Secure Implementation of SRAM PUF for Private Key Generation.

Raja Adhithan Radhakrishnan
r.rajaadhithan@gmail.com
Society For Electronic Transactions and Security(SETS)

Abstract. This paper endeavors to securely implement a Physical Unclonable Function (PUF) for private data generation within Field-Programmable Gate Arrays (FPGAs). SRAM PUFs are commonly utilized due to their use of memory devices for generating secret data, particularly in resource-constrained devices. However, their reliance on memory access poses side-channel threats such as data remanence decay and memory-based attacks, and the time required to generate secret data is significant. To address these issues, we propose implementing n cross-coupled inverters in Verilog to generate n secret bits, followed by syndrome for error correction hardcoded in the hardware itself. This approach improves side-channel security and reduces time consumption, albeit at the expense of additional area utilization.

Keywords: SRAM PUF-LDPC-Hashing

1 Background

SRAM PUF is the more favored choice, primarily due to the widespread availability of SRAM resources in embedded devices. Typically, SRAM PUFs involve the extraction of raw random data from the SRAM Memory. This data is then processed through a fuzzy extractor to produce a key and helper data, followed by hashing to ensure high entropy. However, research indicates potential vulnerabilities within SRAM PUFs, such as susceptibility to data remanence decay attacks [4], stemming from their extraction from SRAM memory. Furthermore, the fuzzy extractor[1] [2] itself is prone to side-channel attacks.

Another prevalent option, the Arbiter PUF, presents advantages in terms of CRPs space compared to SRAM PUFs. Nevertheless, it's essential to note that Arbiter PUFs also face risks, including vulnerabilities to side-channel and modeling attacks [3]. While Arbiter PUFs excel in CRPs space, SRAM PUFs surpass them in terms of speed.

Given the focus of this paper on enhancing the speed and side-channel security of private key generation, it would be advantageous to explore improvements in SRAM PUF. Fig. 1 illustrates the existing approaches to SRAM PUF.

1.1 PROPOSED SCHEME USING SRAM PUF

In this setup, I've implemented a series of cross-coupled inverters to generate raw random data from SRAM PUF, followed by LDPC error correction that

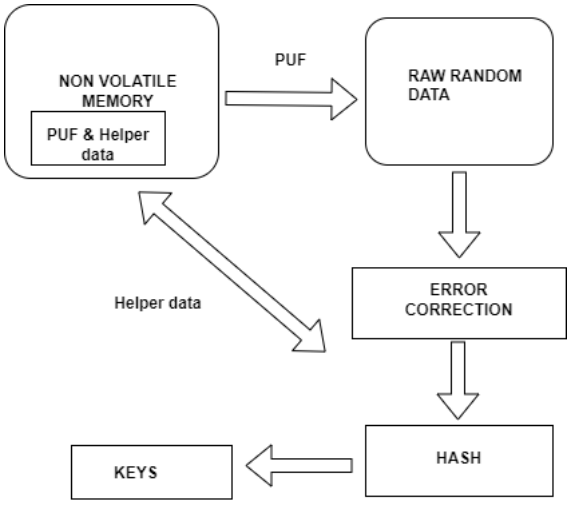

Fig. 1. Existing Private Key generation using PUF

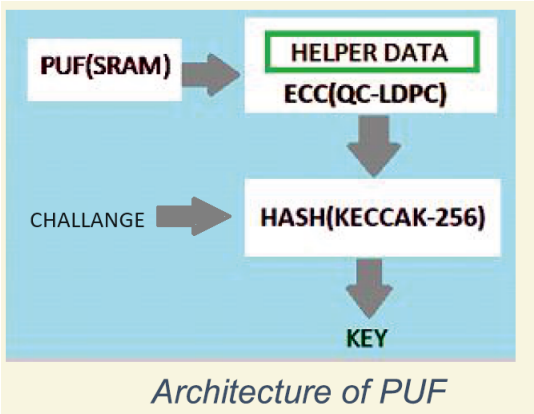

Fig. 2. Proposed Private Key generation using PUF

specifically targets errors in this raw random data. The key's robustness stems from variations in the manufacturing process rather than relying solely on the strength of the error correction block. Subsequently, a hashing process is employed to further enhance the entropy. Additionally, an extra input (challenge) is introduced into the hash function to produce multiple private keys, as depicted in Fig.2 and a comparision of this approach with the existing design is discussed in Table.1. This approach helps limit the accessibility of SRAM PUF, thereby bolstering the security factor. The validation of uniqueness and randomness has been conducted across three VC707 boards.Figures 3 and 4 depict the architecture and area implementation of the PUF, respectively. The time required

Table 1. Comparision of Proposed and Existing design of PUF

Existing Design of SRAM PUF	**Proposed Design of SRAM PUF**
Raw random data is extracted from in-built SRAM memory	Specific design is implement to extract the raw random data
Random data is given to error correction generate the key and helper data	LDPC is used to correct the error with helper data
Side channel threat is available	Side-channel attack is complex, since it is limited to accessible
Private Key generation is low	Private Key generation is high

for the error correction module ranges from 80ns to 800ns, contingent upon the error rate, which spans from 0 to 10 percent. Table 2 showcases the throughput of the PUF architecture concerning various error ranges.

