



**HAL**  
open science

## Scan chain encryption in Test Standards

Mathieu da Silva, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre

► **To cite this version:**

Mathieu da Silva, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre. Scan chain encryption in Test Standards. SURREALIST: SecURity, REliAbiLity, test, prIvacy, Safety and Trust of Future Devices, May 2018, Bremen, Germany. , Workshop on SecURity, REliAbiLity, test, prIvacy, Safety and Trust of Future Devices, 2018. <lirmm-01882578v2>

**HAL Id: lirmm-01882578**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-01882578v2>**

Submitted on 10 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization



# SCAN CHAIN ENCRYPTION IN TEST STANDARDS

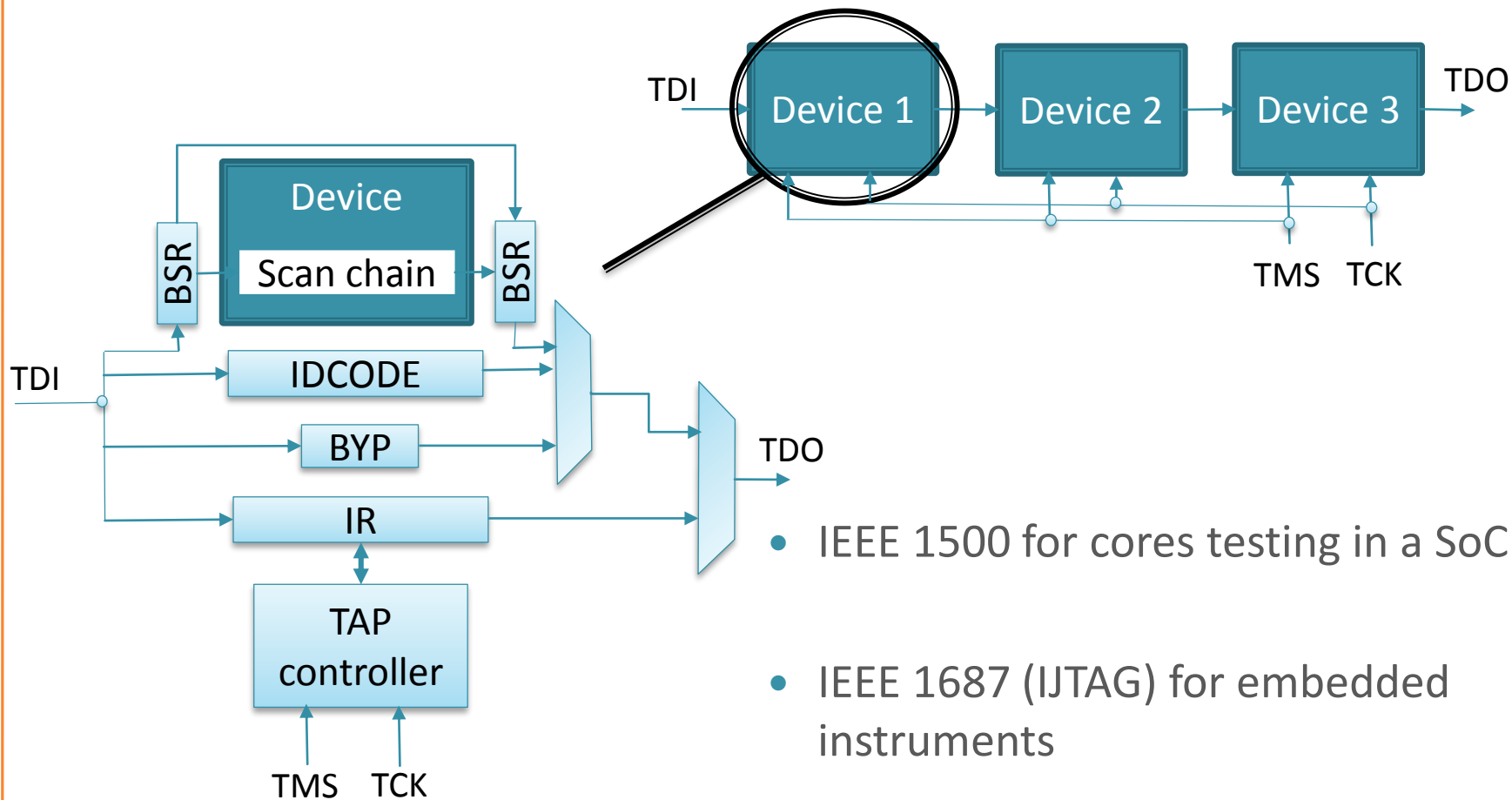
Mathieu Da Silva, Marie-Lise Flottes, Giorgio Di Natale, Bruno Rouzeyre

SURREALIST 2018

# CONTEXT

- Test standards

- IEEE 1149 (JTAG) for board testing

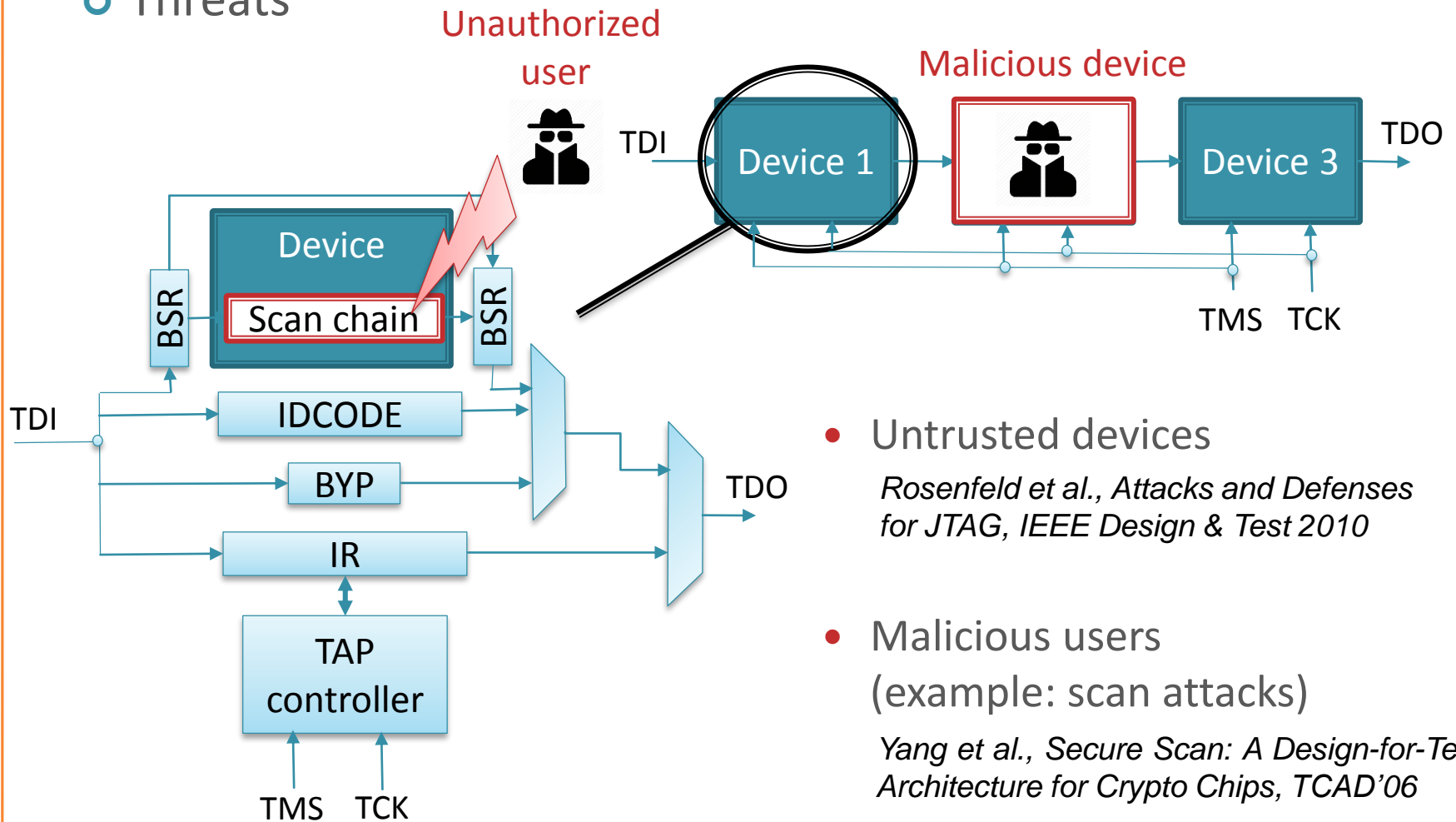


- IEEE 1500 for cores testing in a SoC
- IEEE 1687 (IJTAG) for embedded instruments



# CONTEXT

## ○ Threats



- Untrusted devices

*Rosenfeld et al., Attacks and Defenses for JTAG, IEEE Design & Test 2010*

- Malicious users  
(example: scan attacks)

*Yang et al., Secure Scan: A Design-for-Test Architecture for Crypto Chips, TCAD'06*



# SUMMARY

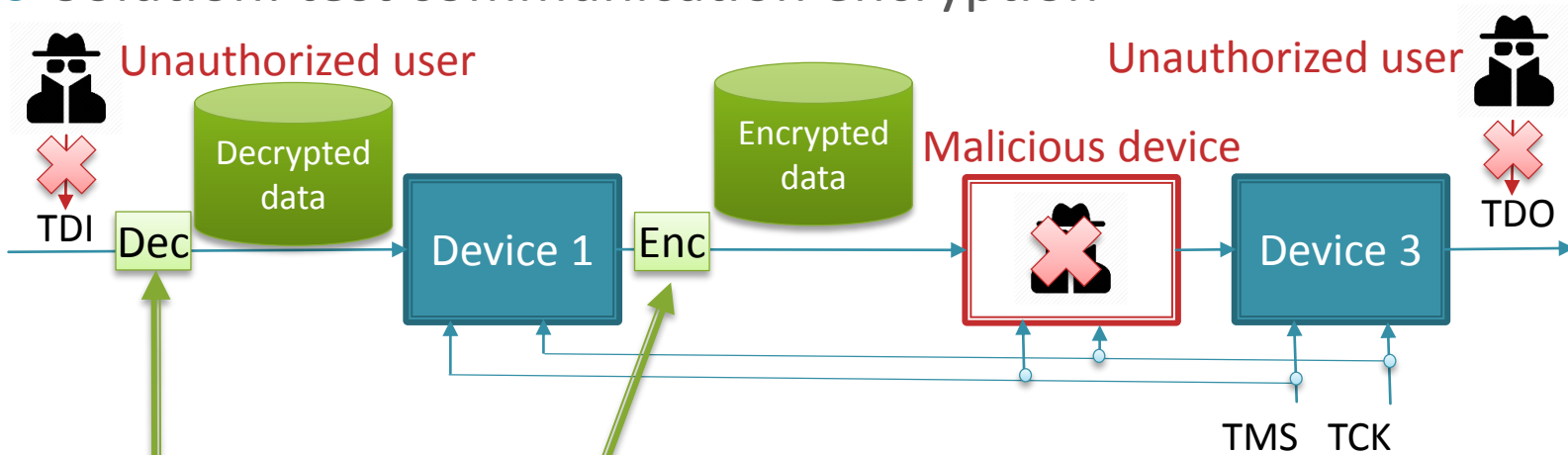
---

- 1) **Scan chain encryption**
- 2) State-of-the-art based on test communication encryption
- 3) Implementation with block cipher
- 4) Implementation with stream cipher
- 5) Conclusion



