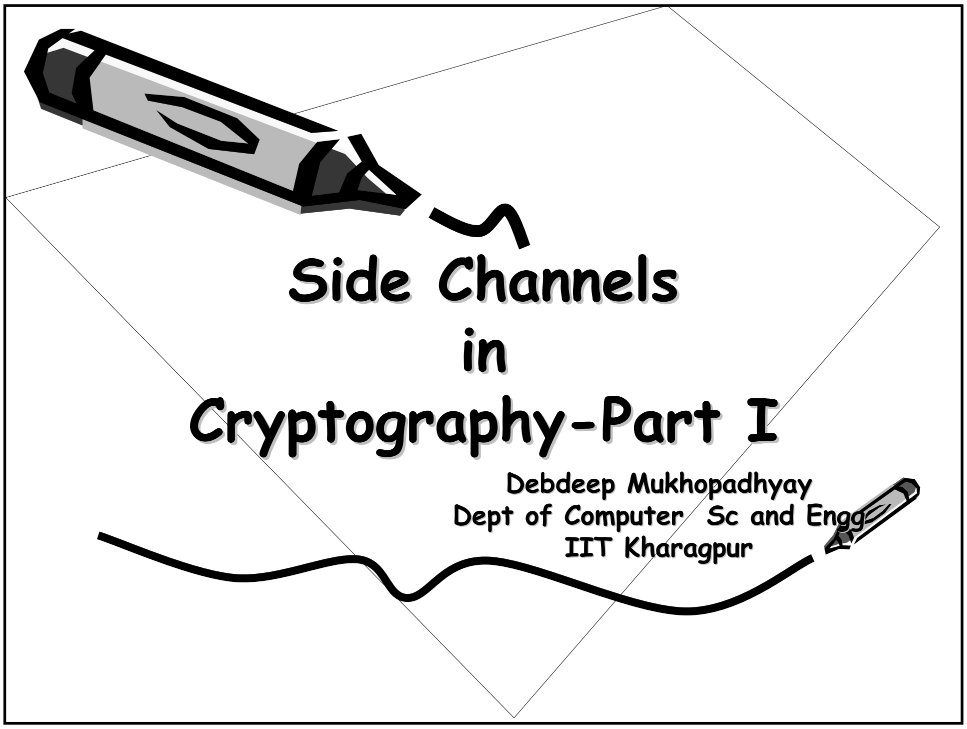
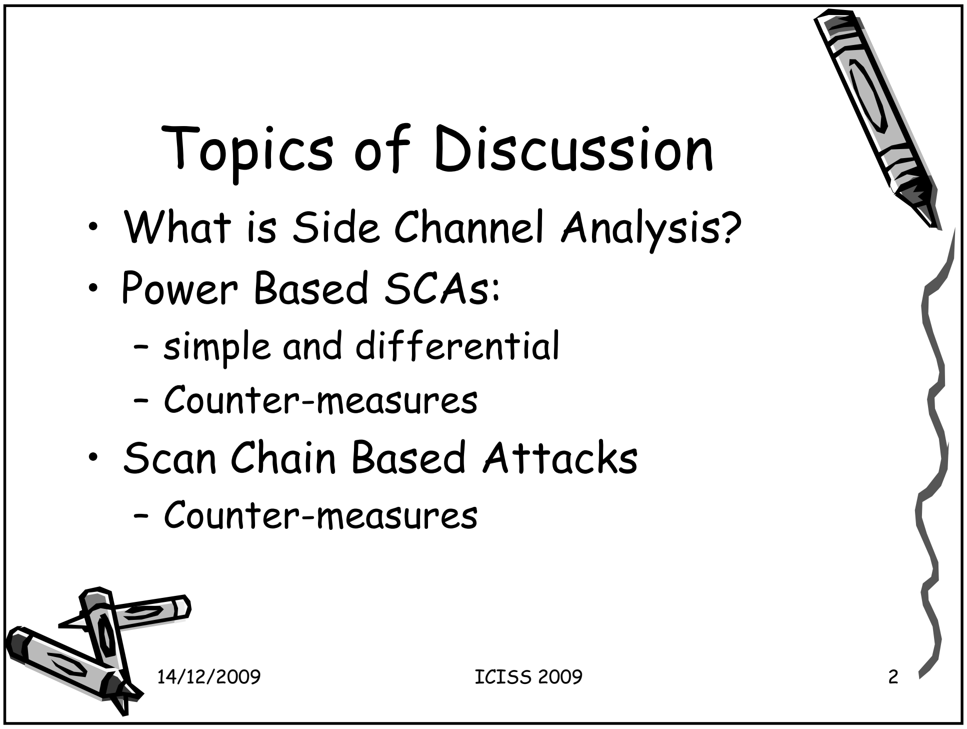


