Testing of Cryptographic Hardware

Presented by:

Debdeep Mukhopadhyay

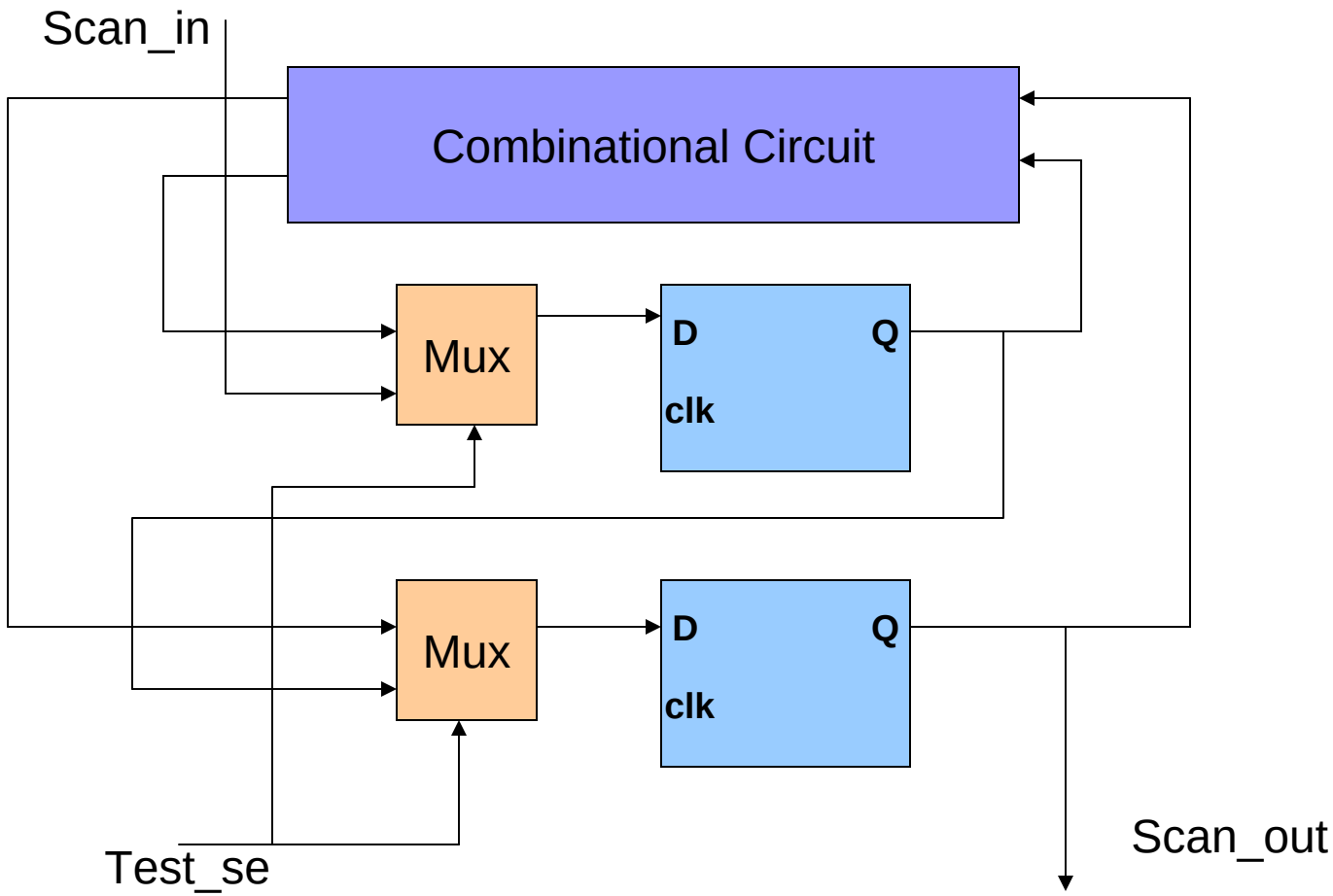
Dept of Computer Science and Engineering,

Indian Institute of Technology Madras

Motivation Behind the Work

- VLSI of Cryptosystems have become popular
- High complexity raises questions about reliability
- Scan Chain Based testing is powerful and popular method
- Double Edged Sword: Opens up side-channels for cryptanalysis!!

What is a Scan Chain ?



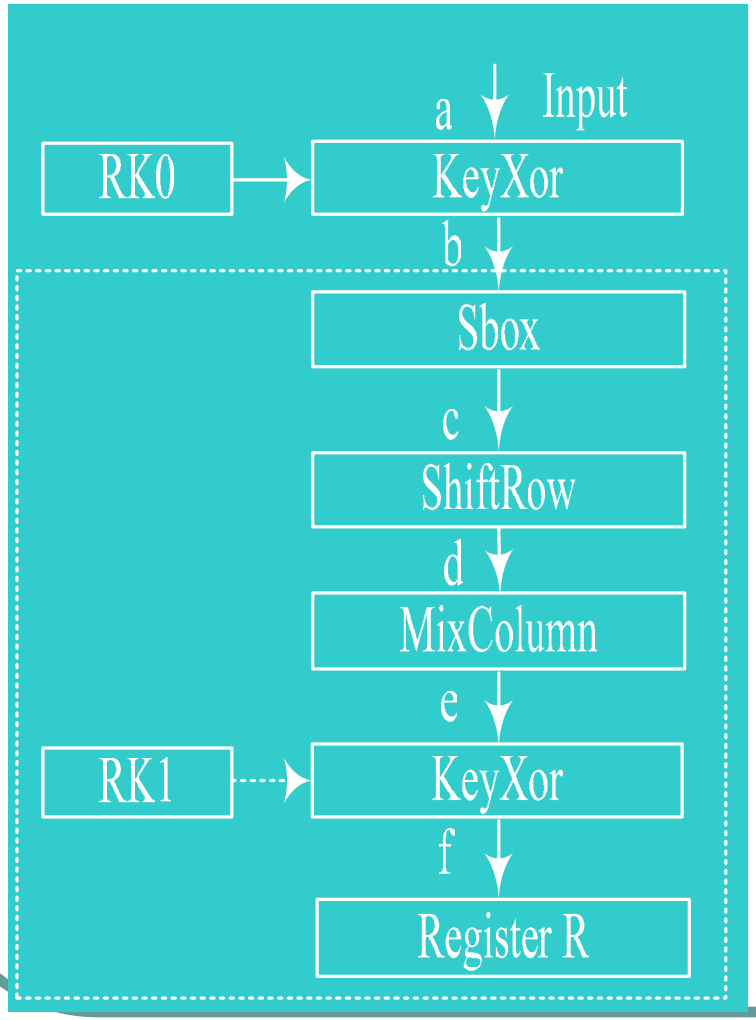
Overview of contemporary research

- Yang, Wu, Karri, *"Scan Chain Based Side Channel Attack on dedicated hardware implementations of Data Encryption Standard",* ITC Oct 2004 : ATTACKED A BLOCK CIPHER
- D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury, and B. Bhattacharya, "*Cryptoscan: Secured Scan Chain Architecture*", 14th IEEE Asian Test Symposium 2005: ATTACKED A STREAM CIPHER
- Emphasizes the need for new type of scan chains…
- Idea:
  - Increased controllability and observability for the authorized user
  - Reduced controllability and observability for the unauthorized user
  - Not Trivial

Scan Based Attacks!!!