# SCAN CHAIN ENCRYPTION

## ○ Solution: test communication encryption

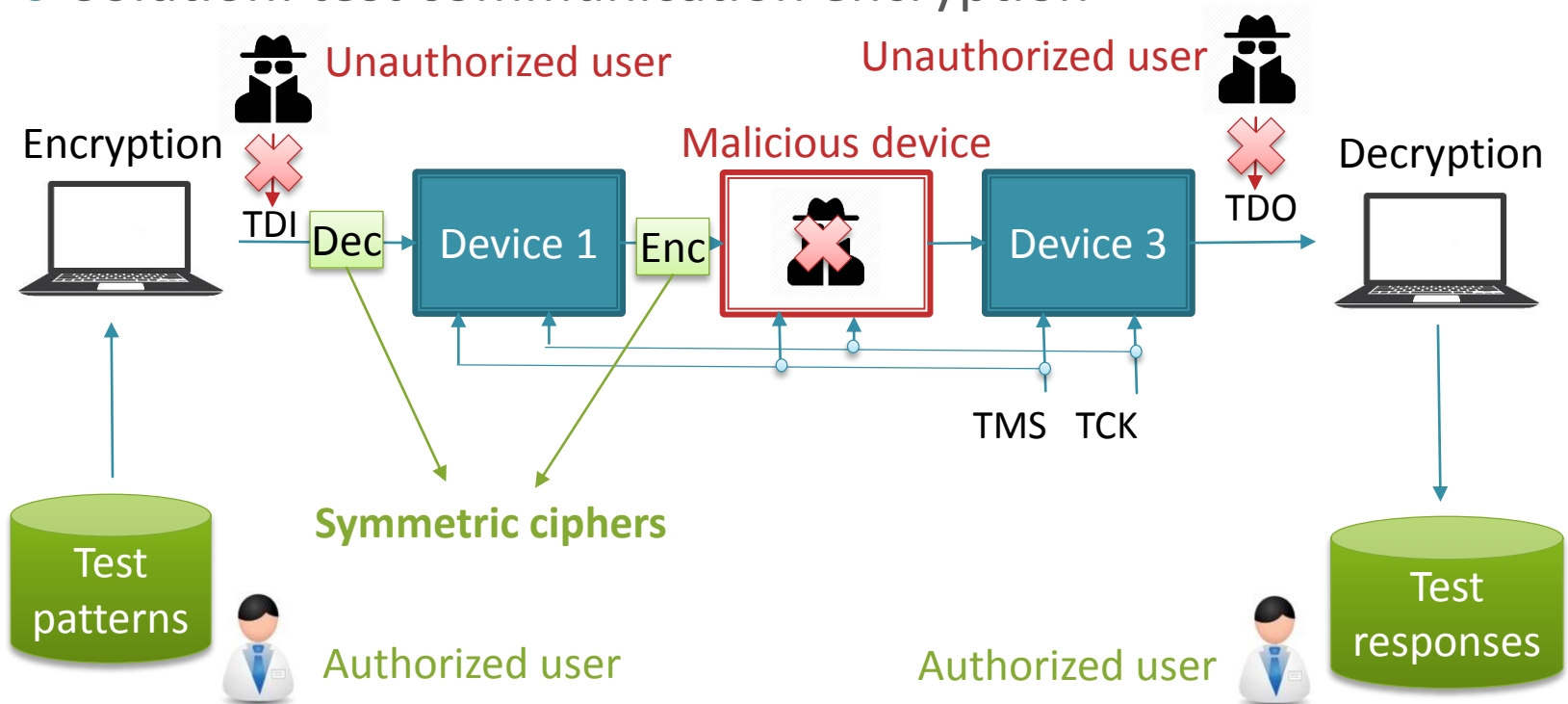


- **Input decryption** prevents sending desired test data
- **Output encryption** prevents reading plain test responses



# SCAN CHAIN ENCRYPTION

## ○ Solution: test communication encryption

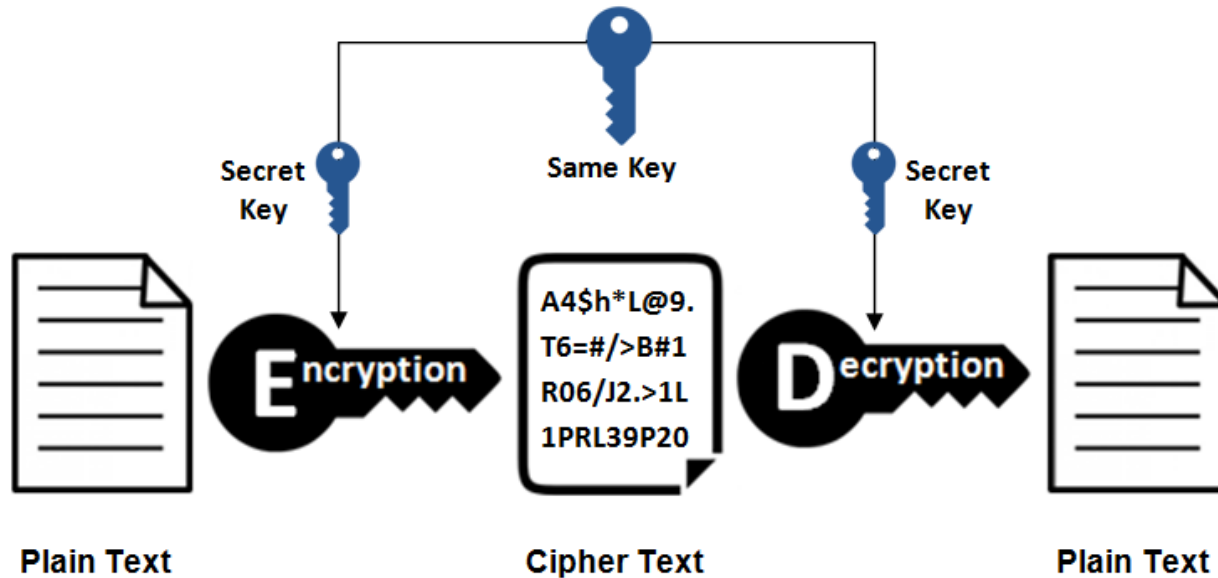


- Input decryption prevents sending desired test data
- Output encryption prevents reading plain test responses
- Test/debug only possible by authorized user knowing the secret key



# SYMMETRIC CIPHER

## Symmetric Encryption



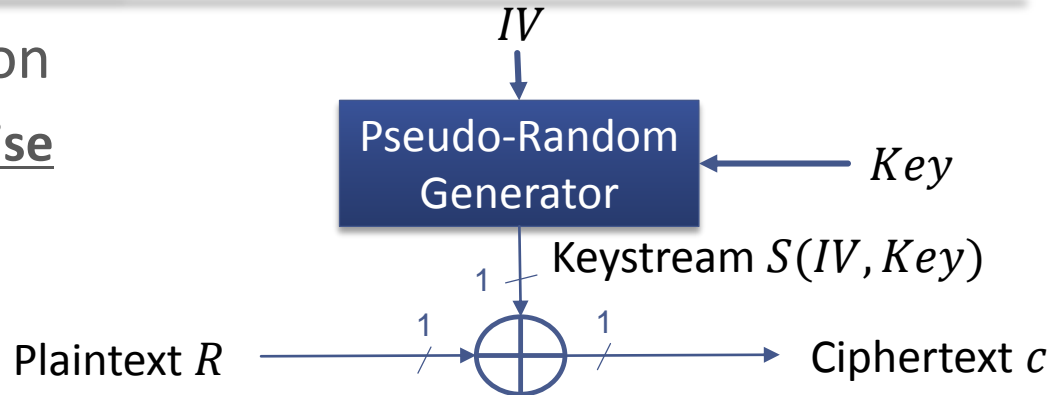
- 2 types of symmetric cipher: stream and block ciphers



