A sequence dependent challenge-response PUF using 28nm SRAM 6T bit cell
Supreet Jeloka1, Kaiyuan Yang1, Michael Orshansky2, Dennis Sylvester1, and David Blaauw1
1University of Michigan, Ann Arbor, MI, 2University of Texas, Austin, TX

Abstract
Conventionally, SRAM PUFs are only used for chip ID. The proposed sequence dependent PUF expands the challenge-response space of an SRAM PUF by an order of row^4(sequence length)^3, making it suitable for authentication. In addition, it has a sequence dependent non-linear behavior making it more immune to machine learning attacks. In 28nm, the 64x64 SRAM-based PUF has a bit area of 388F^2 with energy ranging from 30J/bit - 880J/bit at 0.6V. It also provides high throughput, from 2.2Gbps to 6.8Gbps at 0.9V.

Introduction
Secure electronic devices have become more ubiquitous than ever before, from banking infrastructure to critical communication links. This creates enhanced security risks of setting up trustworthy communication links and data privacy, raising the need for hardware level security features. Physically Unclonable Functions (PUFs) have become a popular hardware primitive for authentication. A PUF is like a device fingerprint, which produces a unique response on each chip for the same challenge. PUF can be based on delay variations [1], a bi-stable element resolving to one state [2,3], or measurement of other physical attributes on-chip. Existing PUFs are linear and use only a single access of the underlying hardware structure (SRAM in our case). This makes it easier to model and learn with Machine Learning (ML) algorithms. The motivation for the proposed PUF design is to compose a challenge/response from a sequence of accesses that makes the response depend on the order of the sequence, making it much more difficult to learn with ML algorithms and simultaneously increasing the challenge-response space. The proposed SRAM-based PUF has a bit area of 388F^2, high throughput of 1.1Gbps at only 97J/bit energy, and is demonstrated under ML attack, showing increased attack resistance for higher order sequences.

Proposed PUF design
Fig. 1 shows a conventional SRAM PUF cell, which is usually based on the start-up value at power-up. The start-up value is determined by the relative strength of the two inverters in the cross-couple. The challenge-response space of a conventional SRAM PUF is equal to number of bits in the memory, and it can only be used as a chip ID.

The proposed PUF is also an SRAM-based design, but is independent of the power-on state. The basic concept is to connect any two bit cells in the SRAM with complementary data initialization by simultaneously asserting their word-lines. The value they resolve to depends on the relative strength of all the 12 transistors of the two bit-cells, and their initial value. To illustrate, Fig. 2 shows a small array with checkerboard initialization. Word lines WL1 and WL4 are asserted and the bit cells in the two rows with opposite states fight over the bit lines in each column, and resolve to a single value. Consequently, the challenge-response space is increased as we can choose any two rows from the array. For n rows, we have n choose 2 = n(n-1)/2 choices of pair-wise row selection.

In addition, a sequence of such pairwise shortcomings can be applied, as shown in Fig. 3. Starting from the same initialization, the top sequence connects rows 1&2, followed by rows 2&3, represented as ((1,2);(2,3)) whereas the bottom sequence is ((3,2);(2,1)). The darker bits in the figure overwrite the gray bits. Even in this simplified example, the two differently ordered sequences result in a final value in row 1 or row 3 with a hamming distance (bit difference of 2). Therefore, both row selection and the order (permutations) in which the selected rows are sequentially connected determine the response of the PUF, making it more difficult to learn by ML algorithms.

If we build a sequence using 3 rows out of n rows, we have n choose 3 = (n^3-n)/6 choices of selecting rows a, b, and c and then we can connect them in 6 different permutations. Fig. 4 shows two such permutations. We can similarly make higher sequence length patterns. In general, number of challenge-response-pairs (CRPs) for an n-row m-column array with a sequence length of r is [n*(n-1)*...*(n-r)]/m. In summary, the response of the proposed PUF depends on SRAM initialization, sequence length, and sequence order. The sensitivity to each of the 3 factors is analyzed and measured using a 28nm test chip.