- Attack on AES (Presented in DAC'05)
- -Attack on Stream Cipher (Presented in ATS'05)

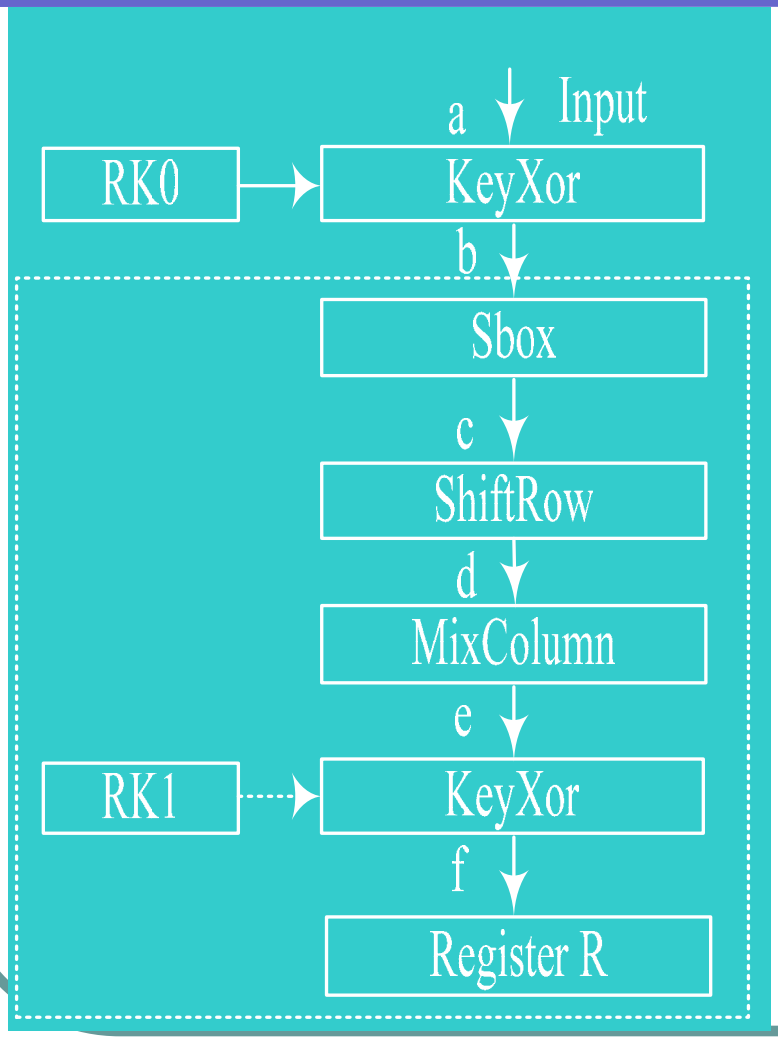
Step 1: Determine scan chain structure



- Input is partitioned into 16 bytes a_{11},… a_{14}, a_{21},… a_{24}, a_{31},… a_{34}, a_{41},… a_{44}
- Register R is fed back to point b ten times with RK1 to RK10
- 128-bit Round register R is in scan chains
- The complexity of AES is reduced to one round
- Can we determine RK0?

…..Yang, Wu and Karri, “Secure Scan: A Design for Test Architecture for Crypto-chips”, DAC 2005…

Step 1: Determine scan chain structure



- The locations of flip-flops of R in the scan chains are unknown
- Change in a_{11}→ change in b_{11}→ change in c_{11}→ change in d_{10}→ change in e_{i0}→ change in f_{i0}→ 4 byte at R
- On average, 15 patterns are enough applied at a_{11} to determine all the 32-bit in Register R (f_{i0}) by comparing the scanned out bit streams

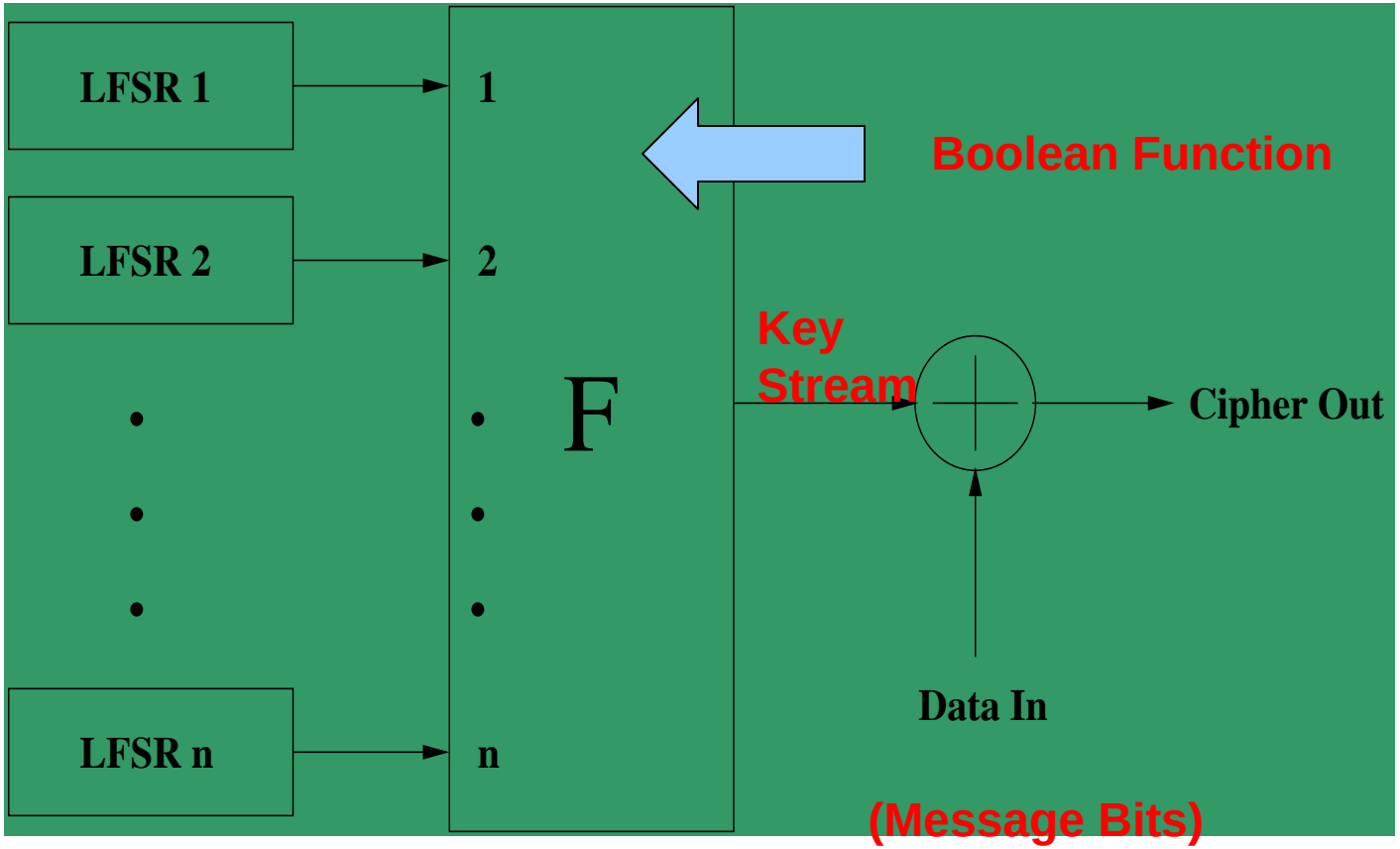
…..Yang, Wu and Karri, “Secure Scan: A Design for Test Architecture for Crypto-chips”, DAC 2005…

