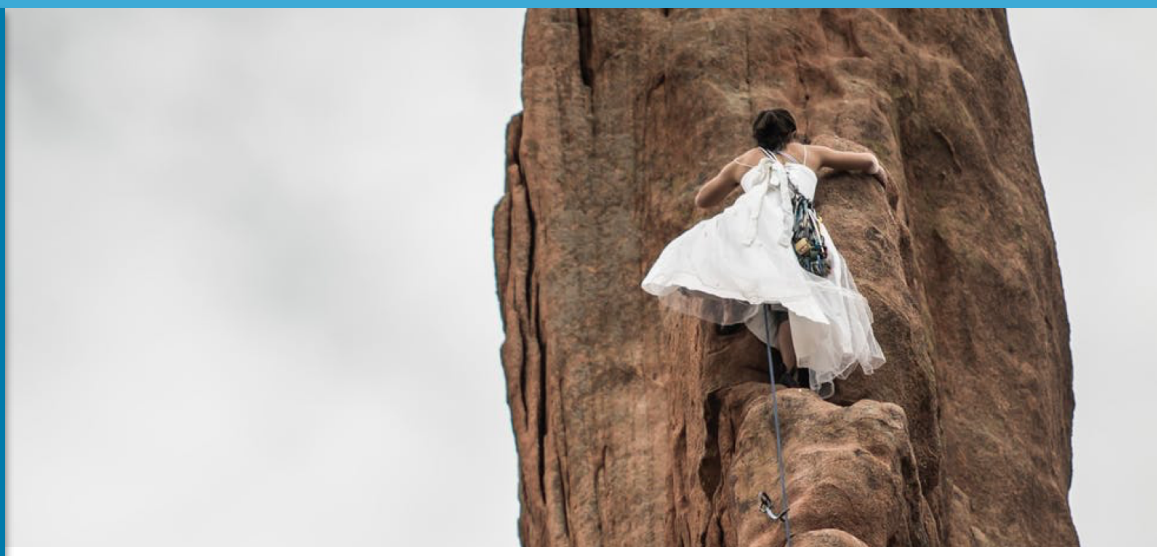
INTRINSIC ID

WHITE PAPER

The Reliability of SRAM PUF



SRAM PUF Benefits

- Uses standard SRAM
- Device-unique, unclonable key
- No secrets reside on the IC
- No key material programmed
- Flexible and scalable
- Highly reliable across large range of operating environments and on every technology node
- Lifetime >25 years

Certifications

- EMVCo, Visa
- CC EAL6+
- US and EU Governments

Markets

- IoT
- Secure Transactions
- Government & Defense

Securing Billions of IoT Devices with Reliable HW-based Keys that are Never Stored.

Physical Unclonable Functions or PUFs are increasingly being deployed as a hardware root-of-trust to secure IoT devices, data and services. They often outcompete traditional non-volatile memories (e.g. flash, EEPROM, anti- fuses, etc.) on different performance metrics such as security, flexibility and cost. The main strength of using PUF is that device-unique keys are generated using the entropy of the manufacturing process of an integrated circuit (IC). Therefore, no external sensitive key needs to be injected and/or programmed on the IC. Moreover, keys are not visible when the device is powered off. This explains the increasing use of PUF as a highly secure yet efficient key storage solution.

Among all PUF systems, the Static Random Access Memory (SRAM) PUF is the most mature. Another important benefit, which is not self-evident and often badly understood, is its reliability. The reliability of a PUF system is rather complex since it depends on the implementation of the PUF, the PUF behavior and the PUF post-processing or key extraction. In this document, we explore the reliability and all its aspects for Intrinsic ID's SRAM PUF system and show that it is a very reliable storage medium for a cryptographic key, even under extreme conditions and for the entire lifetime of the IC.

Introduction

Given its advantageous properties, PUFs are increasingly used as a hardware root-of-trust[1] to secure IoT devices, data and services. To enforce secure applications, the manufacturing process variation of an IC can be leveraged directly as a source of randomness for the generation of device-unique cryptographic keys. For such applications, a high-quality PUF is needed that is reliable and whose outputs (or responses) from different devices are unpredictable. The reliability of a PUF system depends on the implementation of the PUF, the PUF behavior and the PUF post-processing or key extraction.

The foundation of a reliable and secure PUF system is the probabilistic behavior of the PUF itself. In the case of SRAM PUF this is determined by the power-up values of the SRAM cells. Analytically, each SRAM cell has two stable states that represent a 1 or a 0. When a cell is powered up, the resulting state is unpredictable, but it turns out that random sub-microscopic differences between the transistors in the cell give every cell a preference to come up as a 0 or a 1. For a block of cells this results in a random pattern, like a silicon fingerprint, that is unique per IC and unclonable. This pattern can be used to generate a hardware-based device-unique key.

However, some of the cells that are closely balanced can be unstable during SRAM power cycles and generate inverted bit values on the initial pattern of zeros and ones (cell flipping). The number of bits that are inverted divided by the total number of bits in the pattern is defined as the SRAM PUF noise. Cell flipping is prone mainly to local temperature differences, local supply voltage variation and aging. For this reason it is important to conduct an extensive reliability study to understand the effects that lead to SRAM PUF noise.

Figure 1 outlines the different elements that contribute to the reliability of the SRAM PUF system. The probabilistic behavior of the PUF itself, i.e. the noise in the SRAM PUF power-up values, will be investigated in Part A of this document. Part B covers the post processing or key extraction. In Part C we look at the probability for key failures.

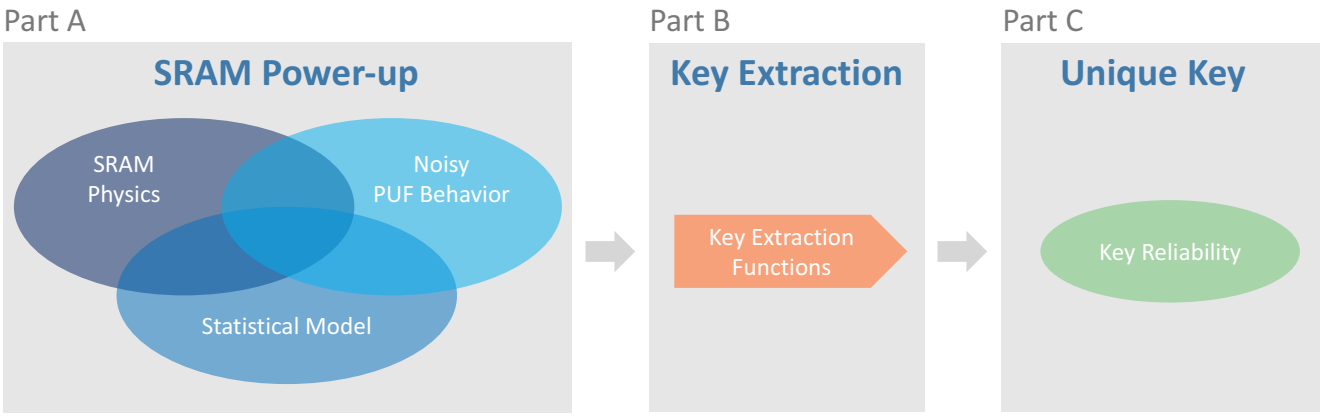


Figure 1. Elements that contribute to the reliability of a PUF system.

