Innovative Dynamic SRAM PUF Authentication for Trusted Internet of Things

Pascal Urien
Telecom Paris
19 Place Marguerite Perey 91120 Palaiseau, France
Pascal.Urien@Telecom-Paris.fr

***Abstract*—This paper presents innovative dynamic SRAM PUF (Physical Unclonable Function) authentication for micro-controller units (MCU) embedding static RAM. We introduce a low cost PUF extractor, designed to monitor the MCU power-up and to read SRAM content. Some SRAM cells, referred as flipping-bits, have a fix content dependant on the voltage rising time. Experiments, using double ramp power-up waveform, show that these cells switch at low voltage values (below 500mV) according to a distribution, which seems to be linear. Because the digital part of the MCU is not working when flipping-bits switch, we propose a simple SRAM dynamic authentication protocol, based on slope and square power-up waveform, whose goal is to detect cloned MCUs.**

***Keywords*—** PUF; SRAM*;* Security

I. INTRODUCTION

Physical integrity of electronic devices is a critical challenge for emerging cyber physical systems (CPS). We would like to detect non genuine micro-controllers (MCUs) used in Internet of Things (IoT) environments. According to the state of art [1][2], technologies based on *Physical Unclonable Function* (PUF), can be used to extract fingerprints from silicon chips embedding static RAM (SRAM).

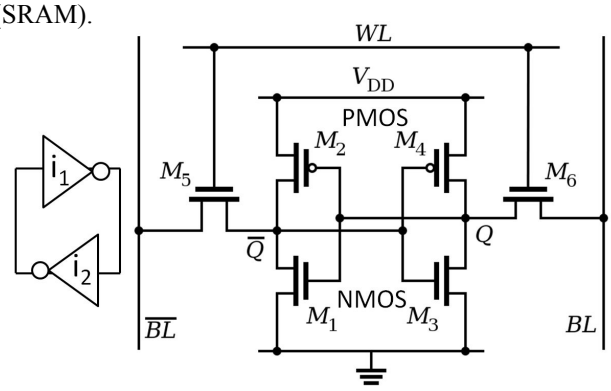

Fig. 1. SRAM memory cell architecture.

A SRAM memory cell (see figure 1) is designed with 6 CMOS transistors, and includes two inverters (i_1 and i_2) connected in series (i.e. head to tail). If these logical inverters have not identical electrical characteristics, the memory cell will switch to a fix value upon power-up. This effect enables to extract a set of bits from a SRAM memory, after its powering up. Typically these PUF bits are used to forge secret keys, associated to symmetric or asymmetric cryptographic algorithms. The underlying prerequisite is that the content of the SRAM keep secret, in order to avoid its knowledge after power-up. For example it is possible to download SRAM probe in a MCU, and thereafter to collect the memory fingerprint. Classical SRAM PUF acts as static authentication procedure. A non genuine chip may store a copy of this fingerprint, for further malicious use. The target of our work is to find and use power-up waveforms that modify in a reliable and reproducible way the SRAM fingerprint, in order to perform dynamic authentication. The key is the physical SRAM. The challenge is a power-up waveform that a non-genuine chip cannot guess. In this paper we present waveforms that modify SRAM PUF bits at low voltage, for which the MCU is not operational.

This paper is organized according to the following outline. Section 2 introduces SRAM PUF. Section 3 describes a simple SRAM PUF extractor and introduces flipping-bits. Section 4 presents *double slopes* power-up waveform (Sy) use to determine flipping-bits voltage threshold, and experimental results. Section 4 describes *slope & square* (Rx) power-up waveforms. Section 5 presents a dynamic authentication protocol working with two Rx signals. Section 6 concludes this paper.

II. ABOUT SRAM PUF (PHYSICAL UNCLONABLE FUNCTION)

Given f_i is the transfer function $V_{out}=f(V_{in})$ of an inverter i (see figure 1), the stationary state implies that $f_1(x) = f_2^{-1}(x)$. If the two inverters are similar, then $f(x) = f^{-1}(x)$ this is what defines three solutions (see figure 2, left part). When the SRAM cell is switched, the output voltage is either V_L (logical 0) or V_H (logical 1). Due to NMOS and PMOS symmetry, $V_{DD}/2$ (i.e. half of inverter feeding voltage) is always a possible solution.

