Recombination of Physical Unclonable Functions

Meng-Day (Mandel) Yu
myu@verayo.com
Verayo, Inc
San Jose, CA

Srinivas Devadas
devadas@mit.edu
MIT
Cambridge, MA

Abstract: *A new Physical Unclonable Function (PUF) construction is described, by treating silicon unique features extracted from PUF circuits as "genetic material" unique to each silicon, and recombining this chip-unique material in a way to obtain a combination of advantages not possible with the original PUF circuits, including altering PUF output statistics to better suit PUF-based key generation and authentication.*

Keywords: FPGA and ASIC authentication; key generation; rights management; supply chain TRUST

Introduction

Over the past decade, the concept of net-enabled operations has become a cornerstone of our national defense strategy. An issue of growing concern is the possibility of cyber attacks that allow an adversary to obtain sensitive information or possibly take either partial or full control of remotely operated systems. Silicon-based Physical Unclonable Functions (PUFs) serve as a critical design primitive to secure these microelectronics systems [1]. This paper describes a new PUF construction by treating silicon unique features extracted from PUF circuits as "genetic material" unique to each silicon device, and recombining this chip-unique material in a way to obtain advantages not present with the original PUF circuits. These advantages include:

- *De-biased PUF Outputs* that pass NIST Statistical Tests for Randomness, resulting in excellent raw material for key generation and for authentication.
- Enabling a *Fully-Challengeable Real-Valued PUF*, supporting both 1) *a large challenge space* (e.g., 2^{64} bits) suitable for authentication based on challenge / response pairs; and 2) *real-valued outputs* suitable for soft decision error correction, which increases environmental stability and reduces implementation complexity for key generation.
- *Realizable in both FPGAs and ASICs*, reducing risks of ASIC deployments via rapid FPGA prototyping and emulation, and offering protection for FPGA as well as ASIC-based devices and systems.

Introduction to Physical Unclonable Functions

The use of Physical Unclonable Functions (PUFs) as a silicon-unique root of trust was first proposed by researchers at MIT [2], enabling authentication based on chip-unique responses as well as generation of chip-unique cryptographic keys. Multiple silicon-based PUF circuits have since been realized. An *arbiter-based PUF*, which has a large challenge space (e.g., 2^{64} bits or more) was prototyped in an ASIC [3], and a variant was commercialized in the form of an "unclonable" RFID IC [4], opening the door for use of PUFs for authentication with virtually unlimited challenge and response pairs. A *ring-oscillator PUF*, was built and tested in [1]. Use of initial SRAM state as a PUF was explored and tested in [5].

Introduction to Recombination

Recombination builds on prior work in Physical Unclonable Functions, and can be applied to various types of silicon PUFs as well as other noisy pseudo-random sources, including biometric sources. Recombination can also be applied to a system containing different types of PUFs or a system containing both PUF and biometric sources. In genetic engineering, recombination refers to the process where genetic material is rearranged and joined to other genetic material. The resulting output may possess genetic combinations or characteristics not previously present. In a similar way, chip-unique characteristics extracted from a PUF circuit (or from different types of PUF circuits and biometric sources) may be recombined to produce characteristics not natively present in the original circuit, e.g., outputs with different statistical characteristics, with bias removed. This is achieved by treating silicon-unique features extracted by PUF circuit as "genetic material" that is then recombined, subject to an input stimulus (i.e., challenge), as shown in Figure 1.

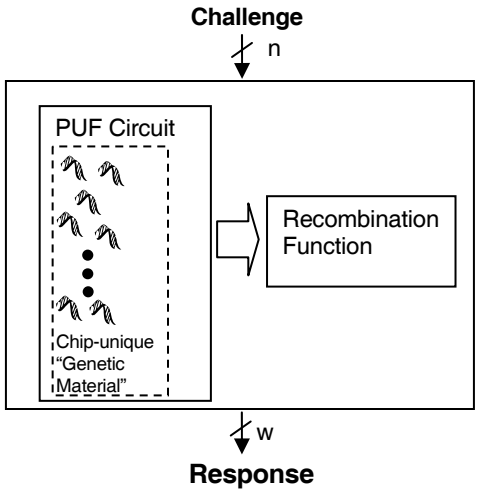

Figure 1: Using recombination to alter PUF characteristics

For the purposes of this paper, the discussion shall be focused on use of recombination on silicon-based PUFs, and in particular, oscillator PUFs. Oscillator PUF in [1] has an average 0^{th} order (dc) bias value of 46.15%. By using a simple recombination function that groups oscillators into *n* stages, and ensuring that the output for each stage s_i statistically results in a bias neutral output, the sum of all stages *S* is then also statistically bias neutral.

