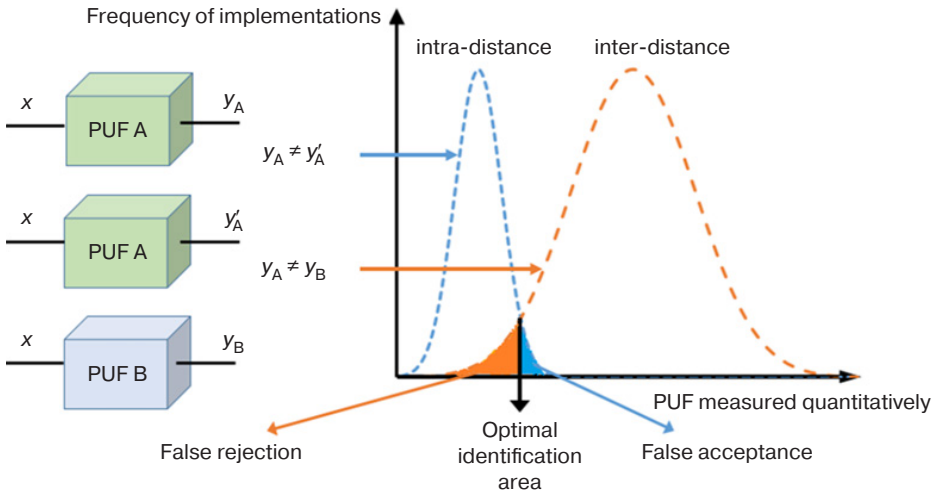


**Fig. 1.** Quantitative characteristics of a PUF [6].
Here, $x$ is the challenge; $y_A$ and $y'_A$ are the responses of a specific PUF instance to the challenge; $y_B$ is the response of a different PUF instance to the same challenge

the smaller the $\mu_{intra}$ value, the more reliable the responses implemented by a given PUF. Conversely, $\mu_{inter}$ expresses uniqueness; it measures the average uniqueness of two systems based on their PUF responses. If the responses are bit strings, the best uniqueness is achieved when half of the bits differ on average. In particular, if $\mu_{inter}$ is expressed as a relative value of the Hamming distance, the optimal result is a value close to 50%. In practice, implementing the minimum $\mu_{intra}$ alongside 50% $\mu_{inter}$ strikes a balance between the methods used to implement PUF. Figure 1 depicts the practical application of these concepts on an example of using a PUF for identification purposes [6].

Since an obtained PUF response is usually associated with physical measurement, there are a number of undesirable side effects that can affect the result. These include random noise and measurement errors. Consequently, the same challenge does not necessarily elicit the same response, resulting in what is known as intra-distance (see definition) between PUF responses. External factors, such as temperature or supply voltage when the PUF is implemented in an integrated circuit (IC), also systematically influence the response measurement. Therefore, in order to correctly compare different results from the literature, it is necessary to consider the conditions under which the $\mu_{intra}$ values have been obtained. An example of the influence of temperature on the reproducibility of a PUF response is given in [8]. If the environmental impact is systematic, methods can be employed to minimize its effect on the PUF response. Other options include the introduction of compensation coefficients [9] and special PUF implementation strategies that minimize dependence on the environment [7, 10].

In terms of effectiveness, PUFs can be classified as weak or strong. A PUF is considered weak if there are only a few combinations of CRP with reactions that are generally insensitive to changes in the environment. Although weak PUFs have relatively low resistance to hacker attacks, they can be used to create secret keys due to their high stability. If a PUF is strong, the number of CRPs in the device is sufficiently large that an attacker cannot destroy the identification system in real time. Therefore, when physically implementing a PUF, special attention should be paid to protection against attacks aimed at cracking the PUF, particularly those using machine learning methods.

The standard procedure for characterizing a PUF involves running statistical tests to determine the degree of randomness of binary sequences generated by hardware or software random number generators. These tests, which are developed by the Information Technology Laboratory at the National Institute of Standards and Technology (NIST)[1], are based on statistical properties that are unique to random sequences.

## FORMAL PUF DESCRIPTION

An attempt to formally describe the PUF based on a description of the physical procedure for responding to challenges was carried out in [7]. Rather than being regarded as a purely abstract concept, the creation of any PUF instance is always associated with a specific physical object. PUF is defined as a procedure Π represented by some input-output functionality,

[1] https://www.nist.gov/. Accessed July 19, 2025.

which can be formally expressed as the c-response transformation of the PUF Π: X → Y: Π($x$). A challenge-response procedure can be formally classified as a PUF if it exhibits the following properties:

1. Computability. Given procedures Π and arguments $x$ from set X, it is possible to compute Y = Π($x$) in polynomial algorithmic time. When it is integrated into a chip, it is therefore necessary for a PUF to be created with minimal effort, for example, under conditions of limited time, space, power, and energy consumption. Clearly, if a PUF is computable, it can be constructed. It is also clear that all versions of PUF that provide experimental results can be constructed and, at least in theory, evaluated.
2. Uniqueness. Π($x$) contains certain information about the identity of the physical object implementing Π. If a set of PUF instances is clearly defined, identification can be made from this set based on the PUF Π($x$) response. Sequential responses enable identification uncertainties to be reduced until a single PUF instance is optimal for identification. In this case, the challenge-response set under consideration will uniquely identify the PUF in the set of objects. Depending on the size of the set and the characteristics of the PUF responses, such an unique identification may or may not be possible. One possible indicator of uniqueness is the histogram of intermediate inter-distance metrics, summed by their $\mu_{inter}$ mean value, as shown in most experimental results.
3. Reproducibility. $y$ = Π($x$) can be reproduced with sufficient error for PUF identification. Responses to different challenges $x$ within the same PUF Π should be similar in terms of the response difference metric under consideration. When interpreting experimental results, these are primarily measured using an intra-distance histogram and summed by their mean value, $\mu_{intra}$. Reproducibility is a property that differentiates PUFs from true random number generators (TRNGs).
4. Unclonability. For a given procedure Π, there is no other procedure Γ that is not equivalent to Π and such that $\forall x \in X\ \Gamma(x) \approx \Pi(x)$ with an accuracy of implementation error. It should be noted that the cloning procedure Γ is not necessarily physically implementable; physical and mathematical unclonability differ. If it is difficult to find a physical object containing another PUF $\Pi_\Gamma \neq \Pi$, so that $\forall x \in X\ \Pi_\Gamma(x) \neq \Pi(x)$, then it is claimed that implementing the PUF is physically impossible. Even the manufacturer of the original PUF would find it difficult to create a physical clone. This is referred to as "manufacturer resistance." If an abstract mathematical procedure $f_\Gamma$ cannot be devised such that $\forall x \in X\ f_\Gamma(x) \approx \Pi(x)$, then it may be claimed that Π is mathematically undecidable. Physical and mathematical unclonability are fundamentally different properties; i.e., a structure may be able to be easily cloned physically but not mathematically, or *vice versa*. For a PUF to be truly unclonable, its implementation procedure Π must be both physically and mathematically unclonable. It should be noted that physical cloning can be very difficult or even impossible, whereas proving unclonability in theory is also very challenging. In principle, systems based on quantum physics are impossible to clone in practice.
5. Unpredictability. For a set of procedures Q = $\{f(x_i, y_i) = \Pi(x_i)\}$ having an error margin, it is impossible to determine $y_c \approx \Pi(x_c)$ for a random challenge $x_c$ such that $(x_c) \notin$ Q. If the PUF response to a random challenge could be accurately predicted simply by observing the CRP set, it would be easy to create a mathematical clone with access to the full catalogue of PUF response options.
6. One-wayness. For any argument $y$ and procedure Π within an acceptable error margin, it is impossible to find an $x$ such that Π($x$) = $y$. In some papers, PUFs are simplistically described as physical versions of one-way cryptographic functions [11].
7. Obvious interference. Modification of the physical object described by procedure Π during the implementation of transforming Π → Π′, such that $\forall x \in X\ \Pi(x) \neq \Pi'(x)$ is true with high probability even when considering the Π implementation error. A distinction should be made between systems that protect against unauthorized access (i.e., systems where interference does not actually result in the acquisition of any useful information) and systems in which interference is obvious and harmful (i.e., systems in which interference with a physical object containing a PUF changes the CRP behavior).

## PUF IMPLEMENTATIONS BASED ON ANALYSIS OF SIGNAL TIME CHARACTERISTICS

A major advantage of embedding a PUF in a digital chip consists in the possibility for responses to challenges to be used directly by other applications running on the same device. Notably, the device can query and read its own PUF without requiring external tools such that the challenge and response never leave the device. Furthermore, the PUF can be implemented using only the technological operations and components used to manufacture the device in which the PUF node is located. This requires virtually no additional cost.

Several options for classifying PUFs are provided in review publications [2, 3, 12–18]. These include the time of the first developments, physical

design properties (hybrid or fully electronic) and implementation technologies (electronic, optical, radio frequency, or magnetic), CRP pair sizes, and areas of application. In summary, there are three main types of PUF implementation in ICs: (1) based on signal timing characteristics; (2) based on memory circuits; (3) analog and passive.

### Ring oscillator (RO) PUFs

These devices exploit the frequency mismatch effect of ring oscillators based on inverters to create the PUF [19]. Due to manufacturing variations, even two nominally identical ring oscillators implemented on a single chip will have a detectable frequency difference. As shown in Fig. 2, an array of *N* oscillators is embedded in RO-PUF.

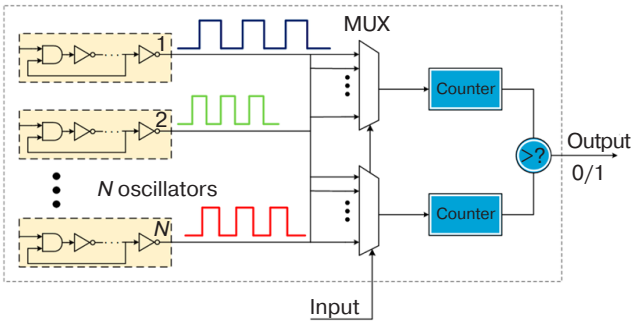


**Fig. 2.** Example of RO-PUF implementation [3]. MUX is a multiplexer

By comparing the frequencies of two ring oscillators, a response bit is produced. Since the challenge consists of the number or location of ring oscillators, the response is the result of the difference in their oscillation frequencies. To reliably compare frequencies, counters are used to count the number of pulses from each oscillator over a specified time interval. An alternative approach is to connect the outputs of the two oscillators to the inputs of a SR-latch.