# STREAM CIPHER / BLOCK CIPHER

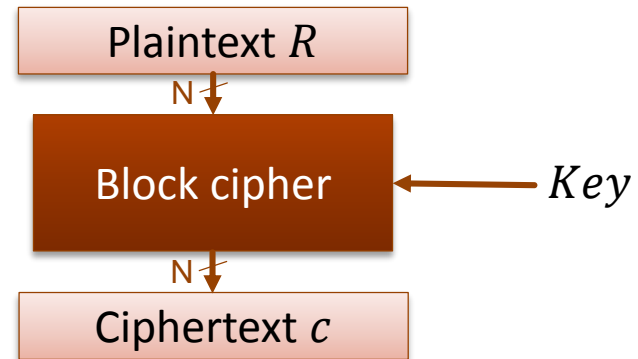
## Stream cipher encryption

- Keystream XORed **bitwise** with the plaintext



## Block cipher encryption

- Confusion and diffusion on a **block** of plaintext



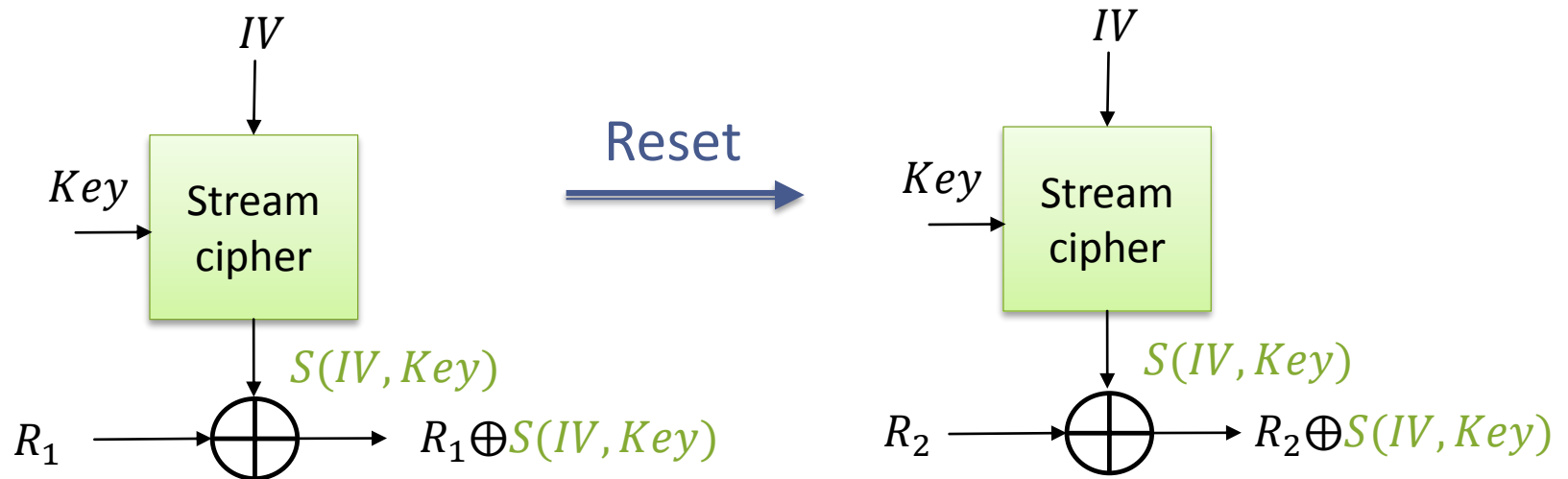
## Preference for stream ciphers

- "Naturally" adapted to serial test communication (JTAG, IEEE 1500, IJTAG)
- Smaller area footprint compared to block ciphers
- But ..



# TWO-TIMES PAD: STREAM CIPHER REQUIREMENT

- Two-times pad: same key and  $IV$  re-used  $\Rightarrow$  same keystream generated to encrypt different data



$\Rightarrow$  Possible to carry out attacks if requirement is not fit

$$R_1 \oplus \cancel{S(IV, Key)} \oplus R_2 \oplus \cancel{S(IV, Key)}$$

$\Rightarrow$  Solution:  $IV$  generated randomly at each circuit reset

$$R_1 \oplus S(IV_1, Key) \oplus R_2 \oplus S'(IV_2, Key)$$



# SUMMARY

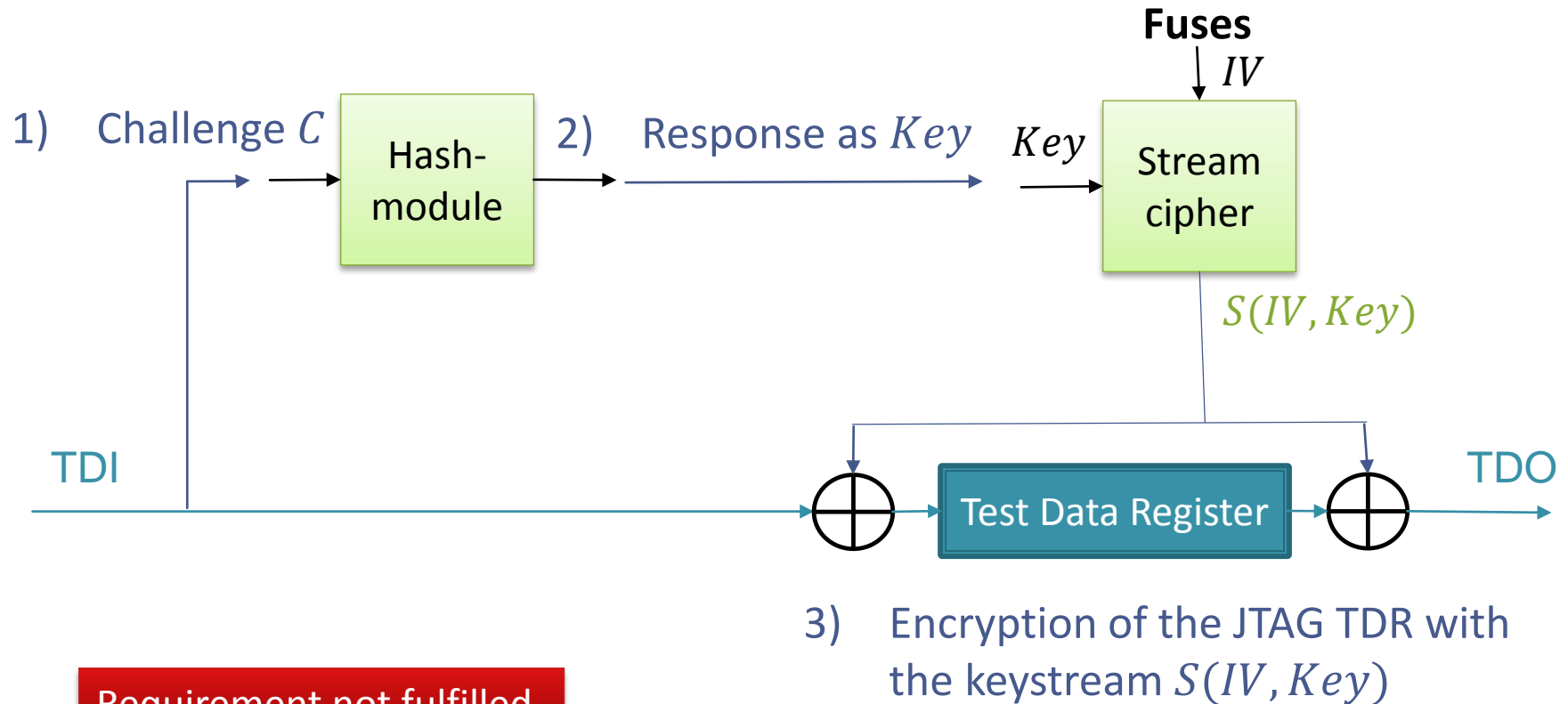
---

- 1) Scan chain encryption
- 2) State-of-the-art based on test communication encryption**
- 3) Implementation with block cipher
- 4) Implementation with stream cipher
- 5) Conclusion



