Ultra-lightweight Symmetric-key Cipher for Resource Constrained Systems

Teng Xu, James B. Wendt, Miodrag Potkonjak
Computer Science Department
University of California, Los Angeles
{xuteng, jwendt, miodrag}@cs.ucla.edu

***Abstract*—In this paper, we present a new ultra lightweight reconfigurable security primitive that consumes orders of magnitude less energy than current state of the art ciphers. The essential idea is to use a matrix of randomly connected lookup tables (LUTs) to form the input-output mapping, whose configuration serves as the shared secret key. We apply our lookup table-based cipher (LUTC) to the NIST randomness benchmark suite and demonstrate that it passes all tests. Finally, we compare the energy and area overhead to existing symmetric key ciphers including AES, HIGHT, and PRESENT, and show orders of magnitude reduction in energy.**

I. Introduction

Since the establishment of AES as the standard symmetric key algorithm, the need for new symmetric ciphers has greatly diminished over the last decade. However, the increasing emergence of extremely resource constrained systems, such as sensor networks, has lead to a need for ultra-low power cryptographic primitives.

Data integrity and security is of the upmost concern for sensor networks deployed in remote locations. The increase proliferation of remote low power systems, such as sensor networks, requires that we develop new security primitives that have low overhead. New solutions such as HIGHT [1] and PRESENT [2] have reduced the energy footprint over the standard AES by an order of magnitude. However, for extremely resource constrained systems it is imperative that we develop ultra-low overhead solutions in terms of energy, area, and time.

Physical unclonable function (PUF) is a deterministic multiple-input multiple-output system that is hard to reverse engineer and simulate. As the name suggests, the device is also impossible to physically replicate. The idea to use PUF in the domain of sensor network security is proposed in [3][4]. Meguerdichian et al. demonstrated the matched PUF, which requires each party in a wide class of security protocols to conduct only a single cycle computation [5]. The key idea is to use device aging to create two completely identical PUFs in such a way that the probability that a third PUF can have the same characteristics is negligible. However, one important weakness of the PUF-based security primitive is its instability against environmental variations.

Our system aims at leveraging the above problems in terms of both power consumption and stability. Our design consists of two pieces of randomly connected LUT network with exactly the same configuration. The two PUFs are matched before communication such that they implement the same functionality, e.g., their input-output mappings are identical. Their function remains complex and unpredictable in terms of confusion and diffusion. Based on the security primitive, we also propose low power security protocols for authentication and cryptographic communication.

II. Related Work

We now briefly survey the most directly related literature on PUFs and the efforts to use PUFs in sensor networks.

PUF is first proposed by Papu et al. in 2001 [6]. Devadas and members of his research group in MIT first observed and proposed to use PUF as a security primitive [7][8]. A great variety of PUFs that employ different physical entities (e.g. delay and leakage energy), different architectures (e.g. ring oscillator, feed-forward, obfuscated parallel, differential, SRAM), and target different types of security protocols (e.g. secret key and public key) have been proposed and evaluated [9][10][11][12][13]. O'Donnell is the first to use PUFs as random number generators and use NIST test to test PUF randomness [14].

The use of PUF in the domain of sensor networks is proposed in [15][16][17][18]. The basic idea is to use the intrinsic input-output mapping of the PUF for encryption and decryption. However, the proposed approach employes the problem of instability and this is due to the fact that the proposed PUF takes advantage of the physical entities, e.g., delay and leakage power which are in the analog domain [19][20]. Our main difference is to propose a security solution in digital domain while maintaining the low power consumption.

III. LUTC Architecture and Operation

In this section, we first propose the architecture of LUTC. Then we analysis the property of confusion and diffusion of the given structure. Afterwards, we present the procedure to match the two LUTCs to form a symmetric cipher.

