An Alternative Permuting Function for the Enhanced Sponge Function (ESP)

Magdy Saeb

Arab Academy of Science, Technology and Maritime Transport

Computer Engineering Department

mail@magdysaeb.net

Abstract: In this short correspondence, we shed some light on the permuting function, used by Bertoni et al. in their Sponge Function Design, indicating some of its limitations. In addition, we provide our proposed alternative permuting function that overcomes some of these limitations.

Keywords: Sponge Function, Permuting Function, Cryptography

1. Introduction

The permuting function applied in the Sponge Function design introduced by Bertoni et al. [1], [2], [3], [4] shown in Figure 1, has several limitations.

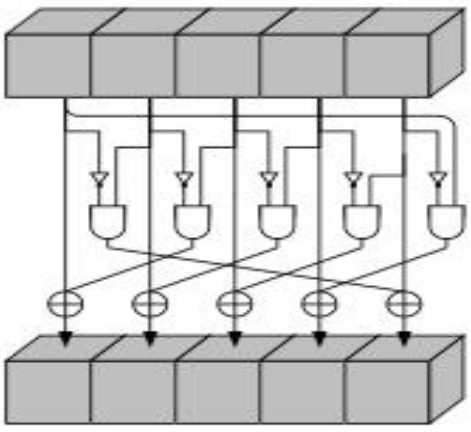

Figure 1: The Permuting Function introduced by Bertoni et al. [1], [2], [3], [4]

Received 9/ 2, Reviewed 9/ 20, 2013

We argue that this permuting function may not provide the required shuffling of the message bits with no bias in the number of ones or zeroes. This argument can be established by noticing the following:

1. Assuming that sub-block bits are all zeroes, then it is clear that the output will remain indefinitely zero regardless of the number of rounds to be used. The AND gate output is zero and if the input to the XOR gate is also zero, then the XOR output will be zero.
2. Now if all inputs are all ones, the outputs will be all ones since the AND gates outputs are all zeroes and the XOR with one and zero inputs will output ones! In other words, if all inputs were ones, then by tracing the outputs, they will be also ones.
3. From the above, one concludes that in the boundary case previously discussed (all inputs are zeroes or all inputs are ones), the permuting function will noticeably perform poorly. The boundary case, while it appears just a theoretical case yet

it more or less occurs in audio and video transmissions.

4. In addition, the AND gate inherently gives a bias in its output to the zeroes with a 3:1 ratio and the XOR does not provide any bias to the ones or zeroes. Therefore, there is an expected bias to the zeroes in the final output.
5. Since the inputs, the outputs and the operations performed on the block bits are all fixed, there is a significant probability of cyclical behavior of the output. This, in turn, will greatly compromise security.
6. Assuming that the delay per one logic gate is 1 ns, then the average delay of this approach is about 2.5 ns per bit.

2. The Proposed Permuting Function

In view of the above argument, we will provide a permuting function design that avoids all of the discussed limitations of the function introduced by Bertoni et al. The proposed permuting function is shown in Figure 2.

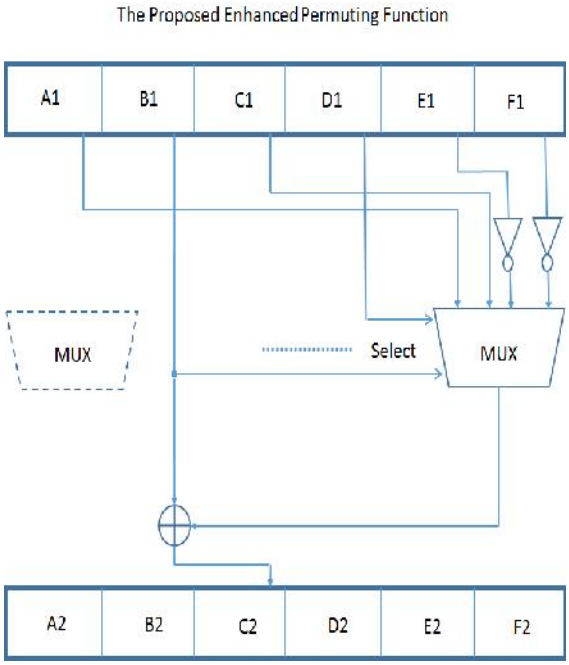

Figure 2 The Proposed Permuting Function

In this Figure, we use six sub-blocks along with six multiplexers. The multiplexer inputs are taken from four sub-blocks and the select input of the multiplexer is taken from the remaining two block. Two of the inputs of each multiplexer are inverted to ensure that there is bias to the zeroes or to the ones. The multiplexer output bit is then XORed with one of the select bits of the multiplexer. This bit is permuted to a different sub-block; we choose the other sub-block that was providing the select bit. In Figure 2, we have only shown one multiplexer with its inputs and output bit to simplify the schematic. However, the same basic procedure is to be applied to all other five multiplexers.

The average or expected delay in our proposed approach is about 3.5 ns per bit which is not significantly different from the Bertoni's approach.

Conclusion:

We have demonstrated the limitations of the permuting function used in the original Sponge Function Design. Our proposed permuting function provides better shuffling of message bits with no bias in the number of ones or zeroes. Since the selected inputs are varied randomly, which is equivalent to a rotation operation, then the proposed permuting function will provide no cyclical effects similar to the probable cyclical effect of the original permuting function. Along with the Enhanced Sponge Function Design outlined in [5], we believe that the proposed permuting function will improve the security of the sponge function.

References:

[1] G. Bertoni, J. Daemen, M. Peeters and G. Van Assche, *Sponge Functions*, in ECRYPT Hash Workshop, 2007.

[2] G. Bertoni, J. Daemen, M. Peters, G. V. Assche, Cryptographic sponge functions, (Report), http://sponge.noekeon.org/

[3] G. Bertoni, J. Daemen, M. Peeters and G. Van Assche, The Sponge Functions Corner, http://sponge.noekeon.org/, accessed June, 2012.

[4] Guido Bertoni, Joan Daemen1, Michaël Peeters, Gilles Van Assche, "The Keccak sponge function family," http://keccak.noekeon.org/, accessed Sept, 2013.

[5] M. Saeb, "An Enhanced Sponge Function," International Journal of Computer Science and Communication Security (IJCSCS), Vol 2, July, 2012.

Magdy Saeb received the BSEE. School of Engineering, Cairo University, in 1974, the MSEE, and Ph.D. Degrees in Electrical & Computer Engineering, University of California, Irvine, in 1981 and 1985, respectively. He was with Kaiser Aerospace and Electronics, Irvine California, and The Atomic Energy Establishment, Anshas, Egypt. Currently, he is a professor in the Department of Computer Engineering, Arab Academy of Science, Technology & Maritime Transport, Alexandria, Egypt; He was on-leave working as a principal researcher in the Malaysian Institute of Microelectronic Systems (MIMOS). His current research interests include Cryptography, FPGA Implementations of Cryptography and Steganography Data Security Techniques, Encryption Processors, Mobile Agent Security. www.magdysaeb.net.