# STREAM-BASED ENCRYPTION ON JTAG INTERFACE

- Challenge/Response protocol to encrypt JTAG test communication



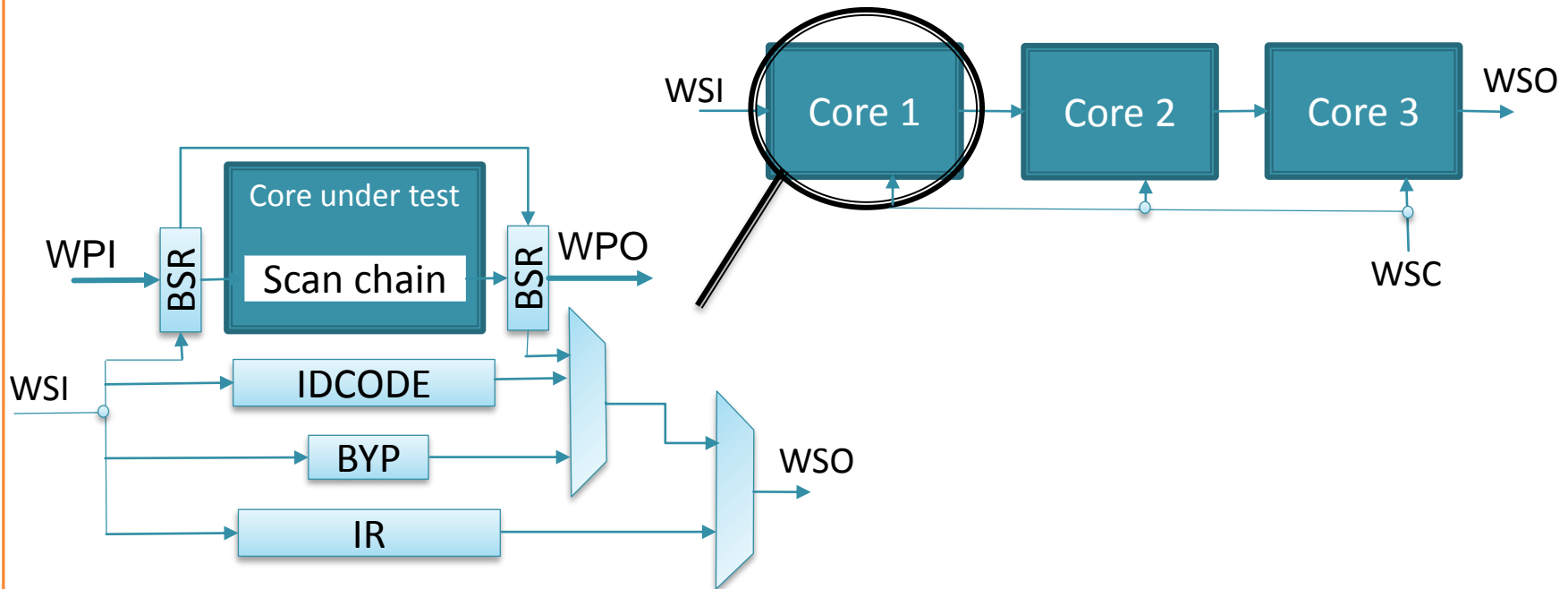
Requirement not fulfilled



# STREAM-BASED ENCRYPTION ON IEEE 1500 INTERFACE

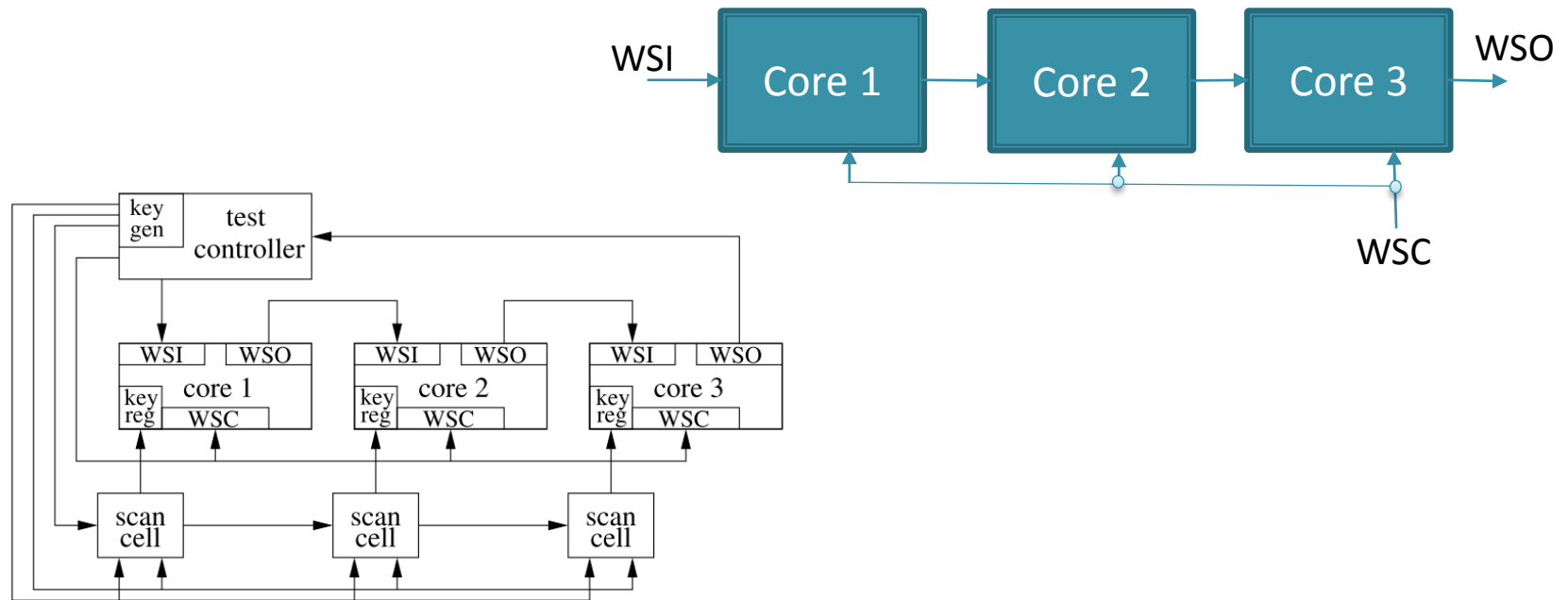
- IEEE 1500 standard

- Similar as JTAG standard, but for SoC wrappers
- Parallel test inputs WPI and parallel test outputs WPO



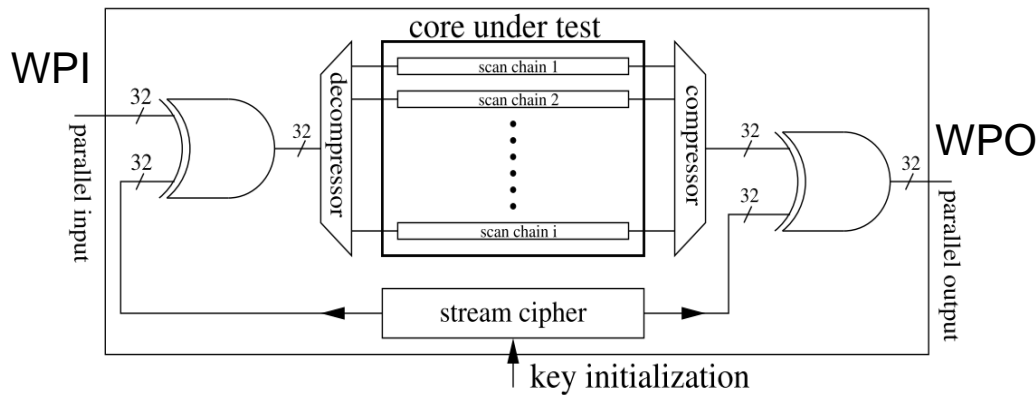
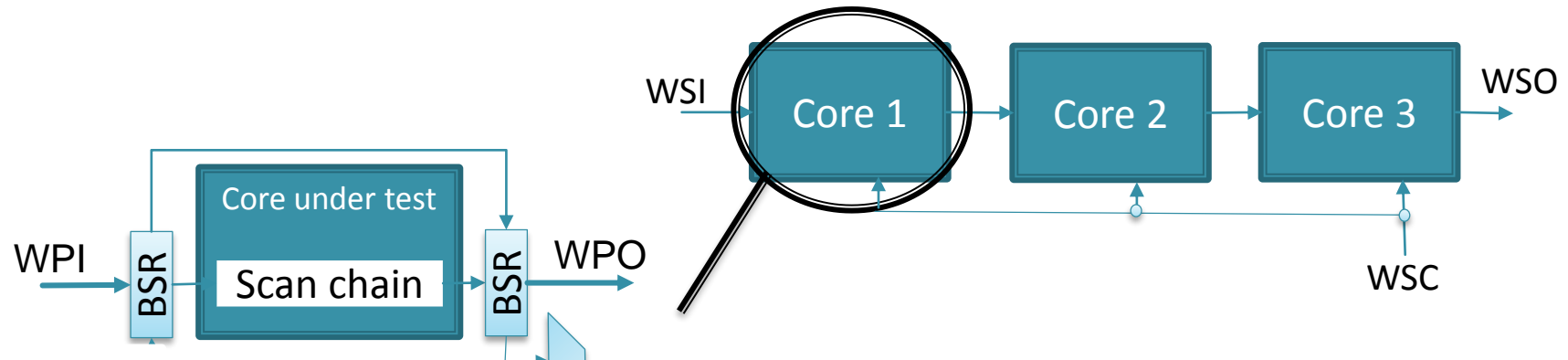
# STREAM-BASED ENCRYPTION ON IEEE 1500 INTERFACE

- Encrypt test data on a targeted core (IEEE 1500)
  - 1) Send the key to the core via specific scan chain non-visible from the others cores



# STREAM-BASED ENCRYPTION ON IEEE 1500 INTERFACE

- Encrypt test data on a targeted core (IEEE 1500)
  - 1) Encrypt the parallel input/output (WPI and WPO)