Step 2: Recovering Round Key RK0

- 32-bit in the scanned-out bit stream correspond to flip-flops f_{i0} are known, but one to one correspondence is unknown
- Applying $(a_{11}, a_{11}+1)$ to generate (e^1_{i0}, e^2_{i0}) and (f^1_{i0}, f^2_{i0}) we found:
  - # of 1s in $f^1_{i0} \oplus f^2_{i0}$ is equal to that in $e^1_{i0} \oplus e^2_{i0}$: the effect of RK1 is canceled
  - Some # of 1s in $f^1_{i0} \oplus f^2_{i0}$ is uniquely determined by a pair of $(b_{11}, b_{11}+1)$. Example: 9→(226, 227)
- $RK0_{11}$ is determined by $a_{11} \oplus b_{11}$

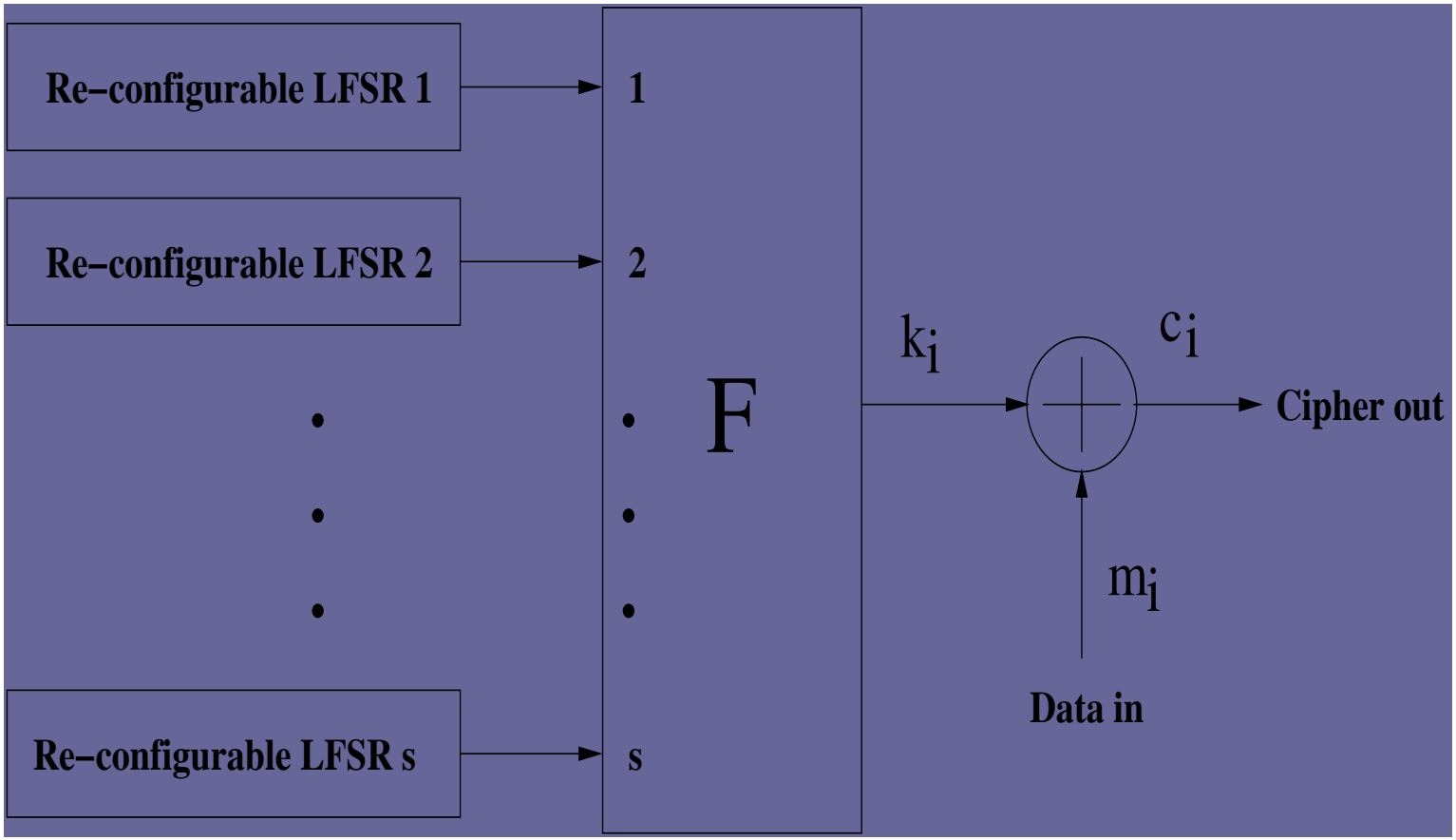
…..Yang, Wu and Karri, "Secure Scan: A Design for Test Architecture for Crypto-chips", DAC 2005…

Classical Structure of Stream Cipher



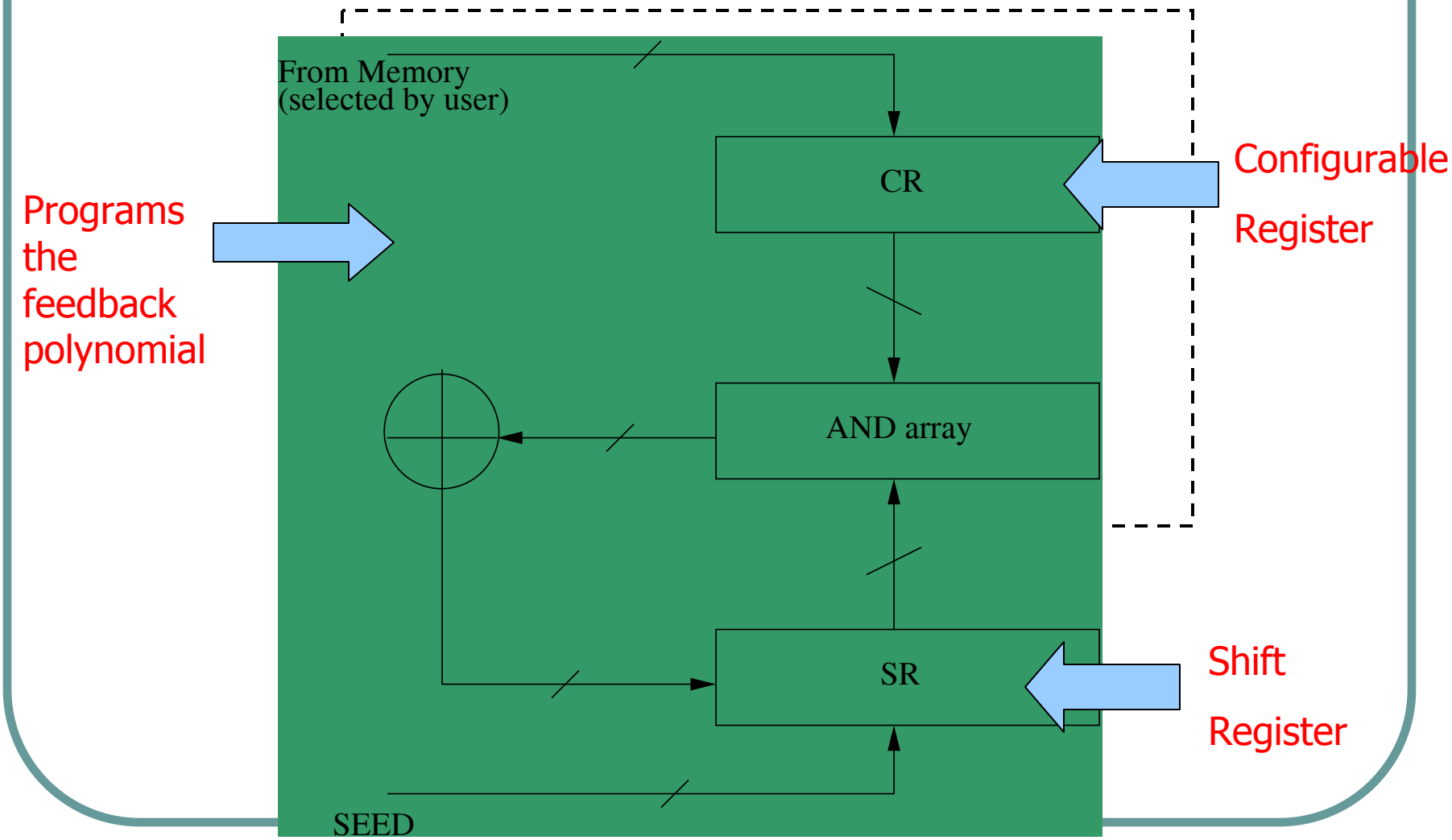
D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury and B. Bhattacharya, "CryptoScan: Secured Scan Chain Architecture", ATS 2005