[1] Every computer/embedded system is built with multiple layers of abstraction, such as hardware, firmware, operating system and applications. To provide secure operations, the higher layers should trust the lower layers and the initial source of trust at the bottom of the system is called root-of-trust, http://csrc.nist.gov/groups/SMA/ispab/documents/minutes/2012-02/feb1_mobility-roots-of-trust_regenscheid.pdf

The behavioral characteristics of an SRAM PUF depend on the environment it is exposed to, such as the ambient temperature, voltage supply variation, and electromagnetic interference. Furthermore, it is well-known that the behavior changes over time, which is called aging. All these aspects will influence the SRAM PUF noise, and will be examined in Part A of this document. For a PUF system to be reliable it is important that the PUF noise remains small enough under a wide variety of environmental conditions for the entire lifetime of the IC.

Due to its noisy behavior, an SRAM PUF response cannot be used directly as a key. Post-processing of the PUF response is needed. This can be done by key extraction algorithms based on error-correction functions and randomness extractors[2]. The key extractor must be able to compensate for the noise of the PUF and derive the same cryptographic key each time it is queried. How this is done for an SRAM PUF will be explained in Part B.

In Part C we look at the SRAM PUF key reliability and the probability for key failures. It is shown that an SRAM PUF is an extreme reliable storage medium for a cryptographic key.

Part A – SRAM PUF Bit Error Rates

The Operation Principle of SRAM PUF

An SRAM PUF evaluates the power-up pattern of a standard 6T SRAM array. Each SRAM cell in the array comprises two nominally matched CMOS inverters which are cross-coupled (see Figure 2). Uncontrollable CMOS process variations introduce deep sub-micron variations that give each transistor slightly random electric properties. The power-up state of the SRAM cell will mainly be determined by the difference between the threshold voltages (V_{th}) of both PMOS transistors P1 and P2. For instance, consider the case when random variations cause $|V_{th,P1}|$ to be slightly smaller than $|V_{th,P2}|$. As a result, at power-up (rising V_{dd}) P1 will start conducting before P2, causing A to go logically high and preventing P2 from switching on. The power-up state of the cell is hence $A = 1$. The larger the mismatch between $V_{th,P1}$ and $V_{th,P2}$, the stronger the power-up preference of a cell and hence the smaller the probability to power-up in a different state.

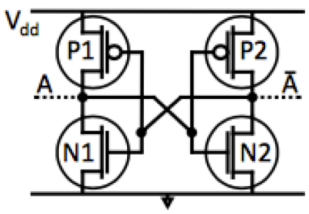


Figure 2. The core of an SRAM cell comprising two nominally matched CMOS inverters which are cross-coupled.

As a result, the SRAM cells with a large difference between the threshold voltages will be stable. The cells with $V_{th,P1} \approx V_{th,P2}$ have a higher probability to change their output from one evaluation to the other, hence causing bit flips or noise in the SRAM PUF response. Since threshold voltages are sensitive to temperature, voltage supply, aging etc., the SRAM PUF noise will be sensitive to these changing conditions as well. This will be illustrated in the next sections.

[2] Fuzzy Extractor, https://en.wikipedia.org/wiki/Fuzzy_extractor

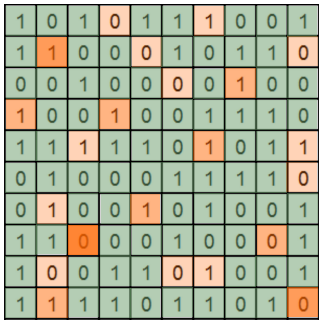


Figure 3b. Typically almost 80% of the SRAM PUF cells are very stable (green) and only a minority is less stable (orange). The large majority of SRAM PUF noise is hence caused by a small minority of cells which are different very often (dark orange).

Figure 3a shows the result of an experiment performed at room temperature where more than 50 million distinct but identically implemented SRAM cells were evaluated 60 times. For each cell the error count, i.e. the number of responses that were different from the reference response, was calculated. It shows that almost 80% of the SRAM PUF cells are stable and never change their output on power-up in the 60 evaluations. Only 0.2% are different about half the time and about 0.06% are always different from the original measurement in 60 evaluations. Hence, as illustrated in Figure 3b, the large majority of errors in a PUF response is caused by a small minority of cells which are different very often. This SRAM PUF behavior was found consistently for SRAM from different designers, technology nodes from 350 nm to 14 nm, and under various conditions[3].

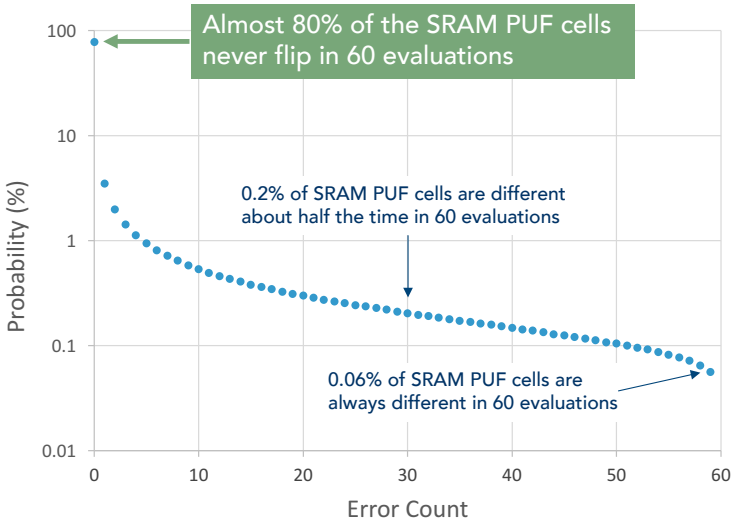


Figure 3a. Fifty million SRAM cells were evaluated 60 times. For each cell the error count, i.e. the number of responses that were different from the reference response, was calculated. The probability for each error count is shown.