RO-PUF implementations demonstrate moderate complexity involving a circuit that consists of repeating blocks of oscillators and simple digital counters/comparators. Since each bit requires a unique RO pair, at least 2*N* oscillators are required to obtain *N* bits. More economical circuits are often used; for example, frequencies can be sorted within a set of ROs and a set of bits can be generated by comparing different pairs in an ordered list. Although this approach allows several bits to be obtained from *N* oscillators, it can complicate analysis. According to [3], the uniqueness of a well-designed RO-PUF is close to 50% due to random frequency spreads giving equal probability of one oscillator being superior to another. Repeatability can also be high; if the frequency difference between the selected pairs is large enough, the order of their comparison is preserved even when temperature and voltage change. Under experimental conditions, 95–99% reliability is achieved. However, under unfavorable conditions (for example, when frequencies converge due to temperature drift), some response bits may be inverted. Therefore, a frequency margin should be introduced—or response encoding should be used—to increase reliability.

In modern technical processes, the scaling of RO-PUF to higher frequencies requires consideration of increased period fluctuations due to errors comparable to the frequency difference between ring oscillators arising as a result of noise at the nanometer scale.

Studies have demonstrated successful RO-PUF implementation in field-programmable gate arrays (FPGAs) [20, 21]. Several proposed developments for improving RO-PUF characteristics are aimed at bringing them into the category of strong PUF [22]. Other works have described the architecture of a configurable RO-PUF (CRO-PUF) which uses frequency and phase shift changes [23, 24]. Each delay block, $S_0$, …, $S_n$, is made of logic elements formed by pairs of the *n*- and *p*-MOS (metal–oxide–semiconductor) transistors such that the total delay time increases in accordance with the tightening of technological tolerances for their manufacture. Figure 3a depicts an example in which each delay block S is configured to form a 4-bit challenge. Figure 3b shows *N* delay cascades connected in series where the output data from each cascade is used to switch the counter start signals. The values of these signals are then compared by a comparator to generate a response signal.

As well as eliminating the duplication of ring oscillators, the proposed design reduces switching activity and introduces inter-stage delay as an additional source of randomness. The PUF is implemented in 22 nm mode using fully depleted silicon-on-insulator technology and Synopsys[2] tools. Tests carried out on eight chips successfully passed NIST tests, achieving intra-HD and inter-HD values of 9.95% and 45.5%, respectively.

The CRO-PUF proposed in [25] is implemented as a modification of the basic circuit, which consists of XOR2 elements acting as controlled delay elements. A complete set of challenges can be applied to this circuit. It is shown that the delay depends not only on the challenge value but also on the configuration of the interconnections of the circuit structural elements with the configurable ring oscillator. A proposed temporal model of the modified CRO PUF allows the influence of interconnections on the frequency

[2] Synopsis Electronic Design Automation Solutions. https://www.synopsys.com/silicon-design.html. Accessed July 19, 2025.

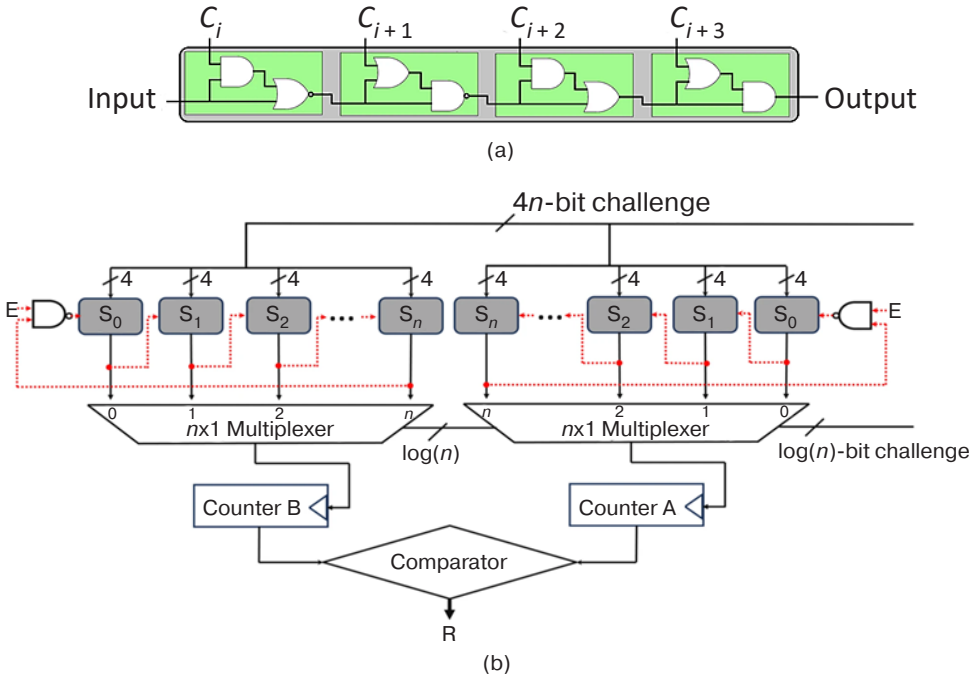


**Fig. 3.** RO-PUF microarchitecture:
(a) delay generation module; (b) response bit generation device for a configurable 4-bit challenge [24]. Here, $C_i$, ..., $C_{i+3}$ are the challenges; E is the enable signal; *n*x1 is the multiplexer with *n* inputs and 1 output; and R is the response

of the generated signal to be proven analytically as confirmed by experiment using the Xilinx Zynq-7000 series FPGA (Xilinx, USA).

The problem of compensating for the influence of temperature on PUF is discussed in [26, 27]. The authors analyze the impact of metal-oxide-semiconductor field-effect transistor temperature characteristics on RO-PUF properties without altering the original circuit structure. Simulations of the 55-nm process performed using Cadence Virtuoso[3] tools and the Monte Carlo method show that temperature changes having the least effect on generation frequency occur when using an *N*-type high voltage *n*MOS transistor in the generation block. In this case, the frequency of the ring oscillator changes by 7.83% over an operating temperature range of 50 to 200°C, which is less than the 14.35% change observed in a classic RO-PUF. When several ring oscillators are implemented in parallel and the frequency is measured by counting the rising edges, the measurement remains the same. However, when a challenge is applied to the PUF, an arbitrary pair of oscillators is selected to form a response as a logical function of the comparison of the two obtained counter values (Fig. 3).

The successful implementation of RO-PUF in FPGA is demonstrated in [28, 29] through experiments conducted on 15 programmable logic ICs (FPGAs) comprising 1024 circuits that yield values of $\mu_{inter} = 46.15\%$ and $\mu_{intra} = 0.48\%$. The authors apply a method for eliminating metastable states that takes into account only the most stable response bit from eight pairs of oscillator cycles. The source of variation is a random difference in signal propagation delay along nominally identical paths. In [29], the authors propose a pseudo linear feedback shift register (LFSR) PUF architecture in which the LFSR structure is implemented as a closed chain of inverters and XOR elements. This forms a single circuit that allows the reliable extraction of signal propagation delay variations that are unique to each chip.

In a transient effect ring oscillator (TERO) PUF, changes in the frequency and duration of signals in signal lines and logic elements are analyzed depending on the type of their manufacture [30]. As shown in Fig. 4, a transient effect ring oscillator consists of two series-connected bistable ring oscillator circuits.

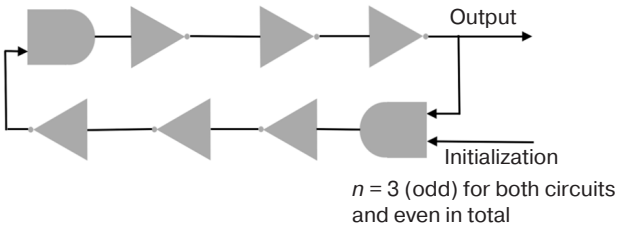


**Fig. 4.** Transient effect ring oscillator [30]

[3] https://cadence-ds.ru/virtuoso/. Accessed July 19, 2025.

The ring oscillator formed with an even number of inverters results in a transient effect. This causes the output signal of the cell to transition to a stable state, which is similar to the behavior of a bistable ring or a bus keeper memory cell. However, prior to this transition, certain temporary oscillations of the circuit must stabilize. After counting the number of oscillations that occur in each TERO cell before it transitions to a stable state, the values for several cells are combined to form a characteristic response for the TERO-PUF. In this case, the challenge is the number or location of the TERO cell (if there is more than one), while the response is the transient oscillations that occur when the system stops.

TERO cells should be designed with a symmetrical structure, requiring the careful selection of control elements and connection delays. Since the FPGA structure prevents developers from automatically selecting connections between elements, implementing such components in FPGAs is a specific task. However, the required symmetry can be achieved by setting constraints manually and using specific features of the target FPGA family. Study [30] describes the TERO-PUF design for two different FPGA technologies: 45 nm Xilinx Spartan 6[4] and 28 nm Altera Cyclone V[5]. Statistical processing of the TERO-PUF using these two target FPGAs produced uniqueness results of 48.46% and 47.62%, and stability results of 2.63% and 1.8%, respectively. These results are close to those obtained in several studies using the RO-PUF ring oscillator, which is considered the best candidate for implementing a PUF on FPGA. It should be noted that TERO-PUF is less susceptible to side-channel attacks than RO-PUF. Additionally, unlike RO-PUF, TERO-PUF can generate multiple bits (from one to three) for each challenge. The authors demonstrate that TERO-PUF provides between 0.85 and 1 bit of entropy per response bit. This study shows that TERO-PUF is a promising alternative to RO-PUF for implementing a PUF on FPGA.

Although the design of a bistable ring (BR) PUF [22] is similar to that of a ring PUF, it can maintain a stable state for a certain period of time. Like a ring oscillator-based PUF, it consists of a chain of NOT gates (or inverters); however, in this case an even number of gates form a bistable system instead of an oscillatory one (see Fig. 5).

When restarted, such a system transitions to one of several stable states. This occurs after a period of time determined by the unique technological variations in ring manufacture. There are many possible configurations of the ring, each of which strives independently for a preferred state. The preferred state is the response; the configuration—particularly that of the bistable ring—is determined by the PUF challenge (in this case, the reset signal).

As first demonstrated in the original work [22], in which the BR-PUF architecture is proposed, the number of topologically distinct rings is $2^n$. This enables this primitive to be categorized as a strong PUF, while the natural variation in technological parameters ensures good inter-chip uniqueness and a broadly distributed spectrum of ring convergence times. In FPGA experiments, the BR-PUF demonstrates near-ideal uniqueness (around 50%) and a reliability of approximately 97%. The long “tails” of the stabilization time distribution form a basis for the rejection of slow or unstable CRPs, thereby increasing repeatability. Conversely, it has been found that a single ring can be modelled using a machine learning algorithm trained on a million challenge-response pairs, which can predict the responses of 64-, 128-, and 256-stage instances with an error rate of less than 5%. Resilience can be increased by simply XOR-combining more than four independent rings in parallel [31].