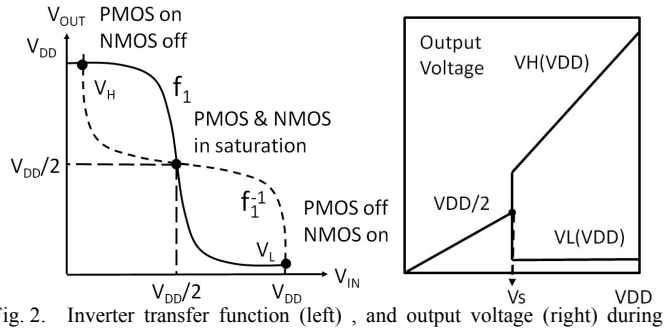

Fig. 2. Inverter transfer function (left) , and output voltage (right) during power-up

During the powering-up (see figure 2, right part) the SRAM output voltage follows $V_{DD}/2$, until the inverter gain is enough for switching towards V_L or V_H. For *Fully-skewed* cell, i.e. highly mismatched SRAM, the cell always takes a fix value. The voltage (Vs) at which the transition occurs is dependent on the cell [3]. The paper [4] demonstrates voltage ramp effect on partially skewed SRAM cells. Given a ramp $V(t)=t.V_{DD}/T$ (slope=V_{DD}/T) flipping-bits are observed for slope high values. Flipping-bits are created by capacitance dissymmetry, while most of PUF bits are due to voltage threshold (V_{TH}) differences. In other words voltage ramps reproducibly switch the value of some SRAM cells at power-up. A test RAM chip was designed with 180nm technology, and simulated. Flipping-bits have been observed for T value less than 15ms.

III. A SIMPLE PUF EXTRACTOR

We focus on MCUs such as ATMEGA processors likely designed (according to several WEB references) with 180nm fabrication process. For example the ATMEGA8 chip is clocked at 12MHz and includes 8KB FLASH, 1KB SRAM, and 512B EEPROM. Our interest for this chip comes from the fact that it is used by popular SPI USBASP programmer, acting as root of trust for bare metal devices [5]. Furthermore multiple papers comment SRAM PUF extraction for AVR processors [6][7].

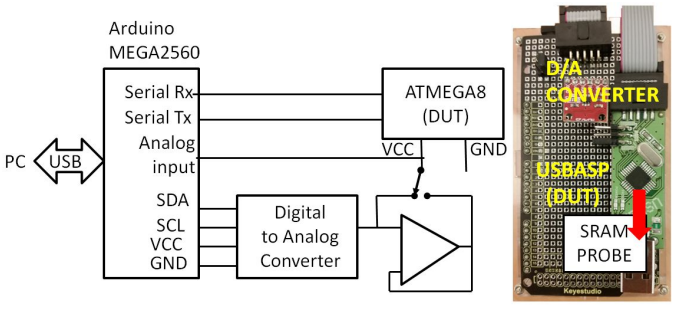

Fig. 3. A low cost PUF-Extractor

The low cost PUF-Extractor is illustrated by figure 3. The *device under test* (DUT) is running a *SRAM Probe* software, which communicates thanks to a serial link. It is power-up by a *Digital to Analog Converter* (DAC) with a 4096 bits resolution, whose output is optionally associated to an amplifier. The PUF-Extractor controls the power-up waveform (about four outputs/ms), and reads the SRAM content of the tested MCU. We previously detailed in [8] results for ATMEGA8 processor. We log, for each SRAM cell (k), the number of 1 occurrence for N power-ups $S^N{}_T(k)$. Some cells are always seen at 1 ($S^N{}_T(k)=N$) or 0 ($S^N{}_T(k)=0$), and define the *PUF-Domain*. The remaining cells are referred as *noisy* ($0<S^N{}_T(k)<N$). For example, for N=250, we get a set of measures $S^N{}_T(k)$ for slopes V_{DD}/T (V/ms), with T expressed in ms. The comparison of a measure $S^N{}_T$ to a reference $S^{Nr}{}_{Tr}$, is a set of tuples $(S^N(k), S^{Nr}{}_{Tr}(k))$. The *common PUF-Domain* is the set of tuples either equal to (N, Nr) or (0,0). Flipping-bits are the set of tuples (N,0) or (0,Nr). Noisy cells comprise the tuples (noisy, noisy), (noisy, 0/Nr) and (0/N, noisy). For low rising times, there are no flipping-bits [4]. We choose a reference $S^{250}{}_{1024}(k)$, i.e. a 1024 ms voltage rising time and 250 power-ups.