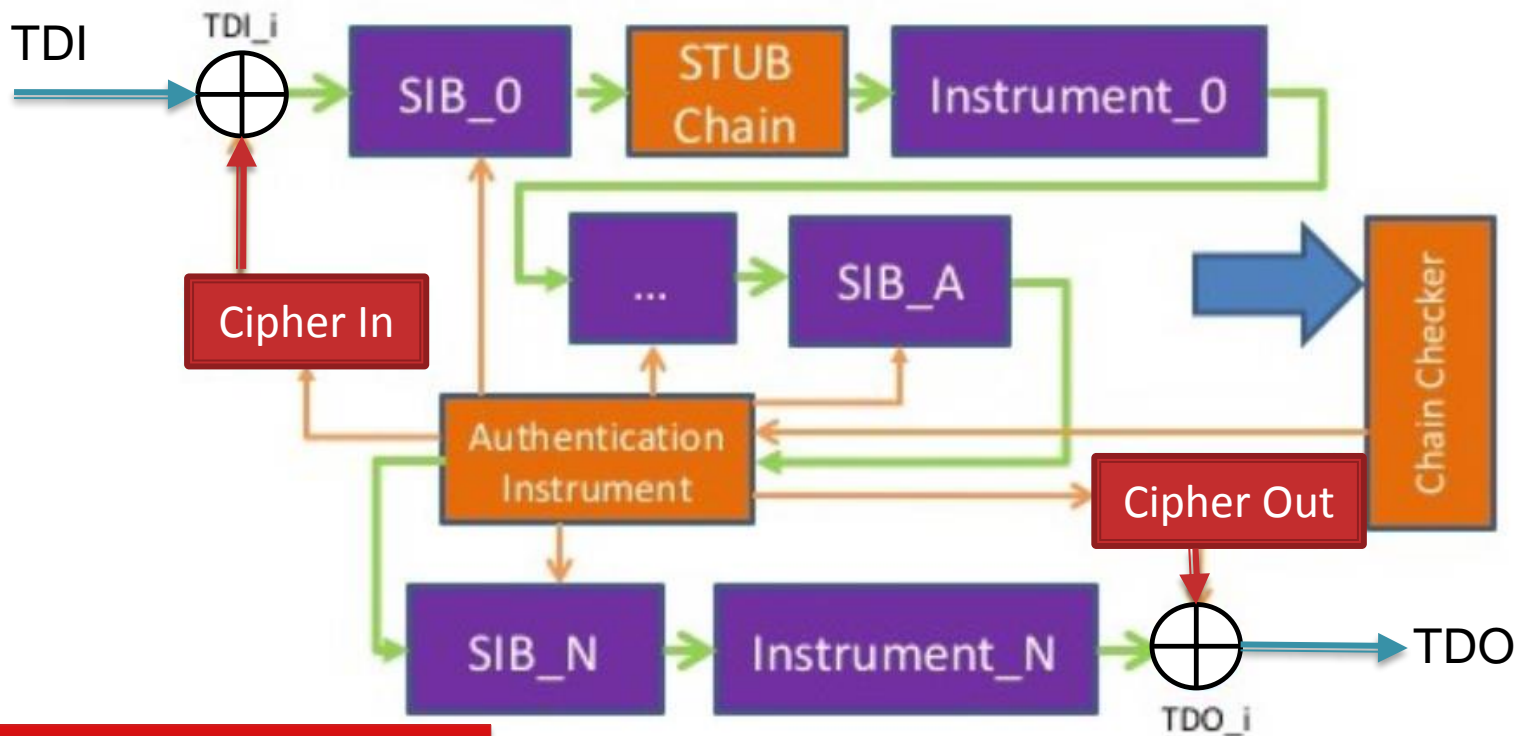


Requirement not fulfilled



# STREAM-BASED ENCRYPTION ON JTAG INTERFACE

- Encryption of Test Data Register associated to Instruments in the JTAG network

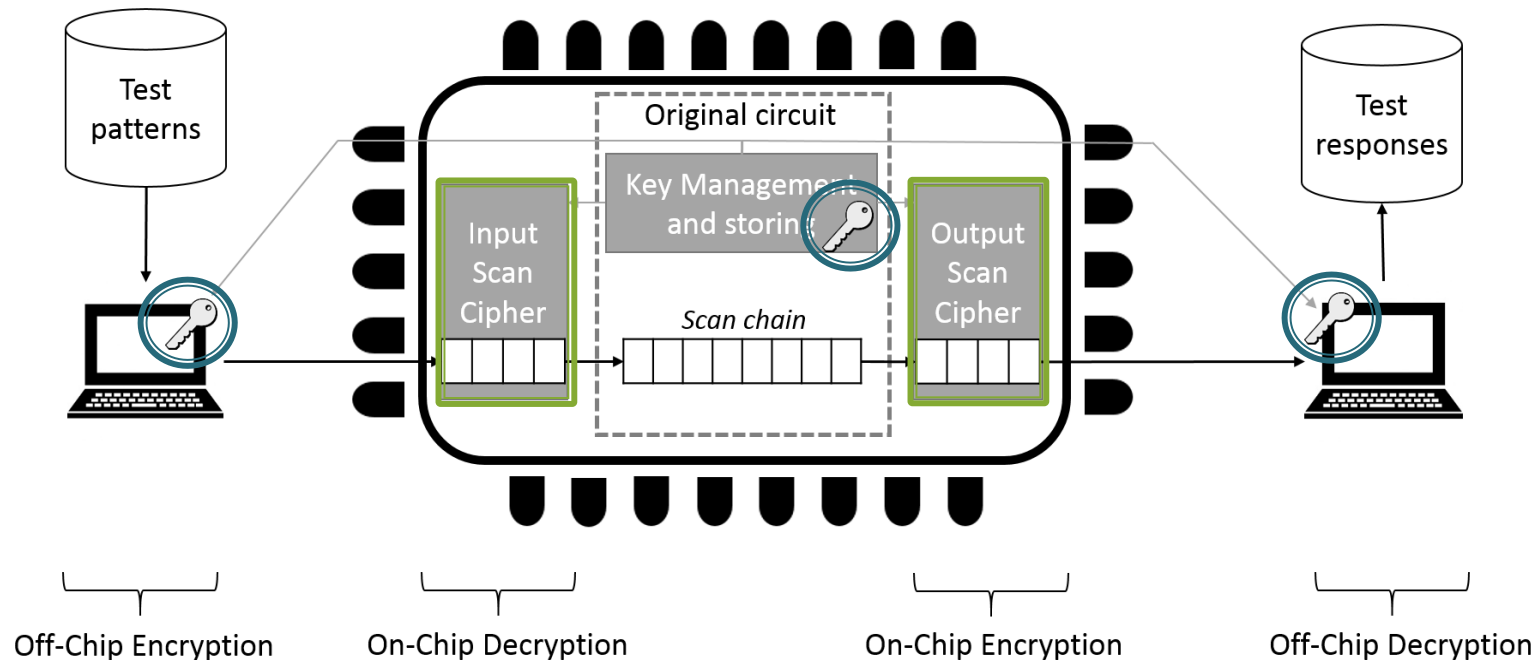


Requirement not fulfilled



# OUR PROPOSITION

- Insertion of block or stream ciphers at Scan-In and Scan-Out



- Assumption: original circuit embedded a crypto-core with its key management and storing
- Scan chain encryption solution shares the key management and storing already implemented



# SUMMARY

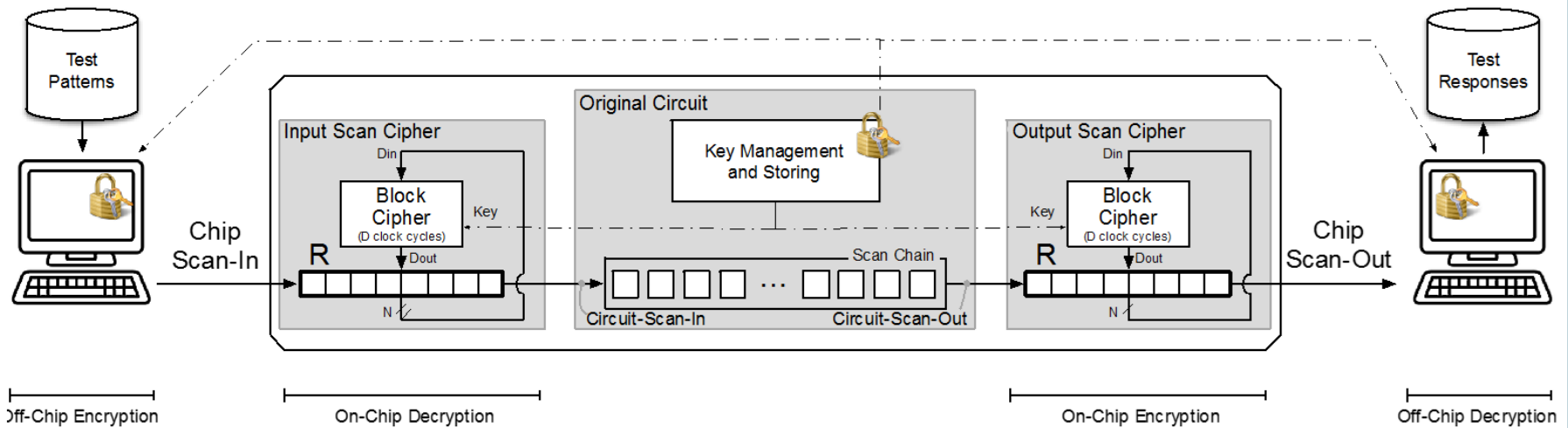
---

- 1) Scan chain encryption
- 2) State-of-the-art based on test communication encryption
- 3) Implementation with block cipher**
- 4) Implementation with stream cipher
- 5) Conclusion



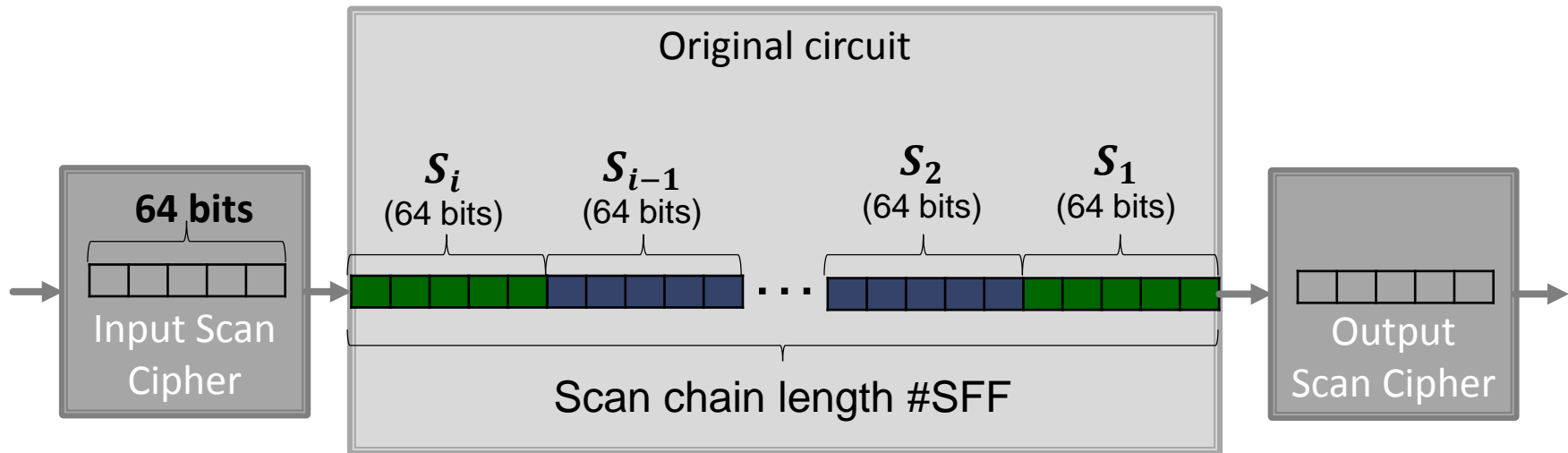
# BLOCK CIPHER-BASED SCAN ENCRYPTION

- Implementation on scan chain with 2 PRESENT block ciphers:
  - Lightweight (1 PRESENT = 2 139 GE)
  - Encryption by 64-bits block size