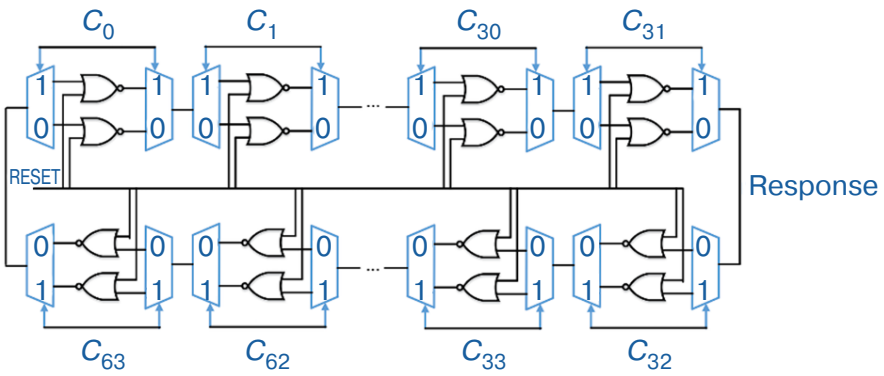


**Fig. 5.** Bistable ring PUF. $C_0$, $C_1$, $C_{30}$, $C_{31}$, $C_{63}$, $C_{62}$, $C_{33}$, $C_{32}$ are challenges

[4] Spartan 6 FPGAs. https://www.amd.com/en/products/adaptive-socs-and-fpgas/fpga/spartan-6.html. Accessed July 19, 2025.
[5] Cyclone V FPGA and SoC FPGA. https://www.altera.com/products/fpga/cyclone/v. Accessed July 19, 2025.

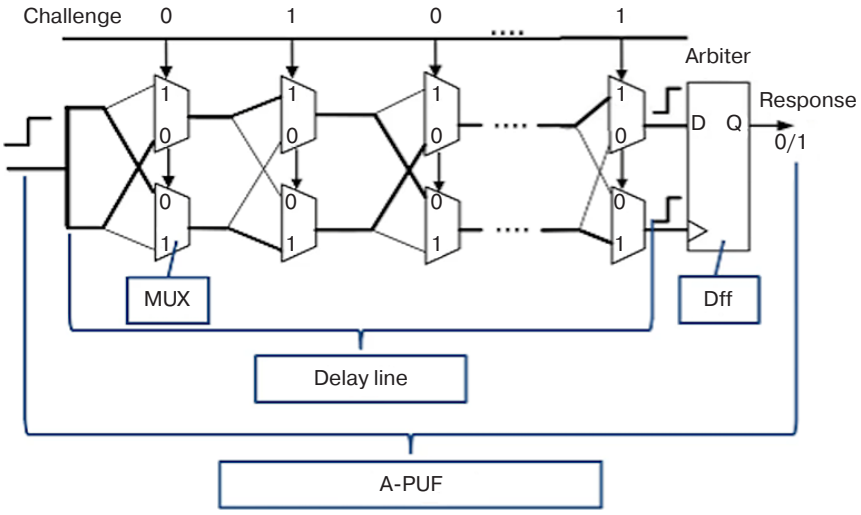


**Fig. 6.** Example of an arbiter PUF implementation. D is input; Q is output; Dff is a D flip-flop

The chaotic-BR-PUF hybrid scheme further strengthens this architecture by obfuscating the output of the basic BR ring through a nonlinear logistic mapping. This reduces the effectiveness of attacks to the level of random guessing (50–60%), while using comparable FPGA resources [32].

As evidenced by the use of ternary BR-PUF on carbon nanotube field-effect transistors (CNTFETs), which generates ternary responses, the current trend is towards multi-valued logic. This expands the CRP space to increase entropy without significantly raising hardware costs. Modeling on a 32-nm library of standard elements shows that it is more unpredictable and resistant to machine learning-based attacks than the binary prototype. Furthermore, the high-temperature stability of CNTFET transistors makes this approach particularly promising for IoT devices [33].

## Arbiter PUFs (A-PUFs)

This type of PUF operates by comparing the transit times of two signals propagating along theoretically symmetrical paths. As shown in Fig. 6, the A-PUF module consists of several cells that connect the signal source to the arbiter component.

Each cell contains a switch that can route signals through other signal lines. The arbiter component outputs a binary signal whose value depends on which of the two input signals, which are separated from the signal source, reaches the component first. Due to random variations in the conductors and switching elements through which the signals pass, their speeds will vary relative to each other. Therefore, the challenge is based on the on/off switching nature of the routing switches (and the multiplicity of the number/position of the arbiter in these systems), while the response depends on the faster path after switching.

If the setup/hold time is violated, the arbitrator may enter a metastable state, which can affect its operation. This state is not determined by the comparison of signal propagation times, but rather by random noise in the responses (see [8] for more details).

This type of PUF has the advantage of being simple to implement and having a small footprint, with one $n$-stage A-PUF consisting of $2n$ multiplexers and one arbiter implemented as a D flip-flop. With $n = 64$, the circuit occupies only a few hundred gates. Since the spread of delays sufficient to generate random differences is preserved as process technology norms decrease, the corresponding device can be easily scaled up on silicon. Practical implementations of A-PUFs, particularly in more complex design modifications, demonstrate a level of uniqueness close to the theoretically optimal value of 50%. An analysis of publications on A-PUFs reveals a variety of implementation methods. In addition to the basic A-PUF structure, the review [3] provides brief descriptions of modifications to this architecture, including two-channel and multi-channel variants with XOR elements, multiplexer-based circuits and combinations with RO-PUF. However, the vulnerability of conventional A-PUF functions to machine learning-based attacks significantly limits their applicability in secure environments with limited resources.

The architecture of the arbiter-type improved PUF is presented in [34–37]. Figure 7 [28] shows an example of an implementation where several independent modules

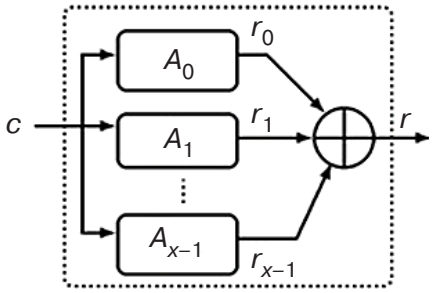


**Fig. 7.** A-PUF modification with response combination using the XOR function.
Here, $c$ is challenge, $A_0$, ..., $A_{x-1}$ represent A-PUF instances; $r_0$, ..., $r_{x-1}$ are responses from A-PUF instances; and $r$ is the XOR response from the A-PUF

are combined by the XOR function to form a single response.

Another architecture[6] involves a feed-forward memory-based arbiter (FF-MB-A) PUF module which combines weak PUF modules based on volatile memory with nonlinear feedback logic in order to increase entropy and enhance resistance to simulation attacks. A comprehensive experimental system was developed to evaluate the proposed solution, using up to 50 mln call-response pairs (CRPs). The results showed that the number and exact location of feedback cycles critically affect simulation resistance. To enable comparison with similar implementations, this work implemented and tuned the most advanced modeling strategies, including deep neural networks and the covariance matrix adaptation evolution strategy. The optimized FF-MB-A-PUF configuration, which incorporated 63 feedback cycles, exhibited robust resistance to simulation-based attacks, enhanced randomness (49.23%), and improved uniqueness between devices (49.20%), leading to balanced output distribution and high entropy. These results suggest that the FF-MB-A-PUF is a scalable, hardware-efficient, and secure primitive ideally suited to next-generation embedded systems and low-power IoT systems.

Typical methods for stabilizing A-PUF against noise, aging, and voltage and temperature fluctuations are described in [37], in particular averaging and masking unstable bits.

### PUFs based on clock signal delays (clock PUF)

In synchronous circuits, clock PUF [38] analyzes changes in the speed of signal propagation from the clock generator to various parts of the circuit itself on the basis of differences in manufacturing during its physical implementation. While modern IC designs aim to eliminate these parasitic phase shift effects, variations and distortions still occur. In this PUF variant, the delays of paired signals in presumably similar circuits are compared to uniquely characterize the circuit in a manner similar to an A-PUF. In this case, the challenge is the clock signal lines, and the response is the delay in each corresponding line.

In [39], a lightweight, symmetric version of the PUF based on a three-phase D flip-flop with increased uniqueness is proposed and implemented in an IC using standard 40-nm complementary MOS (CMOS) technology. Following the chip layout, simulation results predict the device's uniqueness to be 0.4994, the highest among all the considered architectures. Compared to an A-PUF, this device consumes 73.3% less power, occupies 93.6% less area, and uses 95.7% less energy per bit. The corresponding figures for RO-PUF are 98.3%, 96.9%, and 99.9%, respectively. Additionally, unlike other flip-flop-based PUFs, the proposed variant does not require a post-processing block to eliminate bias voltage, thus contributing to savings in the overall implementation area and system power. To demonstrate the concept, the device was implemented on a FPGA. As a means of comparing the performance of the considered architectures, a proposed figure of merit (FOM) considers power, reliability, delay, silicon area, and uniqueness. It should be noted that the proposed architecture provides the highest FOM of all the considered PUF architectures.

In [38], new PUF technologies are presented that extract bits from pairwise distortions between IC clock network domains. An algorithm was implemented to select equidistant receivers, route the reverse network and

[6] Mishra A. *Enhancing the security scalability of Arbiter PUFs using memory-based weak PUFs*. Thesis. West Lafayette (IN): Purdue University; 2025. https://hammer.purdue.edu/articles/thesis/enhancing_the_security_scalability_of_arbiter_pufs_using_memory-based_weak_pufs/28899152. Accessed July 19, 2025.