Hardware Implementation



D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury and B. Bhattacharya, "CryptoScan: Secured Scan Chain Architecture", ATS 2005

Re-configurable LFSR

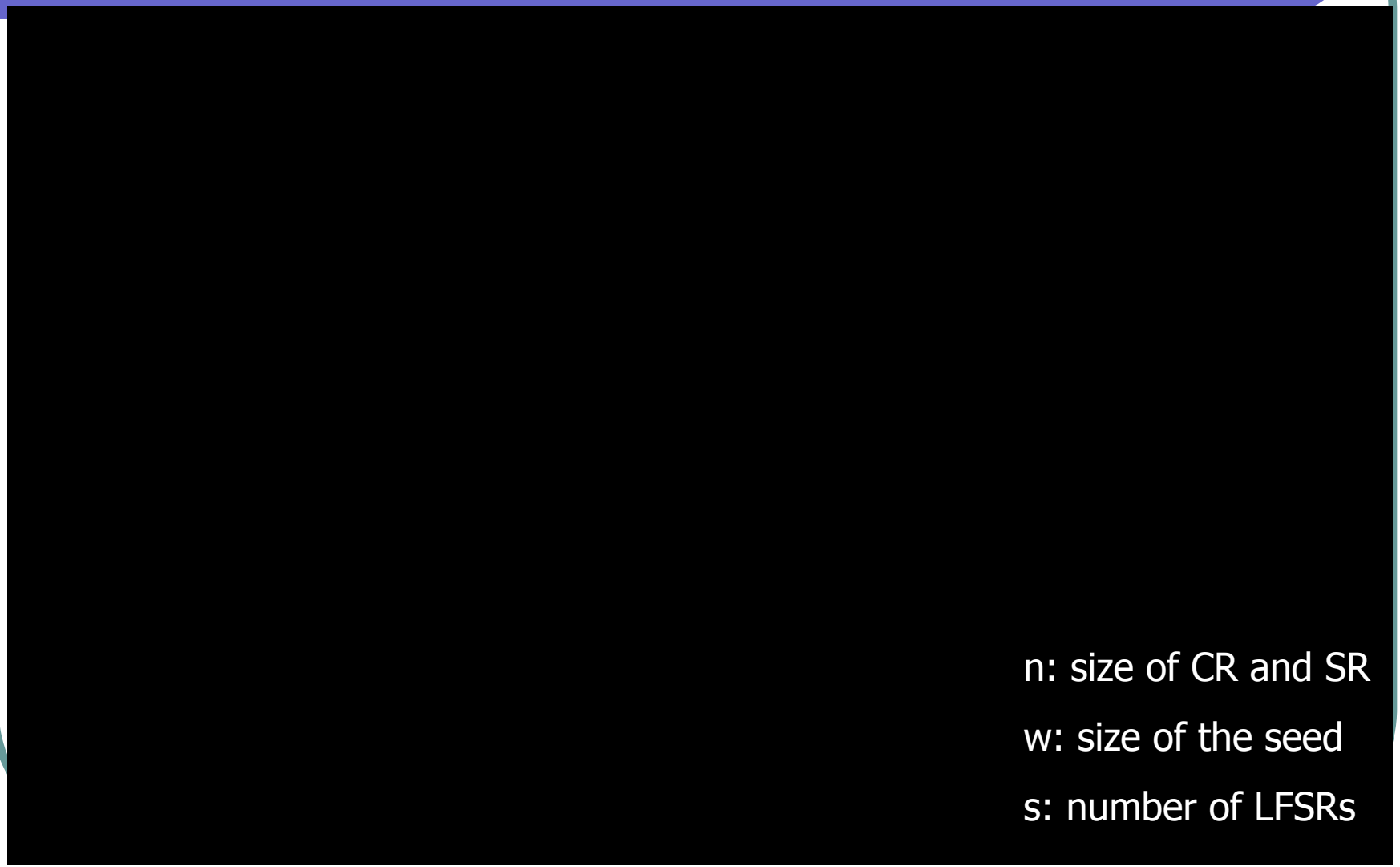


Attacking the Stream Cipher Using Scan Chains

- **Objective of the attacker:** To obtain the message stream $(m_1, m_2, \ldots, m_l)$ from the stream of ciphertexts $(c_1, c_2, \ldots, c_l)$
- Three Stage Attack
  - Ascertain the Structure of the seed
  - Ascertain the positions of the registers
  - Deciphers the cryptogram

D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury and B. Bhattacharya, "CryptoScan: Secured Scan Chain Architecture", ATS 2005

Attacking Environment



D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury and B. Bhattacharya, "CryptoScan: Secured Scan Chain Architecture", ATS 2005

Attacker's Knowledge

- What he knows?
  - Stream Cipher Algorithms which is in public domain
  - High Level Timing Diagram
  - Total size of the seed
  - Number of Flip Flops in the circuit
- What he does not know?
  - Primitive Polynomials stored in memory
  - Structure of the Scan Chains
  - Initial seed

D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury and B. Bhattacharya, "CryptoScan: Secured Scan Chain Architecture", ATS 2005

Ascertain the Structure of the Seed

- Scans out the state of the SR and CR registers
  - However does not know the correspondence of the registers with the scan patterns
- Loads the seed with all zero and applies one clock cycle
- Scans out in test mode, no of ones = *s.wt(m(0))*

D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury and B. Bhattacharya, "CryptoScan: Secured Scan Chain Architecture", ATS 2005

Ascertain the Structure of the Seed….

- Next, the attacker sets the first bit of seed to 1 and the rest to 0 and apply one clock cycle
- The bit with value 1 can go either to the memory or to the SRs
- Scan out the data in test mode.
- If the bit goes to the SR,
  no of ones = *s.wt(m(0))+1*
  else no of ones = *s.wt(m(p))*

  Not Equal

  (as s > 1)
- Repeat the same for all the *w* bits of the seed

D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury and B. Bhattacharya, "CryptoScan: Secured Scan Chain Architecture", ATS 2005

Thus the attacker has ascertained the following….