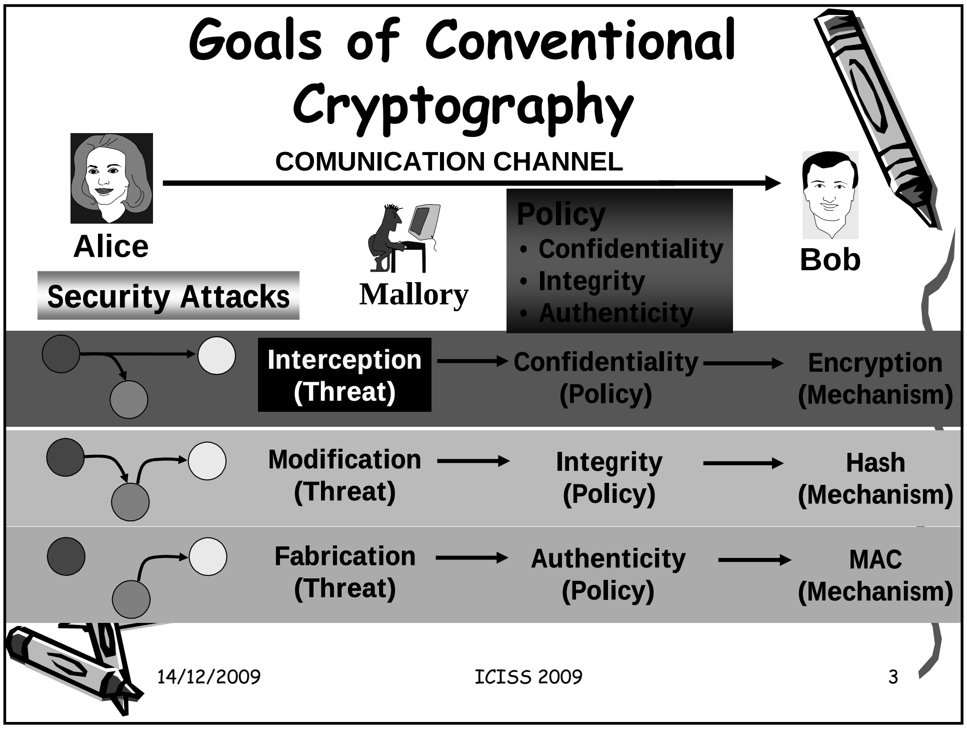
Side Channels in Cryptography-Part I

Debdeep Mukhopadhyay
Dept of Computer Sc and Engg
IIT Kharagpur



Topics of Discussion

- What is Side Channel Analysis?
- Power Based SCAs:
  - simple and differential
  - Counter-measures
- Scan Chain Based Attacks
  - Counter-measures



Goals of Conventional Cryptography

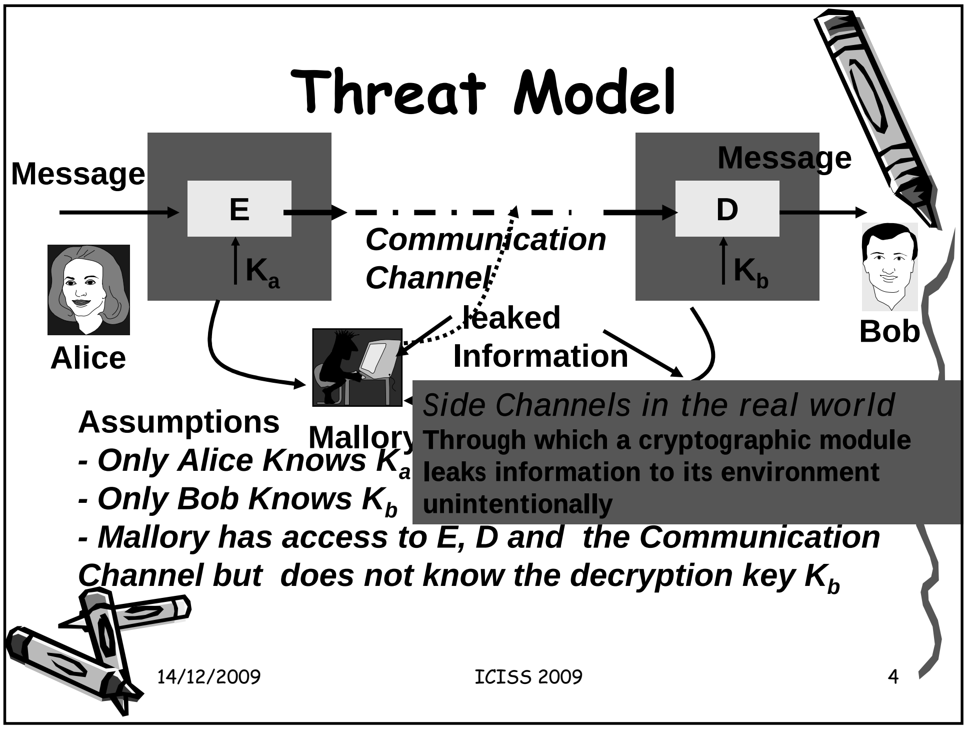
Policy
- Confidentiality
- Integrity
- Authenticity