- $$S = \sum_{i=0}^{n-1} s_i$$

An example of this simple construction uses

- $s_i = (c_i * 2 - 1) \times (f_{2i} - f_{2i+1})$

where

- $0 <= i < n$, and
- $c_i \in \{0,1\}$, is an instantaneous challenge bit that is a result of a seed challenge processed by a mixer (e.g., a linear feedback shift register with a primitive polynomial or a hash function).

This is shown in Figure 2. *S* is a signed quantity.

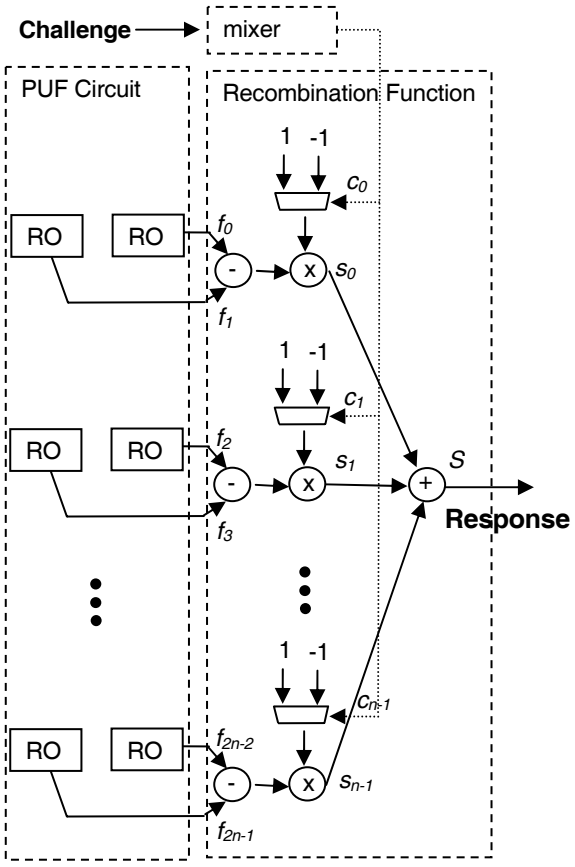

Figure 2: Example of recombined oscillator PUF

More complex examples include use of Response to perturb Challenge (output feedback) or affect subsequent Response values (output feed forward), use of nested recombination functions, use of recombination functions that vary with prior Response values, or the use of XORs, majority functions, or other logical or mathematical functions. As an example of output feed forward recombination, values can be grouped in groups of 6, with the first two bit values pointing to one of the four subsequent values as the (final) Response value:

- Response value = (S_1,S_0 == 0,0)? S_2: (S_1,S_0 == 0,1)? S_3: (S_1,S_0 == 1,0)? S_4: S_5;

As a part of the design derivation process, various recombination functions were first emulated in software, with oscillator PUF data obtained in hardware from Xilinx Virtex-4 LX25 devices. Table 1 shows randomness test results by applying recombination to PUFs in four representative LX25 devices, and analyzing 100 million recombined PUF output bits per device. The recombination structure shown in Figure 2 was used, with a 64-bit challenge and 64 ring oscillator pairs. The 100 million bits for each device was derived from the same starting 64-bit seed challenge that was loaded into an LFSR to obtain subsequent challenges. Success rates for each of the 15 NIST tests across the four chips were comparable with success rates derived from a NIST-recommended set of random numbers (last column). The results show that a recombined oscillator PUF has a negligible dc bias, since a dc bias of more than 51% or less than 49% would result in low NIST test pass rates (e.g., in the single digit percentages). The results contrast with the native dc bias of 46.15% found in oscillator PUF in [1], which readily fails NIST Statistical testing. Recombination, by this measure, produced better raw material for use in key generation and authentication, producing a PUF bias that is negligible based on NIST Statistical Testing, and is superior to the 46.15% bias found in an oscillator PUF in [1], 23% bias found in an early version of an arbiter PUF in [3], and the approximately 46% bias found in a memory PUF in [5].