- The number of bits *(w_1)* in the seed and their positions in the seed which are used to address the memory. Thus, the attacker also knows the bits in the seed which are used to initialize the SRs
- The attacker also identifies the positions of the CR resisters in the scan chains. He also identifies the positions of the SR resisters in the scan out data, however the order is not known
- Complexity : *O(wns)*

D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury and B. Bhattacharya, "CryptoScan: Secured Scan Chain Architecture", ATS 2005

Ascertain the position of the SR and CR registers

- Ascertains the group of SR[i] of the LFSRs
  - Sets all the register bits to 1 through scan chain (in test mode)
  - Apply one clock cycle in normal mode
  - Put the chip in test mode and scan out the data
  - Note the position of 0's in the scanned out data : ascertains the positions of SR[n] bits
  - Return to normal mode and apply another clock cycle
  - Note the position of 0's in the scanned out data : ascertain the positions of the SR[n-1] bits and so on…
  - Complexity: $O(n^2 s)$

D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury and B. Bhattacharya, "CryptoScan: Secured Scan Chain Architecture", ATS 2005

Ascertain the position of the SR and CR registers….

- Identification of the SR bits of a particular LFSR in the scan out data….
  - Attacker knows the group of SR[1] bits
  - Set one of SR[1] to 1 and rest SR[1] bits to 0
  - Set the CRs to 100…001 (through scan chain in test mode)
  - After n clock cycles in normal mode all the SR bits of the particular LFSR (whose SR[1] was set) will become 1
  - Observing this in the scan out data serves the purpose
  - Repeat the above process for the other (s-1) SR bits
  - Complexity : $O(ns^2)$

D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury and B. Bhattacharya, "CryptoScan: Secured Scan Chain Architecture", ATS 2005

Deciphering the Cryptogram

- Decoding c_l : The attacker knows the values of the SR registers of all the LFSRs: {SR[n],SR[n-1],……SR[2],SR[1]}
  - The previous state of the LFSRs can be computed as: {SR[n-1],SR[n-2],…,SR[1],SR[n]⊕SR[1]} (as CR[1] is always 1)
  - He sets the message bit of the device to zero and the device in normal mode. One clock cycle is applied and the output is observed. The output is the value of k_l. Thus $m_l = c_l \oplus k_l$

D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury and B. Bhattacharya, "CryptoScan: Secured Scan Chain Architecture", ATS 2005

Deciphering the cryptogram…

- Decoding $c_1, c_2, \ldots, c_{l-1}$: For decoding c_{l-1}, similarly the attacker computes the previous stage of the SR register of all the LFSRs. Continuing the step for l times leads to the decoding of the entire cryptogram. Thus, the time complexity is *O(nsl)*

D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury and B. Bhattacharya, "CryptoScan: Secured Scan Chain Architecture", ATS 2005

Coming back to …Why Non-trivial???

- Scrambling Technique (Dynamic Re-ordering of scan chains)
  - Separate test key to program the inter-connections
  - Wiring complexity increases fast with the number of flops
  - Control circuit uses themselves flip-flops
  - Statistical Analysis may reveal the ordering

Who tests them ?

Lock and Key Technique

- Test Key
- Test Security Controller (TSC): compares the key
- If wrong key is entered, design goes to an insecured mode unless reset
- Demerits:
  - Large Area Overhead
  - TSC uses flip-flops…
  - Use of additional key, overhead on key exchange

Observations…

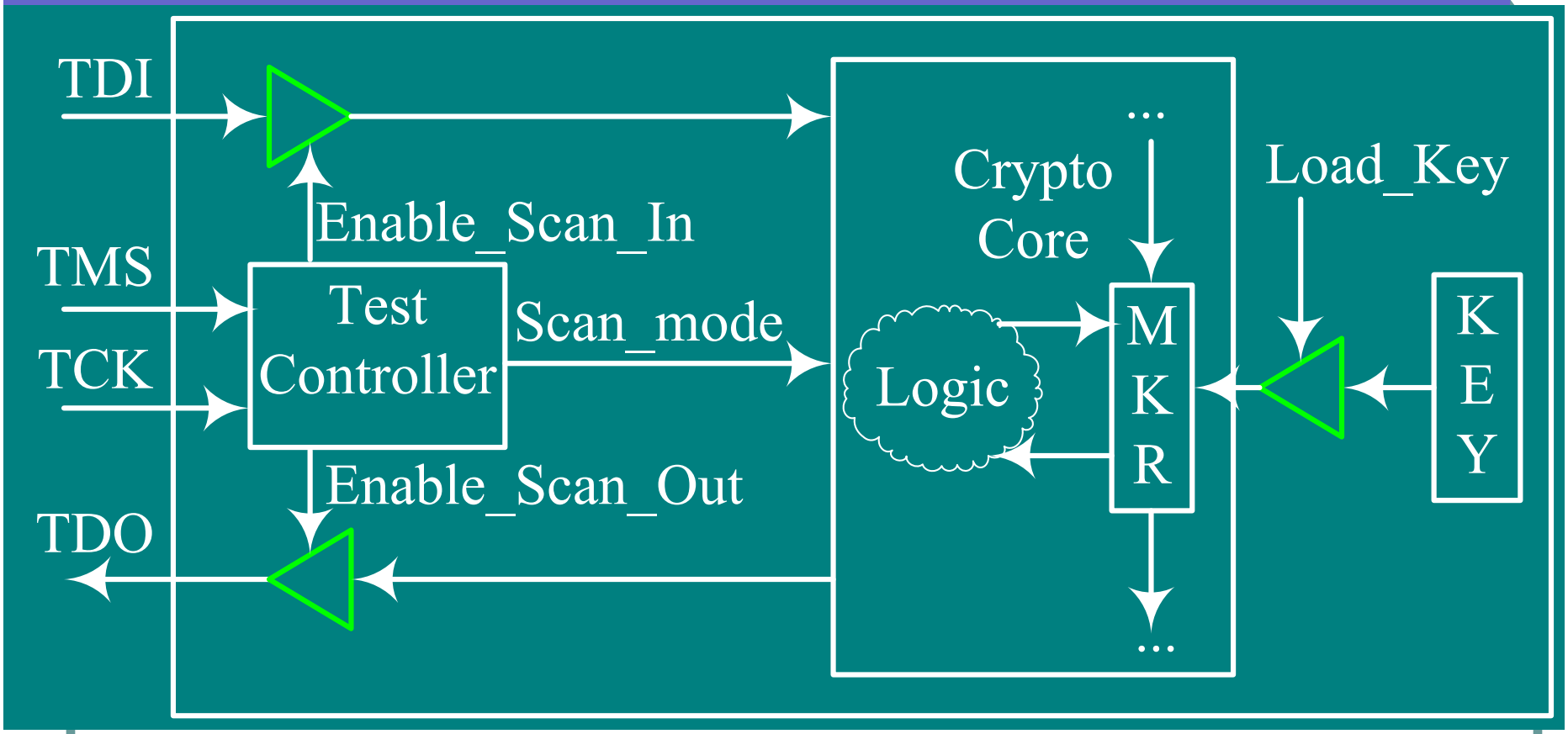
- Any Flip-flops related to secret lead to attacks
- Use of additional key not desirable
- Area Overhead should be less
- On-line testing should be possible

Non-trivial….