Security Attacks

Interception (Threat) → Confidentiality (Policy) → Encryption (Mechanism)

Modification (Threat) → Integrity (Policy) → Hash (Mechanism)

Fabrication (Threat) → Authenticity (Policy) → MAC (Mechanism)

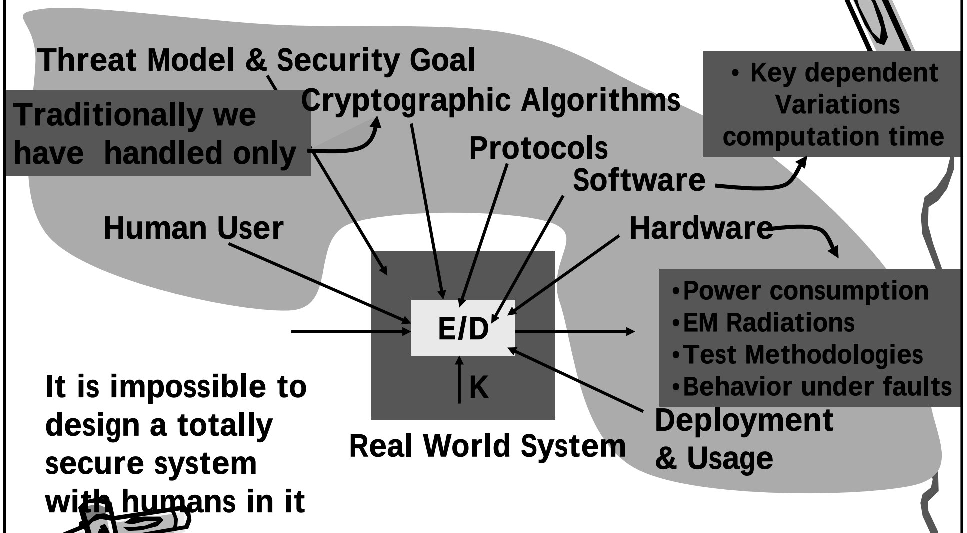


Threat Model

Side Channels in the real world
Through which a cryptographic module leaks information to its environment unintentionally

Assumptions
- Only Alice Knows K_a
- Only Bob Knows K_b
- Mallory has access to E, D and the Communication Channel but does not know the decryption key K_b

Side Channel Sources



Threat Model & Security Goal

Traditionally we have handled only

Cryptographic Algorithms

Protocols

Software

Hardware

Human User

E/D

K

Real World System

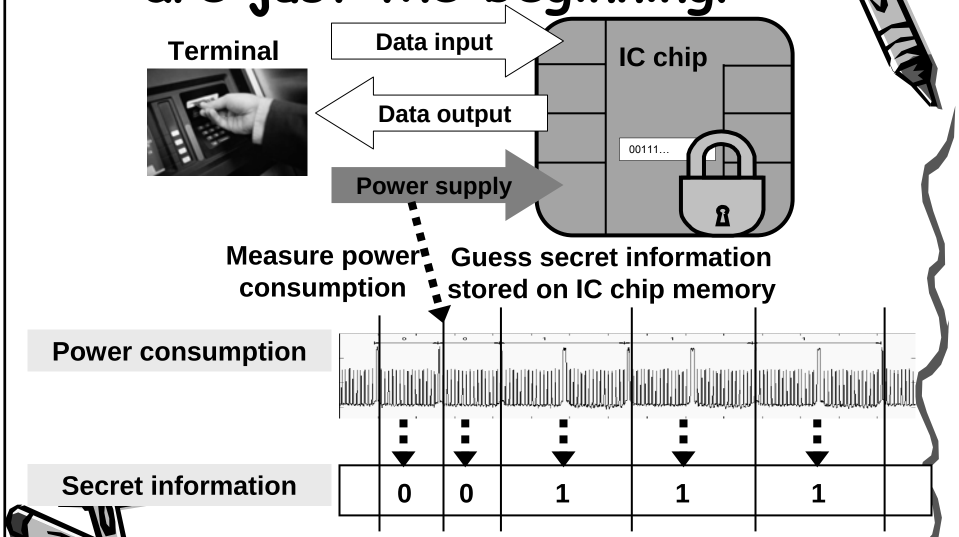
Deployment & Usage

- Key dependent Variations computation time

- Power consumption
- EM Radiations
- Test Methodologies
- Behavior under faults

It is impossible to design a totally secure system with humans in it

Strong cryptographic algorithms are just the beginning!



