

Rocca S

A Symmetric Key Block Cipher

Introduction

- Rocca S is a cryptographic algorithm focused on efficiency with many opportunities for parallelization.
- It uses a state-based approach and ciphers in 256-bit blocks of data.
- Focus on secure, high-speed data encryption.

Algorithm Structure

- ROCCA_STATE is composed of 7 elements of 128-bits each.
- The state gets initialized based on the input key and some predefined variables.
- Z0 and Z1 are 128bit values used in the initialization of State.
- SubBytes Array is the same array that's used in substitution phase of AES.

Algorithm Structure

- Key is 256bit for maximum security. Its divided into K0(left) and K1(right) each 128bits for use in algorithm.
- Nonce is an initializing vector. Varies from 96bits to 128bits.
- Associated Data (2^{61} bytes max_size) is used also used to update State and is sent without encryption.
- Plaintext (2^{125} bytes max_size) is the text to be ciphered.

Rocca S - Overview

Rocca S - Overview

Input:

1. Plain text
2. User Key - 256 bit
3. Nounce(iv) - 96bit to 128 bit value
4. AD(additional data)

Rocca S

Output:

1. Cipher text
2. Tag - 256 bit value

Algorithm variables:

1. State - 7 elements of 128bit each
2. Z0 - 128bit value
3. Z1 - 128bit value
4. Arrays for AES round:
 - a. Sub bytes array - 256 elements 8bit each
 - b. Finite field $GF(2^8)$ - 16 elements 8 bit each

Padding Explanation

- $\text{PAD}(M)$ pads M to have a length divisible by 256 by adding 0s to the right.
- $\text{PADN}(N)$ pads N to have length divisible by 128 by adding 0s to the right.

PAD - Explanation

Plaintext - M
Suppose it has
 $256+256+2 = 514$ bits

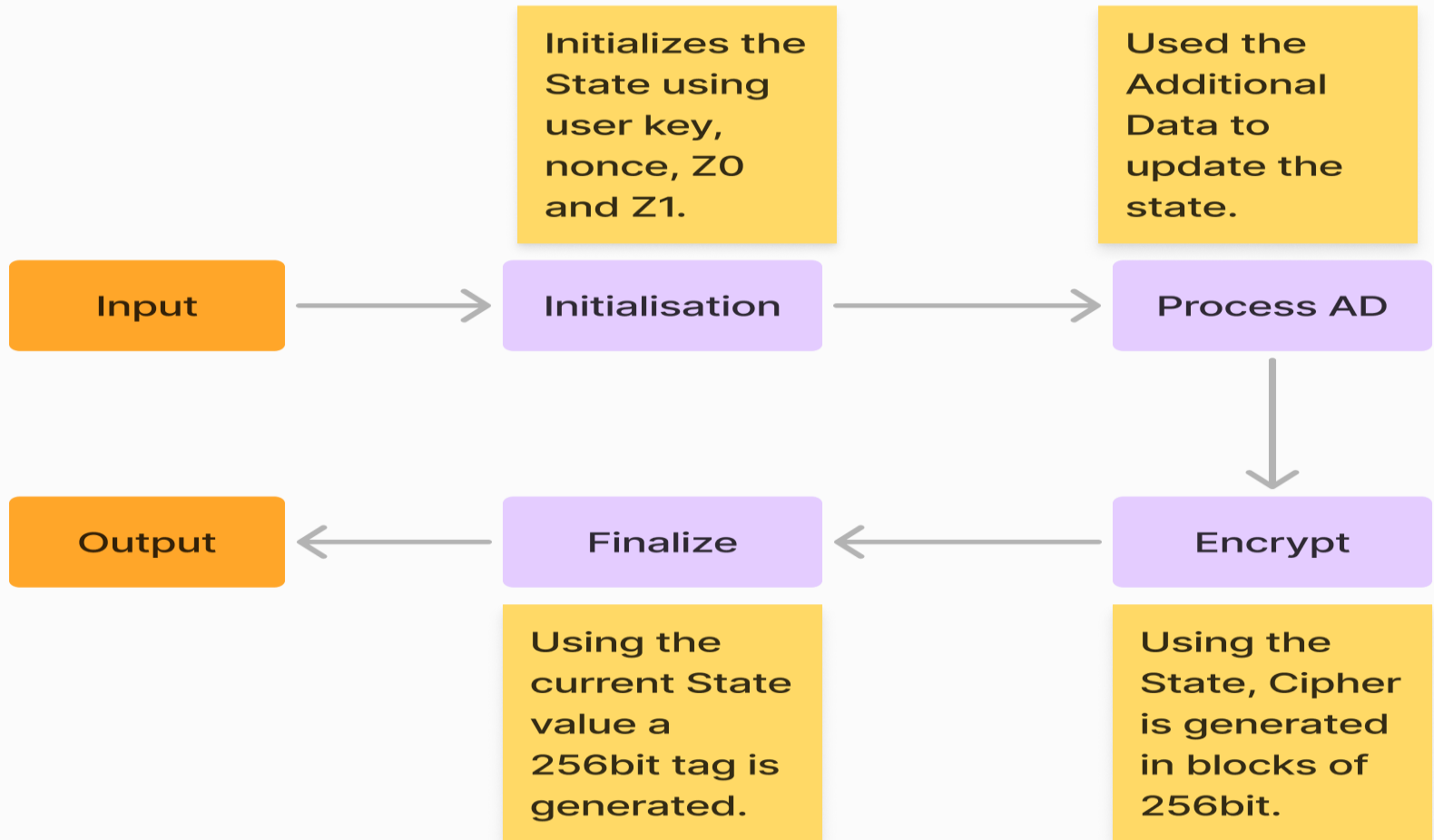
0	1	2	254	255	0	1	2	254	255	0	1
0	1	1	0	1	0	1	1	1	0	0	1	0	1