The Impact of Silicon Aging

For SRAM PUF silicon aging can be counteracted

An operational IC slowly but gradually changes over time, i.e. it ages. Eventually the induced physical changes affect the circuit's operation, typically in a degrading manner, and ultimately even lead to circuit failures. The main degradation effects that lead to SRAM failure are indicated in Figure 4a: Negative Bias Temperature Instability (NBTI), Hot Carrier Injection (HCI), Electromigration, and Time-Dependent Dielectric Breakdown (TDDB). HCI mainly affects N channel MOSFETs and deteriorates when the circuit is operating. Hence it has minor impact on the SRAM power-up values. TDDB will bring the SRAM cell in a fixed state which is no problem for the PUF since the cell is fully stable. In the next sections we will show that the error correction algorithm corrects up to 25% of the SRAM PUF bit values if needed.

[3] For more details we refer to the Intrinsic ID Reliability Report - Available under NDA

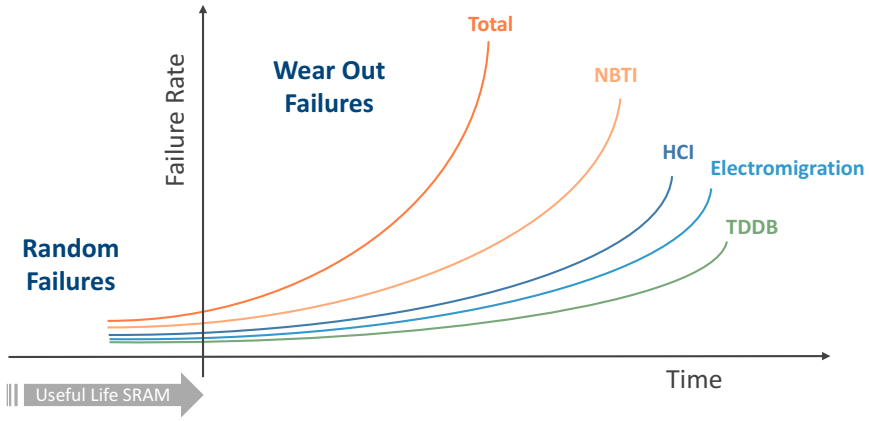


Figure 4a. The main degradation effects that lead to SRAM failure: Negative Bias Temperature Instability (NBTI), Hot Carrier Injection (HCI), Electromigration, and Time-Dependent Dielectric Breakdown (TDDB).

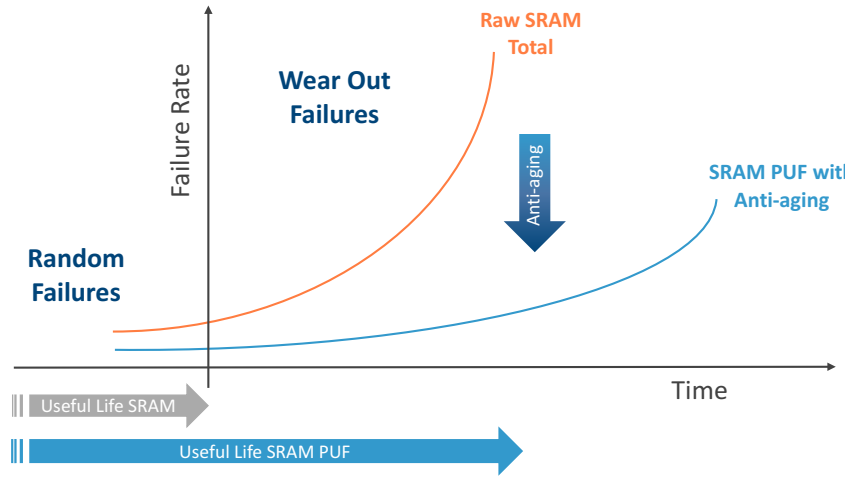


Figure 4b. For an SRAM PUF with anti-aging features, wear out failures are strongly reduced compared to raw SRAM. Anti-aging corrects for the main degradation effect, i.e. NBTI. Furthermore, effects such as HCI, radiation, and TDDB don't affect the working of the SRAM PUF system.

The dominant effect in modern ICs that at the same time has a large impact on the noisy behavior of the SRAM PUF is NBTI. It causes a gradual increase in the threshold voltage, which is most evident in switched-on PMOS transistors. The effect of NBTI aging for SRAM cells depends on the bit value stored in the cell. When the cell stores a zero ($A = 0$), P1 is switched off and P2 is switched on. As a result, $V_{th,P2}$ will increase over time due to NBTI while $V_{th,P1}$ is unaffected. For $A = 1$ the opposite effect occurs. Combined with the power-up behavior, the situation is such that the PMOS with the smallest V_{th} tends to turn on at power-up and will subsequently experience a gradually increasing V_{th} due to NBTI. The natural tendency of an SRAM cell is hence to age such that $|V_{th,P1} - V_{th,P2}|$ grows smaller over time. From an SRAM PUF perspective, this is a disadvantage since a decreasing $|V_{th,P1} - V_{th,P2}|$ means a higher probability of a PUF response bit error. In other words, when no countermeasures are taken, SRAM PUFs tend to become less reliable over time due to NBTI.

How aging effects the SRAM PUF noise is illustrated in Figure 5. Aging can be accelerated by exposing the IC to high temperatures and high voltage for a long period of time. In this experiment three ICs were exposed to a temperature of 80°C and a supply voltage of 1.1 * Vdd for 130 days. For each device, the SRAM was split in two areas. For each area the SRAM PUF was evaluated every hour and the response was compared to a reference value at room temperature. On the SRAM parts where no countermeasures were taken (figure on the left), the noise on the SRAM PUF increases from 5% to almost 15%.