Secure Scan : Karri's Curry☺

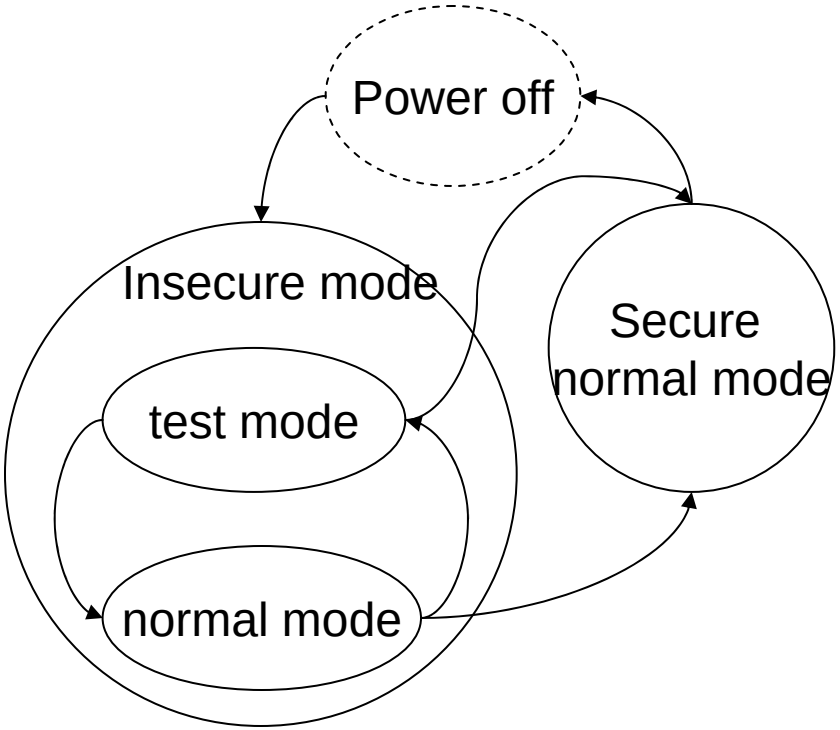
- Test and debug crypto chips using general scan based DFT
  - Information obtained from scan chains should not be useful in retrieving the secret key
- Two copies of the secret key
  - Secure key: hardwired or in secure memory
  - Mirror Key (MKR): used for testing
- Two modes of operation: Insecure and Secure
  - Insecure mode: secure key is isolated, MKR is used and debug allowed
  - Secure mode: secure key is used and debug disabled

Secure Scan Architecture



- Insecure Mode
  - Enable_Scan_In=1, Enable_Scan_Out=1, Load_Key=0
- Secure Mode
  - Enable Scan In=0. Enable Scan Out=0. Load Key=1

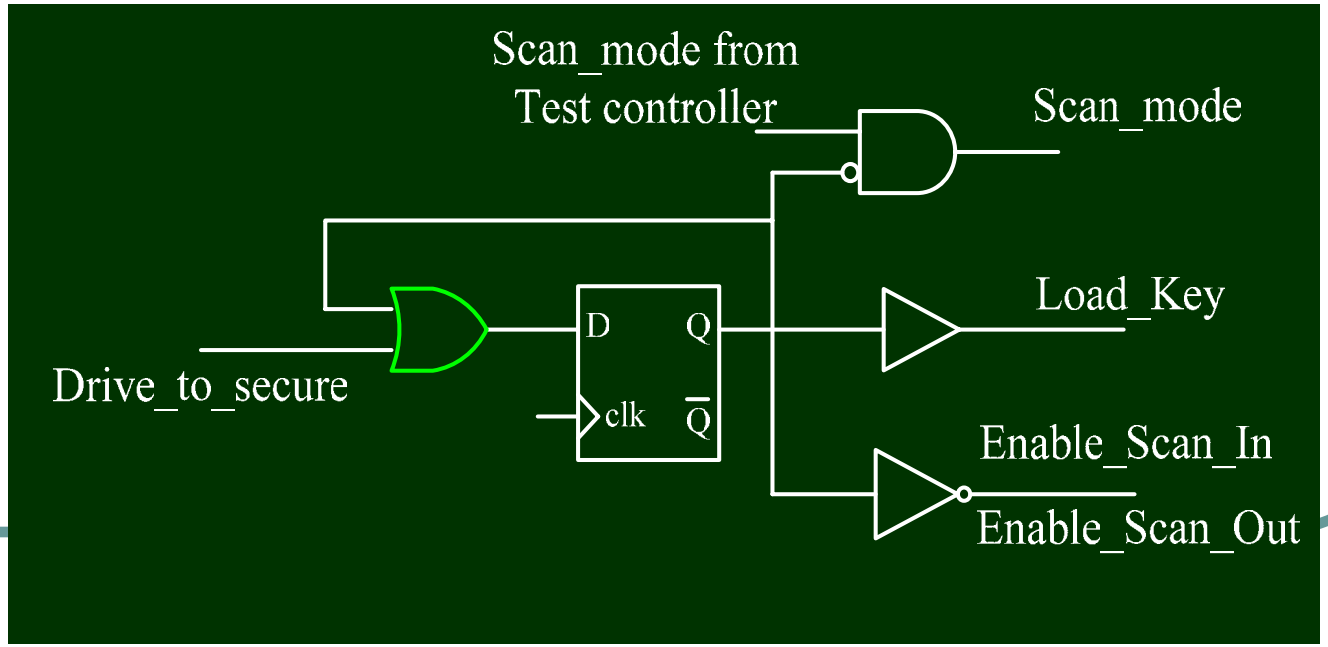
Secure Scan: State Diagram



- Enable Scan if Load_Key = ‘0’, Enable_Scan_In = ‘1’and Enable_Scan_Out = ‘1’
- Disable Scan if Load_Key = ‘1’, Enable_Scan_In = ‘0’and Enable_Scan_Out = ‘0’

Secure Scan: Test Controller

- Modify IEEE 1149.1 Test Controller
  - New instruction: Drive_to_secure
  - Three new output control signals
- Dedicated Secure Control Circuit



Overhead Analysis

Architecture	Area (gates)	Area overhead (gates)	Ratio
Iterative (with KS)	31,234	412	1.32%
Iterative (without KS)	30,854	412	1.34%
Pipelined (with KS)	273,187	412	0.15%
Pipelined (without KS)	282,120	4620	1.64%

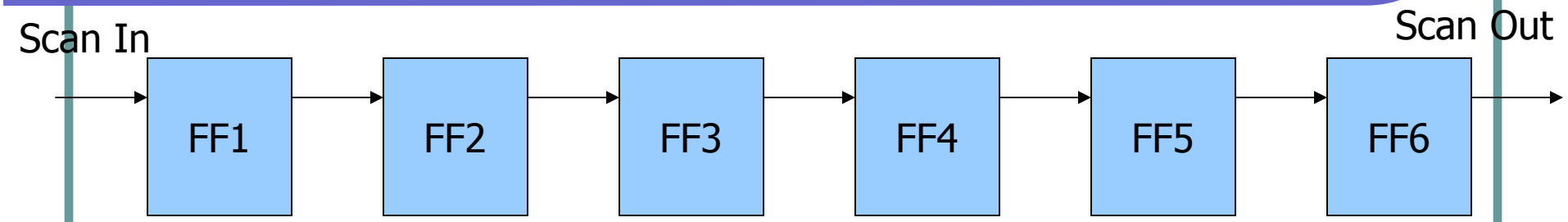
Analysis of Secure Scan

- **Merits:**
  - Does not degrade test speed
  - Circuit incurred by secure scan is easy to test
  - Easy to integrate into current scan DFT flow
    - Specify MKRs to corresponding secret key bit and do secure synthesis (Secured CAD??)
  - Area overhead is very small
- **Demerits:**
  - If secret is permanently stored like credit card nos.
  - On-line testing not possible
  - If device is part of a critical system it should remain on continuously
  - Testing of MKR not straight-forward
  - In-convenient if the AES engine is used in a Cipher Block Chaining Mode

