Quantum Implementation of ASCON Linear Layer

Soham Roy **Anubhab Baksi** **Anupam Chattopadhyay**

Indian Institute of Technology, Madras

Nanyang Technological University

June 20, 2023

Table of Contents

Introduction

ASCON is a lightweight cryptographic primitive.

It was recently selected as the winner of the LWC project by NIST.

In recent years, quantum cryptography has become an active research area.

Exploring the potential of quantum computing to improve the performance of ASCON is of significant interest to the cryptography community.

Contribution

In this work, we present possibly the first-ever in-place implementation of the ASCON linear layer. The related implementations are available as an open-source project[1].

The linear layer is described in terms of rotation and XOR of five 64-bit registers. Thus, it can be equivalently expressed as a 320×320 binary matrix.

In total, we show 3 in-place implementations of the ASCON linear layer.

The naïve implementation in quantum doubles the number of qubits. When the qubit count $\times$ CNOT count metric is considered, the implementation reported by us is 16.93% cheaper.

[1] https://github.com/sohamroy19/ascon-linear-layer/

ASCON Linear Layer

Definition

The linear layer p_L applies $\Sigma_i(x_i)$ to each word x_i, defined as follows ('$\ggg$' indicates right rotation):

$$
\begin{aligned}
x_0 &\leftarrow \Sigma_0(x_0) = x_0 \oplus (x_0 \ggg 19) \oplus (x_0 \ggg 28) \\
x_1 &\leftarrow \Sigma_1(x_1) = x_1 \oplus (x_1 \ggg 61) \oplus (x_1 \ggg 39) \\
x_2 &\leftarrow \Sigma_2(x_2) = x_2 \oplus (x_2 \ggg 1) \oplus (x_2 \ggg 6) \\
x_3 &\leftarrow \Sigma_3(x_3) = x_3 \oplus (x_3 \ggg 10) \oplus (x_3 \ggg 17) \\
x_4 &\leftarrow \Sigma_4(x_4) = x_4 \oplus (x_4 \ggg 7) \oplus (x_4 \ggg 41)
\end{aligned}
$$

This can be written as a binary non-singular matrix of dimension 320×320.
The Hamming weights of each row and column are 3, and the multiplicative order is 64.

ASCON Linear Layer

Matrix

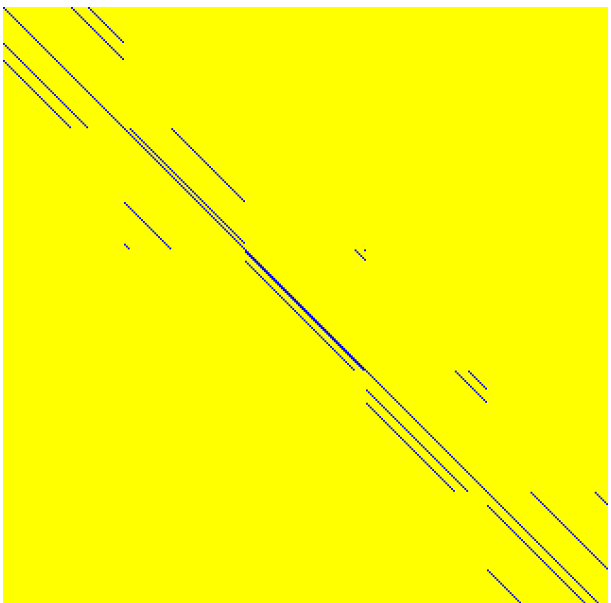

Figure. ASCON linear layer (320×320 binary matrix in graphical form: blue indicates 1 and yellow indicates 0)

Gates

Analogous to the concept of bit in the classical computing, we use *quantum bit*s (*qubit*s for short).

The *Controlled NOT* (*CNOT* for short) and *SWAP* gates form the basis of linear operations.

(a) CNOT **(b)** SWAP

Figure. Basic quantum gates

A CNOT gate is like an in-place XOR gate. A SWAP gate can be implemented using 3 CNOT gates.