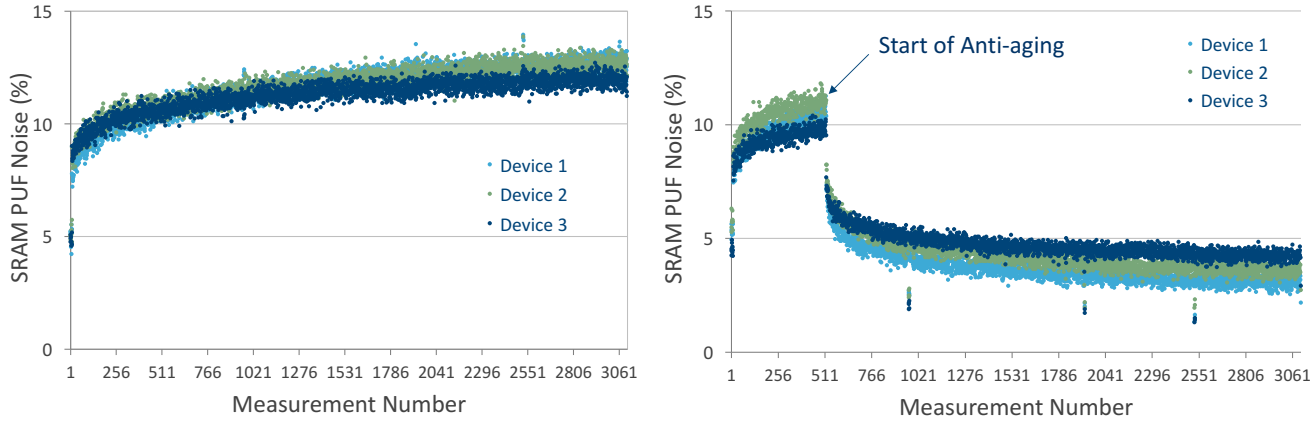


Figure 5. SRAM PUF noise during burn-in. The SRAM of three devices is split into two areas. The figure to the left shows the SRAM PUF noise versus time on SRAM areas where no aging countermeasure is applied. The figure to the right shows SRAM PUF noise versus time on areas where the countermeasure is applied after 20 days of burn-in. Dips correspond to measurements taken at +20°C, whereas the other measurements are taken at +80°C.

SRAM Aging Mitigation (Anti-aging)

Since the NBTI is well understood, there are several ways to counteract the aging tendency. An evident solution is to let each cell store the inverse of its power-up value, since this will increase $|V_{th,P1} - V_{th,P2}|$ and hence make the corresponding SRAM PUF response bit effectively more reliable over time.

In the past 10 years, anti-aging strategies have been developed that cause SRAM PUF to become more reliable over time, without degrading the other PUF quality measures such as security and efficiency[4]. The effect of anti-aging is illustrated in Figure 5 (right) where it is applied on the SRAM after 20 days of burn-in. The PUF noise drops almost immediately below 10% and holds a decreasing trend even after more that 100 days of burn-in. It should be noted that, when applied from the beginning, this anti-aging strategy even reduces the natural SRAM PUF noise (see Figure 8).

A major practical advantage of SRAM PUF anti-aging solutions is that they do not require any circuit changes or pre-deployment effort. They are hence usable for standard SRAM implementations in a regular development flow. Furthermore, it is important to mention that due to anti-aging the wear-out failure rates of the SRAM PUF will be much lower than the failure rates of SRAM under typical use (raw SRAM) and hence the useful life of an SRAM PUF is much longer than that of the raw SRAM (see Figure 4b).

[4] R. Maes and V. van der Leest, "Countering the effects of silicon aging on SRAM PUFs", Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust (HOST), pp. 148-153 available at http://www.Intrinsic.id.com/wp-content/uploads/2014/09/PUF_aging.pdf

Empirical Data

A PUF is reliable when the PUF responses are repeatable with limited noise over time and under operating specifications. From the previous section it is clear that operating conditions such as environmental temperature and supply voltage as well as lifetime of the IC will affect the SRAM power-up values and hence the PUF's reliability. When no anti-aging measures are taken, the worst-case reliability of an SRAM PUF is expected at a point in the future at the end of the device's lifetime, after years of silicon aging. But thanks to the anti-aging solution for SRAM PUF, as explained in previous section, the worst-case can now be investigated at the beginning of the device's lifetime, immediately after manufacturing. Over time the PUF's reliability will stay constant or even improve, which means that the reliability requirements for the PUF-based application can be significantly relaxed, resulting in a gain in efficiency (e.g. less complex error-correcting codes and using less SRAM) for the post-processing.

To determine the worst-case reliability the SRAM power-up behavior has been qualified under a wide variety of circumstances and foundry processes[5]:

- Semiconductor technology nodes ranging from 350nm down to 14nm
- Semiconductor process optimizations for low power, high speed, and high density
- Temperature range for PUF reading from -50°C to 150°C [-58°F to 300°F]
- Voltage supply variation +/- 20%
- Humidity up to 80%
- EMC tests at 3V/m (EN55020 0.15–150 MHz and IEC 61000-4-3 80-1000MHz)

The noise properties of SRAM PUFs can be investigated by measuring the PUF responses (SRAM power-up values) under a wide variety of operating conditions and comparing these PUF responses with their reference value i.e. the first measurement typically taken at room temperature (also called the enrollment measurement cfr. infra). When the SRAM PUF is used as a key storage medium, the post-processing or key extractor has to deal with the SRAM PUF noise under all operating conditions in order to reconstruct the same key every time it is required.

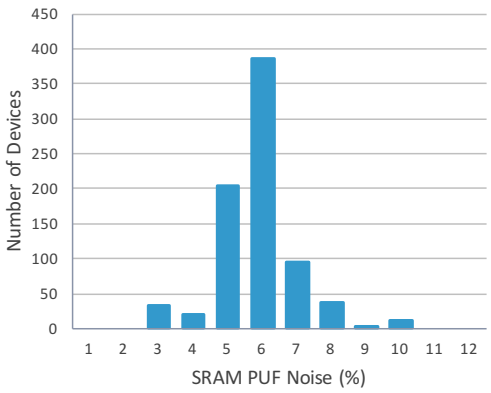


Figure 6. Distribution of noise levels measured in SRAM PUF at room temperature

SRAM PUF at Room Temperature

Since 2003, billions of SRAM PUF measurements have been conducted by Intrinsic ID, its partners and customers. For ICs produced in different factories around the world (Global Foundries, Intel, Samsung, UMC, Cypress, TSMC, IBM, Renesas, etc.) and SRAM from different designers (TSMC, ARM, Dolphin, Synopsys, etc.), power-up values have been taken under various conditions. Technology nodes varied from 350 nm to 14 nm. Figure 6 shows a distribution of the maximum SRAM PUF noise measured at room temperature on all these different ICs. The average noise is about 6%

[5] For more details we refer to the Intrinsic ID Reliability Report - Available under NDA

and the maximum noise measured at room temperature never exceeded 11%. Neither the IC type nor the technology node has a significant impact on the noise of an SRAM PUF.