Terminal

Data input

Data output

Power supply

IC chip

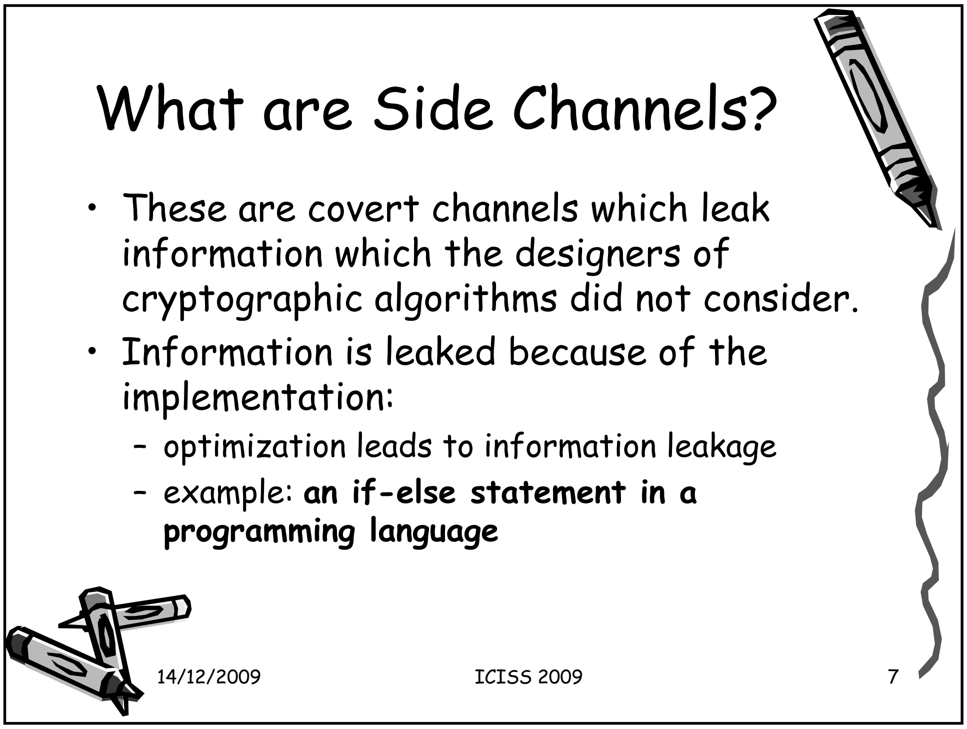
00111...

Measure power consumption

Guess secret information stored on IC chip memory

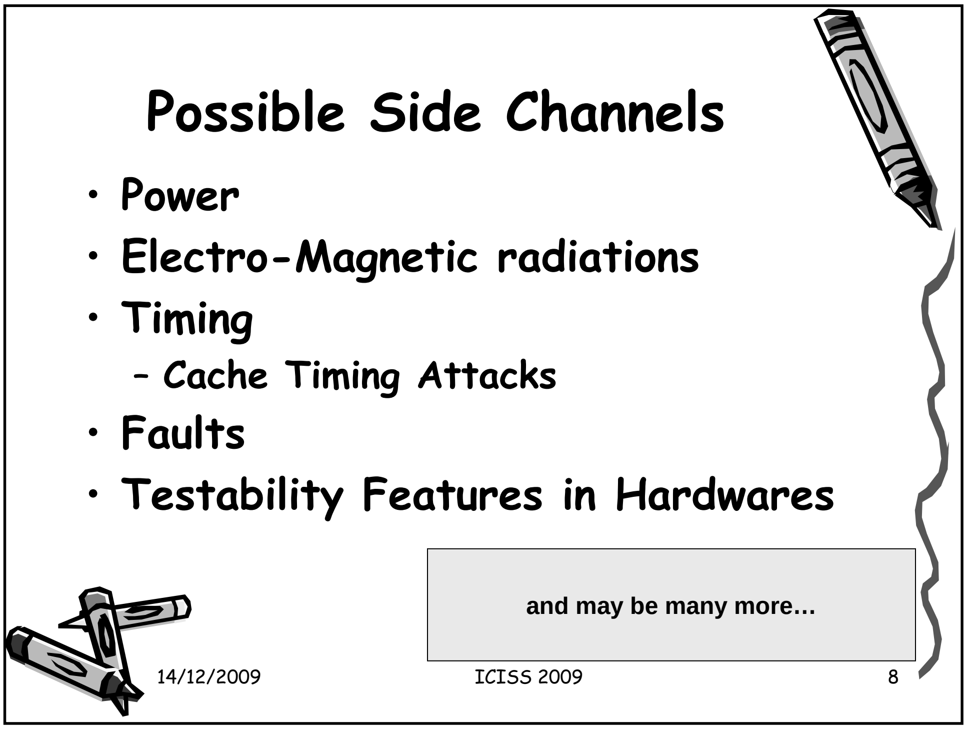
Power consumption

Secret information 0 0 1 1 1



What are Side Channels?