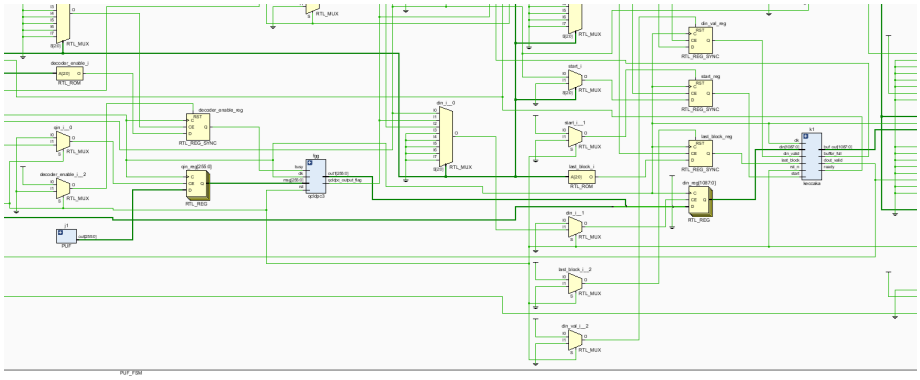

Fig. 3. Implementation of PUF Architecture

2 Conclusion

In summary, this paper introduces a secure implementation of a Physical Unclonable Function (PUF) for private key generation in Field-Programmable Gate Arrays (FPGAs). Addressing vulnerabilities in SRAM PUFs, the proposed scheme enhances both speed and security through cross-coupled inverters, LDPC error

Name	Slice LUTs (303600)	Slice Registers (607200)	F7 Muxes (151800)	F8 Muxes (75900)	Slice (75900)	LUT as Logic (303600)	Bonded IOB (700)	IBUFDS (672)	BUFGCTRL (32)
N TOP_PUF	139225	101011	34	16	40133	139225	6	1	2
b1 (PUF_FSM)	138351	100393	0	0	39836	138351	0	0	0
fgg (qcldpc3)	133989	97212	0	0	38374	133989	0	0	0
j1 (PUF)	397	0	0	0	143	397	0	0	0
k1 (keccaka)	3960	2130	0	0	1122	3960	0	0	0
L1 (RECEVIER)	337	313	0	0	133	337	0	0	0
u1 (TRANSMITTER)	130	303	34	16	109	130	0	0	0

Fig. 4. Area of PUF Architecture

Table 2. Throughput of PUF design with respect to error percentage

Error rate	**Throughput of PUF Architecture**
0%	85.33 Mbps
10%	25.64Mbps

correction, and hashing. Validation through implementation and testing shows promising improvements in security and efficiency. This work marks a significant advancement towards secure private key generation in resource-constrained devices, emphasizing the importance of FPGA-based solutions in enhancing overall security.

References

1. B. M. S. Bahar Talukder, F. Ferdaus and M. T. Rahman, "Memory-Based PUFs are Vulnerable as Well: A Non-Invasive Attack Against SRAM PUFs," in IEEE Transactions on Information Forensics and Security, vol. 16, pp. 4035-4049, 2021, doi: 10.1109/TIFS.2021.3101045. keywords: Physical unclonable function;Protocols;Entropy;Silicon;Hamming distance;Fabrication;Error correction;PUF;PUF attack;PUF modeling;non-invasive attacks on weak PUF,
2. Dominik Merli, Dieter Schuster, Frederic Stumpf Georg Sigl "Side-Channel Analysis of PUFs and Fuzzy Extractors" International Conference on Trust and Trustworthy Computing Trust 2011: Trust and Trustworthy Computing pp 33–47.
3. Georg T. Becker, Raghavan Kumarctive and A Passive "Side-Channel Attacks on Delay Based PUF Designs" eprint.
4. Chintala Yehoshuva, R. Raja Adhithan, N. Nalla Anandakumar: A Survey of Security Attacks on Silicon Based Weak PUF Architectures. SSCC 2020: 107-122