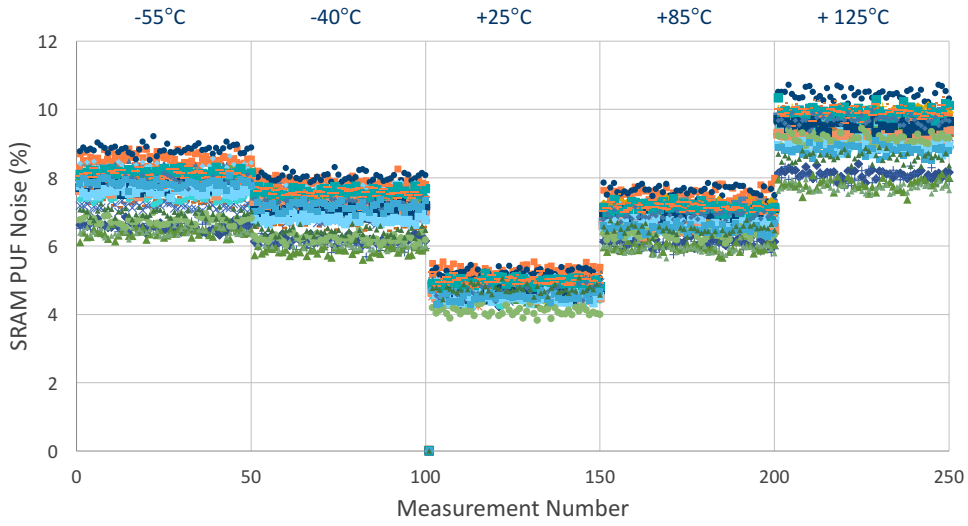


Figure 7. A typical temperature variation measurement taken for military applications, where temperatures ranged from -55°C to +125°C showing the sensitivity of the SRAM PUF noise on changing temperatures.

The Influence of Temperature Variation

Figure 7 shows the results of a a typical temperature variation measurement taken for military applications, where temperatures ranged from -55°C to +125°C. In this experiment SRAM memory on TSMC 65nm technology node[6] was placed in a climate chamber and stabilized at the indicated temperatures. The SRAM PUF response was compared to its reference value at 25°C in order to calculate the noise. As expected, the SRAM PUF noise was higher at extreme temperatures (-55°C and +125°C) but still lower than 11%.

Accelerated Lifetime Test

A typical SRAM aging test or burn-in test is set up as follows:

- The SRAM memory of several ICs is kept powered at an ambient temperature of +125°C. The SRAM memory is split into two areas: 1) an area where no aging countermeasure is applied; 2) an area where Intrinsic ID anti-aging is applied.
- The devices are all powered with a core voltage that is 20% higher than the nominal operating voltage.
- Every 6 hours the memories are re-powered and a measurement is taken at the same high temperature of +125°C.

[6] UMC 65nm CMOS process

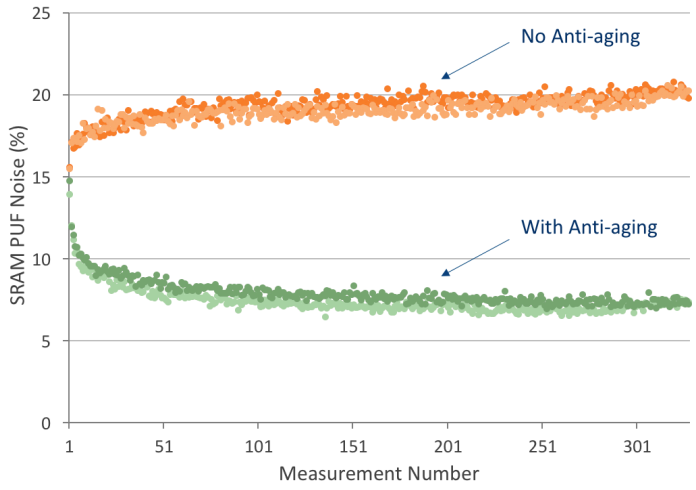


Figure 8. Burn-in test at T = +125 °C and increased core voltage (+20%), showing the noise level of SRAM PUF over time when no aging countermeasure is applied (red), and when applying the aging countermeasure from the start (green).

The duration of the burn-in test indicated in Figure 8 was 2,000 hours. Exposing the IC to such a high temperature and voltage will accelerate the aging effect due to NBTI. Depending on the application, this lifetime test will simulate an effective aging of more than 4 years. In this case, the difference between an enrollment measurement taken at 25°C increases toward 21% after 2,000 hours of aging (T = +125°C, V_{dd} = 1.31V). When applying the aging countermeasure, the noise with respect to the enrollment measurement decreases to a level below 8% (green data points).

In most applications the SRAM PUF noise will be lower than 10%, and in worst-case scenarios it might rise to 18%. Furthermore, this noise is coming from a minority of SRAM cells that are flipping very often. Hence, from a reliability point of view, SRAM PUFs are very well suited as a secure storage medium for a cryptographic key. It will be shown in Part B that it is possible to extract a secure key from PUFs even for noise levels up to 25% and with a very high reliability.

Statistical Models

To get more insight into a PUF's reliability, a statistical model that closely fits the empirical statistics is of great importance. Such a statistical model serves multiple needs:

- Understand where the behavioral characteristics are coming from.
- Extrapolate predictions to unobserved points.
- Analyze the design space of a PUF system and converge to an optimized solution.

In 2013 a PUF reliability model was introduced[7] that takes into account the observed heterogeneous nature of PUF cells. A substantial experimental validation demonstrates that the predicted distributions describe the empirically observed data statistics almost perfectly, even considering sensitivity to operational temperature. This allows studying PUF failure behavior in full detail, including the average and the worst-case probabilities. For more details we refer to literature[7].

[7] R. Maes: "An Accurate Probabilistic Reliability Model for Silicon PUFs," in Cryptographic Hardware and Embedded Systems, CHES 2013, LNCS vol. 8086, pp. 73–89, Aug. 2013.

Part B

Key Extraction from SRAM PUF

To turn the noisy SRAM PUF responses into a reliable and secure device-unique key, a Fuzzy Extractor or Helper Data Algorithm is used[8]. Such an algorithm implements two processing steps: i) Error Correction (also referred to as information reconciliation) and ii) Privacy Amplification (also referred to as randomness extraction). In the next sections we will concentrate on the error correction component and show that an SRAM PUF is a very reliable storage medium for a cryptographic key. For Privacy Amplification, we refer to literature[9].

Error Correction – Toy Example

To illustrate the idea of error correction, consider a very small SRAM PUF consisting of one byte (eight bits). The SRAM PUF response can be thought of as a bit string of eight bits. For the sake of simplicity, we extract one secret bit (1 or 0) from this eight-bit SRAM PUF, which corresponds to a code rate of 1/8. The error correction code that we use in this example is a repetition code of length eight. This means that the secret bit '0' will be encoded into the code word $C_0 = (00000000)$ and the secret bit '1' into the code word $C_1 = (11111111)$.