$\text{PAD}(M)$ will add 0s to
the right so that its
length is multiple of
256.

0	1	2	254	255	0	1	2	254	255	0	1	2	254	255
0	1	1	0	1	0	1	1	1	0	0	1	0	1	0	0	0	0

Rocca S – Steps Involved

Rocca S - Steps Involved



Initialization

Initialization

RoundFunction(S,Z0,Z1)

Initialize

RoundFunction

Xor Phase

$S[0] = K[1]$
 $S[1] = \text{PADN}(N)$
 $S[2] = Z0$
 $S[3] = K[0]$
 $S[4] = Z1$
 $S[5] = \text{PADN}(N) \wedge K[1]$
 $S[6] = \text{Zero}(128)$

16 rounds

Z0 and Z1 as
parameters in
each iteration

$S[0] = S[0] \wedge K[0]$
 $S[1] = S[1] \wedge K[0]$
 $S[2] = S[2] \wedge K[1]$
 $S[3] = S[3] \wedge K[0]$
 $S[4] = S[4] \wedge K[0]$
 $S[5] = S[5] \wedge K[1]$
 $S[6] = S[6] \wedge K[1]$

RoundFunction

RoundFunction

X_0 and X_1 are
128bit
parameters of the
round function

$S_{new}[0] = S[6] \wedge S[1]$
 $S_{new}[1] = \text{AES}(S[0], X_0)$
 $S_{new}[2] = \text{AES}(S[1], S[0])$
 $S_{new}[3] = \text{AES}(S[2], S[6])$
 $S_{new}[4] = \text{AES}(S[3], X_1)$
 $S_{new}[5] = \text{AES}(S[4], S[3])$
 $S_{new}[6] = \text{AES}(S[5], S[4])$

S = Snew

The State gets
updated to
newState after
every round
function.

AES - Overview

AES - Overview

State : array of 16
elements each
8bits.(128bit)
Key: array of 16
elements each
8bits.(128bit)