# MODE OF OPERATIONS

- 64 bits encrypted every 32 clock cycles



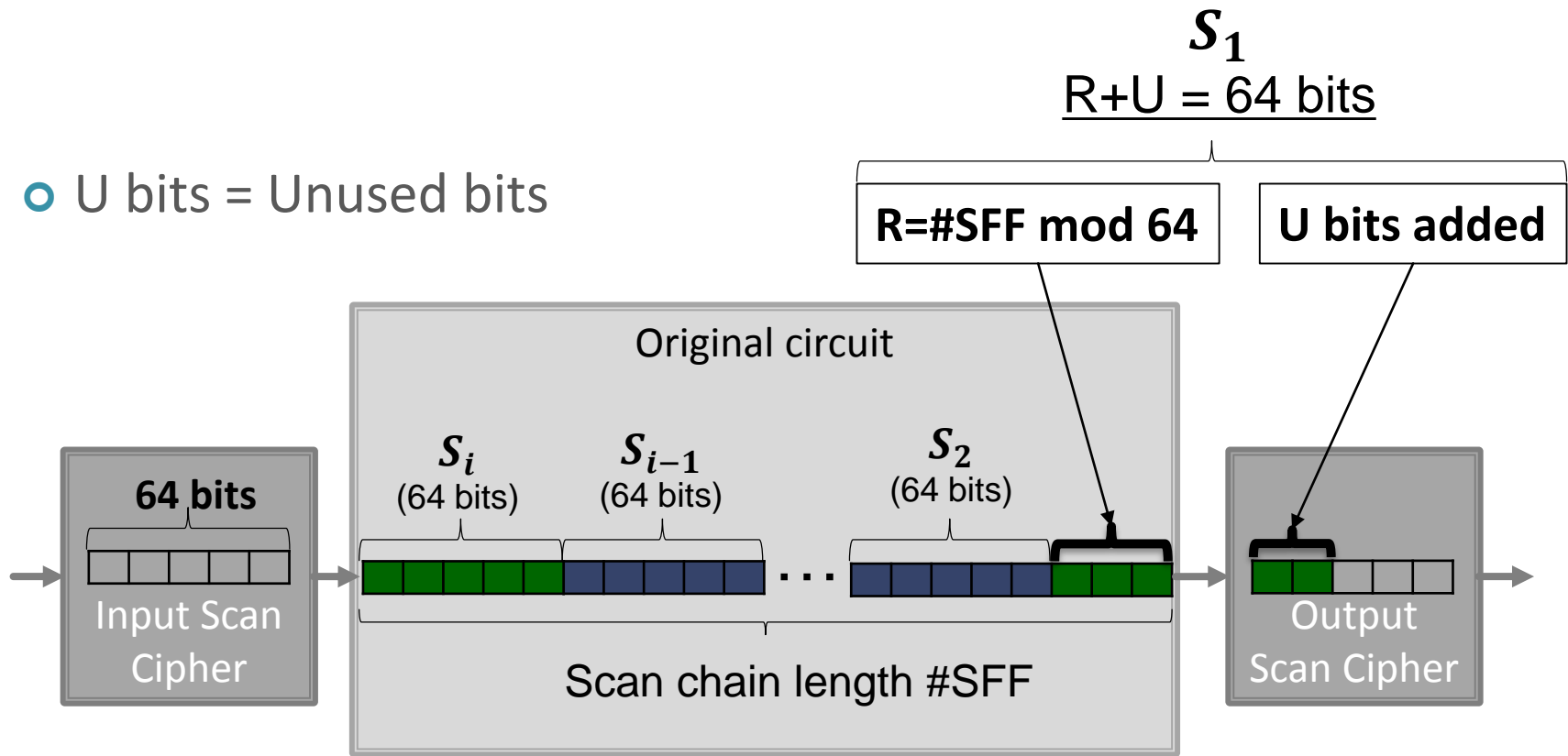
⇒ **#SFF = P x 64**

⇒ **No test time overhead on each pattern**



# MODE OF OPERATIONS

- U bits = Unused bits



$\Rightarrow \#SFF = P \times 64 + R$

$\Rightarrow$  Loss of U clock cycles per pattern



# SUMMARY

---

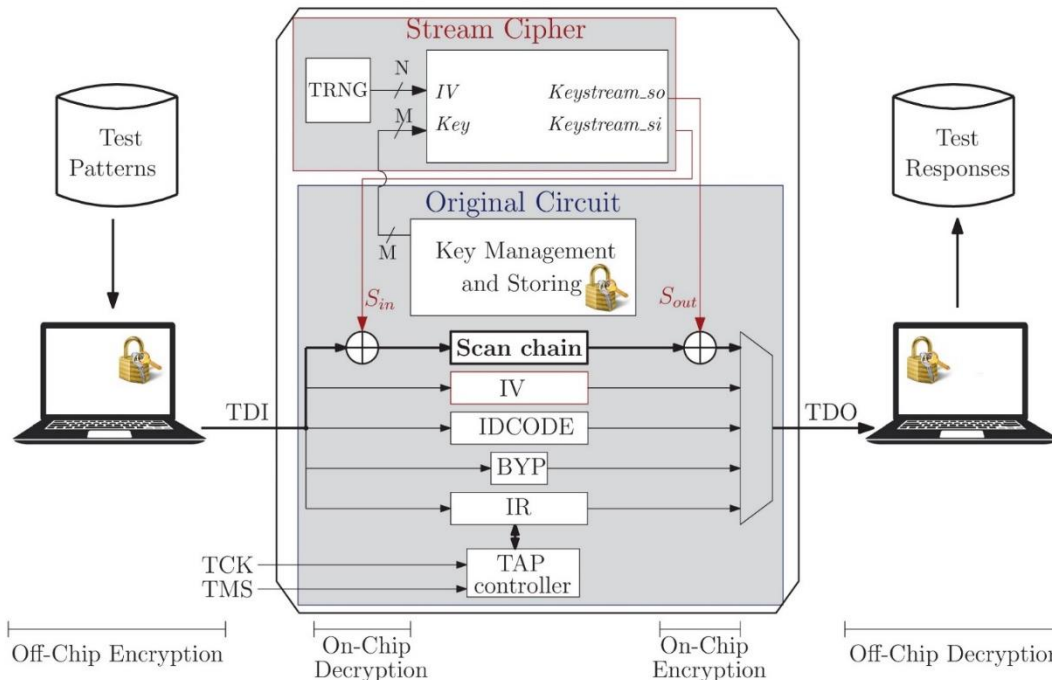
- 1) Scan chain encryption
- 2) State-of-the-art based on test communication encryption
- 3) Implementation with block cipher
- 4) Implementation with stream cipher**
- 5) Conclusion



# STREAM CIPHER-BASED SCAN ENCRYPTION

## Implementation on JTAG:

- 1 TRIVIUM stream cipher (2 016 GE)
- TRNG to generate random IV
- New instruction *GetIV* with a test data register IV