Fully-Challengeable Real-Valued PUF

In addition to de-biasing PUF output, the recombined PUF in Figure 2 is also a *Fully-Challengeable Real-Valued PUF* supporting *both* of the following features:

- *a large challenge space* (e.g., 2^{64} bits), suitable for authentication based on challenge / response pairs; and
- *real-valued outputs* suitable for soft decision error correction, to increase environmental stability and reduce complexity for key generation. (The msb of *S* in Figure 2 indicates a bit polarity of 1 or 0, and the remaining bits of *S* indicate strength of that bit.)

It is difficult to produce both of these characteristics with other PUF implementations described in existing open literature. Arbiter PUFs (with multiple arbiters and output processing) in [1, 3] for example, have a large challenge space but natively do not produce real-valued outputs of sufficient resolution (e.g., at least 4 bits), thus complicating error correction. Neither the oscillator PUF realized in [1] nor the memory PUF as described in [5] has a large

Table 1: NIST Statistical Tests for Randomness success ratio for recombined PUF output bits

Statistical Test	Block/Template Length	Success ratio (chip #100)	Success ratio (chip #101)	Success ratio (chip #102)	Success ratio (chip #103)	Reference bitstream[1]
Frequency	-	99%	99%	98%	99%	98%
BlockFrequency	128	100%	100%	99%	99%	97%
CumulativeSums	-	99% - 99%	99% - 100%	97% - 98%	99% - 99%	98% - 99%
Runs	-	97%	99%	100%	99%	100%
LongestRun	-	100%	100%	99%	99%	97%
Rank	-	100%	98%	100%	100%	100%
FFT	-	100%	100%	100%	100%	100%
NonOverlappingTemplate	9	94% - 100%	95% - 100%	95% - 100%	95% - 100%	95% - 100%
Overlapping Template	9	98%	98%	99%	98%	97%
Universal	-	97%	98%	100%	96%	100%
ApproximateEntropy	10	100%	99%	99%	99%	100%
RandomExcursions	-	98%-100%	97% - 100%	97% - 100%	98% - 100%	98% - 100%
RandomExcusionVariant	-	97% - 100%	97% - 100%	97% - 100%	96% - 100%	93% - 100%
Serial	16	99% - 99%	99% - 100%	99% - 100%	98% - 98%	98% - 100%
LinearComplexity	500	100%	99%	99%	99%	100%
Cumulative p-values		100% (188/188) pass	100% (188/188) pass	100% (188/188) pass	100% (188/188) pass	100% (188/188) pass
Cumulative proportions		99% (187/188) pass	99% (187/188) pass	99% (187/188) pass	99% (187/188) pass	98% (184/188) pass

[1] From George Marsaglia's *Random Number CDROM*.

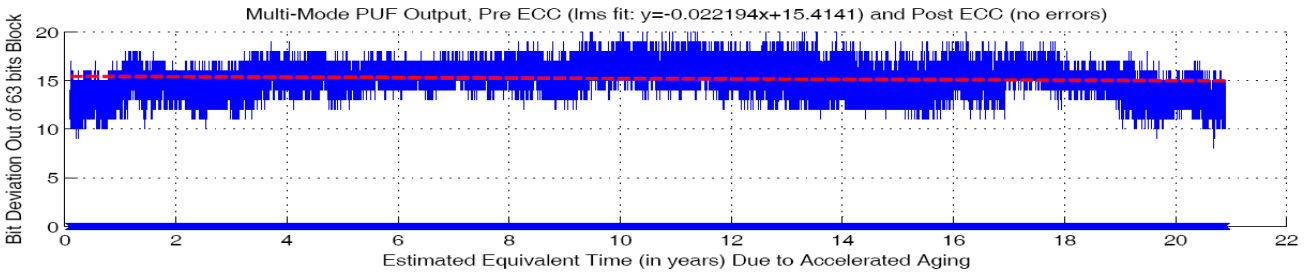

Figure 3: Accelerated aging data, key generation. Provisioning at 25°C, 1.0V, regeneration at 145°C, 1.05V. No errors for over 3 million regenerations. Worst case bit deviation for each time interval shown.