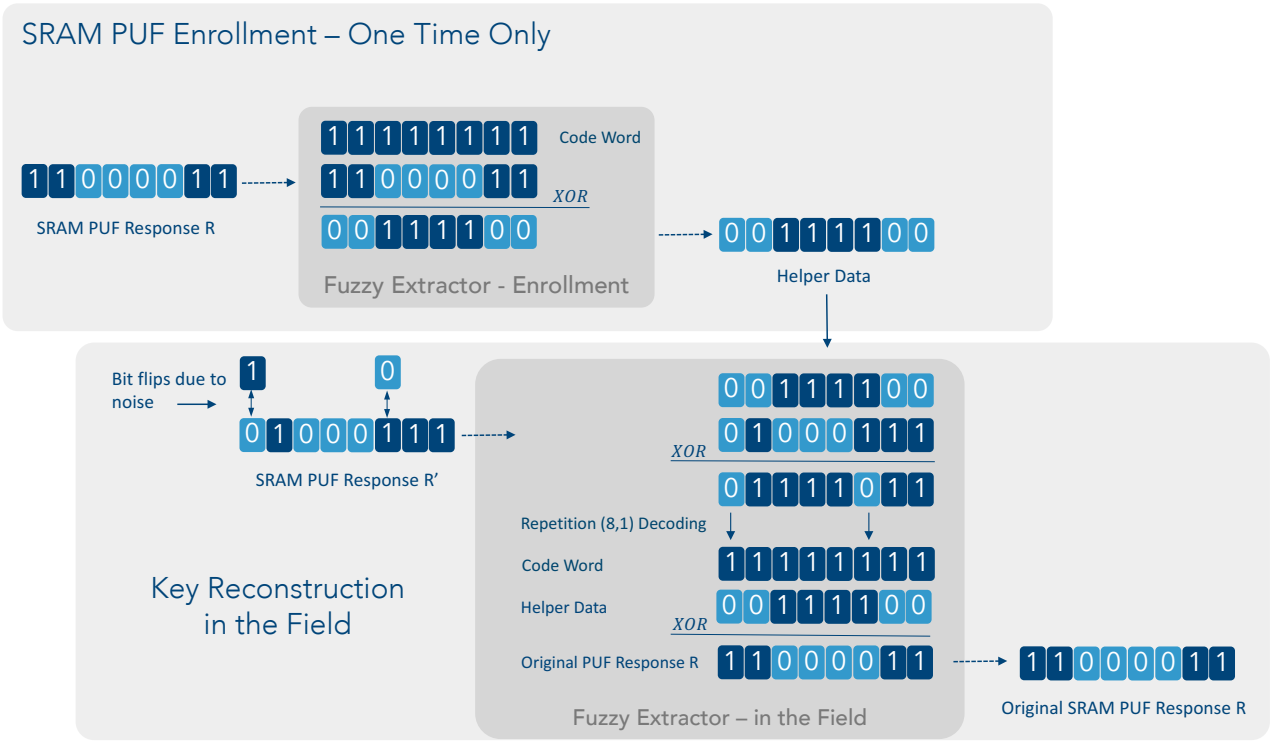


Figure 9. The enrollment phase (top) and the reconstruction phase (bottom) of an SRAM PUF.

[8] Fuzzy Extractor, https://en.wikipedia.org/wiki/Fuzzy_extractor

[9] P. Tuyls, B. Škorić and T. Kevenaar: "Security with Noisy Data: Private Biometrics, Secure Key Storage and Anti-Counterfeiting," Springer, 2007.

Enrollment Phase – One Time Only

The life of an SRAM PUF starts with an enrollment phase. This is a one-time process during which the following steps are performed, as illustrated in the upper part of Figure 9. First, an SRAM PUF response is taken, e.g. $R = (11000011)$. Second, a code word is chosen randomly in the error correction code space and the SRAM PUF response is mapped onto this code word. In this simple example we only have two code words: the all-one and all-zero strings: $C_1 = (11111111)$ and $C_0 = (00000000)$. In case C_1 is selected, the SRAM PUF response R is mapped onto C_1. This is done by computing the difference (bitwise XOR) between the code word C_1 and R:

$$C_1 \oplus R = (00111100)$$

The resulting bit string (00111100) is called Helper Data. This is non-sensitive data and can be stored, e.g. in the internal memory of the IC, in external memory, in the cloud, etc., in an accessible manner so it can be used in the field. Once this storage has been finalized, the enrollment phase is finished.

Key Reconstruction – In the Field

Later, in the field when the secret key has to be reconstructed, a new 'noisy' response R' is measured. Assume that the new SRAM PUF response is $R' = (01000111)$. Note that R' is different from R in two positions (position 1 and position 6). To retrieve the original response R we add the Helper Data to R' and obtain:

$$(00111100) \oplus R' = (01111011)$$

This result lies closed to the code word C_1. Hence this word is decoded to C_1 and we can compute the original Response R as follows:

$$R = C_1 \oplus \text{Helper Data } = (11000011)$$

The decoding algorithm of this code is easy. When a word of length eight is obtained one checks whether it has mostly ones or mostly zeros. When it has more zeros than ones, it is decoded to C_0, otherwise to C_1. Note that this code can correct up to three errors.