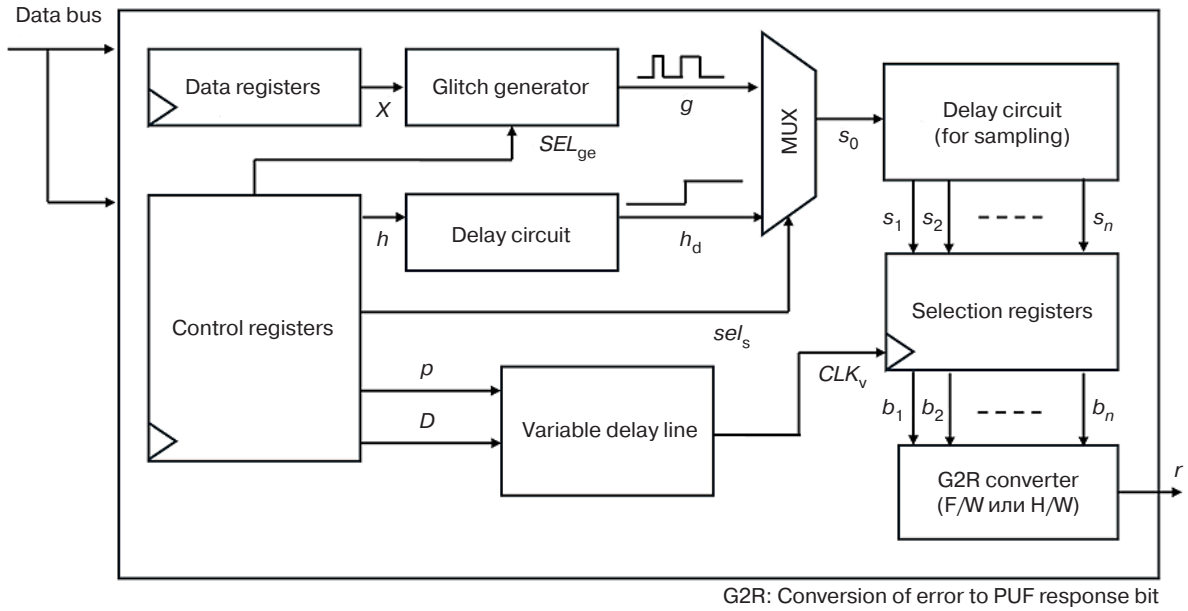


**Fig. 8.** Example of a device with glitch PUF [41].
Here, $X$ is input data fed to the glitch generator; $SEL_{ge}$ is the glitch generator output selection signal; $g$ is the selected glitch generator output bit; $s_0$ is the initial Glitch-signal; $s_1, ..., s_n$ are signals after the delay circuit (sampling points); $h$ is the calibration pulse; $h_d$ is the calibration pulse after the delay circuit; $sel_s$ is the input selection signal; $p$ is the trigger signal; $D$ is the time delay control code; $CLK_v$ is the clock signal after variable delay; $b_1, ..., b_n$ are values read by the selection registers and representing the form of glitches; G2R is Glitch-to-Response (glitch-to-response conversion device); F/W is firmware; H/W is hardware; and $r$ is response

then extract random bits for a specific chip. Evaluation of clock pulses based on a SPICE[7] simulation of 45 nm CMOS technology confirms the feasibility, stability, uniqueness, randomness, and low overhead of this implementation.

The analysis of clock signal phase shifting is also considered as an option in [40]. The proposed software-based PUF (S-PUF) option causes the video encoding circuit to malfunction using a clock signal. The response key with circuit characteristics is generated using the dependence of the response on the synchronization path. The video encoding circuit, which forms part of the IP core[8] of an open-source video encoding microcircuit, serves as the carrier circuit for the PUF. Based on an analysis of the timing path of the encoding circuit, a trigger signal is selected to place the circuit into abnormal operation mode. Random data is generated and subjected to video encoding and compression operations, which are then masked using Gray code and false bit exclusion operations. Test results show that this implementation of the proposed S-PUF variant passes the NIST test with 48.87% uniqueness and an autocorrelation coefficient of 0.0204 at 95% confidence.

## PUFs based on transient processes (glitch PUFs)

These PUFs are more complex than RO-PUF and A-PUF schemes due to their analyzing the transient characteristics of signals that cause device malfunctions. In this PUF concept, the circuit itself is the challenge, while the response is the specific implementation of the glitches that occur and how they evolve over time (see the example in Fig. 8 [41, 42]).

In the glitch PUF architecture described in [41], glitches arising from changes in the delay between transistor gates in the circuit, which affect the pulse propagation characteristics from each gate, are used to form the PUF. The paper describes a procedure for simulating a simple implementation of such a circuit at the design stage. The results coincide well with the data obtained during the hardware implementation of such a PUF in real microcircuits.

As noted in [43], susceptibility to noise is an inherent issue in the hardware implementation of glitch PUF. To address this, the paper proposes a fault control module with a multi-level, parallel architecture

[7] Simulation program with IC emphasis is an open-source simulator of general-purpose electronic circuits.

[8] IP cores (intellectual property) are ready-made blocks for designing microchips.

for generating multi-bit stable information entropy. A noise reduction circuit has also been developed which uses a hysteresis effect, a feedback mechanism with Schmitt triggers and a pulse width detection circuit to filter out noise and extract output data from glitch signals during transient processes. A 128-bit glitch PUF circuit has been implemented using the TSMC[9] 65 nm CMOS technology. Experimental results demonstrate that the randomness (intra-distance = = 0.01) is 99.9%, while the uniqueness (inter-distance) is 50.03%. These results suggest that the proposed design could be widely adopted to enhance the security of IoT devices.

As stated in [44], the performance of such PUFs varies slightly due to environmental changes, necessitating error correction methods to eliminate these variations. One option for such a method is proposed. To demonstrate the effectiveness of this method quantitatively, evaluation experiments are conducted using an FPGA.

## IMPLEMENTATION OF PUF BASED ON STATIC RANDOM-ACCESS MEMORY (SRAM)

### SRAM-based PUFs

The implementation of PUFs based on static SRAM memory [45, 46] is rooted in the random distribution of the 6T memory cell states (Fig. 9), which determine their behavior when activated. This distribution is directly related to the manufacturing conditions and tolerances of the technological processes involved in forming the cells. The random initial state of the SRAM cells, which acts as a "fingerprint" of the chip, can be used either directly as a key or as a basis for generating responses to challenges. Therefore, compared to a standard SRAM array, only a controlled reset or power-up of the memory is required, and no additional energy is consumed in idle mode.

Studies have demonstrated that SRAM-PUFs have the capacity to provide near-perfect uniqueness. However, their reliability, which is limited by noise and environmental influences, is approximately 88–90% without correction [16]. To enhance the reliability of the output, averaging circuits (multiple reads at power-up) and error correction algorithms (e.g., Reed–Solomon code or fuzzy extractor) are employed [17].

Scaling of SRAM-PUF can be achieved as demonstrated by implementations in processes down to 7 nm, which show continued operability [18, 47]. However, reducing the size of transistor leads to a decrease in the absolute values of mismatches. This may necessitate more sophisticated bit processing (e.g., discarding unstable cells) to ensure reliability. Data from SRAM-based PUF experiments involving the inclusion of 8190 bytes of SRAM from various memory blocks on different FPGAs show that the average difference between two different blocks under fixed environmental conditions is $\mu_{inter}$ = 49.97%, while the average difference between multiple measurements in a single block is $\mu_{intra}$ = 3.57%. However, $\mu_{intra}$ increases to 12% for large temperature variations. The authors estimate the entropy content of SRAM turn-on states is to be 0.76 bits per SRAM cell.

Similar results are presented in [46, 48], which examine the behavior of SRAM when enabled on two different platforms. For 5120 blocks of 64 SRAM cells

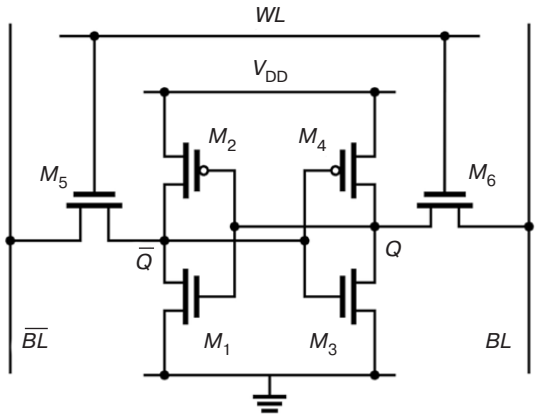


**Fig. 9.** 6T SRAM cell.
Here, *WL* is the word line, which controls two access transistors; $V_{DD}$ is the power supply; $M_1$, ..., $M_6$ are MOS transistors; $Q$, $\overline{Q}$ are complementary data storage nodes; and $BL, \overline{BL}$ are complementary bit lines, which are used for writing and reading data

[9] The Taiwan Semiconductor Manufacturing Company (TSMC) is a manufacturer of ICs and semiconductor wafers.

measured on eight commercial SRAM chips, $\mu_{inter}$ and $\mu_{intra}$ values of 43.16% and 3.8%, respectively, are obtained. For 15 blocks of 64 SRAM cells from the built-in memory of three microcontroller chips, $\mu_{inter}$ and $\mu_{intra}$ values of 49.34% and 6.5%, respectively, are obtained.

One disadvantage of implementing a PUF on a FPGA in this way is due to the fact that, in the most common type of FPGA, all SRAM cells are reset to zero immediately after powering up, resulting in a loss of randomness. To address this issue, butterfly-type circuits and latch circuits have been proposed for use with PUFs. The butterfly circuit proposed in [49] consists of two cross-linked latches with a clock signal set to one, thereby simulating a combinational bistable element (Fig. 10).

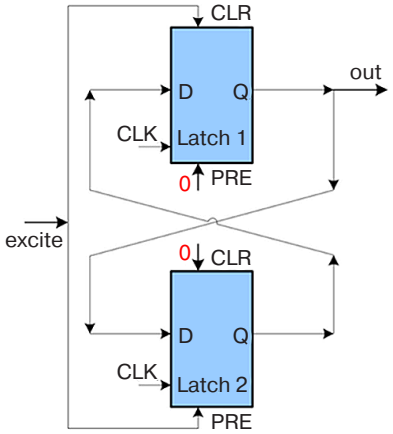


**Fig. 10.** Butterfly-type PUF [49].
Here, Latch 1/2 is a latch; excite is an initialization signal; CLR (clear) is an asynchronous latch reset input; PRE (preset) is an asynchronous latch preset input; CLK (clock) is a clock signal input; D is a latch data input; Q is a latch output; and out is a circuit output

The excite signal produces different levels (0/1) at the outputs of both elements, placing the cell in a metastable state. When the signal is removed, the circuit transitions to a random stable state. The specific value of this state is determined by the physical implementation of the latches and cross-connections forming the response bit. The measurement results presented in [50] are obtained using 64 circuits with a butterfly-type PUF on 36 PLDs. The values $\mu_{inter}$ = 50% and $\mu_{intra}$ < 5% are obtained at significant temperature fluctuations. As with other PUFs, error correction methods are employed in this type of PUF.

Another option for identifying the IC is proposed in [51]. Here, rather than cross-connecting two inverters or two latches, two NOR gates are cross-connected to form a NOR latch. When a reset signal is applied, the latch enters an unstable state and returns to one of two stable states depending on the mismatch between the internal electronic components. Tests involving 128 NOR latches are carried out on 19 ultra-large ICs produced using CMOS technology with 0.130 μm technology, resulting in values of $\mu_{inter}$ = 50.55% and $\mu_{intra}$ = 3.04%.

## PUFs based on dynamic random-access memory (DRAM)

DRAM-PUF, another type of PUF based on volatile memory, is present in almost all modern computing devices, including resource-constrained embedded systems and IoT platforms. This makes it an attractive basis for embedded hardware security primitives. The large number of cells provides a significant pool of entropy and a potentially wide range of challenge-response pairs; moreover, DRAM can be accessed both during boot-up and while the system is running. These properties are particularly important in cases where classic discrete security modules are impossible or expensive to use. Additionally, DRAM typically consumes less power than SRAM for comparable capacity.

