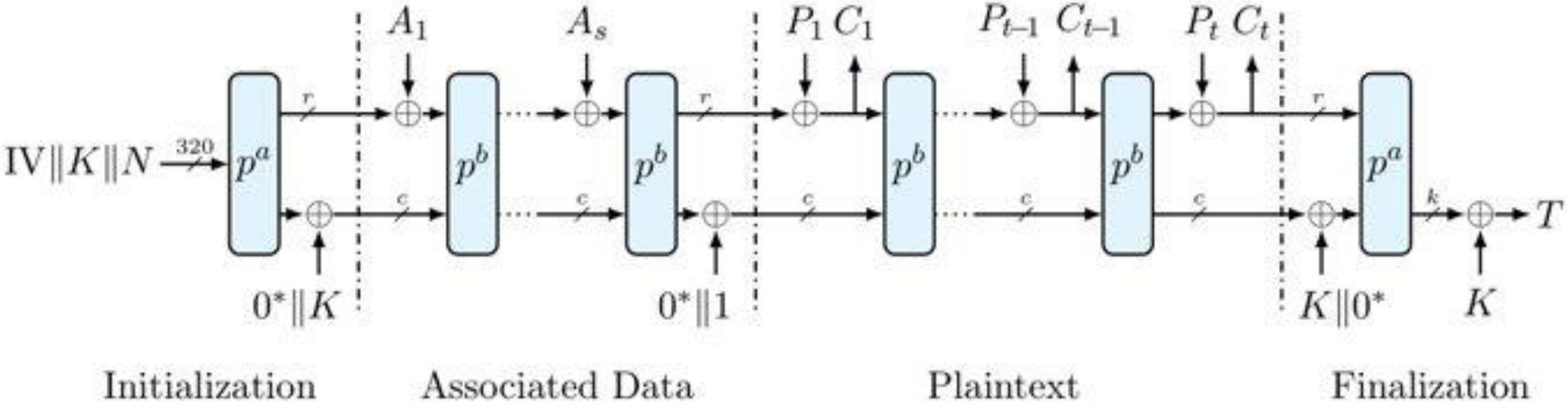
Hardware Implementation of ASCON

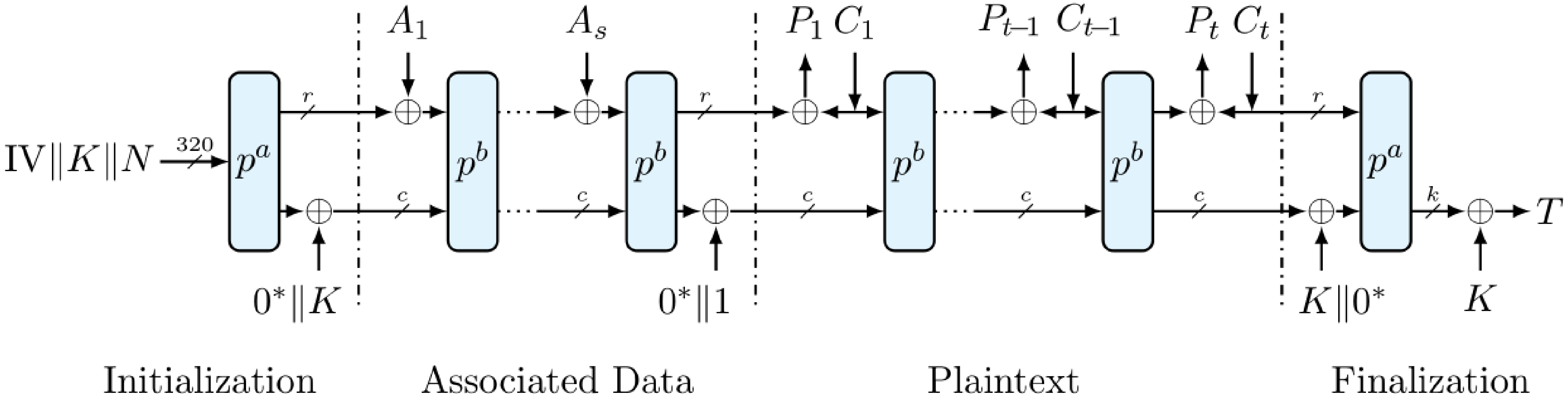
ASCON Description

- ASCON family has 2 variants – ASCON 128 (block size = 64) and ASCON 128a (block size = 128).
- Both of those take 128-bit key and nonce and have a 320-bit state.
- A 128-bit tag is generated after encryption + tag generation and it is verified in decryption + tag verification.
- In the ASCON permutation, the 320-bit state is divided into 5 registers: x0, x1, x2, x3 and x4.
- One round of the permutation consists of 3 different layers:
  - Round Constant Layer: A constant term which depends on the round number is added to x2 register
  - Substitution Layer: A 5-bit SBox operation is applied to each bit-slice comprising of 5 bits from all the registers
  - Linear Diffusion Layer: The bits of each register are internally shuffled with the help of right rotation and XOR.