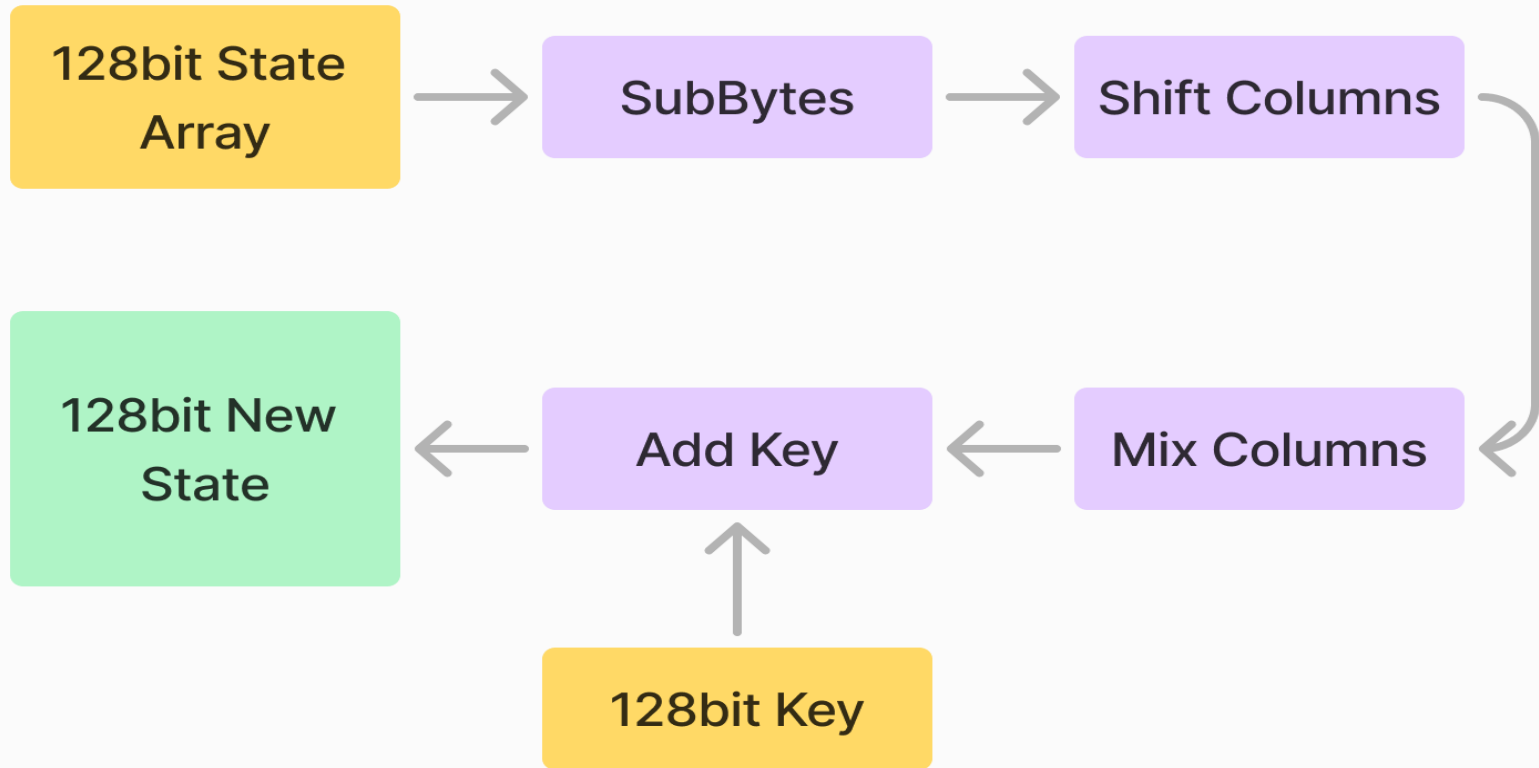
AES

Updated State :
array of 16
elements each
8bits.(128bit
value is returned)