A typical 1T1C[10] DRAM cell comprises a transistor and a capacitor (see Fig. 11), and requires periodic refreshing due to charge leakage.[11]

The unique signature of a cell is created by differences that occur during manufacturing due to variations in capacitance, leakage currents, and threshold voltages. Leakage can occur within a row and between adjacent rows/lines. Active interaction between neighboring rows enhances charge flow, leading to bit inversions that increase entropy. The phenomenon of variable retention time is also observed, whereby the same cell switching unpredictably between states with high and low retention capacity. Finally, shortened read/write delays (e.g., tRCD[12] and tRP[13]) can cause cells to fail to settle correctly, forming a characteristic error pattern. These mechanisms of varying stability enable the creation of both reproducible PUFs (where unstable bits are discarded and/or errors are corrected) and high-entropy random number generators (TRNGs), which operate in modes where the result is as unstable as possible.

The DRAM-PUF studies are traditionally grouped according to the physical effect used, with start-up values when power is turned on, retention/decay when refresh is stopped or power is turned off (start-up DRAM-PUF,

[10] One-transistor, one-capacitor is a RAM cell consisting of one field-effect transistor and one capacitor.

[11] DRAM Scaling Challenges Grow. https://semiengineering.com/dram-scaling-challenges-grow/. Accessed July 19, 2025.

[12] Time row address to column address delay is row activation delay.

[13] Row precharge time is row refresh time.

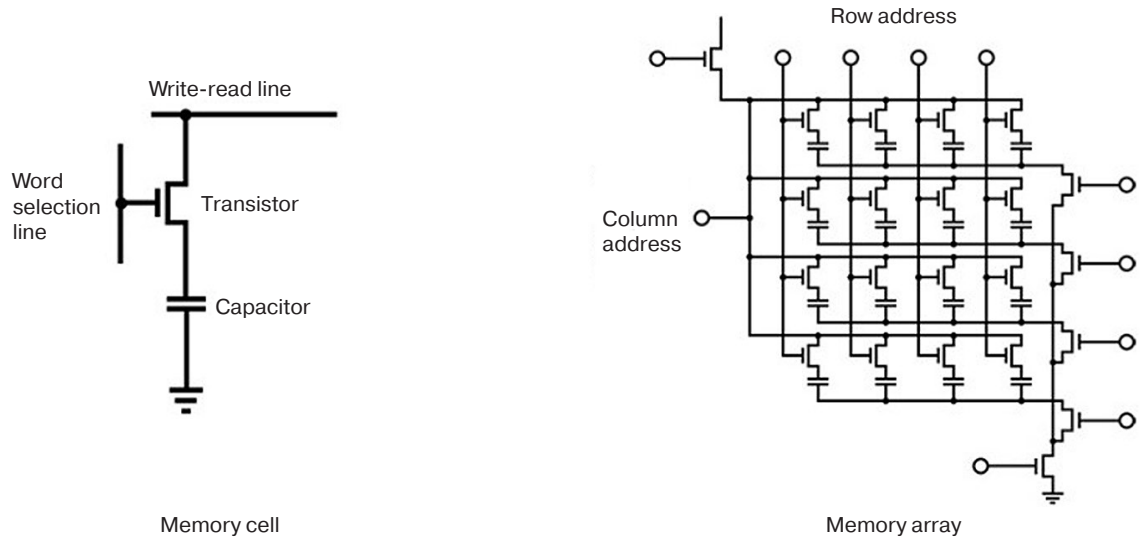


**Fig. 11.** DRAM memory and 1T1C cell

retention-based DRAM-PUF), and timing violations (latency-based DRAM-PUF). Solutions involving more specific effects are also being studied, including Rowhammer PUF and combined solutions.

The operation of the start-up DRAM-PUF is based on the initial distribution of cell charges after power is applied. In [52], the PUF's stability is examined in terms of temperature (0–80°C), power supply (4.5–5.5 V), and accelerated aging. The PUF, which is shown to be capable of forming 2048-bit responses (16 × 128-bit keys), features increased reliability through an algorithm for selecting cells with stable neighbors. Subsequent analysis shows that reducing the temperature and the supply voltage has a greater impact on stability than increasing them; the aging effect (negative bias temperature instability, NBTI) is moderate

Similar works have utilized the entire DRAM array as an "image" (converted to grayscale), with subsequent feature extraction by an artificial neural network for identification. A classification accuracy of ~98.8% was achieved, albeit in the absence of a detailed evaluation of classical PUF stability metrics [53]. The paper also discusses an attempt to improve randomness by post-processing LFSR.

Retention-based DRAM-PUF uses variations in charge retention time when auto-refresh (or power) is disabled and the array is read at specified intervals. The resulting pattern of bit inversions is unique to the memory segment. In [54], the possibility of reducing PUF waiting time to 20–60 s is demonstrated if the response is formed as a map of bit inversion locations. Uniqueness is demonstrated not only at the microcircuit level, but also between memory blocks within a single DRAM module. Reliability is maintained when the temperature changes within the range of 20–40°C ± 10°C; the effect of aging at 85°C for 48 h is additionally studied.

In latency-based DRAM-PUF, entropy is extracted from differences in cell access speeds. Implementing a PUF involves the controller deliberately underestimating the timings (tRCD, tRP, etc.), causing individual cells to produce read/write errors in characteristic positions. This architecture greatly speeds up the operation of the PUF, with a reported time of ~0.875 ms, which is more than 10000 times faster than typical retention DRAM-PUFs (approximately 10 s). The implementation example [54] uses tRP/tRCD variation to write a known pattern, reading with disrupted timings, and constructing entropy maps. Subsequently, XNOR filtering is employed to eliminate unstable bits.

The Rowhammer effect involves repeatedly accessing specific DRAM rows over a short period of time, causing accelerated charge leakage in neighboring rows and deterministic bit inversions. In [28], this phenomenon was first used to implement a PUF. The Rowhammer PUF method involves selecting two PUF memory rows from the DRAM array that are pre-filled with opposite patterns: 0x55 and 0xAA. At a certain point, the refresh of these rows is disabled, and they begin to discharge rapidly under the influence of two actively working (activation + precharge) neighboring Rowhammer rows to form a unique pattern.

Combined solutions, such as the combination of SRAM-PUF and DRAM-PUF, allow the advantages of different memory types to be combined and the disadvantages of each to be partially compensated. Implementations with high entropy, numerous CRPs,

and stable authentication over a wide temperature range (20–60°C) have been demonstrated.

A comparison of DRAM-PUF types reveals that, while most studies achieve an inter-HD value close to the ideal 50%, reliability varies depending on the class. Retention-based DRAM-PUFs are more sensitive to temperature and voltage, while latency-based DRAM-PUFs are usually more reproducible when the timing thresholds are selected correctly, and start-up DRAM-PUFs occupy an intermediate position, requiring algorithms to select stable cells [55].

Although a wide CRP space can be provided in principle by a large amount of DRAM, the number of stable bits suitable for use is sharply reduced in practice after reliability and randomness filtering. In application systems, most DRAM PUFs are therefore classified as weak PUFs (with a limited number of stable CRPs), which are primarily suitable for key generation and episodic authentication.

## NVM-PUFs

NVM-PUFs use random variations in memory cell characteristics to create unique fingerprints of microcircuits. The key physical effects determining the uniqueness and stability of such PUFs are variations in materials and leakage currents, as well as stochastic switching processes.

In resistive random-access memory (ReRAM) PUF [56], the randomness comes from the stochastic process of formation of conductive sections in a dielectric; the switching thresholds and resistance of the high/low state vary slightly from cell to cell due to technological tolerances. In other ReRAM-PUF designs, the resistance spread in a pre-formed array is measured without overwriting it [57].

Similarly, the unique differences in magnetoresistive random-access memory (MRAM) are due to variations in the resistance of magnetic tunnel junctions (MTJs) that occur during production [58]. In MRAM, variability is influenced by the thickness of the barrier and the magnetic anisotropy of the MTJ; small variations result in differences in cell resistance within the same state. The reproducibility of the response is improved due to the significant stability of these technological variations over time.

In phase-change memory (PCM), in which each cell has a slightly different level of conductivity due to fluctuations in the structure during manufacturing or after switching to an amorphous state, the basis is taken from the differences in the material's states. This makes it possible to read the memory state as a random pattern [59]. To ensure reproducibility, the resistance drift of the amorphous state over time must be considered. Studies have shown that selection of the appropriate reading mode (e.g., differential) and calibration can minimize the impact of drift and generate stable bits [60].

The physical processes that generate randomness, such as the random formation of conducting channels, variations in tunnel resistance, and the dispersion of phase states, generate high entropy. Engineering solutions aim to minimize the instability of these effects to ensure reproducibility. Thus, ReRAM-, MRAM-, and PCM-PUF form the response bit either by directly reading the uncontrolled initial state of the cells or by using a special read/write mode that highlights the spread of cell characteristics. Recently, many PUF architectures based on these types of NVM have been proposed to improve their characteristics. The basic scheme is a weak PUF based on a memory array, where the response is formed from the state of numerous cells (for example, an *N*-bit starting vector is read from uninitialized memory). Although this approach is simple and economical, it provides a limited number of challenge-response pairs (a fixed fingerprint) [61].

The challenge-response space is expanded by using modified architectures to transfer them to the strong PUF class. ReRAM, for example, uses 2T2R and 1T4R cells to enable different methods of reading and comparing resistances. This results in a vast number of possible responses [59]. The paper proposes an approach in which the ReRAM array operates in the in-memory calculation mode. This is achieved by combining several cells *via* XOR or by reading using a special template to create an output that is insensitive to modeling and resistant to machine learning attacks.

In terms of circuit design, emphasis is also placed on the suppression of side channels. In particular, symmetrical structures (such as mirror-switched differential pairs of cells) eliminate systematic bias to increase the difficulty for an attacker to predict the outcome [57]. Modern implementations also include a self-destruct mode, whereby an increased voltage is provided to the ReRAM-PUF, irreversibly destroying part of the cells if an attacker attempts to access them, thus destroying the key [56].

NVM-PUFs can be found in a range of architectures, from compact embedded memory cells that produce a single hardcoded key to large-scale tunable arrays capable of generating multiple responses and resisting simulation attacks. NVM-PUFs are capable of providing near-ideal performance. In experiments with ReRAM, MRAM and PCM, uniqueness is usually around 50%, while the difference between repeated reads is less than 1–2%. Several papers report bit error rates of 0.01–0.1% without correction. For example, values of ~49.96% and ~0.98%, respectively, are obtained

for inter-HD and intra-HD in an MRAM-PUF based on an MTJ array. The entropy of the generated responses approaches the maximum value; the 0/1 distribution is typically around 50% (uniformity ~50%), as confirmed by NIST statistical randomness tests for PUF bit sequences. It is noted in reviews that, after unstable bits have been rejected, information saturation of approximately 1 bit of entropy per cell is provided by modern ReRAM-PUFs.