- The ASCON encryption/decryption is divided into four sub-routines:
    - Initialization: A state of 320 bits is created by concatenating the fixed initialization vector, key, and the nonce, which are then passed through a rounds of permutation and XOR operation
    - Processing Associated Data: Associated data is absorbed in the algorithm by dividing it into datasets of r bits each and the last dataset is padded with a 1 followed by 0’s to make the length equal to r.
    - Processing Plain/Cipher Text: It is similar to the previous routine, but a cipher text is generated for encryption and a plain text for decryption.
    - Finalization: Generates a tag in encryption which is used in the finalization stage of decryption to verify if the processed data is correct.
- The ASCON hashing is based on sponges and is divided into three sub-routines:
    - Initialization: A 320-bit state is created by using the initialization vector
    - Absorbing Message: The message is absorbed into the algorithm similar to the plain text processing stage mentioned
    - Squeezing: The hash data is generated from multiple permutation stages



ASCON Encryption



ASCON Decryption

Side Channel Attacks

- Side channel attacks pose a significant threat to cipher security, particularly through power consumption or electromagnetic emanation.
- Ciphers with strong classical security claims often fall short against side channel attacks.
- Understanding these attacks and developing low-cost countermeasures are research priorities.
- Masking is a prominent countermeasure that introduces randomness to break the linkage between models and leaked intermediate variables.
- Depending on the strength of the attacker, various degrees of masking can be adopted.

SCA Countermeasures

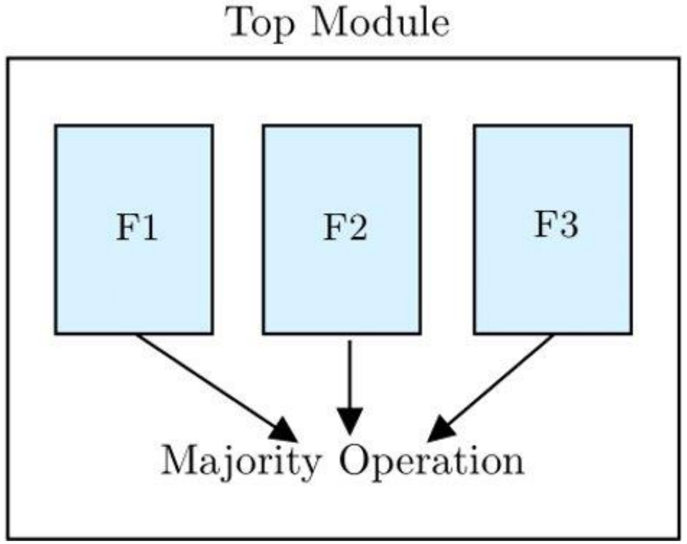
- Threshold Implementation (TI) is one of the prominent side channel protection
- For example, the ASCON permutation state is divided into three shares The shares must obey the following rule: S = S0 + S1 + S2 where + represents XOR operation and S represents the state.
- Each share has a distinct round constant layer, substitution layer, and linear layer that are cleverly designed so that the output of all three shares can be merged at the conclusion of the permutation phase to yield the same state value as in unprotected ASCON.
- The round constant and linear layers are linear processes, while the substitution layer uses a non-linear SBox.
- The minimum number of shares needed is 1 more than the algebraic degree of the SBox, thus we need atleast 3 shares.

Fault Injection

- Fault injection attacks exploit error propagation, making redundancy a crucial defence mechanism.
- Replicating the same circuit in temporal or spatial domains helps combat fault injection attacks.
- Depending on the power consumption and complexity, we can either use duplication or triplication:
  - Duplicate and compare technique is effective against Differential Fault Attacks (DFA).
  - Triplicate and majority-based approach is suitable for countering Statistical Ineffective Fault Attacks (SIFA), though duplication based countermeasures exist.
- The implemented countermeasure involves executing all procedures multiple times and selecting the majority output, with a random number as output in cases of disagreement.