Design of Crypto-Scan

- Hardware Designs of Ciphers are insecure with conventional scan chains
- Require Scan Chains for cryptographic chips!
- Objectives:
  - Modify the Scan Structure so that testing features are maintained
  - The Scan Structure does not open up a side-channel

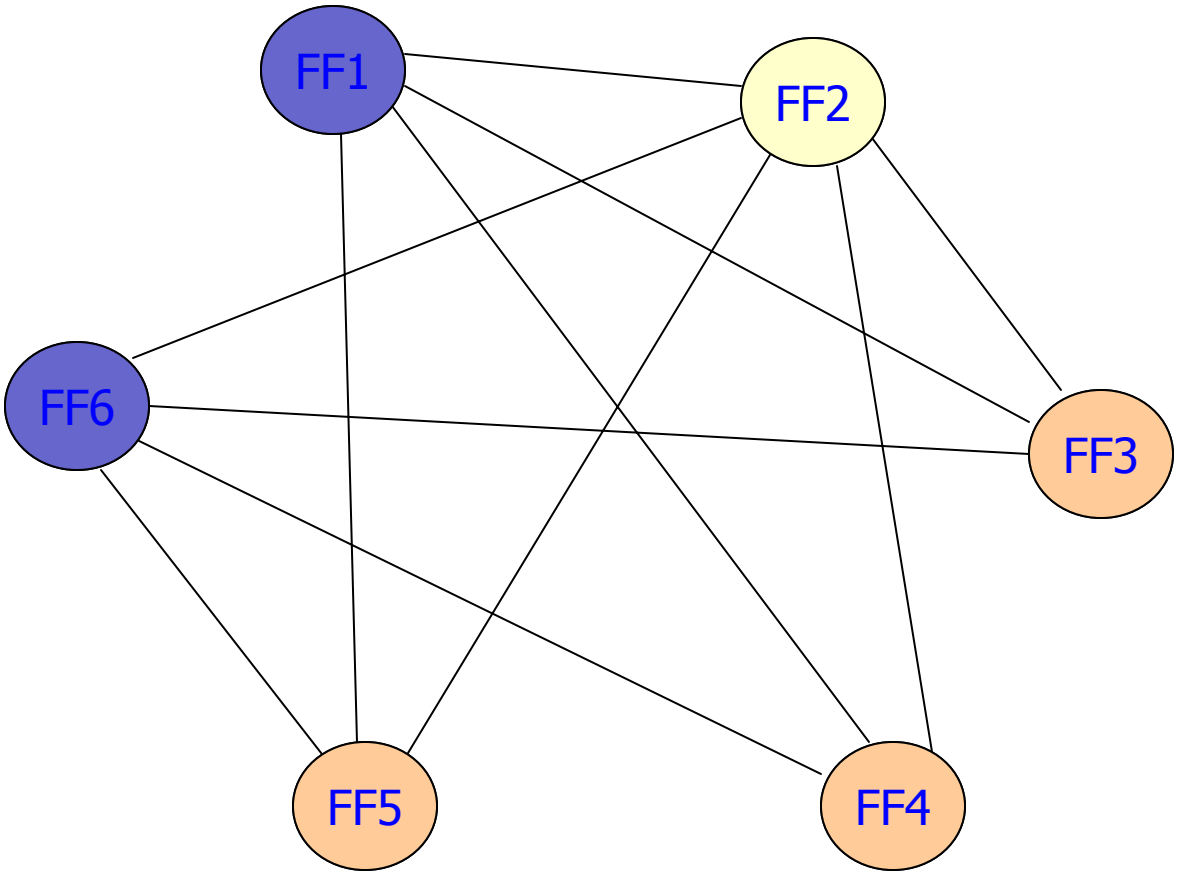
Scan Tree Architecture



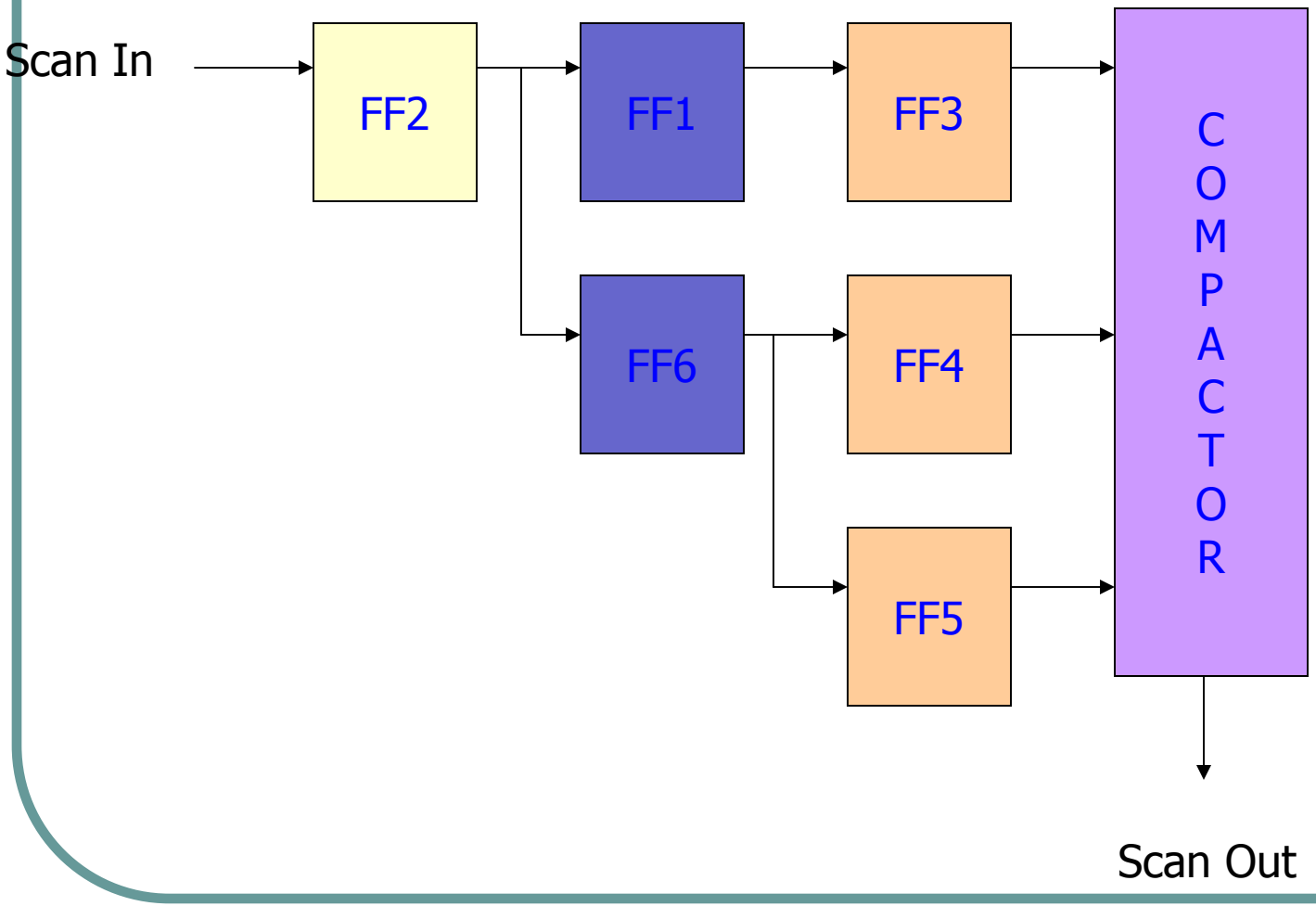
t1	1	0	X	0	0	1
t2	0	0	1	X	1	X
t3	X	1	0	0	X	X

Scan Tree Architecture…..