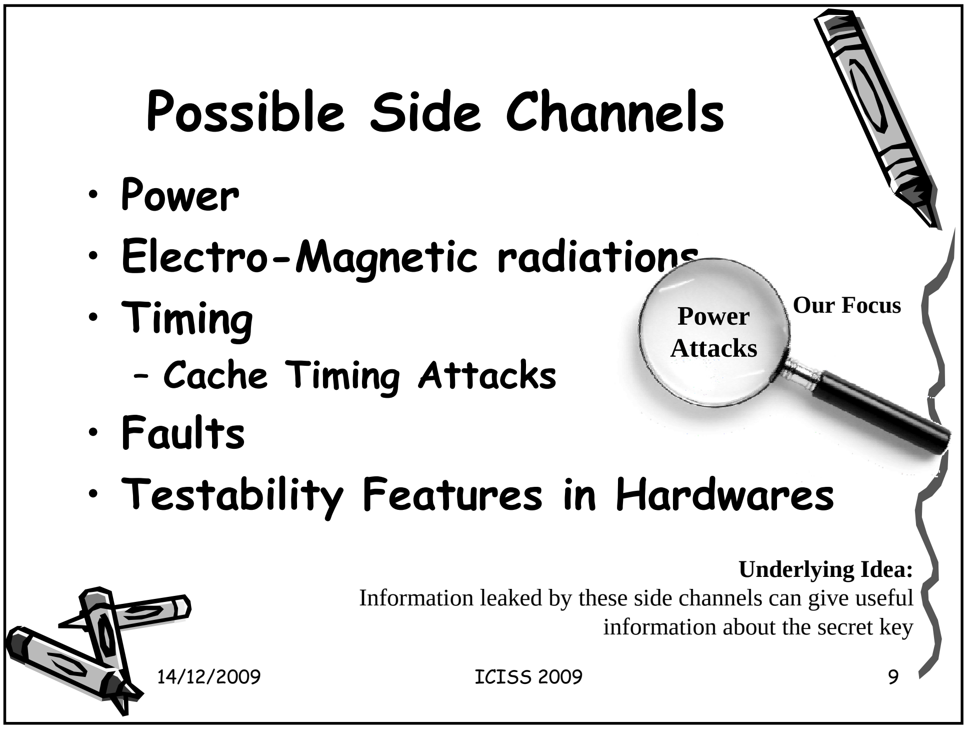
- These are covert channels which leak information which the designers of cryptographic algorithms did not consider.
- Information is leaked because of the implementation:
  - optimization leads to information leakage
  - example: **an if-else statement in a programming language**



Possible Side Channels

- **Power**
- **Electro-Magnetic radiations**
- **Timing**
  - **Cache Timing Attacks**
- **Faults**
- **Testability Features in Hardwares**

and may be many more…



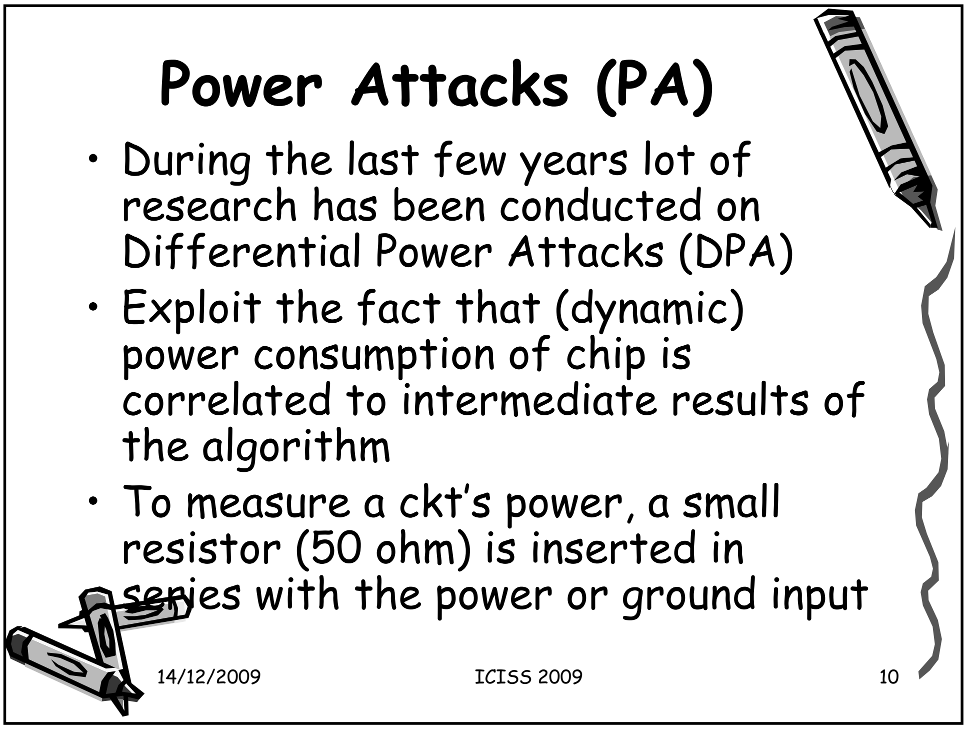
Possible Side Channels

- Power
- Electro-Magnetic radiations
- Timing
  - Cache Timing Attacks
- Faults
- Testability Features in Hardwares