Gates

The quantum circuits are composed of the quantum gates and those operate on the qubits.

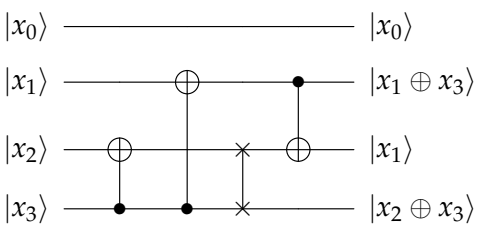

Figure. A quantum circuit (with CNOT and SWAP gates)

The depth of a circuit is the number of combinational logic gates along the longest path of the circuit. Classical depth is simpler to calculate than quantum depth, as quantum gates can have only 1 fan-out.

Examples

Naïve quantum

Consider the binary matrix $M^{4\times4}$:

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Our original qubits are $\{x_0, x_1, x_2, x_3\}$. In the first step, 4 ancilla qubits are created and initialized to 0:

$$x_4 \leftarrow |0\rangle; x_5 \leftarrow |0\rangle; x_6 \leftarrow |0\rangle; x_7 \leftarrow |0\rangle$$

After this, the ancilla qubits are updated in-place, so that each copies one original qubit:

$$\begin{aligned} x_4 &\leftarrow x_4 \oplus x_0 \\ x_5 &\leftarrow x_5 \oplus x_1 \\ x_6 &\leftarrow x_6 \oplus x_2 \\ x_7 &\leftarrow x_7 \oplus x_3 \end{aligned}$$

Equivalently, M is augmented with the 4×4 identity matrix to create this 8×4 matrix M':

$$M = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} \longrightarrow \left(\begin{array}{cccc} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right) = M'$$

Finally, we update the original qubits in-place (in M') with the help of the ancilla qubits:

$$x_0 \leftarrow x_0 \oplus x_5, x_1 \leftarrow x_1 \oplus x_6, x_1 \leftarrow x_1 \oplus x_7, x_2 \leftarrow x_2 \oplus x_4,$$
$$x_2 \leftarrow x_2 \oplus x_5, x_2 \leftarrow x_2 \oplus x_7, x_3 \leftarrow x_3 \oplus x_4$$

Thus, the naïve implementation takes 8 qubits and 11 CNOT gates, with 3 quantum depth. □

Legacy Algorithms

Two legacy algorithms are known to have a quantum-friendly outcome:

1. Gauss-Jordan elimination;
2. PLU factorization.

Legacy Algorithms

Gauss-Jordan

Consider the following binary matrix:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

By applying the Gauss-Jordan elimination we obtain the following factorization of the matrix:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The corresponding implementation incurs 3 CNOT and 1 SWAP gates with 4 quantum depth:

$$\begin{aligned} x_2 &\leftarrow x_2 \oplus x_3 \\ x_1 &\leftarrow x_1 \oplus x_3 \\ x_2, x_3 &\leftarrow x_3, x_2 \\ x_2 &\leftarrow x_2 \oplus x_1 \end{aligned}$$

By applying the PLU factorization on the following binary matrix, we obtain the following:

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The corresponding implementation incurs 5 CNOT and 1 SWAP gates, with 4 quantum depth:

$$\begin{aligned} x_0 &\leftarrow x_0 \oplus x_2 \\ x_1 &\leftarrow x_1 \oplus x_3 \\ x_2 &\leftarrow x_2 \oplus x_3 \\ x_3 &\leftarrow x_3 \oplus x_1 \\ x_2 &\leftarrow x_2 \oplus x_0 \\ x_2, x_3 &\leftarrow x_3, x_2 \end{aligned}$$

□

BENCHMARKS

Table. Quantum benchmarks of ASCON linear layer

Method	Qubit count	CNOT count	Quantum depth
Naïve	640 (out-of-place)	960	26
Gauss-Jordan	320 (in-place)	2413	358
PLU		2413	288
Modified XZLBZ		1595	119