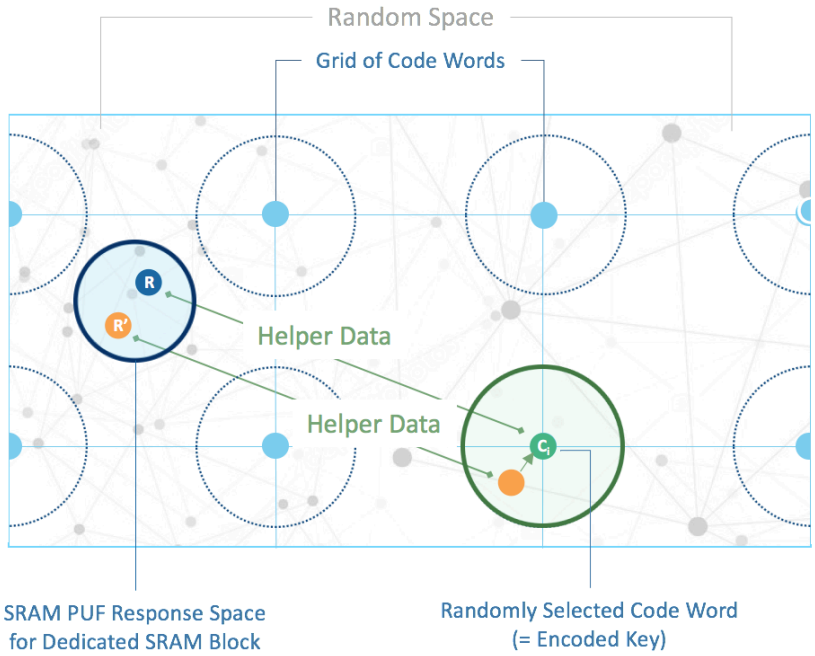


Figure 10. The set of responses of a specific SRAM PUF can be considered as lying within a random sphere in a high dimensional space. A grid in this space is defined whose vertices are the code words of a well-constructed error correction code. During enrollment, Helper Data is created to map the SRAM PUF response R onto a randomly chosen code word C_i. Later, when used in the field, the Helper data will map the new SRAM PUF response R' into the error correction sphere of the same code word C_i. Once the code word is recovered, the reference SRAM PUF response R is recovered too.

Error Correction – General Approach

In essence, an SRAM PUF response is a very long random bit string. Therefore, it can be considered a random binary vector in a high-dimensional space (see Figure 10). Since all responses of one specific SRAM PUF are close to each other we can consider the set of likely SRAM PUF responses as lying within a sphere (of finite radius) in this high-dimensional space.

Error correction techniques enable reliable key reconstruction even under worst-case conditions

However, since we don't have any structure in this space, we have no means to map a new measurement onto a previous or reference measurement. To deal with this, we provide structure in this space by defining a grid in the space whose vertices are the code words of a well-constructed error correction code. The code words of this code are bit strings of the same length as the SRAM PUF responses and are located at the centers of the error correction sphere. All bit strings that are close to the code words (which are within the error correction sphere) can be mapped by the decoding algorithm onto the center of the error correction sphere or the code word (= encoded key). The choice of the error correction code, can be considered a system parameter that is used during all phases of the SRAM PUF.

The error correction procedure is similar to the toy example above and illustrated in the Figure 10. During enrollment, helper data is created to map the SRAM PUF response R onto a randomly chosen code word C_i. Later, when used in the field, the helper data will map the new SRAM PUF response R' into the error correction sphere of the same code word C_i. By running the decoding algorithm, the noise is removed and the code word is recovered. With the help of the helper data, the SRAM PUF response R, taken during enrollment, is recovered, too.

For IoT applications the SRAM PUF system must be small, fast, reliable and secure

Note that, in practice, error correction algorithms are much more complicated than a single repetition code and able to cope with much higher noise. The choice of the right error correction code and its implementation is non-trivial since it must satisfy the following requirements in an efficient way:

1. **Security:** The error correcting code has to contain many code words. In case the number of code words is too small, the entropy of the generated device-unique key would be too small.
2. **Reliability:** The error correction code needs to be able to correct all SRAM PUF bit errors even in the worst-case circumstances.
3. **Speed:** The error code has to have an efficient decoding algorithm. This guarantees that the keys are sufficiently fast created after power-up of the IC and meet market requirements.

4. **Size:** The error correction code needs to have a small implementation in software and hardware. In that way the cost of implementing it can be kept to a minimum and it can be integrated in limited-resource environments such as IoT devices.

For an SRAM PUF, these requirements can be satisfied by use of the following measures:

1. **Concatenation of existing error correction codes:** Build an efficient error correction code based on the concatenation of existing well-known and understood error correction codes (such as Reed-Solomon codes etc.)
2. **Soft Decision Decoding:** The code can be improved by using a decoding strategy based on soft-decision information.

In commercial applications, 1 KB of SRAM is needed for a 256-bit key

As such, typical error correction codes will use 32 bits for 1 bit of key material. Note that in the toy example 8 bits were needed for 1 bit of key material. In commercial applications 1KB or 8,192 SRAM cells are needed for a 256-bit key, and 0.5 KB or 4,096 bits are needed for a 128-bit key.

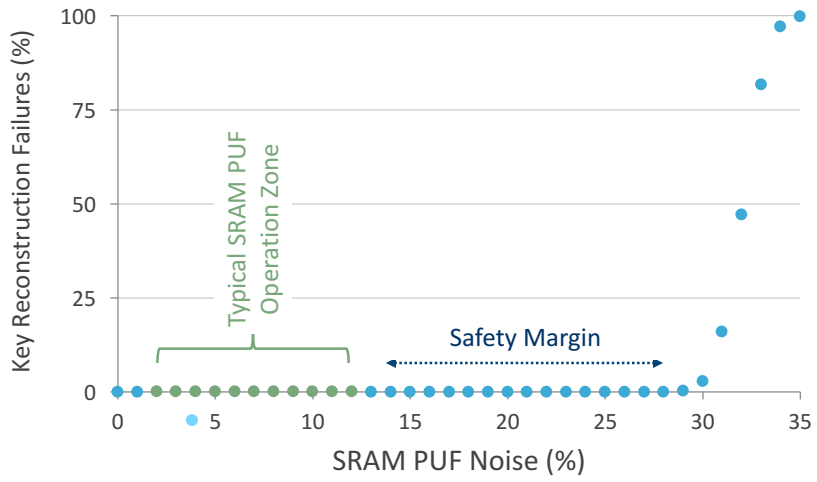


Figure 11. The SRAM PUF key-reconstruction algorithm tested on simulated data. For each noise level from 1% to 35%, 100,000 SRAM PUF responses were simulated. For each simulation the key was reconstructed and compared with the reference key. The results show that at 28% SRAM PUF noise or less, key reconstruction never fails. Note that in typical situations, SRAM PUF has a noise level in the 2%-12% range.

Part C

Key Reliability

A key generation fails when the key extractor is unable to correct all the PUF response bit differences that simultaneously occur in a single evaluation. The key failure rate is the probability of this happening and should be very small for practical applications (typically smaller than 10^{-9}).

By combining the SRAM PUF anti-aging (Part A) with error correction algorithms as explained in the previous section, the Intrinsic ID SRAM PUF implementation is designed to reconstruct a key with a failure rate of less than 10^{-9} even under extreme conditions when the noise in the PUF would rise up to 25%.

This has been tested by running 10 billion 256-bit key reconstructions on simulated SRAM PUF data (of 1KB) where 25% noise was introduced. The results indicated only four key reconstruction failures. Note that in a realistic situation the SRAM PUF noise levels will often be lower than 10% and the key reliability will even be much better and the failure rate much less than 10^{-9}.