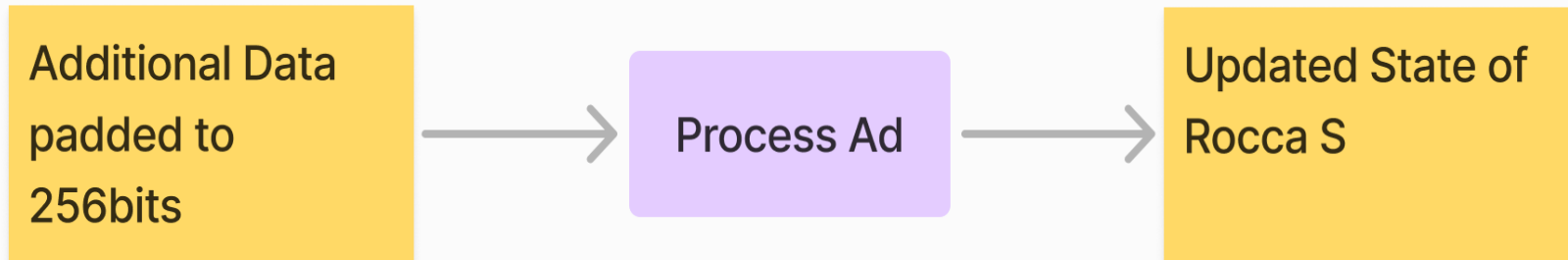
AES - Details

AES - Details



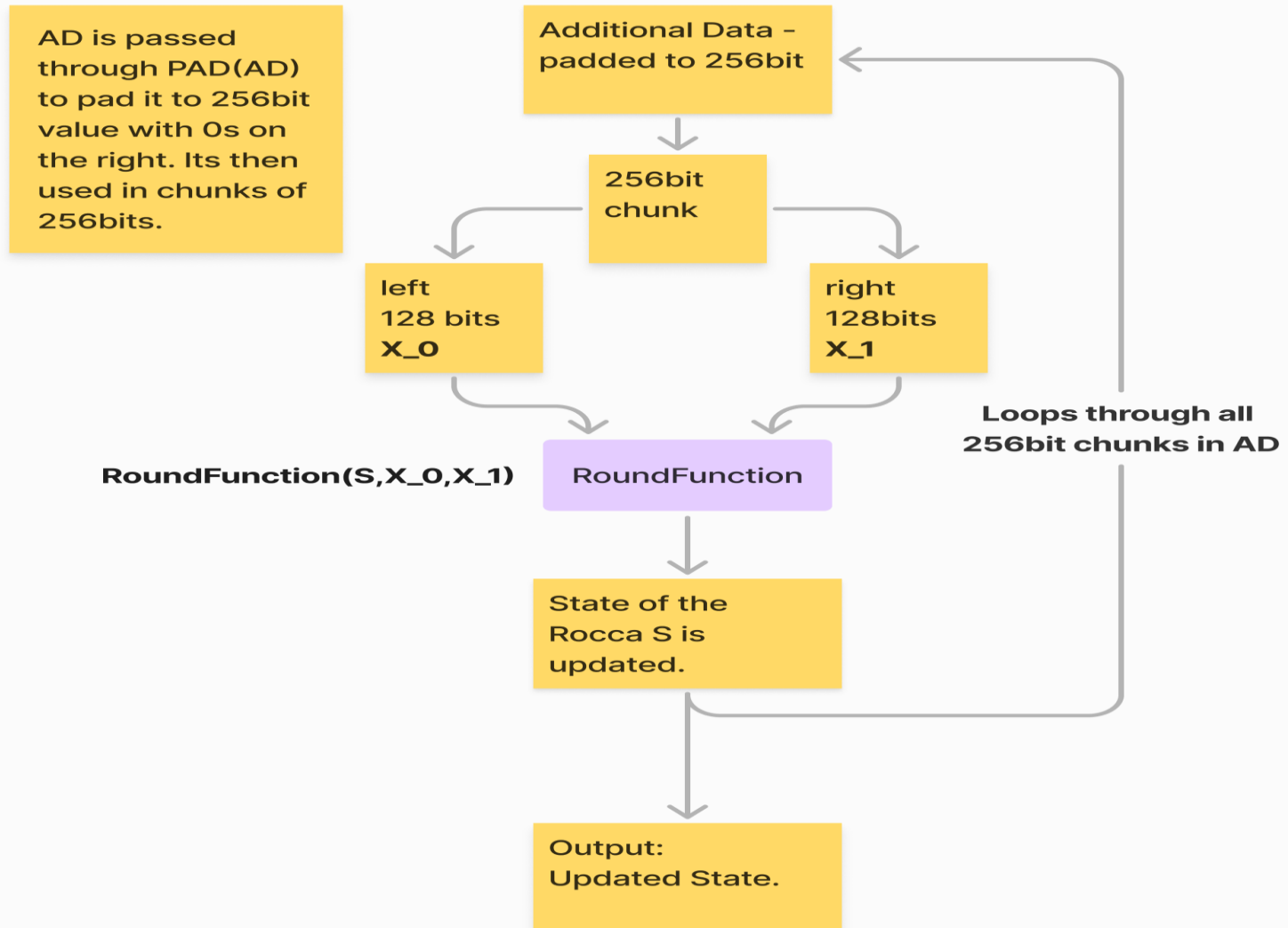
Process AD - Overview

Process AD - Overview



Process AD - Details

Process AD - Details