Power Attacks

Our Focus

Underlying Idea:
Information leaked by these side channels can give useful information about the secret key

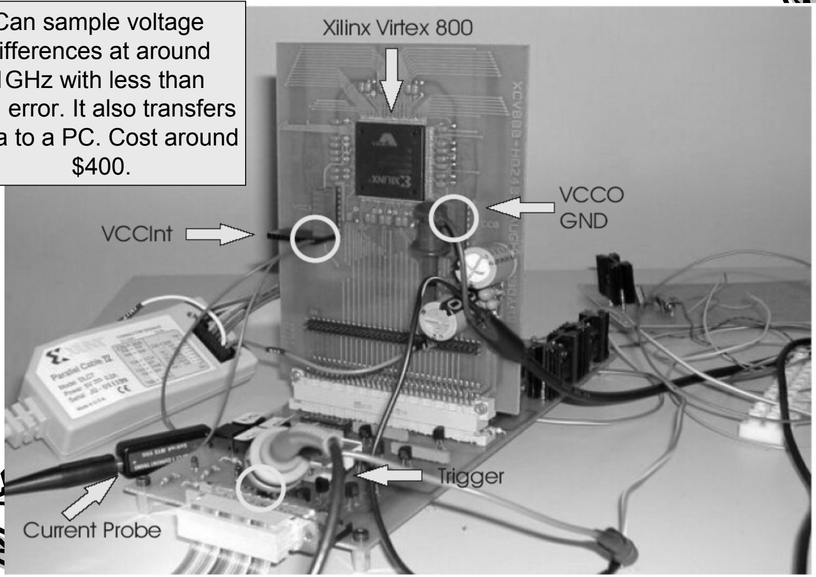


Power Attacks (PA)

- During the last few years lot of research has been conducted on Differential Power Attacks (DPA)
- Exploit the fact that (dynamic) power consumption of chip is correlated to intermediate results of the algorithm
- To measure a ckt’s power, a small resistor (50 ohm) is inserted in series with the power or ground input