Another way to examine key failure is to randomly flip more and more SRAM bits in the SRAM PUF response and see how long the key extractor is able to recreate the original key. Figure 11 shows the results of a test on simulated data. For each noise level from 1% to 35%, 100,000 SRAM PUF responses were simulated. For each simulation the key was reconstructed and compared with the reference key (from the reference SRAM PUF response). The results show that at 28% SRAM PUF noise or less, the key reconstruction never fails. Note that in typical situations, SRAM PUF has a noise level in the 2%-12% range.

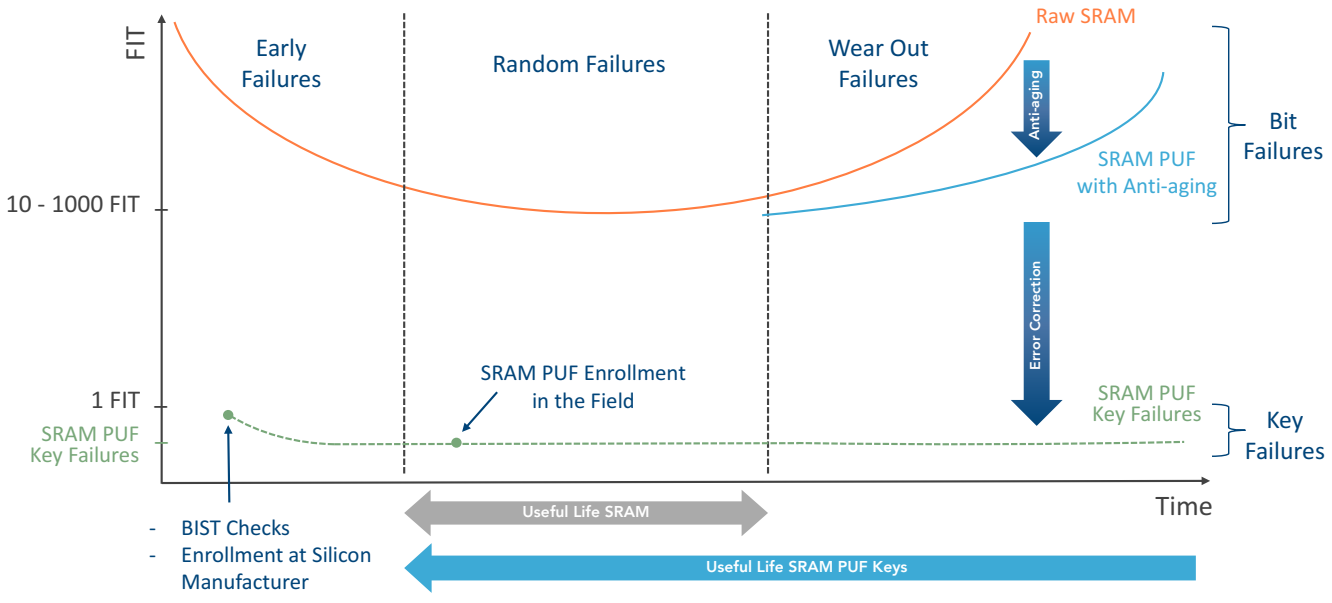


Figure 12. Failures In Time (FIT) for raw SRAM bits, SRAM PUF with anti-aging, and key reconstruction, showing the reliability of SRAM PUF as a key storage medium.

The reliability of SRAM PUF as key storage medium is illustrated in Figure 12. Intrinsic ID key extractor IP is typically delivered with Built-In Self Test (BIST). This can be logic BIST to check if the gates in the module are working correctly, and/or SRAM PUF BIST to check the operation of the bits in the SRAM PUF. These BIST are typically used in an early stage to detect early failures.

Note that there are several options for SRAM PUF enrollment. This can be done at the silicon manufacturer immediately after production, or later in the field. Hardware, software and hybrid implementations of the Intrinsic ID key extractor exist[10].

As explained in Part A, the main degradation effects contributing to wear-out failures of raw SRAM don't affect the working of the SRAM PUF or can be avoided by using the aging countermeasure. Hence, as indicated in Figure 12, an SRAM PUF with anti-aging has a longer useful life than SRAM under typical use. Furthermore, bit failures are corrected for by the error correction algorithm and don't lead to failure of the key reconstruction. E.g. in case of a 256-bit key, more than 2000 random bits (25%) need to fail before a key reconstruction starts failing. The error correction of the SRAM PUF is designed such that, when 2,000 random bits flip with respect to the enrollment value, the key failure rate is still less than 10^{-9}.

In applications where the device is powered every hour and the key is reconstructed, the failure rate for an SRAM PUF key is still lower than 1 Failure In Time (FIT). In most more realistic situations, where the SRAM PUF has a noise level lower than 10% and the device is powered only a few times a day (e.g. financial transactions) or a few times in its life (e.g. sensors), the key failure rate will be orders of magnitude lower than 1 FIT.

In Conclusion

SRAM PUF-based device security is proven to be a reliable method to store cryptographic keys for very long time periods and under a wide variety of circumstances. The combination of anti-aging and sophisticated error-correction techniques makes SRAM PUF technology such as Intrinsic ID's the most reliable component in an IC, enabling reliable key reconstruction even under worst-case conditions and ensuring a 25-year lifetime. This reliability makes SRAM PUF suitable for a wide range of use cases that have very strict requirements, such as automotive and military applications. Diverse SRAM PUF-enabled products have achieved EMVCo Visa and CC EAL6+ certification, and have been vetted by U.S. and EU governments.

[10] https://www.intrinsic-id.com/products/

Glossary

CC EAL6+	Common Criteria Evaluation Assurance Level 6 augmented: resistance to attackers with high attack potential
CMOS	Complementary Metal-Oxide Semiconductor
EEPROM	Electrically Erasable Programmable Read Only Memory
EMV	Europay, MasterCard and Visa standard for inter-operation of IC cards, for authenticating credit and debit card transactions
EMVCo, Visa	Certificate issued by EMVCo for secure payment transactions (based on EMV specifications and related testing processes)
FIT	Failure In Time (a failure rate of 1 per billion hours)
HCI	Hot Carrier Injection
HW	Hardware
IC	Integrated Circuit
IoT	Internet of Things
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
NBTI	Negative Bias Temperature Instability
PMOS	Positive channel Metal-Oxide Semiconductor
PUF	Physical Unclonable Function(s)
SRAM	Static Random Access Memory
TDDB	Time-Dependent Dielectric Breakdown
Vdd	Positive supply voltage of the SRAM cell

info@intrinsic-id.com www.intrinsic-id.com INTRINSIC ID