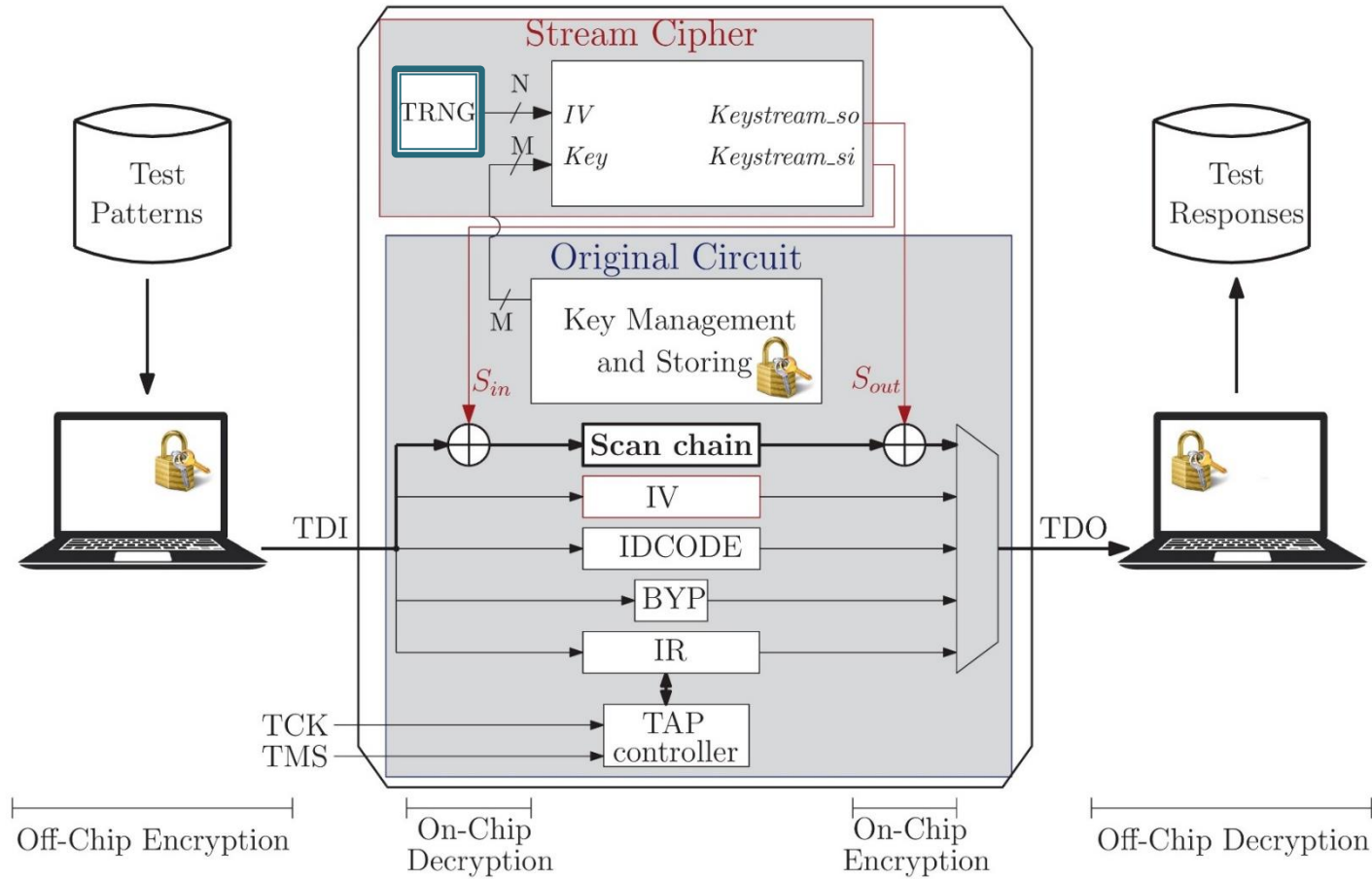


## Mode of operations in 2 phases: initialization and encryption

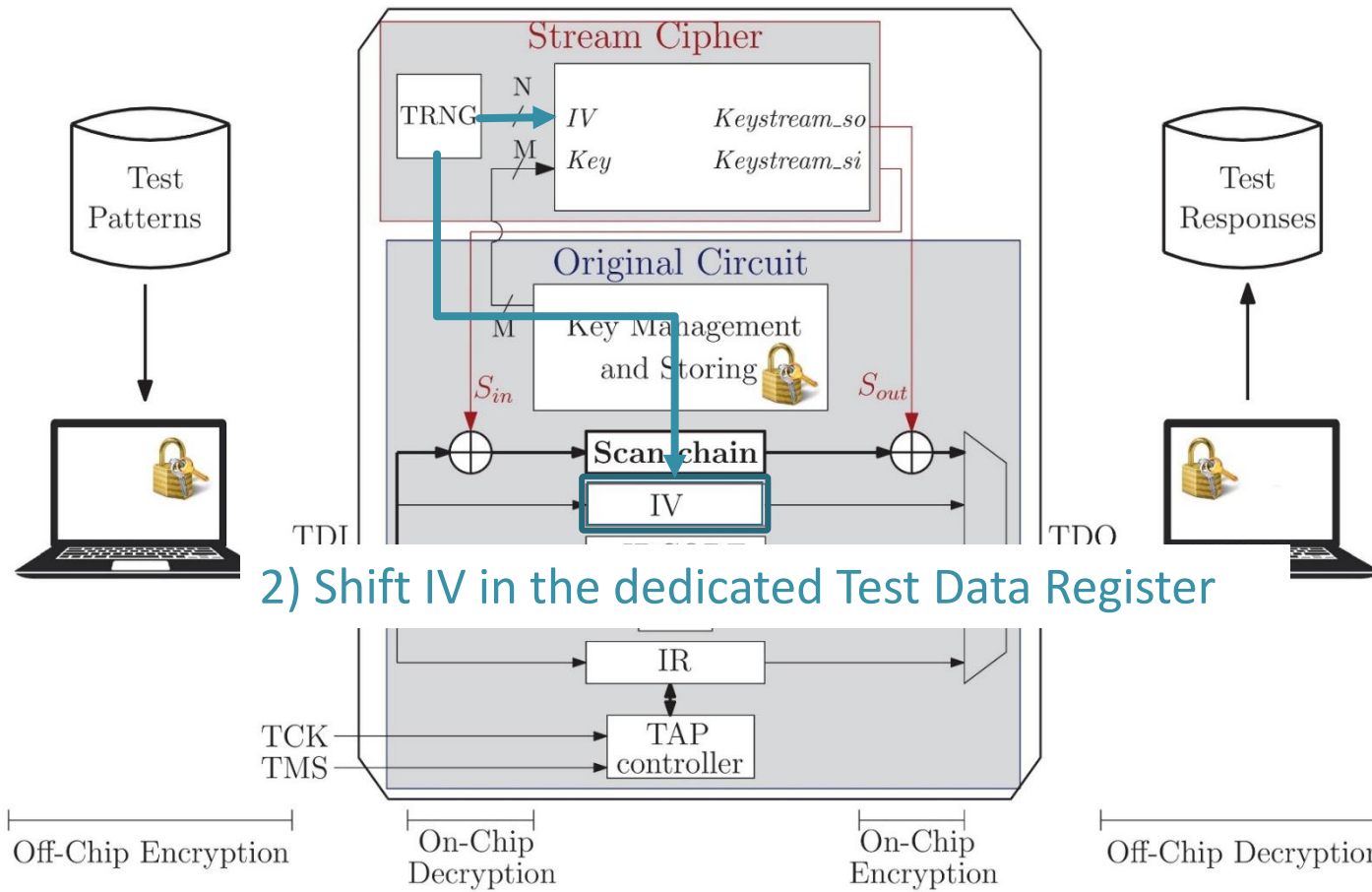


# INITIALIZATION PHASE

## 1) TRNG initialization: reach sufficient entropy to generate random number

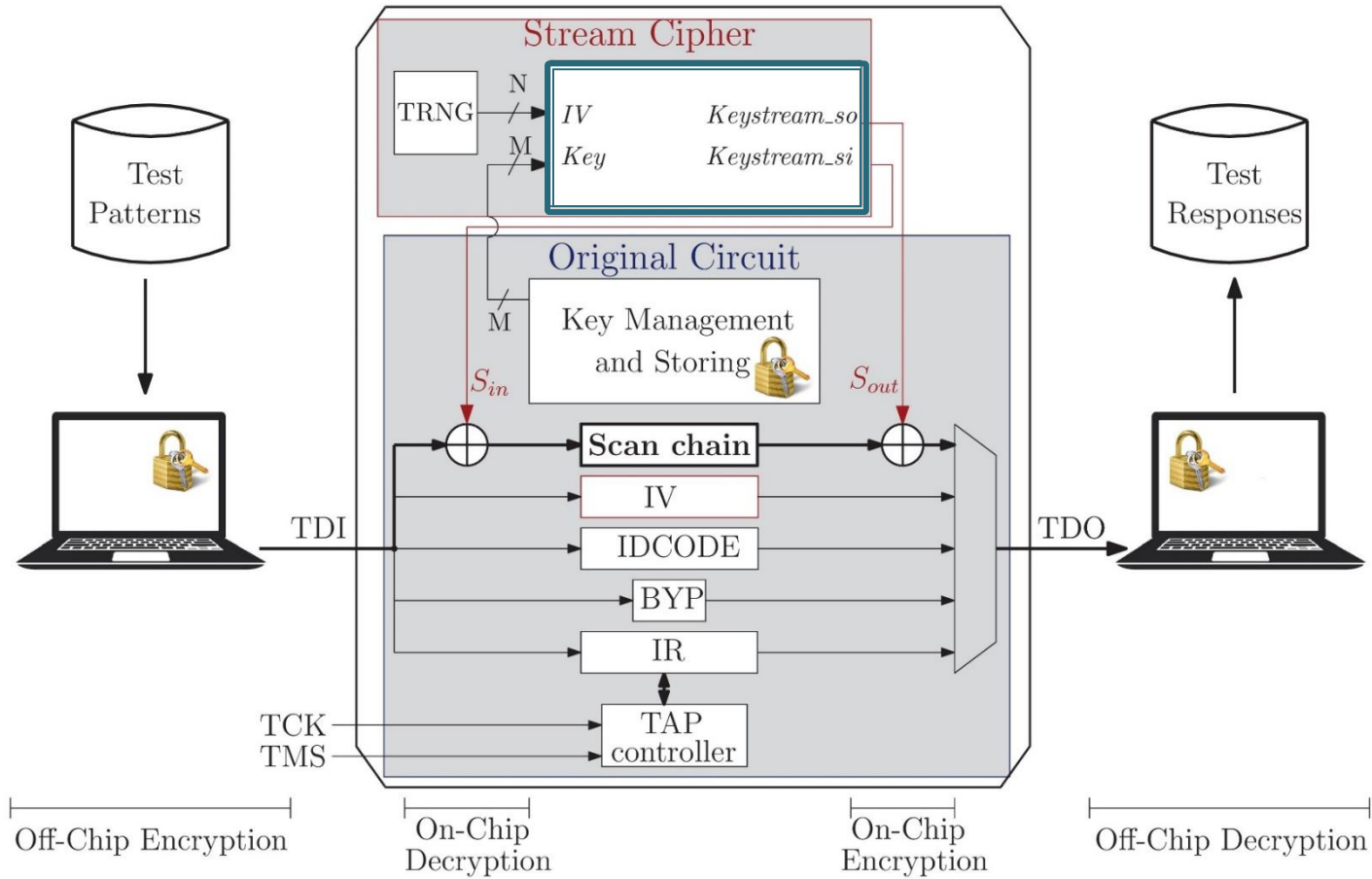


# INITIALIZATION PHASE

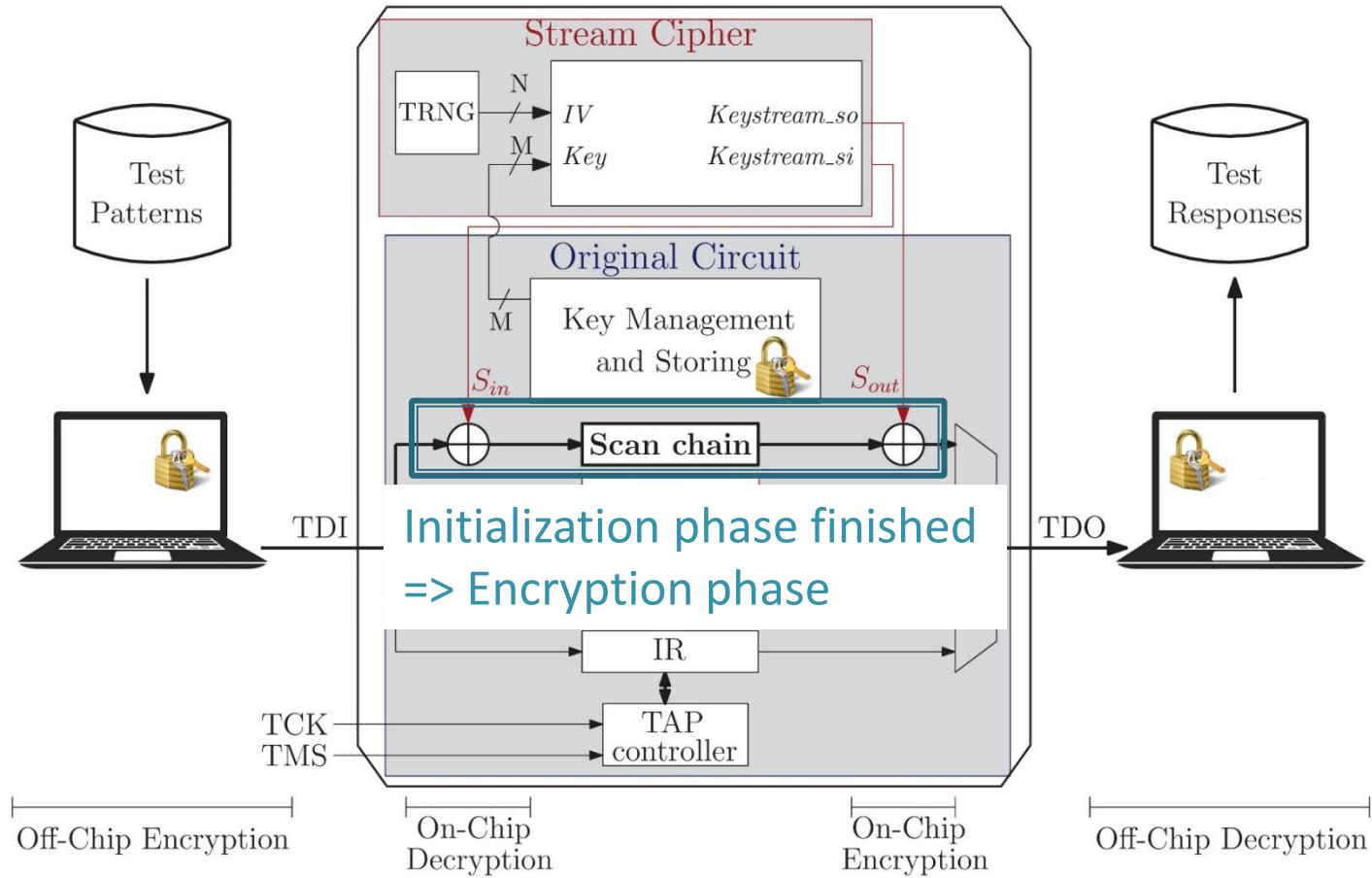


# INITIALIZATION PHASE

## 3) Stream cipher setup

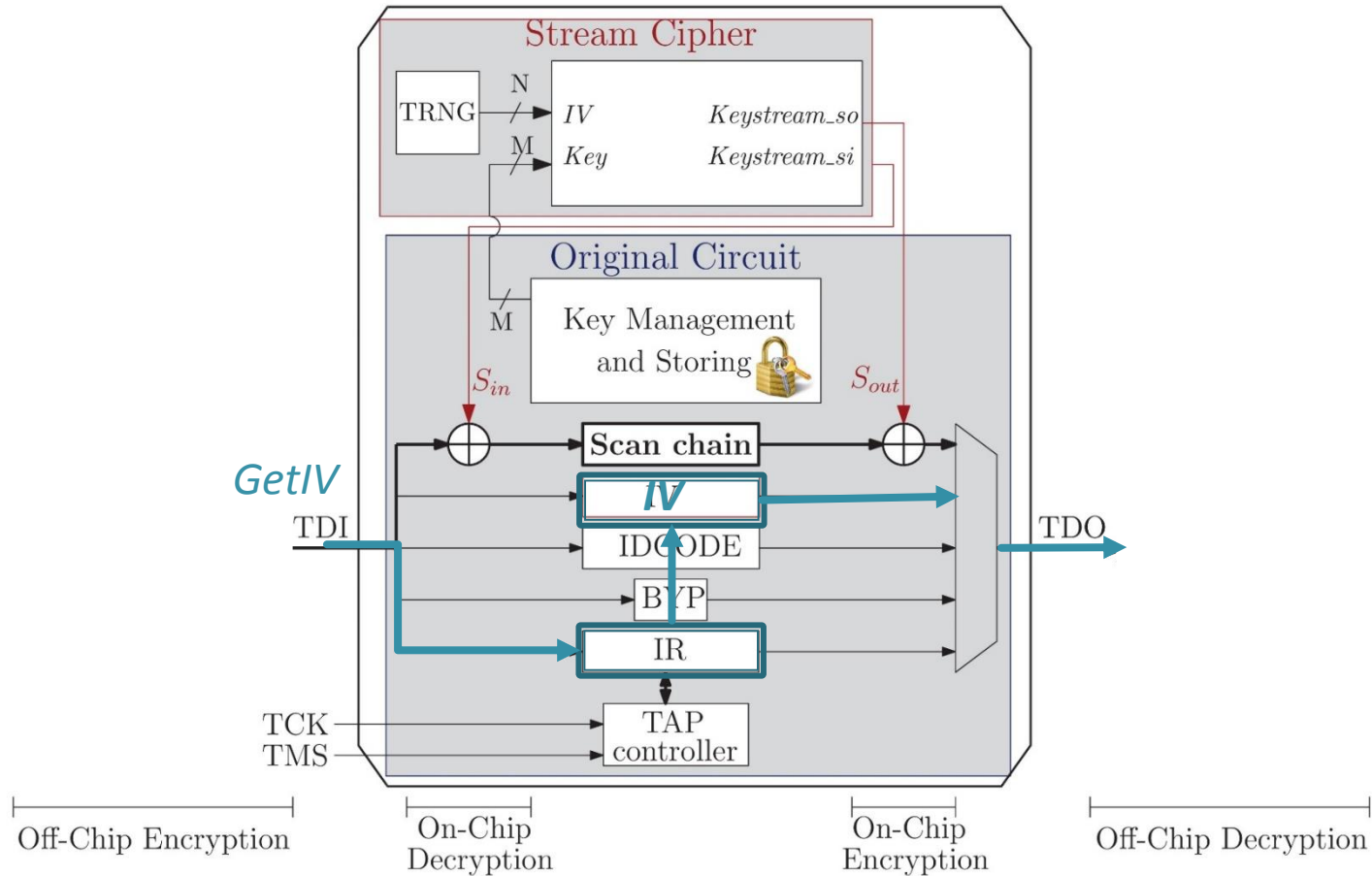


# INITIALIZATION PHASE



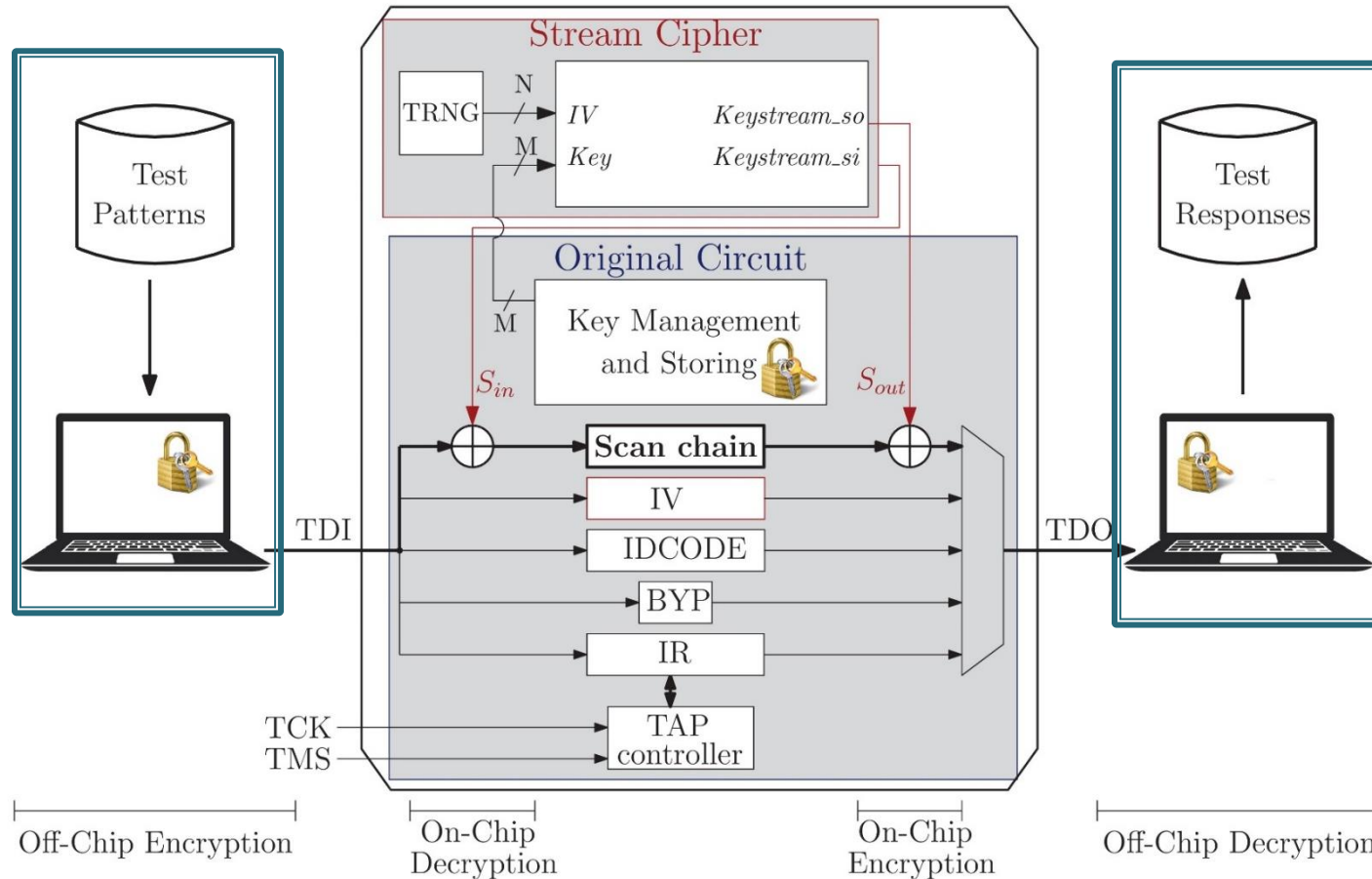
# ENCRYPTION PHASE

- User sends *GETIV* instruction
- ⇒ Shift the content of the IV register out the circuit



# ENCRYPTION PHASE

- User can encrypt and decrypt test data with the **obtained IV** and the **shared secret key**



# TIME FOR THE INITIALIZATION PROCESS

- $T_{TRNG\_init}$  to initialize the TRNG
- 80 clock cycles to shift the  $IV$  in the register
- 1 152 clock cycles for the stream cipher setup

Original circuit	Triple-DES	Pipelined AES-128	Pipelined AES-256	RSA 1024	LEON3
Test time* (clock cycles)	687 101	1 944 877	4 559 845	39 405 239	11 612 051
<b><i>Test time overhead</i></b>					
Block-based solution (%)	+0.31	+0.81	+0.006	+0.33	+0.004
Stream-based solution (%)**	+0.18	+0.06	+0.03	+0.003	+0.01