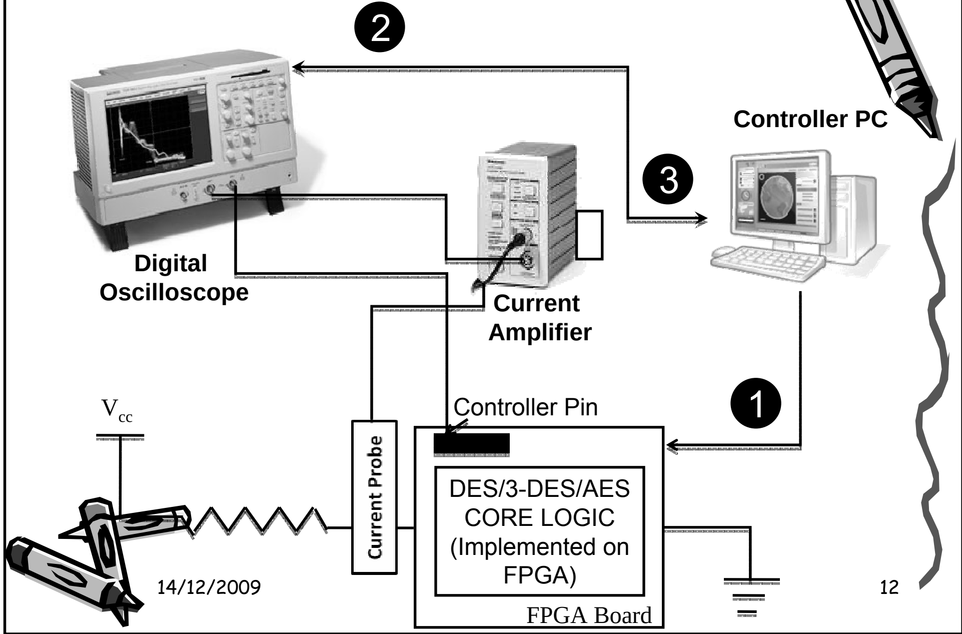
Lab Set Up for Power

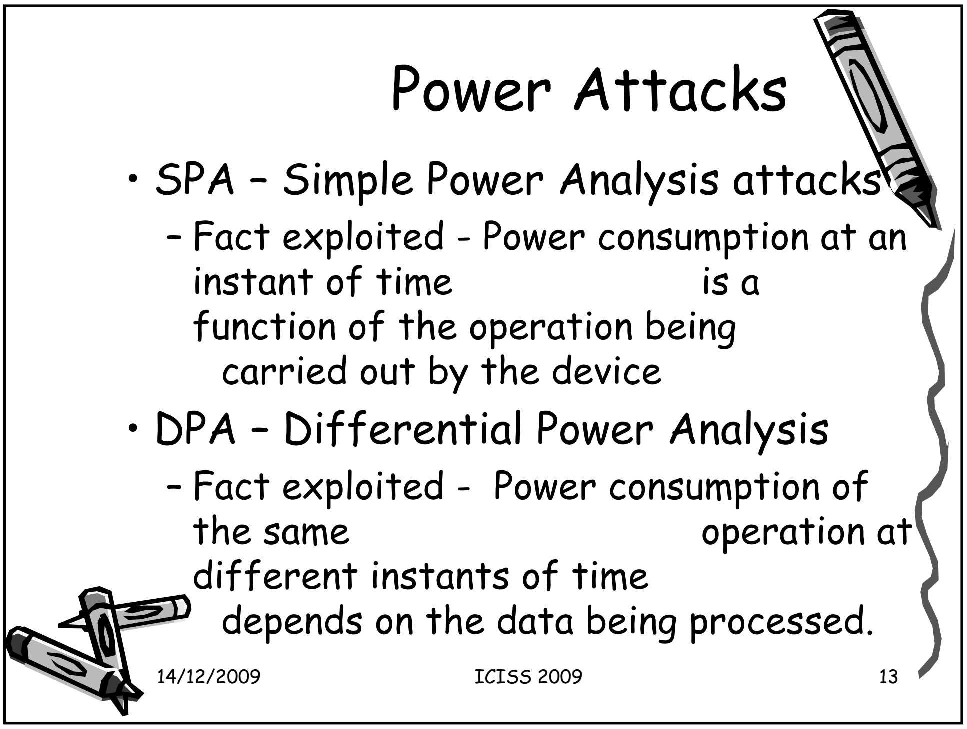
Can sample voltage differences at around 1GHz with less than 1% error. It also transfers Data to a PC. Cost around $400.



Courtesy: Side-Channel Analysis Lab, *Graz University of Technology*

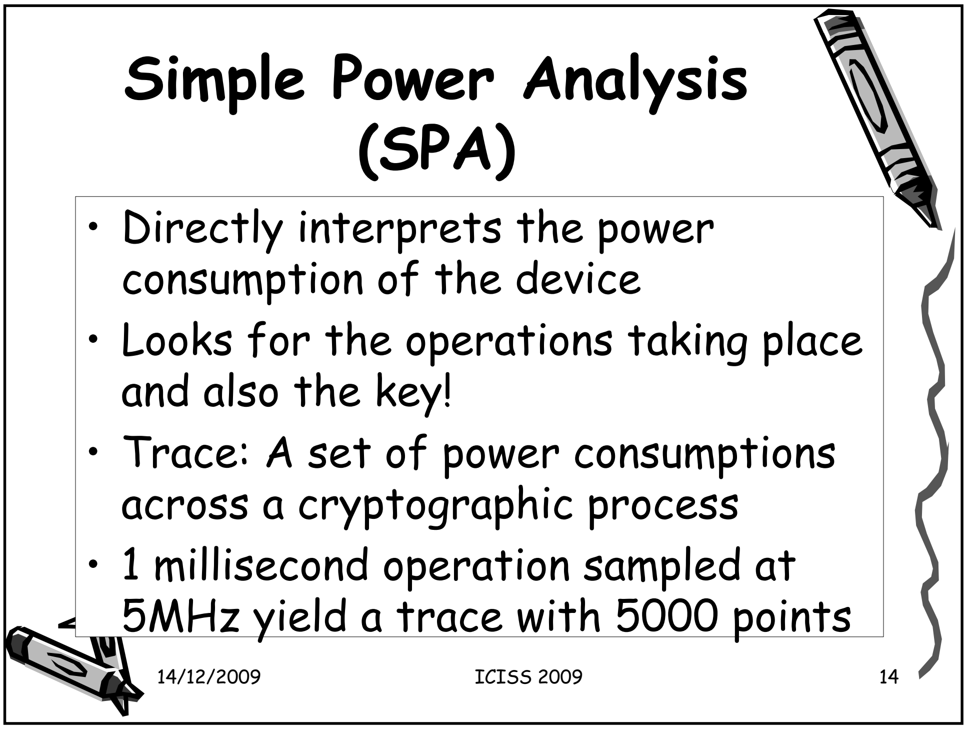
Experiment Set-up @ IIT KGP





Power Attacks

- SPA – Simple Power Analysis attacks
  - Fact exploited - Power consumption at an instant of time is a function of the operation being carried out by the device
- DPA – Differential Power Analysis
  - Fact exploited - Power consumption of the same operation at different instants of time depends on the data being processed.