The reliability of NVM-PUFs is characterized by their ability to maintain a stable response to various disturbances, such as temperature changes, supply voltage, and component aging. Since the state of NVM is physically fixed (e.g., conducting regions in ReRAM, magnetic vectors in MRAM, and phase structures in PCM), such PUFs are less sensitive to external influences. For example, an MRAM-PUF demonstrates stability from −40 to 150°C [59]. ReRAM-PUF shows only a slight increase in resistance spread during temperature cycling (for example, from 25 to 125°C). It has been experimentally confirmed that >91% of bits remain error-free at 125°C [56]. The issue of aging has also been investigated. While slow degradation of the oxide conductive sections or tunnel barriers during the device's service life can increase bit errors, built-in threshold margins and correction algorithms can ensure long-term stability.

Although post-processing in memory-based schemes can reduce bias and improve statistical performance, it adds overhead and introduces potential vulnerabilities if applied incorrectly [55].

## SUMMARY RESULTS FOR THE REVIEWED PUFs

The following table summarizes the characteristics of some of the key PUF options discussed in this series of reviewed papers. It reflects studies that produced original results following the implementation of one or more PUFs, where inter-distance and intra-distance were selected as key metrics. In a number of publications, these are referred to as uniqueness and reliability, respectively.

Sensitivity to external conditions indicates the voltage/temperature variations at which the characteristics are measured and how much they change. If such data is provided, the intra-distance changes are indicated in parentheses.

The "Estimated implementation complexity" column (high, medium, or low) is intended to demonstrate the relative hardware costs of implementing a particular type of a PUF, as well as the technical complexity involved, such as balancing paths, selecting element parameters, and changing technical processes.

**Table.** Summary results for the reviewed PUFs

| PUF type/operation | Publication year | PUF characteristics | | | | | |
|---|---|---|---|---|---|---|---|
| | | Inter-distance | Intra-distance | Platform | Inter-distance | | Estimated implementation complexity |
| | | | | | Temperature | Voltage | |
| RO [7] | 2007 | 46.15% | 0.48% | 15 × Xilinx Virtex-4 LX25 FPGA | −20–120°C | 1.2 V ± 10% | medium |
| Arbiter [7] | 2007 | 23% | 0.7% | ASIC TSMC 180-nm | 20–70°C (+4.8%), 20–120°C (+9%) | ±2% (+3.7%), ±33% (+9%) | medium |
| RO [10] | 2009 | 47.8% | ~0% | SPICE model 90-nm CMOS | −15–65°C | 0.2–1.0 V (+7% at 0.5 V) | medium |
| SRAM PUF (FinFET[14] 16-nm) [17] | 2022 | – | 14% | NVIDIA Jetson, 16-nm LP FinFET | 0–85°C | – | low |
| SRAM PUF (FinFET 14-nm) [17] | 2022 | – | 10% | NXP LPC, 14-nm LP FinFET | 0–85°C | – | low |
| SRAM PUF (FinFET 7-nm) [17] | 2022 | – | 11% | Xilinx Versal, 7-nm HP FinFET | 0–85°C | – | low |

[14] Fin field-effect transistor is a 3D-structured transistor.

**Table.** Continued

| PUF type/operation | Publication year | PUF characteristics | | | | | |
|---|---|---|---|---|---|---|---|
| | | Inter-distance | Intra-distance | Platform | Inter-distance | | Estimated implementation complexity |
| | | | | | Temperature | Voltage | |
| TERO [19] | 2014 | 48.07% | 1.73% | FPGA (Altera Cyclone II) | – | – | medium |
| BR [22] | 2011 | 49.8% | 0.7% | FPGA (Xilinx Virtex-II Pro) | 5–45°C (+2.7%) | ±10% (+2.2%) | medium |
| CRO[23] | 2024 | 45.5% | 9.95% | ASIC 22-nm FDSOI | −40–70°C | 0.72–0.88 V | medium |
| RO [26] | 2025 | – | 0.38% | SPICE-model 55-nm CMOS | −50–200°C (+9.38%) | – | medium |
| Rowhammer [27] | 2017 | – | <5% | DDR3/4 DRAM | −40–60°C | – | low |
| XOR Arbiter [28] | 2022 | 48.69% | 0.59% | FPGA (Xilinx Artix 7) | 0–85°C | 0.95–1.05 V | medium |
| Pseudo-LFSR [29] | 2011 | 65.6% | 1.8% | FPGA (Xilinx Virtex 5) | – | 1 V | medium |
| TERO [30] | 2016 | 47.22% | 2.36% | FPGA (Xilinx Spartan 6) | −15–65°C | 1.1–1.3 V | medium |
| TERO [30] | 2016 | 48.58% | 2.66% | FPGA (Altera Cyclone V) | −15–65°C | 1.05–1.15 V | medium |
| XOR BR [31] | 2015 | 14.8% | 0.8% | FPGA (Xilinx Spartan 6) | 27–75°C | – | medium |
| BR [32] | 2021 | 48.0% | – | FPGA (Xilinx Artix 7) | – | – | medium |
| BR [33] | 2024 | 66.26% | 1.58% | SPICE-model CNTFET 32-nm CMOS | 0–100°C | 0.8–1.0 V | medium |
| Eye-Opening Arbiter[15] [37] | 2025 | 44.99% | 3.49% | FPGA (Xilinx Zynq-7010) | −40–125°C | 0.81–0.99 V | medium |
| Clock [38] | 2013 | 50.11% | 1.19% | SPICE-model 45-nm CMOS | – | – | medium |
| Tri-state Flip-Flop[16] [39] | 2020 | ~49% | ~2% | FPGA (Altera Cyclone) | – | – | low |
| Overclocking clock software[17] [40] | 2025 | 48.87% | – | SPICE model TSMC 65-nm CMOS | −25–125°C | 1.0–1.4 V | low |
| Glitch [41] | 2010 | ~32% | 1.3% | FPGA Xilinx XC3S400A–4FTG256C (16 boards) | 100°C (+5.3%) | – | medium |
| Glitch PUF with a Schmitt Trigger [43] | 2021 | 50.03% | – | ASIC TSMC 65-nm CMOS | −25–125°C | 0.8–1.4 V | medium |
| SRAM [45] | 2007 | 49.97% | 3.57% | FPGA | −20–80°C | – | low |

[15] Arbiter PUF with a phase window.
[16] PUF based on high-impedance flip-flops.
[17] S-PUF based on overclocking.

**Table.** Continued

| PUF type/operation | Publication year | PUF characteristics | | | | | |
|---|---|---|---|---|---|---|---|
| | | Inter-distance | Intra-distance | Platform | Inter-distance | | Estimated implementation complexity |
| | | | | | Temperature | Voltage | |
| Butterfly [49] | 2007 | ~50% | ~10% | FPGA (Xilinx Virtex-5) | −20–80°C | – | medium |
| MRAM [58] | 2024 | 49.96% | 0.98% | MRAM (STT[18]-MRAM array) | −25–100°C | 0.65–0.85 V | medium |
| 3D ReRAM [60] | 2022 | 49.4% | 0.014% | ReRAM (8-layer 3D array) | 0–80°C (+1.93%) | 1.65–2.2 V (+1.93%) | high |
| Dual-Mode[19] ReRAM [61] | 2025 | ~50% | ~1% | ReRAM (1T1R cells + logic) | – | – | high |

## CONCLUSIONS

A class of devices known as PUFs is based on the time characteristics of signals. These devices use frequency, phase, and transient analysis to generate responses due to manufacturing variations. Such solutions typically necessitate a carefully balanced combination of layout, calibration, and post-processing modules to ensure a consistent response. The advantage of these functions is that they offer virtually unlimited space for challenge-response pairs, making them ideal for authentication protocols. However, they are vulnerable to simulation-based attacks.

Memory-based PUFs create a "fingerprint" of the initial states and/or characteristics of SRAM/DRAM/NVM arrays. These functions can be easily integrated into existing on-chip blocks and are highly reproducible at moderate overhead costs. However, the space of the challenge-response pairs is typically limited, restricting their use to tasks such as key generation and device identification.

Stabilization measures for operating conditions (temperature, supply voltage, and aging) and the use of auxiliary data, error correction, and/or analysis modules are important for both classes. The selection of a particular PUF should be predicated on the requirements of the application and the target platform.

## ACKNOWLEDGMENTS

This work was supported by the Ministry of Science and Higher Education of the Russian Federation (State task for universities No. FSFZ-2026-0003) and using the equipment of the Center for Collective Use of RTU MIREA (agreement dated September 01, 2021, No. 075-15-2021-689, unique identification number 2296.61321X0010).

### Authors' contributions

**E.Ph. Pevtsov**—study conceptualization, review outline and structure, manuscript writing.

**T.A. Demenkova**—study conceptualization, review outline and structure, synthesis of the results.

**Yu.A. Korotaev**—literature analysis and systematization, manuscript writing, synthesis of the results.

**A.S. Sigov**—scientific consulting, scientific editing, final approval of the manuscript.

All authors have read and approved the published version of the manuscript.

[18] Spin-torque-transfer.
[19] Dual-channel RAM mode.