A. LUTC Architecture

The architecture of the LUTC is depicted in Figure 1. We choose FPGA as the platform to build the architecture of LUTC because of its configurability. More importantly, the LUTs are elementary component in the FPGA, so that it can be directly configured and connected as shown in Figure 1. Each LUT has 4 inputs and 1 output, it randomly chooses

978-1-4799-0162-3/14/$31.00 ©2014 IEEE

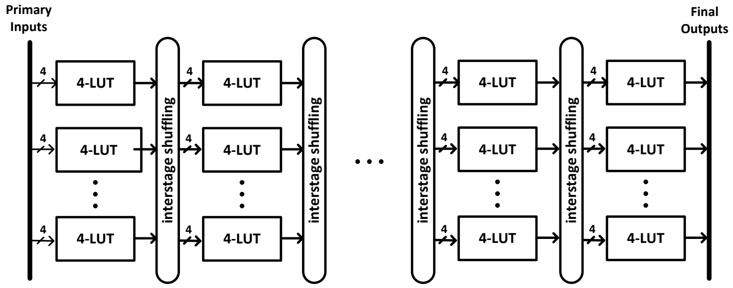

Fig. 1: The architecture of LUTC.

its each input from the output of a LUT in previous levels. Also, the contents in each LUT are independent and can be customized by the users. By repeating the LUT interconnection for levels, the structure is formed.

An important open question for the structure of LUTC is what is the optimal connection of the LUTC. For example, a derivative of structure shown in Figure 1 is that the input of LUTs in level n can not only from the outputs of $n-1$ level LUTs, but also from outputs of earlier level LUTs or even from primary inputs. The derivatives increase the possibility of various configuration. In this paper, our analysis is based on the structure shown in Figure 1.

Another problem is that given a LUTC of m outputs, what is the sufficient number of LUT levels to achieve good security property while maintaining low area/power consumption. For this problem, we adopt the following test. Given a LUTC structure of 64 primary inputs and 64 final outputs, and each level has 64 4-input LUTs, we test the output hamming distance of avalanche effect given different number of levels. The output hamming distance is defined as the number of bits in the output vector changed when one bit is changed in the input vector. The ideal case is 32 which indicates completely no correlation between the inputs and the outputs. Table I clearly shows that the hamming distance exponentially grows in the beginning when number of levels increases and it quickly converges to the ideal case of 32 when the number of levels reaches 9. Therefore, in the case of 64-bits LUTC, 9 levels is enough to achieve good security property. Similar tests can easily be done given other bit size of LUTC.

Levels	1	2	3	4	5
Avg. Hamming Distance	0.46	1.30	5.51	18.82	23.93
Levels	6	7	8	9	10
Avg. Hamming Distance	26.63	28.93	29.44	30.20	30.58

TABLE I: Average output hamming distance of avalanche effect given different number of levels.

B. Confusion and Diffusion

In his seminal security paper, Shannon described two properties that are paramount in designing secure systems: confusion and diffusion [21]. Confusion and diffusion are two main concerns in the design of security cipher. Confusion is achieved when the relationship between the key and the ciphertext is sufficiently complex. Diffusion, on the other hand, refers to the relationship between the plaintext and the ciphertext. In the case of LUTC, the key is a set of LUTs as well as their connections in the structure, the plaintext is the set of initial inputs, and the ciphertext is the outputs. Maximal confusion in a LUTC, then, results in great (ideally exponential) simulation effort for an attacker to compute the outputs for a particular input and key. For ideal diffusion, each output depends on all inputs and on each input equally, resulting in great difficulty for an attacker to predict the output for a particular set of inputs.

The structure achieves excellent confusion and diffusion by employing exponentially connection possibilities between the LUTs as well as an exponential number of LUT configurations. From another point of view, because the LUTs in level n choose inputs from the outputs of LUTs in level $n-1$, and the number of inputs in level n LUTs is 4 times the number of outputs in level $n-1$ LUTs. Therefore, for each output from previous level, it is expected to connect to 4 LUTs in next level, thus influencing the outputs of the 4 LUTs. The influence propagates exponentially with the levels of LUTs which means, a output in level n is expected to influence 4^w LUTs in level $n+w$, resulting in good diffusion effect.