\*: Test time considered for a fault coverage of 100%, except for LEON3 where it reaches 70%

\*\* : test time overhead without the initialization of the TRNG



# SUMMARY

---

- 1) Scan chain encryption
- 2) State-of-the-art based on test communication encryption
- 3) Implementation with block cipher
- 4) Implementation with stream cipher
- 5) **Conclusion**



# COMPARISON BETWEEN BOTH SOLUTIONS

	Block cipher-based solution (PRESENT)	Stream cipher-based solution (TRIVIUM)
<b>Security</b>		
- Scan attacks	Protected	Protected (two times pad not possible)
- Malicious core	Protected	Protected
<b>Cost</b>		
- Area	10 658.96 $\mu\text{m}^2$	5 408.52 $\mu\text{m}^2$ (+ 31 200 $\mu\text{m}^2$ for TRNG)
- Test time	Depends on the scan length (multiple or not of the block size)	Clock cycles required for the initialization phase
<b>Integration</b>		
- Diagnosis & debug	Still possible in-field	
- Key management	Re-use key management already implemented	
- Integration in test daisy-chain	Possible issue with the padding of test data	No issue



Thank You



# ACKNOWLEDGEMENTS

---

- FUI#20 TEEVA Project

- Partners



TRUSTONIC



**LABORATOIRE  
HUBERT CURIEN**  
UMR • CNRS • 5516 • SAINT-ETIENNE