## REFERENCES

1. Khalil K., Idris H., Idriss T., Bayoumi M. *Lightweight Hardware Security and Physically Unclonable Functions: Improving Security of Constrained IoT Devices*. Cham: Springer Nature Switzerland; 2025, 152 p.
2. McGrath T., Bagci I.E., Wang Z.M., Roedig U., Young R.J. A PUF taxonomy. *Appl. Phys. Rev.* 2019;6(1):011303. https://doi.org/10.1063/1.5079407
3. Zerrouki F., Ouchani S., Bouarfa H. A survey on silicon PUFs. *J. Syst. Archit.* 2022;127:102514. https://doi.org/10.1016/j.sysarc.2022.102514
4. Alhamarneh R.A., Mahinderjit Singh M. Strengthening Internet of Things Security: Surveying Physical Unclonable Functions for Authentication, Communication Protocols, Challenges, and Applications. *Appl. Sci.* 2024;14(5):1700. https://doi.org/10.3390/app14051700
5. Tehranipoor M., Pundir N., Vashistha N., Farahmandi F. *Hardware Security Primitives*. Cham: Springer; 2023, 350 p.
6. Maes R., Verbauwhede I. Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions. In: Sadeghi A.-R., Naccache D. (Eds.). *Towards Hardware-Intrinsic Security: Foundations and Practice*. Berlin: Springer; 2010. P. 3–37. https://doi.org/10.1007/978-3-642-14452-3_1
7. Suh G.E., Devadas S. Physical Unclonable Functions for Device Authentication and Secret Key Generation. In: *Proceedings of the 44th ACM/IEEE Design Automation Conference* (*DAC 2007*), San Diego, CA, USA, June 4–8, 2007. New York: ACM; 2007. P. 9–14. https://doi.org/10.1145/1278480.1278484
8. Lebedev V.R., Pevtsov E.F., Demenkova T.A., Maleto M.I., Filimonov V.V. Method for studying the implementation of Physical Unclonable Function in information systems. *Int. J. Open Inf. Technol.* 2024;12(1):28–36 (in Russ.). Available from URL: http://injoit.org/index.php/j1/article/view/1712. Accessed July 10, 2025.
9. Gassend B., Clarke D., van Dijk M., Devadas S. Silicon physical random functions. In: *Proceedings of the 9th ACM Conference on Computer and Communications Security* (*CCS 2002*), Washington, DC, USA, November 18–22, 2002. New York: ACM; 2002. P. 148–160. https://doi.org/10.1145/586110.586132
10. Vivekraja V., Nazhandali L. Circuit-level techniques for reliable physically unclonable functions. In: *Proceedings of the 2009 IEEE International Workshop on Hardware-Oriented Security and Trust* (*HOST 2009*), San Francisco, CA, USA, July 27, 2009. Piscataway, NJ: IEEE; 2009. P. 30–35. https://doi.org/10.1109/HST.2009.5225054
11. Pappu R., Recht B., Taylor J., Gershenfeld N. Physical one-way functions. *Science*. 2002;297(5589):2026–2030. https://doi.org/10.1126/science.1074376
12. Anandakumar N.N., Hashmi M.S., Tehranipoor M. FPGA-based Physical Unclonable Functions: A comprehensive overview of theory and architectures. *Integration*. 2021;81:175–194. https://doi.org/10.1016/j.vlsi.2021.06.001
13. Cao Y., Xu J., Wu J., Wu S., Huang Z., Zhang K. Advances in Physical Unclonable Functions Based on New Technologies: A Comprehensive Review. *Mathematics* (*Basel*). 2024;12(1):77. https://doi.org/10.3390/math12010077
14. Vatalaro M., De Rose R., Lanuzza M., Crupi F. Weak physically unclonable functions in CMOS technology: A review. *Chips*. 2025;4(1):3. https://doi.org/10.3390/chips4010003
15. Sklavos N., Chaves R., Di Natale G., Regazzoni F. *Hardware Security and Trust: Design and Deployment of Integrated Circuits in a Threatened Environment*. Cham: Springer; 2017, 254 p. https://doi.org/10.1007/978-3-319-44318-8
16. Lata K., Cenkeramaddi L.R. FPGA-Based PUF Designs: A comprehensive review and comparative analysis. *Cryptography*. 2023;7(4):55. https://doi.org/10.3390/cryptography7040055
17. Masoumian S., Selimis G., Wang R., Schrijen G-J., Hamdioui S., Taouil M. Reliability analysis of FinFET-based SRAM PUFs for 16 nm, 14 nm and 7 nm technology nodes. In: *Proceedings of the 2022 Design, Automation & Test in Europe Conference & Exhibition* (*DATE 2022*), Antwerp, Belgium, March 14–23, 2022. Piscataway, NJ: IEEE; 2022. P. 1189–1192. https://doi.org/10.23919/DATE54114.2022.9774735
18. Eiroa S., Baturone I., Acosta A.J., Dávila J. Using physical unclonable functions for hardware authentication: A survey. In: *Proceedings of the 25th Conference on Design of Circuits and Integrated Systems* (*DCIS 2010*), Lanzarote, Canary Islands, Spain, November 17–19, 2010. Lanzarote; 2010. Available from URL: https://digital.csic.es/bitstream/10261/96029/1/Using%20Physical.pdf
19. Bossuet L., Ngo X.T., Cherif Z., Fischer V. A PUF based on a transient effect ring oscillator and insensitive to locking phenomenon. *IEEE Trans. Emerg. Top. Comput.* 2014;2(1):30–36. https://doi.org/10.1109/TETC.2013.2287182
20. Brzuska C., Fischlin M., Schröder H., Katzenbeisser S. Physically uncloneable functions in the universal composition framework. In: Rogaway P. (Ed.). *Advances in Cryptology – CRYPTO 2011*, Santa Barbara, CA, USA, August 14–18, 2011. Book Series: Lecture Notes in Computer Science. Berlin: Springer; 2011. V. 6841. P. 51–70. https://doi.org/10.1007/978-3-642-22792-9_4

21. Tuyls P., Schrijen G-J., Škorić B., van Geloven J., Verhaegh N., Wolters R. Read-proof hardware from protective coatings. In: Goubin L., Matsui M. (Eds.). *Cryptographic Hardware and Embedded Systems. CHES 2006*, Yokohama, Japan, October 10–13, 2006. Book Series: Lecture Notes in Computer Science. Berlin: Springer; 2006. V. 4249. P. 369–383. https://doi.org/10.1007/11894063_29
22. Chen Q., Csaba G., Lugli P., Schlichtmann U., Rührmair U. The bistable ring PUF: a new architecture for strong physical unclonable functions. In: *Proceedings of the 2011 IEEE International Symposium on Hardware-Oriented Security and Trust* (*HOST 2011*), San Diego, CA, USA, June 5–6, 2011. Piscataway, NJ: IEEE; 2011. P. 134–141. https://doi.org/10.1109/HST.2011.5955011
23. Abulibdeh E., Saleh H., Mohammad B., Al-Qutayri M., Veeran A. Area and power efficient implementation of configurable ring oscillator PUF. *TechRxiv Preprint*; April 2, 2024. https://doi.org/10.36227/techrxiv.171207533.30573247/v1
24. Abulibdeh E., Saleh H., Mohammad B., Al-Qutayri M., Hussain A. Kernel-based response extraction approach for efficient configurable ring oscillator PUF. *Sci. Rep.* 2025;15:5938. https://doi.org/10.1038/s41598-025-89769-5
25. Ivaniuk A.A., Yarmolik V.N. Configurable ring oscillator with controlled interconnections. *Bezopasnost' informatsionnykh tekhnologii = IT Security* (*Russia*). 2024;31(2):121–133 (in Russ.). https://doi.org/10.26583/bit.2024.2.08
26. Du H., Guo C., Cui S. Optimization design of the RO PUF temperature reliability based on MOSFET temperature characteristics. In: *The International Conference Optoelectronic Information and Optical Engineering* (*OIOE 2024*), Wuhan, China, October 18–20, 2024. Proc. SPIE 13513; 2025. Art. 1351324. https://doi.org/10.1117/12.3045630
27. Schaller A., Xiong W., Anagnostopoulos N.A., Saleem M.U., Gabmeyer S., Katzenbeisser S., Szefer J. Intrinsic Rowhammer PUFs: Leveraging the Rowhammer effect for improved security. In: *Proceedings of the 2017 IEEE International Symposium on Hardware-Oriented Security and Trust* (*HOST 2017*), McLean, VA, USA, May 1–5, 2017. Piscataway, NJ: IEEE; 2017. P. 1–7. https://doi.org/10.1109/HST.2017.7951729
28. Anandakumar N.N., Hashmi M.S., Chaudhary M.A. Implementation of efficient XOR arbiter PUF on FPGA with enhanced uniqueness and security. *IEEE Access*. 2022;10:129832–129842. https://doi.org/10.1109/ACCESS.2022.3228635
29. Hori Y., Kang H., Katashita T., Satoh A. Pseudo-LFSR PUF: A compact, efficient and reliable physical unclonable function. In: *Proceedings of the 2011 International Conference on Reconfigurable Computing and FPGAs* (*ReConFig'11*), Cancun, Mexico, November 30 – December 2, 2011. Cancun: IEEE; 2011. P. 223–228. https://doi.org/10.1109/ReConFig.2011.72
30. Marchand C., Bossuet L., Cherkaoui A. Enhanced TERO-PUF implementations and characterization on FPGAs. In: *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays* (*FPGA 2016*), Monterey, CA, USA, February 21–23, 2016. New York: ACM; 2016. P. 282. https://doi.org/10.1145/2847263.2847298
31. Xu X., Rührmair U., Holcomb D.E., Burleson W.P. Security evaluation and enhancement of bistable ring PUFs. In: Mangard S., Schaumont P. (Eds.). *Radio Frequency Identification. RFIDSec 2015*. Book Series: Lecture Notes in Computer Science. Cham: Springer; 2015. V. 9440. P. 3–16. https://doi.org/10.1007/978-3-319-24837-0_1
32. Thirumoorthi M., Jovanovic M., Mirhassani M., Khalid M.A.S. Design and evaluation of a hybrid chaotic-bistable ring PUF. *IEEE Trans. Very Large Scale Integr.* (*VLSI*) *Syst.* 2021;29(11):1912–1921. https://doi.org/10.1109/TVLSI.2021.3111588
33. Sharifi F., Momeni H., Hosseini A. Ternary bistable ring PUF for high-secure applications. *J. Supercomput.* 2024;80: 12663–12685. https://doi.org/10.1007/s11227-024-05935-y
34. Rührmair U., van Dijk M. On the practical use of physical unclonable functions in oblivious transfer and bit commitment protocols. *J. Cryptogr. Eng.* 2013;3(1):17–28. https://doi.org/10.1007/s13389-013-0052-8
35. Rührmair U. Oblivious transfer based on physical unclonable functions. In: Acquisti A., Smith S.W., Sadeghi A.-R. (Eds.). *Trust and Trustworthy Computing. TRUST 2010*. Berlin: Springer; 2010. V. 6101. P. 430–440. https://doi.org/10.1007/978-3-642-13869-0_31
36. Roy A., Roy D., Stănică P. On combining Arbiter based PUFs. *Cryptogr. Commun.* 2025;17(2):493–510. https://doi.org/10.1007/s12095-024-00769-0
37. Driemeyer B., Mandry H., Wiens D.-P., Becker J., Kauffman J.G., Ortmanns M. An eye-opening Arbiter PUF for fingerprint generation using auto-error detection for PVT-robust masking and bit stabilization achieving a BER of 2e-8 in 28 nm CMOS. In: *Proceedings of the 2025 IEEE International Solid-State Circuits Conference* (*ISSCC 2025*), San Francisco, CA, USA, February 16–20, 2025. Piscataway, NJ: IEEE; 2025. P. 300–302. https://doi.org/10.1109/ISSCC49661.2025.10904785
38. Yao Y., Kim M., Li J., Markov I., Koushanfar F. ClockPUF: physical unclonable functions based on clock networks. In: *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition* (*DATE 2013*), Grenoble, France, March 18–22, 2013. Piscataway, NJ: IEEE; 2013. P. 422–427. https://doi.org/10.7873/DATE.2013.095
39. Khan S., Shah A.P., Chouhan S.S., Roy A., Roy D., Stănică P. Utilizing manufacturing variations to design a tri-state flip-flop PUF for IoT security applications. *Analog Integr. Circ. Sig. Process.* 2020;103:477–492. https://doi.org/10.1007/s10470-020-01642-9