C. Operation

Two pieces of LUTC can easily be matched through FPGA configuration. As long as the LUTs in the LUTC have the same initialization as well as connection, the two LUTCs are matched. Therefore, given the same input vector, the matched LUTC can produce the same output vector, thus the symmetric-key cipher is produced.

IV. Protocols

In this section, we present how to use the LUTC for secret key communication and authentication in sensor networks.

Protocol 1 proposed the protocol of private key communication. Private key communication is one of the most commonly used protocols in the domain of secure message transfer. In the scenario of sensor networks, it is useful when two sensors want to exchange messages. Two sensors can easily communicate with each other after their LUTC are matched.

Protocol 2 describes the procedure of authentication. It takes advantage of the fact that two matched LUTC produce the

Protocol 1 Secret Key Communication

message sender: sensor1
message receiver: sensor2

1. sensor1 and sensor2 match their LUTC.
2. sensor1 chooses a random seed as input vector I for LUTC and computes the corresponding output vector O.
3. sensor1 XOR the output vector O with the message M and gets the result R.
4. sensor1 sends I and R to the sensor2.
5. sensor2 computes output vector O with the received I using its matched LUTC.
6. sensor2 XOR O with the received R to get message M.

same outputs given the same inputs. In sensor networks, a sensor can use this protocol to authenticate another sensor or any party who wants to communicate with the sensor. For example, if a sensor owner wants to read the data of a sensor remotely, he/she needs to be authenticated first.

Protocol 2 Authentication

authenticator: sensor1
supplicant: sensor2

1. sensor1 and sensor2 match their LUTC.
2. sensor1 chooses a random seed as input vector I for LUTC and computes the corresponding output vector O_1.
3. sensor1 sends I to sensor2.
4. sensor2 computes output vector O_2 with the received I using its matched LUTC.
5. sensor2 sends O_2 to sensor1.
6. sensor1 compares O_1 with O_2, only when $O_1 = O_2$, sensor1 authenticates sensor2.

As an extension of the above two protocols, both protocols can easily be extended to multi-party protocols. For example, for the protocol of authentication, if a third party needs to be added as the authenticated party, he/she only needs to acquire an identical piece of LUTC, so that among the three parties, they can mutually authenticate each other.

V. Power Analysis

We compare the delay, area, and power consumption across LUTC and the FPGA implementation of the other types of proposed symmetric-key ciphers in Table II. We can obviously conclude from the table that LUTC, as a symmetric-key cipher, owns ultra-low power implementation that outperforms the traditional ciphers with at least two orders of magnitude.

VI. Security Analysis

A. Output randomness

We test the output randomness by applying the NIST randomness test [23]. NIST is a battery of standard statistical tests to detect non-randomness in binary sequences constructed using either random number generators or pseudo-random number generators.

We simulate our LUTC with 9 levels and 64 inputs/outputs. We generate the output stream in the following way: a random seed is provided as the primary inputs to the LUTC and its outputs are XORed with the current inputs and fed back as the inputs to LUTC in the next iteration. Meanwhile, the outputs of the LUTC are collected as the output stream. After this process, we apply Von Neumann correction on the output stream.

Table III shows the average passing ratio of each NIST statistical test. We can see that the proportion of successful tests is high enough to indicate excellent randomness in the output stream.

Statistical Test	Avg. Success Ratio
Frequency	100%
Block Frequency (m=128)	97.7%
Cusum-Forward	98.6%
Cusum-Reverse	98.6%
Runs	96.6%
Longest Runs of Ones	97.4%
Rank	97.8%
Spectral DFT	99.1%
Non-overlapping Templates (m=9)	96.4%
Overlapping Templates (m=9)	96.8%
Universal	100%
Approximate Entropy (m=8)	98.5%
Random Excursions (x=+1)	97.4%
Random Excursions Variant (x=-1)	97.4%
Serial (m=16)	97.2%
Linear Complexity (M=500)	98.8%

TABLE III: NIST Statistical Test Suite average success ratio. 1000 arrays are tested for each test. Significance Level $\sigma = 0.01$. When $P\text{-}value \geq \sigma$, the array passes test.