Fig. 5 shows the implementation of a single column of the array. The same signal is typically used to both equalize and precharge the bitlines in a conventional SRAM. Here the precharge signal is split from the equalization signal. We keep precharge signal ‘preb’ asserted during WL assertion to remove WL-assertion timing-mismatch from causing systematic row bias. Also, equalization (eqb) is asserted longer than ‘preb’ to avoid any bias between BL and BLB due to the precharge transistors or any other column circuit mismatch.

Analysis & measured results
A 4kb (64x64) PUF array was designed in a 28nm FDSOI CMOS process, using standard push-rule 6T SRAM bit cells. Fig. 6 shows measured data for the sensitivity of the response to the initialization value. The baseline 2-row experiment (a,b) is performed with both data background and flipped data background initialization. The inter-initialization hamming distance (HD) is significant, with a mean of 38%. Hence, new initialization backgrounds, lead to unique response increasing the possible challenge space. Fig. 7 shows the measured sensitivity of the PUF response to sequence order. For this we run three different sequence lengths and measure the inter-sequence HD which is the bit-difference between the responses for the same set of rows interacting on the same chip, but with different orderings. The inter-sequence HD increases with sequence length, and has a mean of 30% for sequence length of 5, showing non-linear behavior similar to that of a state machine and making it more difficult for ML to attack. In addition, for a sequence length of ‘r’, we get an expansion of r-factorial in CRPs.

Fig. 8 shows the measured inter-PUF HD for 2-row, as well as 3 - 5 sequence lengths. The inter-PUF HD is very close to the ideal 50% for all sequence variations, with the mean for the baseline 2-row at 49.5%. In Fig. 9, we applied machine learning attacks on the proposed PUF using the open-source package LIBLINEAR & LIBSVM [7]. Dual support vector classification gave the best prediction results. Despite SRAM being in general a “weak” PUF, the proposed model is able to resist attacks, and its resistance improves as we increase sequence length. Even a smaller sequence length of 3 is >10× more resistant than a 64-bit arbiter PUF [5][4]. The prediction error per column for 10k training patterns for length-5 sequence is measured at 0.106. For a 64-bit output this prediction translates to 7.6-8.4 accuracy.

PUF repeatability is measured as bit-error rate (BER). At nominal operation of 700mV/25°C the BER is 3.17%. The variation of BER across Vdd & temperature are shown in Fig. 10 with the golden data recorded at 700mV/25°C. The BER can be lowered further by using majority voting/masking techniques. Figs. 11 & 12 show the measured circuit performance of the proposed PUF. Energy is optimal at 0.6V supply, and is only 30fJ/bit for the 2-row case and 88fJ/bit for length-5 sequence. The throughput for 2-row can reach 6.8Gbps, as frequency is similar to that of a regular SRAM access. Fig. 13 shows the die photo, and Table 1 compares the proposed PUF with other designs. This work allows reuse of already available SRAMs, benefiting from its high density and throughput. The design demonstrates non-linear behavior, making it more difficult for ML to learn. It is tested with ML attacks showing that the sequence length of 5 is >30× more resistant than the 64-bit arbiter.

Acknowledgements
STMicroelectronics is gratefully acknowledged for IC fabrication.

References
[1] J.Lee et al., VLSI, 2004
Fig. 1. Proposed PUF basics

Fig. 2. Proposed PUF – Small array example

Fig. 3. Sequence dependence example

Fig. 4. Different length sequence and permutation example

Fig. 5. Single column circuit & timing waveform

Fig. 6. Measured Hamming Distance (HD) between complementary initialization

Fig. 7. Measured HD between sequence permutations

Fig. 8. Measured Inter-PUF HD

Fig. 9. Machine learning prediction errors for different sequence lengths using support vector classification

Fig. 10. Measured native bit-error-rate versus supply voltage & temperature with golden value at 700mV, room temperature

Fig. 11. Measured energy versus supply Vdd

Fig. 12. Measured throughput versus supply Vdd

Fig. 13. Die photo and PUF memory layout

Table 1. Comparison with previous works