1	2	4	8	16	32	64	128	256	512
360	373	379	380	353	330	240	140	34	0

Fig. 4. Power-up rising time (in ms, upper line) and number of associated flipping-bits (lower line)

Figure 4 shows some experimental results. For "fast" rising time there are about 360 flipping-bits; this number is divided by two for T=100ms, and reaches zero for T less than 512ms. Figure 5 shows a graphical representation of flipping-bits (colored in red) for $S^{250}{}_{64}$ versus $S^{250}{}_{1024}$. The *PUF common domain* is colored in green (always 1) and yellow (always 0). Noisy cells are colored in white.

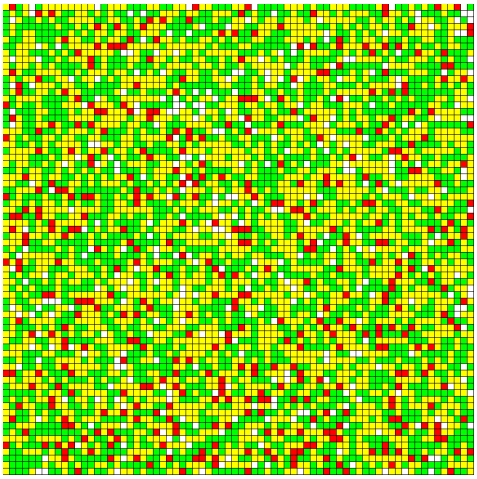

Fig. 5. Flipping-bits (in red) for $S^{250}{}_{64}$ versus $S^{250}{}_{1024}$. Always 1 (green), Always 0 (yellow), noisy (white).

IV. DOUBLE SLOPES SY WAVEFORM

A voltage ramp modifies flipping-bits SRAM cells, whose content is $b^0{}_k$ (either 0 or 1) for a slope S_0, and $b^1{}_k= (1-b^0{}_k)$ for a slope S_1. According to figure 4, we observe about 200 flipping-bits for a $V_{DD}/64$ slope.

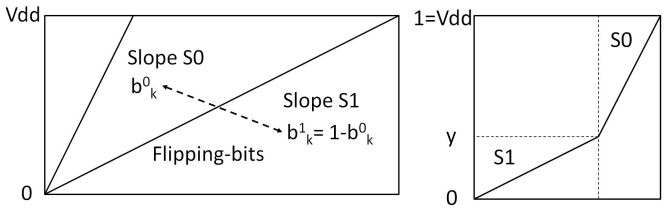

Fig. 6. The double slope, Sy waveform

We would like to use flipping-bits for dynamic authentication, the voltage waveform being a physical challenge. For this purpose we build a voltage waveform that mixes two slopes (Sy). For voltage values ranging from 0 to y we use slope S1, and for voltage values ranging from y to 1 (1 meaning V_{DD}=5V) we use slope S_0 (see figure 6).

At the end of such power-up, we observe a cell content b^y_k. Obviously for y=0 (slope S_0) we should find b^0_k, and for y=1 (slope S_1) we should find b^1_k (i.e. 1-b^0_k). This waveform (Sy) is useful only for flipping-bits; otherwise the y parameter would have no effect on memory cells content.

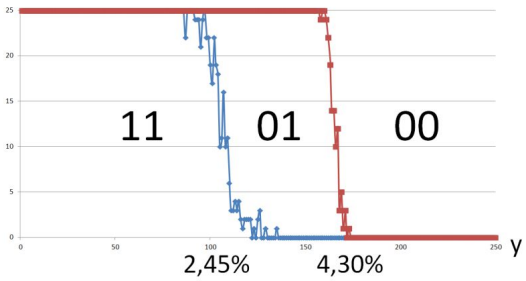

Fig. 7. Switching voltage (Vs) for Flipping-bits

We want to find the y switching value (Vs), for which a k SRAM cell content switches from S_0 output to the S_1 output, and to observe the threshold distribution Vs(k). In order to reach this goal we performed 250x25 measures, with S_0=S_{64} (V_{DD}/64 V/ms) and S1=S_{1024} (V_{DD}/1024 V/ms), for y values starting from 1 bit resolution (y=1/4096) to 250 bits resolution (y=250/4096), with an increment step of one bit resolution (1/4096, about 1.2mV=5V/4096). So the same y is associated to N=25 measures; we store the number of 1 occurrences, ranging from 0 to 25, in B_k(y) records. Figure 7 shows B_k(y) curves for two flipping-bits. Obviously the region around the threshold is noisy; a small y interval is observed to decrease B_k(y) from 25 to 0. The two threshold points (Vs) are about 2,45% and 4.30%.