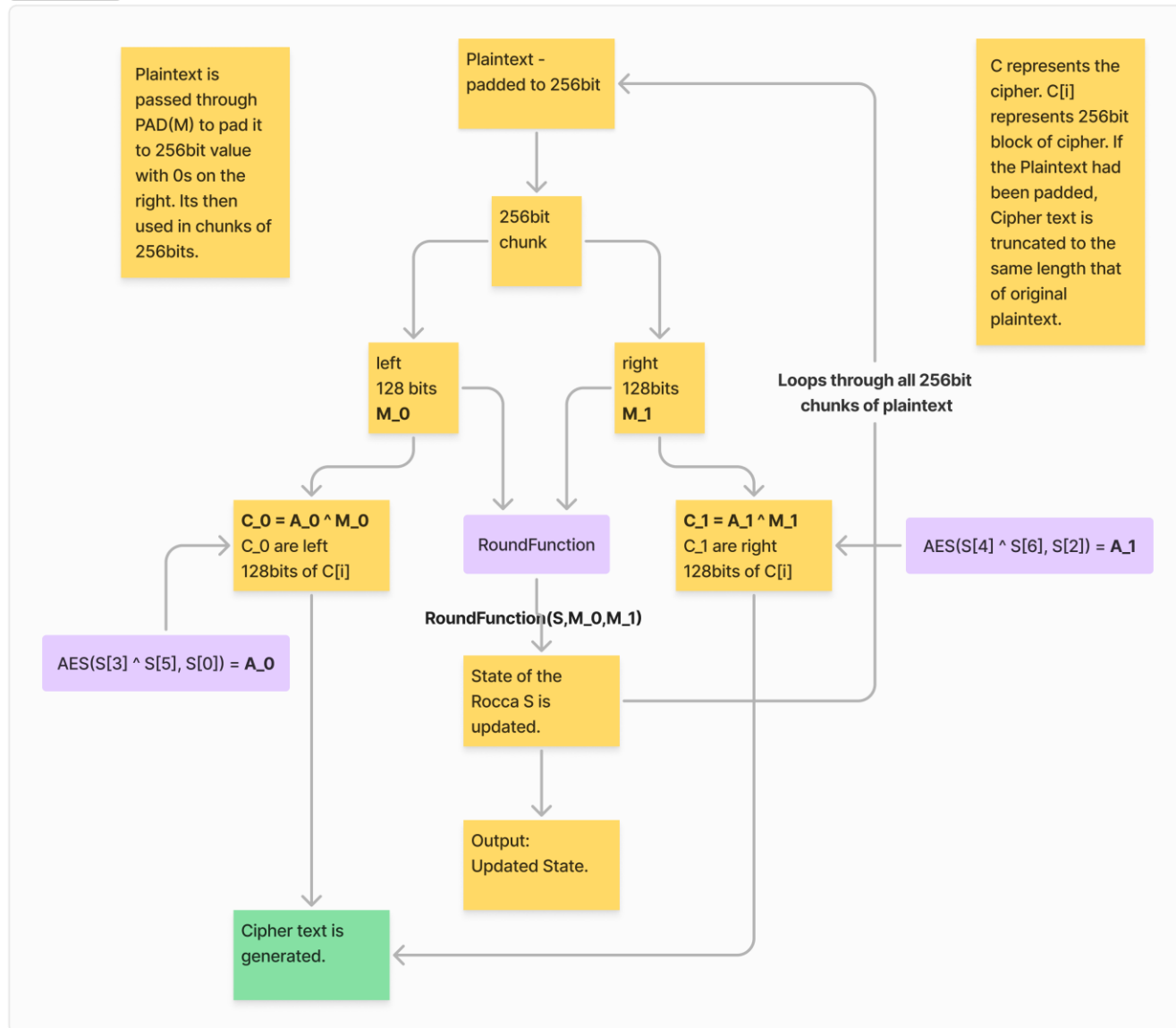
Encrypt - Overview

Encrypt - Overview



Encrypt - Details

Encrypt - Details



Finalize - Overview

Finalize - Overview

Size of AD before padding :