40. Yuan T., Wang P., Zhang Y., Zhou Z. An overclocking clock software PUF circuit with no additional hardware resource overhead based on video coding circuit. *Integration*. 2025;101:102319. https://doi.org/10.1016/j.vlsi.2024.102319
41. Suzuki D., Shimizu K. The Glitch PUF: a new Delay-PUF architecture exploiting glitch shapes. In: Mangard S., Standaert F.-X. (Eds.). *Cryptographic Hardware and Embedded Systems. CHES 2010*, August 17–20, 2010. Santa Barbara, CA, USA. Book Series: Lecture Notes in Computer Science. Berlin: Springer; 2010. V. 6225. P. 366–382. https://doi.org/10.1007/978-3-642-15031-9_25
42. Anderson J. A PUF design for secure FPGA-based embedded systems. In: *Proceedings of the 15th Asia South Pacific Design Automation Conference* (*ASP-DAC 2010*), Taipei, Taiwan, January 18–21, 2010. Piscataway, NJ: IEEE; 2010. P. 1–6. https://doi.org/10.1109/ASPDAC.2010.5419927
43. Ni L., Wang P., Zhang Y., Chen J., Li L., Zhang H. A reliable multi-information entropy glitch PUF using Schmitt trigger sampling method for IoT security. In: *2021 IEEE 14th International Conference on ASIC* (*ASICON 2021*), Kunming, China, October 26–29, 2021. Piscataway, NJ: IEEE; 2021. P. 1–4. https://doi.org/10.1109/ASICON52560.2021.9620406
44. Nozaki Y., Takemoto S., Yoshikawa M. Error correction method for lightweight cipher PRINCE-based physically unclonable function. In: *Proceedings of the 6th International Conference on Information Technology and Computer Communications* (*ITCC 2024*), Xi'an, China, July 5–7, 2024. New York: ACM; 2024. P. 38–42. https://doi.org/10.1145/3704391.3704397
45. Guajardo J., Kumar S.S., Schrijen G-J., Tuyls P. FPGA intrinsic PUFs and their use for IP protection. In: Paillier P., Verbauwhede I. (Eds.). *Cryptographic Hardware and Embedded Systems – CHES 2007*, Vienna, Austria, September 10–13, 2007. Lecture Notes in Computer Science. Berlin: Springer; 2007. V. 4727. P. 63–80. https://doi.org/10.1007/978-3-540-74735-2_5
46. Holcomb D.E., Burleson W.P., Fu K. Initial SRAM state as a fingerprint and source of true random numbers for RFID tags. *Proceedings of the Conference on RFID Security*. 2007;7(2):01–012.
47. Gebali F., Mamun M. Review of physically unclonable functions (PUFs): Structures, models, and algorithms. *Front. Sens.* 2022;2:751748. https://doi.org/10.3389/fsens.2021.751748
48. 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.* 2009;58(9):1198–1210. https://doi.org/10.1109/TC.2008.212
49. Kumar S., Guajardo J., Maes R., Schrijen G-J., Tuyls P. The butterfly PUF: protecting IP on every FPGA. In: *Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust* (*HOST 2008*), Anaheim, CA, USA, June 3–4, 2008. Piscataway, NJ: IEEE; 2008. P. 67–70. https://doi.org/10.1109/HST.2008.4559053
50. Farha F., Ning H., Ali K., Chen L., Nugent C. SRAM-PUF-based entities authentication scheme for resource-constrained IoT devices. *IEEE Internet Things J.* 2021;8(7):5904–5913. https://doi.org/10.1109/JIOT.2020.3032518
51. Su Y., Holleman J., Otis B. A 1.6 μJ/bit stable chip-ID generating circuit using process variations. In: *Proceedings of the IEEE International Solid-State Circuits Conference* (*ISSCC 2007*), San Francisco, CA, USA, February 11–15, 2007. Piscataway, NJ: IEEE; 2007. P. 606–611. https://doi.org/10.1109/ISSCC.2007.373466
52. Tehranipoor F., Karimian N., Yan W., Chandy J.A. Investigation of DRAM PUFs reliability under device accelerated aging effects. In: *Proceedings of the 2017 IEEE International Symposium on Circuits and Systems* (*ISCAS 2017*), Baltimore, MD, USA, May 28–31, 2017. Piscataway, NJ: IEEE; 2017. P. 1–4. https://doi.org/10.1109/ISCAS.2017.8050629
53. Yue M., Karimian N., Yan W., Anagnostopoulos N.A., Tehranipoor F. DRAM-based authentication using deep convolutional neural networks. *IEEE Consum. Electron. Mag.* 2021;10(4):8–17. https://doi.org/10.1109/MCE.2020.3002528
54. Sutar S., Raha A., Raghunathan V. D-PUF: an intrinsically reconfigurable DRAM PUF for device authentication in embedded systems. In: *Proceedings of the 2016 International Conference on Compilers, Architectures and Synthesis of Embedded Systems* (*CASES 2016*), Pittsburgh, PA, USA, October 2–7, 2016. New York: ACM; 2016. P. 1–10. https://doi.org/10.1145/2968455.2968519
55. Chew Y.Y., Lim W.L., Tan J.L., Ooi C.Y. In-depth review and comparative analysis of DRAM-based PUFs. *IEEE Access*. 2025;13:79367–79384. https://doi.org/10.1109/ACCESS.2025.3566068
56. Wilson T., Cambou B. Tamper-sensitive pre-formed ReRAM-based PUFs: Methods and experimental validation. *Front. Nanotechnol.* 2022;4:1055545. https://doi.org/10.3389/fnano.2022.1055545
57. Napolean A., Sivamangai N.M., Sharon N., Naveen Kuma R. Review on resistive random access memory based physical unclonable function circuits for high security. *Procedia Environ. Sci. Eng. Manag.* 2023;10(1):41–52. Available from URL: http://www.procedia-esem.eu/pdf/issues/2023/no1/5_Napolean_22.pdf
58. Adel M.J., Rezayati M.H., Moaiyeri M.H., et al. A robust deep learning attack immune MRAM-based physical unclonable function. *Sci. Rep.* 2024;14:20649. https://doi.org/10.1038/s41598-024-71730-7
59. Go S.X., Wang Q., Lim K.G., Lee T.H., Bajalovic N., Loke D.K. Ultrafast near-ideal phase-change memristive physical unclonable functions driven by amorphous state variations. *Adv. Sci.* (*Weinh.*) 2022;9(36):e2204453. https://doi.org/10.1002/advs.202204453

60. Yang J., Lei D., Chen D., Li J., Jiang H., Luo Q., et al. Machine-learning-resistant 3D PUF with 8-layer stacking vertical RRAM and 0.014% bit error rate using in-cell stabilization scheme for IoT security applications. In: *2020 IEEE International Electron Devices Meeting* (*IEDM*), San Francisco, CA, USA, December 12–18, 2020. Piscataway, NJ: IEEE; 2020. P. 28.6.1–28.6.4. https://doi.org/10.1109/IEDM13553.2020.9372107
61. Li J., Cui Y., Gu C., Wang C., Liu W., Kvatinsky S. A highly reliable dual-mode RRAM PUF with key concealment scheme. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* 2025. https://doi.org/10.1109/TCAD.2025.3536376

## About the Authors

**Evgenii Ph. Pevtsov,** Cand. Sci. (Eng.), Director of Center for the Design of Integrated Circuits, Nanoelectronics Devices and Microsystems, MIREA – Russian Technological University (78, Vernadskogo pr., Moscow, 119454 Russia). E-mail: pevtsov@mirea.ru. Scopus Author ID 6602652601, ResearcherID M-2709-2016, RSCI SPIN-code 1410-2483, http://orcid.org/0000-0001-6264-1231

**Tatyana A. Demenkova,** Cand. Sci. (Eng.), Associated Professor, Computer Technology Department, Institute of Information Technologies, MIREA – Russian Technological University (78, Vernadskogo pr., Moscow, 119454 Russia). E-mail: demenkova@mirea.ru. Scopus Author ID 57192958412, ResearcherID AAB-3937-2020, RSCI SPIN-code 3424-7489, http://orcid.org/0000-0003-3519-6683

**Yuri A. Korotaev,** Postgraduate Student, Department of Nanoelectronics, Institute for Advanced Technologies and Industrial Programming, MIREA – Russian Technological University (78, Vernadskogo pr., Moscow, 119454 Russia). E-mail: korotaevyua@yandex.ru. RSCI SPIN-code 7428-683, https://orcid.org/0009-0000-3976-7872

**Alexander S. Sigov,** Academician at the Russian Academy of Sciences, Dr. Sci. (Phys.–Math.), Professor, President, MIREA – Russian Technological University (78, Vernadskogo pr., Moscow, 119454 Russia). E-mail: sigov@mirea.ru. Scopus Author ID 35557510600, ResearcherID L-4103-2017, RSCI SPIN-code 2869-5663, https://www.researchgate.net/profile/A_Sigov

## Об авторах

**Певцов Евгений Филиппович,** к.т.н., директор структурного подразделения «Центр проектирования интегральных схем, устройств наноэлектроники и микросистем», ФГБОУ ВО «МИРЭА – Российский технологический университет» (119454, Россия, Москва, пр-т Вернадского, д. 78). E-mail: pevtsov@mirea.ru. Scopus Author ID 6602652601, ResearcherID M-2709-2016, SPIN-код РИНЦ 1410-2483, https://orcid.org/0000-0001-6264-1231

**Деменкова Татьяна Александровна,** к.т.н., доцент, кафедра вычислительной техники, Институт информационных технологий, ФГБОУ ВО «МИРЭА – Российский технологический университет» (119454, Россия, Москва, пр-т Вернадского, д. 78). E-mail: demenkova@mirea.ru. Scopus Author ID 57192958412, ResearcherID AAB-3937-2020, SPIN-код РИНЦ 3424-7489, https://orcid.org/0000-0003-3519-6683

**Коротаев Юрий Александрович,** аспирант, кафедра наноэлектроники, Институт перспективных технологий и индустриального программирования, ФГБОУ ВО «МИРЭА – Российский технологический университет» (119454, Россия, Москва, пр-т Вернадского, д. 78). E-mail: korotaevyua@yandex.ru. SPIN-код РИНЦ 7428-6831, https://orcid.org/0009-0000-3976-7872

**Сигов Александр Сергеевич,** академик Российской академии наук, д.ф.-м.н., профессор, президент ФГБОУ ВО «МИРЭА – Российский технологический университет» (119454, Россия, Москва, пр-т Вернадского, д. 78). E-mail: sigov@mirea.ru. Scopus Author ID 35557510600, ResearcherID L-4103-2017, SPIN-код РИНЦ 2869-5663, https://www.researchgate.net/profile/A_Sigov

*Translated from Russian into English by K. Nazarov*
*Edited for English language and spelling by Thomas A. Beavitt*