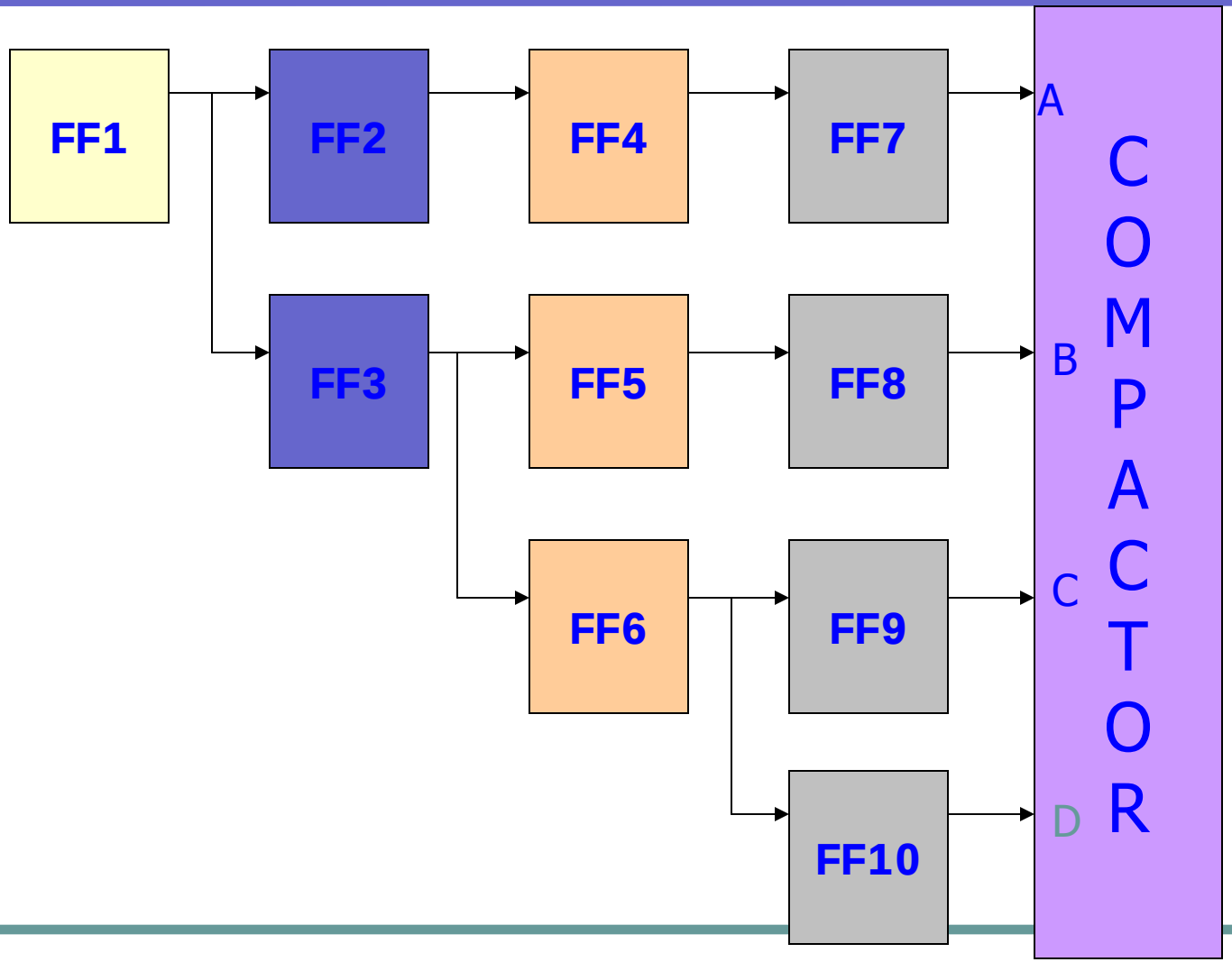
{FF2}, {FF1, FF6}, {FF3, FF4, FF5}



Scan Tree Architecture…



Aliasing Free Compactor…



Expected Responses…

	Test Responses									
Test Patterns	FF1	FF2	FF3	FF4	FF5	FF6	FF7	FF8	FF9	FF10
t1	1	0	1	1	0	1	1	1	0	0
t2	0	1	0	0	1	1	0	1	1	0
t3	1	0	0	0	1	1	1	1	0	0
t4	0	0	1	1	1	0	0	1	0	1

Truth Table for Compactor

Counter-1 (T)		Counter-1 (C)		Inputs				Outputs	
t1	t2	c1	c2	A	B	C	D	Y	Decision
0	0	0	0	1	1	0	0	0	Fault Free
0	0	0	0	0	X	X	X	1	Faulty
0	0	0	0	X	0	X	X	1	Faulty
0	0	0	0	X	X	1	X	1	Faulty
0	0	0	0	X	X	X	1	1	Faulty
0	0	0	1	1	0	1	X	1	Fault Free
0	0	0	1	0	X	X	X	0	Faulty
0	0	0	1	X	1	X	X	0	Faulty
0	0	0	1	X	X	0	X	0	Faulty
…	…	…	…	…	…..			…..	…..

Why is Crypto-Scan Secured?

- d: Compatible Groups
- L= $\{l_1, l_2, \ldots\ldots, l_d\}$
- N : Total Number of flip-flops
- Scan-Tree Characterized: *st(l,d)*
- Normal Scan Chain :
  - N Known
  - Position of flip-flops can be ascertained

Security of Crypto-Scan

- Crypto-Scan:
  - d does not reveal information about N
  - $d \leq N \leq dl_d$
  - Compactor hides the value of l_d, hence N cannot be determined
  - Scan Structure secured because value of L is hidden

Space of Scan Trees

- **Theorem 1:** ***If l is the length of the longest scan chain and n is the number of scan out pins, the probability of guessing the correct tree structure is :***

$$p = \frac{1}{\sum_{l}^{nl} \binom{nl}{r} r^{r-2}}$$

- Proof:
  - Attacker fills up a grid on nxl, in a tree fashion as number of nodes in the tree (r) varies from *l* to *nl.*
  - *No of trees with r nodes:* r^{r-2}
  - *No of ways of choosing r :* $\binom{nl}{r}$

Experimental Setup

- ISCAS’89 Bench Marks
- Solaris-10 Platform
- Synthesized using Design Compiler (Synopsys)
- TetraMax (Synopsys) is used for test pattern generation

Area Overhead Due to Compactor and Scan Tree

Benchmark Circuits Name	Area Overhead %
s298	**21**
s344	**18**
s382	**19**
s400	**19.4**
s5378	**17**
s9234	**17.7**
s13202	**16.4**
s15850	**17**
s35932	**15.8**
s38417	**16.4**

Analysis

- Merits:
  - Fast on-line testing : test compression
  - Testing of components easy
  - No use of flip-flops
- Demerits:
  - Overhead?

Conclusion

- Future research required
- Testability vs Security is indeed non-trivial
- Ideal Scan Chains for Crypto-devices should be:
  1. Easy to implement without extra flip-flops
  2. No extra key should be used
  3. On-line testing should be supported
  4. Overhead on test pattern generation and area should be less

Thank You