challenge space. When applied to the oscillator PUF, recombination results in a PUF with a large challenge space which was not present in the original oscillator PUF.

Note that model-building (e.g., using machine learning) to build a software clone within reasonable time [3] may be possible for PUFs using simple recombination functions. If resistance against model-building attacks is required, a more complex recombination function needs to be used.

To support key generation, the following components were added to the recombined PUF of Figure 2:

- Index-Based Syndrome Codec, supporting 0^{th} to 5^{th} order indices;
- 1x, 3x, and 5x repetition coder and majority decoder; and
- BCH(63) codec supporting t = 1 to 6.

These building blocks are described in detail in [6]. Index Based Syndrome Coding was used, taking advantage of recombined *real-valued* outputs, to achieve a 16x to 64x reduction in error correction code complexity through use of soft decision coding [6].

Figure 3 shows environmental test results in a representative device (Xilinx Virtex-5 LX50). Results show that raw bit flips (pre-ECC) for error correction block size of 63 do not deviate much for over 21 years of accelerated-life operation; the least square fit curve has a negligible slope of -.02. Test parameters for accelerated aging were derived from *MIL-STD-883G Method 1005.8 Steady State Life* as well as accelerated aging parameters obtained from Xilinx. Specifically, .70eV activation energy was assumed, at a confidence level of 60% (same assumptions as those used by Xilinx). Over 3 million error correction blocks were run, with no failures across 21 accelerated-life years, implying error correction block failure rate below .34 ppm. In-flux testing was done, with pre/post ECC output gathered while both temperature and voltage stresses were applied as a part of the accelerated life testing. Provisioning was at 25°C, 1.0V, and regeneration at 145°C, 1.05V, and thereby illustrating stability in key generation across a wide range of temperature and voltage conditions. Data obtained is consistent with the error free performance across millions of tests under mil-spec temperature and extreme voltage conditions in [6].

Multi-mode PUF

The key generation design tested in the previous section is actually a *Multi-mode PUF* design operating in key generation mode. This design is multi-modal in that it can operate in both *C/R authentication mode* (due to large challenge space achieved using recombination) and in *key generation mode* (recombined real-valued outputs results in stable and efficient error correction). To provide flexible multi-modal operation, this design supports multiple oscillator banks and a variety of recombination functions,

including the one shown in Figure 2 as well as 2/4/8 way XORs, multi-LFSR mixing, and other features. This design was successfully implemented and tested in a variety of devices, including Virtex-4 LX60 and Virtex-5 LX50.

An illustrative use case for Multi-mode PUF is shown in Figure 4, where PUF #1 is used in key generation mode and PUF #2 is used in C/R authentication mode; this is to provide a *layered* security approach that is stronger than conventional approaches of using only burn-in keys. PUF #1, by using different challenges, generates multiple root seeds; this is costly with conventional approaches. Conventional approaches also rely on *security by physical obscurity* and are broken if design is physically de-layered and visualized. A PUF is immune to such an attack, since all devices have the same layout and yet produce different keys. The device can be authenticated by an entity that knows the root seed (or a key derived from root seed) by sending a random nonce N as shown in the figure, which is encrypted on the device and can be decrypted on the client side. Optionally, a second multi-mode PUF operating in C/R authentication mode (PUF #2) can be added. PUF #2 offers an extra measure of security by relying on C/R authentication where C/R pairs are used only once and discarded. The response of PUF #2 is encrypted using the root seed or derived key and decrypted on the client side.