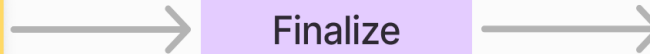
Size_AD

Size of plaintext before padding :

Size_M

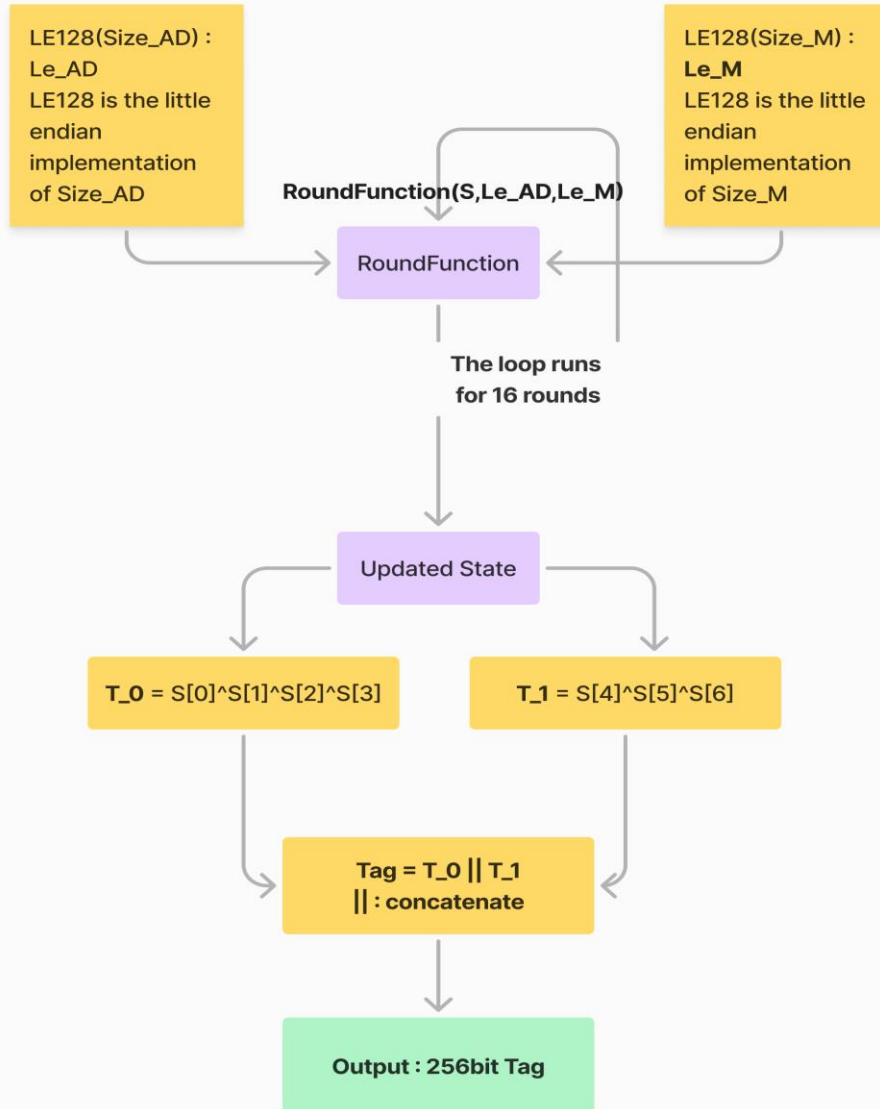
Finalize

Updated States
of Rocca S
256bit tag - T is
generated



Finalize - Details

Finalize - Details



Outputs

- The Cipher text that's generated is truncated to the same length as the original message.
- The Tag generated is a 256-bit value used for Authentication or Integrity?