As illustrated by figure 7, flipping-bits are associated to different switching values Vs;. n flipping bits, with distinct Vs defines n+1 areas. For example in figure 7 two flipping-bits define three regions from left to right: 11 01 and 00.

Figure 8 presents the switching voltage distribution. The S_0 *PUF-Domain* is on the left (green=1 and yellow=0). The S_1 domain is on the right (red color). The frontier between the two domains is noisy (white points). Roughly speaking, these domains are separated by a line, for y ranging from VsMin to VsMax. So for y within [VsMin, VsMax], the Sy signal selects a set of flipping-bits ($\{b^y_k\}$) either in the S_0 ($\{b^0_k,\}$) or in the S_1 ($\{b^1_k,\}$) domain.

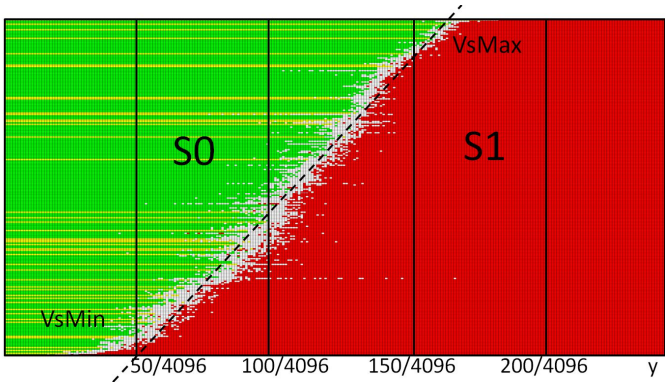

Fig. 8. The swtiching voltage Vs(k) distribution

The parameter y is on the abscissa. The memory cells are on the ordinate. Horizontal line represents the values of a flipping-bit k (1= b^0_k= green, 0= b^0_k = yellow, noisy= white, red= inverted= b^1_k) according to the parameter y, which varies from 1/4096 to 250/4096. A column represents the values of the flipping-bits (1= b^0_k= green, 0= b^0_k = yellow, noisy= white, red= inverted= b^1_k) for a given y.

The addresses of the flipping-bits are sorted by increasing values of threshold voltage (Vs). Figure 8 shows a quasi-linear distribution for switching voltages as function of y.

The average threshold value is about 225mV. So flipping-bits state is determined for low voltage values, less than 500mV. For such voltages the processor is not working. In these conditions, we believe that it should very difficult to design clones, able to sample voltage, in order to set the flipping-bits content accordingly. Without the accurate knowledge of the Sy signal, the probability of selecting a right set of n flipping-bits is 1/(1+n) (as illustrated by figure 7), what allows to define many protocols able to detect cloned devices.

V. Slope & Square Rx Waveform

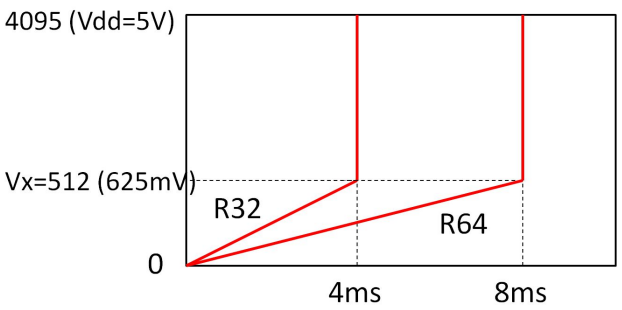

Fig. 9. Slope & Square Rx Waveforms

Figure 8 shows that flipping-bits are created at low voltage Vs. We define Rx waveforms (as illustrated by figure 9), which comprise two parts: a slope Sx from 0V to Vx=625 mV (i.e. 4096/512 * V_{DD}), and then a fast rise from Vx to V_{DD}. The x parameter is expressed in ms, and the associated slope is Vdd/x in Volt/ms.

For fast rising time, Rx waveforms should create flipping-bits. For low rising time, Rx waveform should not create flipping-bits. Therefore we expect that two Rx waveforms such as R_{64} and R_{1024}, can be used in a random way, in order to create SRAM PUF, with and without flipping-bit. Because processors are not working at low voltage, malicious software cannot guess the induced memory fingerprint.

VI. A Simple Dynamic SRAM PUF Authentication Protocol

We perform a single measure of SRAM content, powered by the Rx (R^1x) signal, and compare the results with references based on 250 measures ($S^{250}x$). For these references the voltage waveform is a single ramp, ranging from 0 to V_{DD}, whose slope is V_{DD}/x in Volt/ms (x in millisecond).