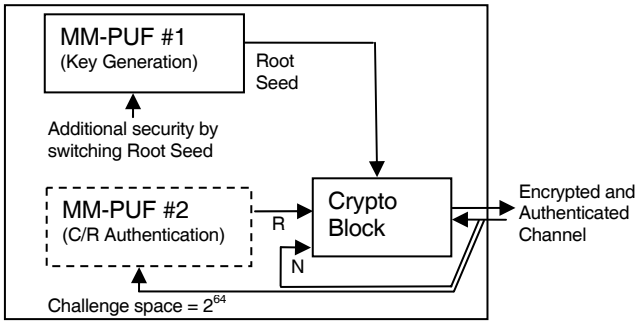

Figure 4: Using two Multi-mode PUFs to provide layered security

Realization in both FPGAs and ASICs

Recombination can be used to create a PUF with large challenge space (e.g., 2^{64} bits) in FPGAs. This is a capability that does not exist for FPGAs using other PUF implementations realized to date as found in open literature, including [1-5]. A memory PUF, for example, would be compromised if an adversary reads initial value of every memory location in the device; a dump of all memory bits in an FPGA is easy to obtain. By contrast, a recombined oscillator PUF has a large challenge space and also has a much greater routing variability, and variance could be made to be even greater (or the PUF can be reconfigured to disappear after use) through use of the partial reconfiguration feature found in Xilinx FPGAs. Another application for a PUF with a large challenge space is Trojan detection in *runtime* FPGA Firmware. In a Xilinx Virtex-4/5/6, the runtime bitstream can be read back, for certain classes of designs, using the Xilinx ICAP facility, and this bitstream can be mixed with a challenge to produce a response that is unique to the runtime bitstream and unique to the FPGA device, as shown in Figure 5.

With FPGA and ASIC using the same recombined PUF design, rapid FPGA prototyping and emulation can be used to reduce ASIC tape-out risks. In addition, de-biasing via recombination can increase yield and reduce risks when PUF is ported across different ASIC process nodes.

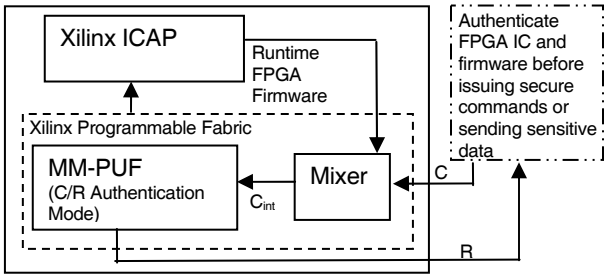

Figure 5: Multi-mode PUF used for FPGA firmware Trojan detection

Conclusion

A new construction of PUF using recombination presents several advantages, including: 1) de-biased PUF outputs; 2) support for C/R authentication as well as efficient and robust key generation; and 3) support for both FPGAs and ASICs. Future work includes applying recombination to a wider range of PUFs and biometric sources, and developing new applications.

References

1. S. Devadas, E. Suh, T. Ziola, "Physical Unclonable Functions and Applications to Device Authentication", Proc. Gov't Microcircuit Applications and Critical Technology Conference (GOMACTech), 2007.
2. B. Gassend, D. Clarke, M. van Dijk and S. Devadas, ''Silicon Physical Random Functions,'' Proceedings of the 9th ACM Conference on Computer and Communications Security, November 2002.
3. D. Lim, ''Extracting Secret Keys from Integrated Circuits,'' Master's thesis, EECS, MIT, 2004.
4. S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, V. Khandelwal, "Design and Implementation of PUF-Based "Unclonable" RFID ICs for Anti-Counterfeiting and Security Applications," Proceedings of the IEEE Int'l Conference on RFID, 2008.
5. D.E. Holcomb, W.P. Burleson, and K. Fu, ''Initial SRAM State as a Fingerprint and Source of True Random Numbers for RFID Tags,'' Proc. Conf. Radio Frequency Identification Security, 2007.
6. M. Yu, S. Devadas, "Secure and Robust Error Correction for Physical Unclonable Functions," IEEE D&T, Special Issue on Verifying Physical Trustworthiness of ICs and Systems, January 2010.