B. Output hamming distance

The output hamming distance of avalanche effect is an important indicator of the diffusion of the symmetric-key cipher. The ideal case is that when the inputs changed by one bit, the outputs would be completely changed in an unpredictable way. For a 9 level 64-bit LUTC, we test the distribution of outputs hamming distance when the inputs changed by one bit. The result in Figure 2 indicates an almost perfect binomial distribution.

C. Input-output correlation

This security test reveals the bitwise correlation between the inputs and the outputs. We use the probability $P(O_i = c_1 | I_j = c_2)$, c_1, c_2=1 or 0 to indicate the correlation. The ideal secure system will have a probability of 0.5 for all conditions. Figure 3 depicts the conditional probability $P(O_i = 1 | I_j = 1)$ of a 9 level 64-bit LUTC. Despite the fact that some probability values are close to 1 or 0, but we can simply not to use these bits in encryption. The majority part of the probabilities are close to 0.5, which is the ideal case in this test.

Design	Flip Flops	LUTs	Area (Slices)	Maximum Delay (ns)	Clock Cycles	Energy (μJ)	Block Size (bits)	Throughput (Mbps) at f_{max}	Device
Present[22]	114	159	117	8.78	256	3.16×10^{-3}	64	28.46	xc3s50-5
HIGHT[22]	25	132	91	6.12	160	1.07×10^{-3}	64	65.48	xc3s50-5
AES[22]	338	531	393	14.21	534	3.58×10^{-2}	128	16.86	xc3s50-5
64-LUTC	64	64	32	67.32	1	2.58×10^{-5}	64	950.69	xc3s50-5
128-LUTC	128	128	64	156.38	1	1.2×10^{-4}	128	818.53	xc3s50-5

TABLE II: Comparisons for LUTC security-cipher with the traditional block cyphers. The results for Present, HIGHT and AES are cited from [22]. The results for LUTCs are tested on the Spartan-3 XC3S50-5 FPGA and generated by the Xilinx ISE Design Suite 14.3.

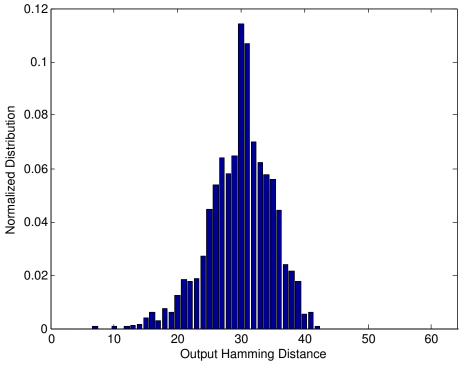

Fig. 2: The output hamming distance of avalanche effect for a 9 level 64-bit LUTC.

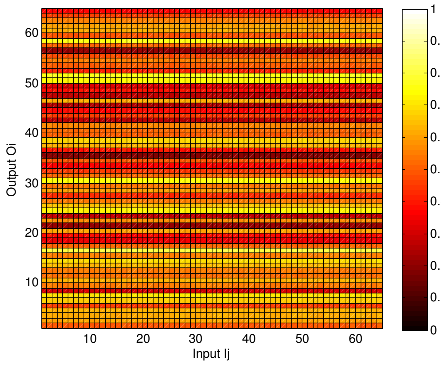

Fig. 3: Conditional probabilities between the inputs and the outputs: $P(O_i = 1|I_j = 1)$.

VII. Conclusion

We have developed an ultra-power security cipher: LUTC in this paper. It employs the randomly connected LUT networks to enable low-power security communication in the sensor networks. A comparison between LUTC and other traditional ciphers indicate that LUTC is at least two orders of magnitude more energy efficiency. Finally, our security test shows that LUTC also owns excellent security properties in terms of confusion and diffusion.

VIII. Acknowledgement

This work was supported by the NSF under Award CNS-0958369, Award CNS-1059435, and Award CCF-0926127.

References

[1] D. Hong et al., HIGHT: A new block cipher suitable for low-resource device, *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pp. 46-59, 2006.