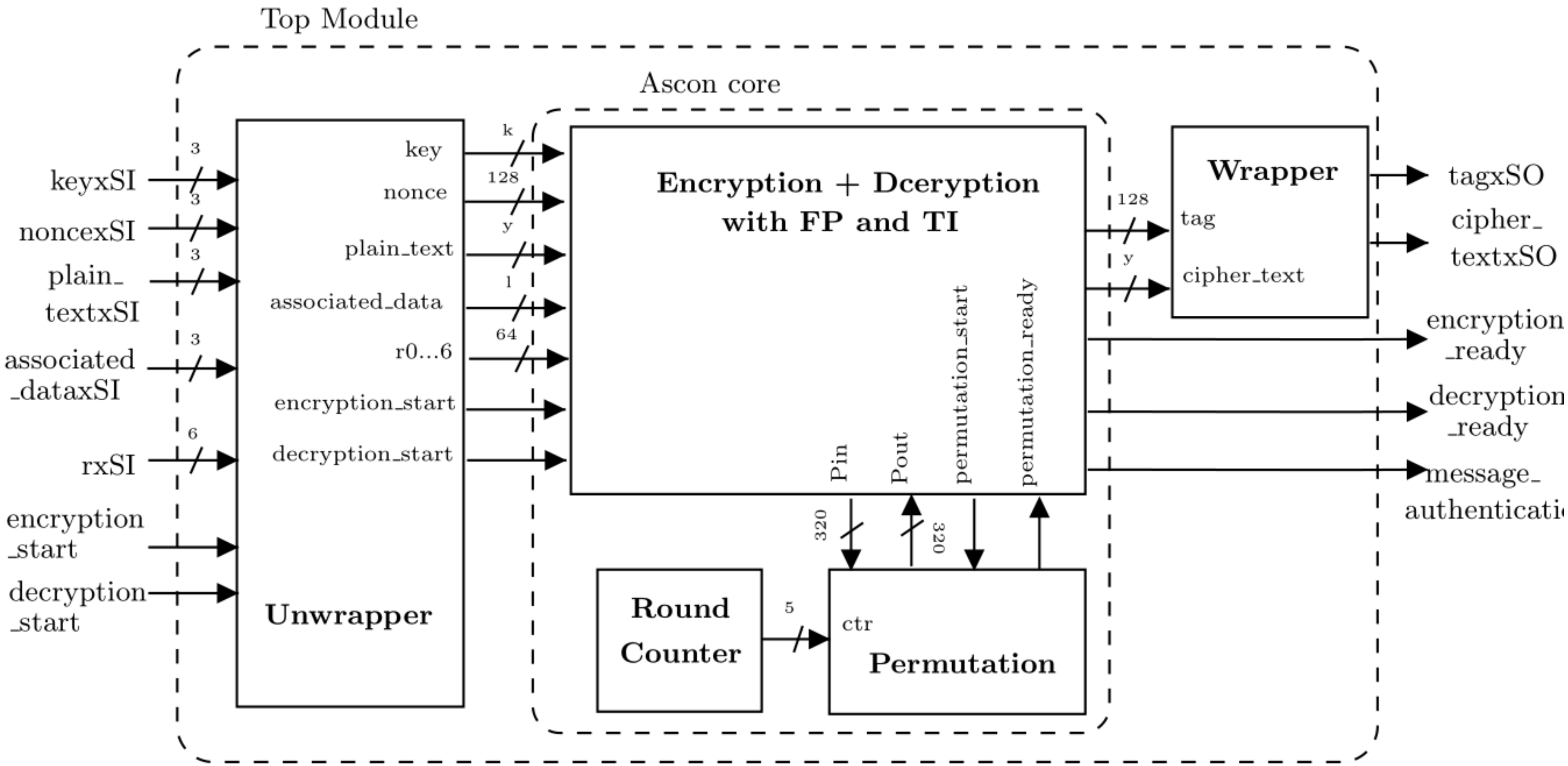
FIA Countermeasures

- In the triplication-based countermeasure, all the procedures are executed thrice, and the final output is determined by selecting the majority output from the three.
- In cases where all three outputs differ, a random number is produced as the output.



Architecture

- The input data consist of the key, 128-bit nonce, associated data (AD), plain text (PT), control signals and random numbers.
- The output data consists of the cipher text (CT), 128-bit tag and ready signals to indicate the end of the processing.
- *keyxSI* signal is of width 3 bits. The LSB bit carries the key information, and the other 2 bits carry random numbers, which is utilised for threshold implementation.
- *noncexSI*, *plain_textxSI*, *associated_dataxSI* signals are of width 3 bits, and the distribution of the bits is similar to key.
- *encryption_startxSI* and *decryption_startxSI* are 1-bit control pulses that signal the start of encryption/decryption, respectively.
- *rxSI* signals is of width 7 bits carrying random numbers which is utilized for threshold implementation.



ASCON Architecture

Parameters

- The code can be customized by adjusting parameters such as key size, block size, and internal permutation rounds based on the desired ASCON variant (128 or 64).
- The lengths of associated data and plaintext can be modified to accommodate different inputs, including the option of having them as zero.
- Four configurations are available: unprotected ASCON, ASCON with TI (Threshold Implementation), ASCON with FP (Fault Protection), and ASCON with both TI and FP.
- These configurations offer flexibility in tailoring ASCON to specific security requirements and performance considerations.

Benchmarks

Design	Cells	Area (µm2)	Critical path (ps)	Dynamic power (nW)
Unprotected	7157	98524	8520	762520.083
ASCON with Fault Protection	19150	258224	8518	2346943.378
ASCON with TI	26248	364320	9830	4369303.426
ASCON with TI and Fault Protection	69692	948544	9832	11124794.73

ASIC benchmarks for protected and unprotected ASCON (STM 130nm)

Design	LUT	Registers	Slice	LUT logic	Max freq. (MHz)
ASCON Hash	870	912	313	870	203
ASCON Hash with TI	3930	2706	1169	3930	171

FPGA benchmarks for unprotected and protected versions of ASCON hash (Kintex-7)

THANK YOU