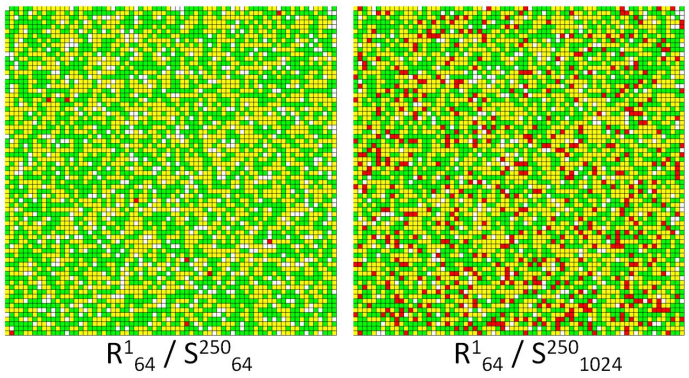

Fig. 10. R^1_{64} memory fingerprint versus references S^{250}_{64} and S^{250}_{1024}.

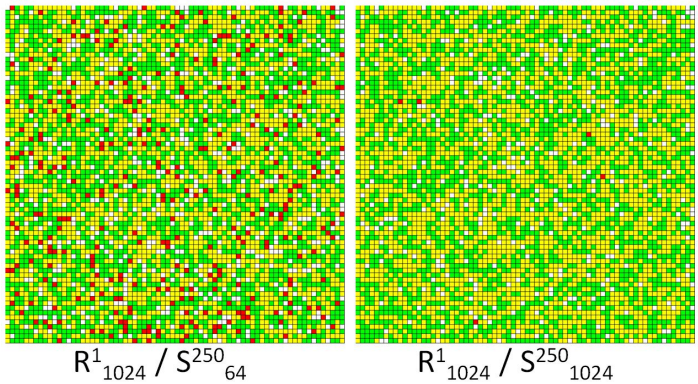

Fig. 11. R^1_{1024} memory fingerprint versus references S^{250}_{64} and S^{250}_{1024}.

Figures 10 and 11, show the flipping-bits (i.e. memory fingerprint mismatch) induced by the R^1x waveforms with the two $S^{250}x$ references. Rx fingerprints are closed to $S^{Nr}y$ references for x=y, while flipping-bits (about 200) creates many errors for x#y.

Finally we deduce from these measures the following simple dynamic PUF authentication protocol:

"Luke has SRAM contents for two Rx power-up waveforms: R_{64} and R_{1024}. The R_{64} SRAM content has about 200 flipping-bits. These contents are determined at low voltage (512mV), before Luke and Vador have a digital life. Vador and Leia know these SRAM contents. In order to authenticate Luke, Leia uses power-up waveforms either R_{64} or R_{1024}, in a random order. Luke will always produce the right response, while Vador will make a random choice; so after n tries so probability of zero error for Vador will be $1/2^n$."

VII. Conclusion

In this paper we propose a dynamic SRAM PUF authentication protocol based on flipping-bits created by dedicated power-up waveforms at low voltage, for which the micro controller unit is not working. We believe that it should be an efficient way to prove MCU cloning in Internet of Things frameworks.

VIII. References

[1] De Holcomb et all, "Initial SRAM state as a fingerprint and source of true random numbers for RFID tags", RFID Security, 2007

[2] Herder, C et all, "Physical Unclonable Functions and Applications: A Tutorial", in IEEE Volume 102, Issue: 8, Aug. 2014

[3] MSc THESIS Modeling SRAM Start-up Characteristics For Physical Unclonable Functions, Apurva Dargar

[4] Abdelrahman T. Elshafiey et all, "The effect of power supply ramp time on SRAM PUFs", IEEE MWSCAS 2017.

[5] Urien, P; " Integrity Probe: Using Programmer as Root Of Trust For Bare Metal Blockchain Crypto Terminal", Fifth International Conference On Mobile And Secure Services, MobiSecServ2019

[6] Mikhail Platonov, "SRAM-Based Physical Unclonable function on an Atmel Atmega Microcontroller", Master's thesis, 2013

[7] Christoph Lipps et all, "Proof of Concept for IoT Device Authentication based on SRAM PUFs using ATMEGA 2560-MCU", ISDIS 2018

[8] P. Urien, "Innovative ATMEGA8 Microcontroler Static Authentication Based on SRAM PUF," 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 2020