[2] A. Bogdanov et al., PRESENT: An ultra-lightweight block cipher, *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pp. 450-466, 2007.

[3] T. Xu, M. Potkonjak, “Lightweight digital hardware random number generators,” *IEEE SENSORS*, pp. 1-4, 2013.

[4] T. Xu, J. B. Wendt, and M. Potkonjak, “Matched Digital PUFs for Low Power Security in Implantable Medical Devices” to appear in *IEEE International Conference on Healthcare Informatics (ICHI)*, 2014.

[5] S. Meguerdichian, M. Potkonjak, “Matched public PUF: ultra low energy security platform,” *IEEE/ACM ISLPED*, pp. 45–50, 2011.

[6] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, “Physical one-way functions,” *Science*, vol. 297, no. 5589, pp. 2026-2030, 2002.

[7] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, “Silicon physical random functions,” *ACM Conference on Computer and Communications Security*, pp. 148-160, 2002.

[8] S. Devadas, V. Khandelwal, S. Paral, R. Sowell, E. Suh, T. Ziola, “Design and Implementation of PUF-Based Unclonable RFID ICs for Anti-Counterfeiting and Security Applications,” *IEEE RFID*, 2008.

[9] N. Beckmann, M. Potkonjak, “Hardware-based public-key cryptography with public physically unclonable functions,” *Information Hiding Conference*, pp. 206–220, 2009.

[10] M. Potkonjak, S. Meguerdichian, A. Nahapetian, Sheng Wei, “Differential Public, Physically Unclonable Functions: Architecture and Applications”, *ACM/IEEE Design Automation Conference*, pp. 242-247, 2011.

[11] T. Xu, J. B. Wendt, M. Potkonjak, “Digital Bimodal Function: An Ultra-Low Energy Security Primitive,” *IEEE/ACM ISLPED*, pp. 292-297, 2013.

[12] T. Xu, M. Potkonjak, “Robust and Flexible FPGA-based Digital PUF, to appear in *International Conference on Field Programmable Logic and Applications (FPL)*, 2014.

[13] R. Yan, V. C. Shah, T. Xu and M. Potkonjak, “Security Defenses for Vulnerable Medical Sensor Network, to appear in *International Conference on Healthcare Informatics (ICHI)*, 2014.

[14] C. W. O’Donnell, G. E. Suh, and S. Devadas, “PUF-based random number generation,” *MIT CSAIL CSG Technical Memo 481*, 2004.

[15] S. Wei, J. H. Ahnn, M. Potkonjak, “Energy attacks and defence techniques for wireless systems,” *WISEC*, pp. 185-194, 2013.

[16] M. Potkonjak, S. Meguerdichian, J.L. Wong, “Trusted Sensors and Remote Sensing”, *IEEE Sensors*, pp. 1104-1107, 2010.

[17] J. B. Wendt, M. Potkonjak, “Nanotechnology-Based Trusted Remote Sensing,” *IEEE SENSORS*, pp. 1213-1216, October 2011.

[18] S. Wei, M. Potkonjak, “Wireless security techniques for coordinated manufacturing and on-line hardware trojan detection,” *WISEC*, pp. 161-172, April 2012.

[19] M. A. Alam and S. Mahapatra, “A comprehensive model of PMOS NBTI degradation,” *Microelectronics Reliability*, vol. 45, pp. 71-81, 2005.

[20] M. Bhargava, C. Cagla, and M. Ken, “Comparison of bi-stable and delay-based Physical Unclonable Functions from measurements in 65nm bulk CMOS,” *Custom Integrated Circuits Conference*, IEEE, 2012.

[21] C. E. Shannon, Communication Theory of Secrecy Systems, *Bell System Technical Journal*, vol. 28, no. 4, pp. 656-715, 1949.

[22] P. Yalla and J-P. Kaps, Lightweight cryptography for FPGAs, *ReConFig*, pp. 225-230, 2009.

[23] “A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications,” National Institute of Standards and Technology (NIST) Special Publication 800-22, Rev. 1a, Apr. 2010.