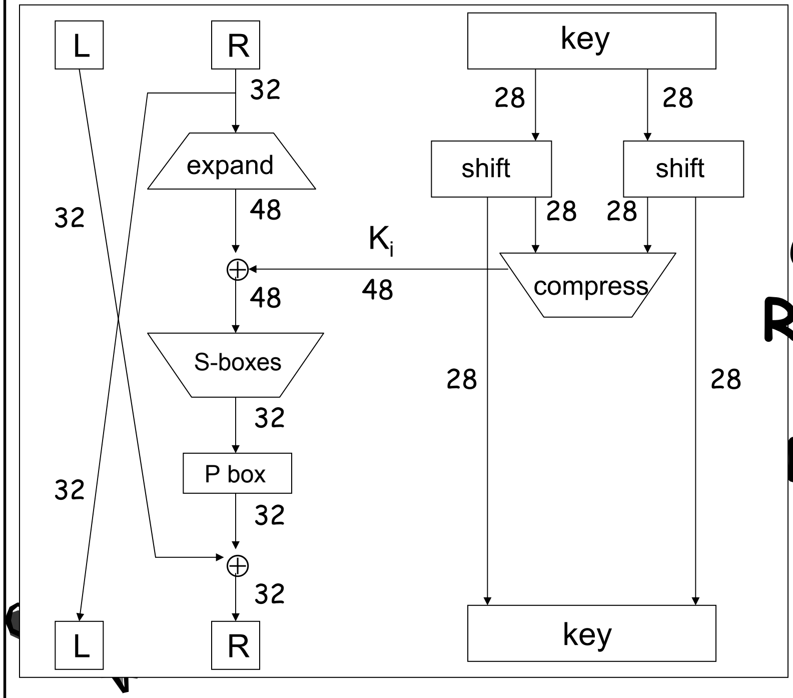
Simple Power Analysis (SPA)

- Directly interprets the power consumption of the device
- Looks for the operations taking place and also the key!
- Trace: A set of power consumptions across a cryptographic process
- 1 millisecond operation sampled at 5MHz yield a trace with 5000 points

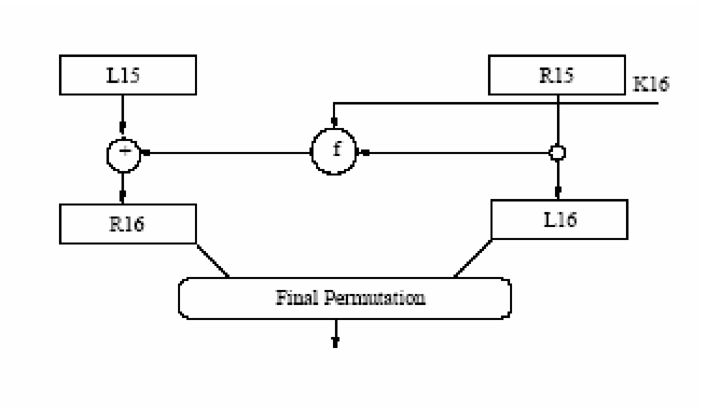
DES Numerology

- DES is a block cipher
- 64 bit block length
- 56 bit key length
- **16 rounds**
- 48 bits of key used each round (subkey)
- Each round is simple (for a block cipher)
- Security depends primarily on “S-boxes”
- Each S-boxes maps 6 bits to 4 bits
- Each S-box has a share of 6 bits of the key

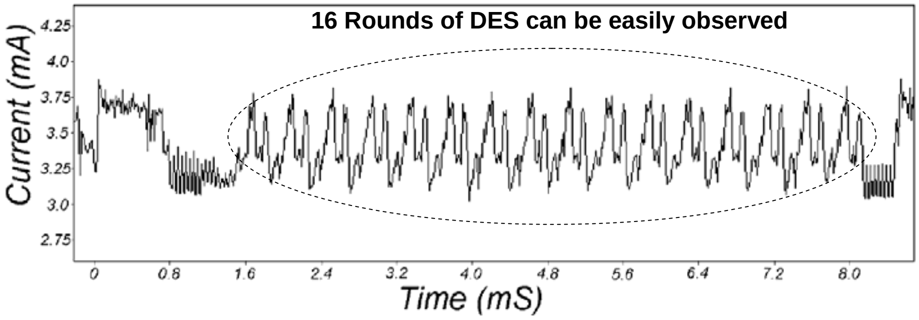
One Round of DES



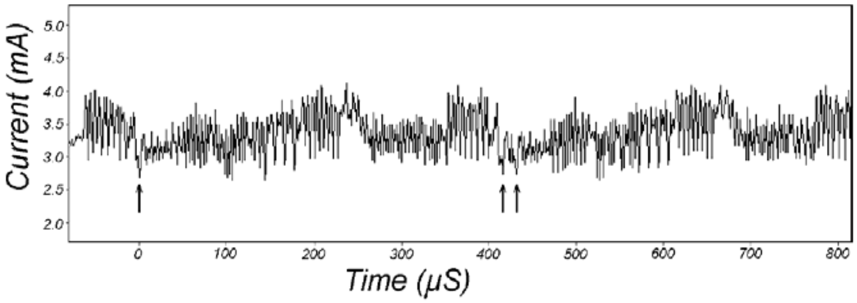
Last Round of DES



Power Traces of DES

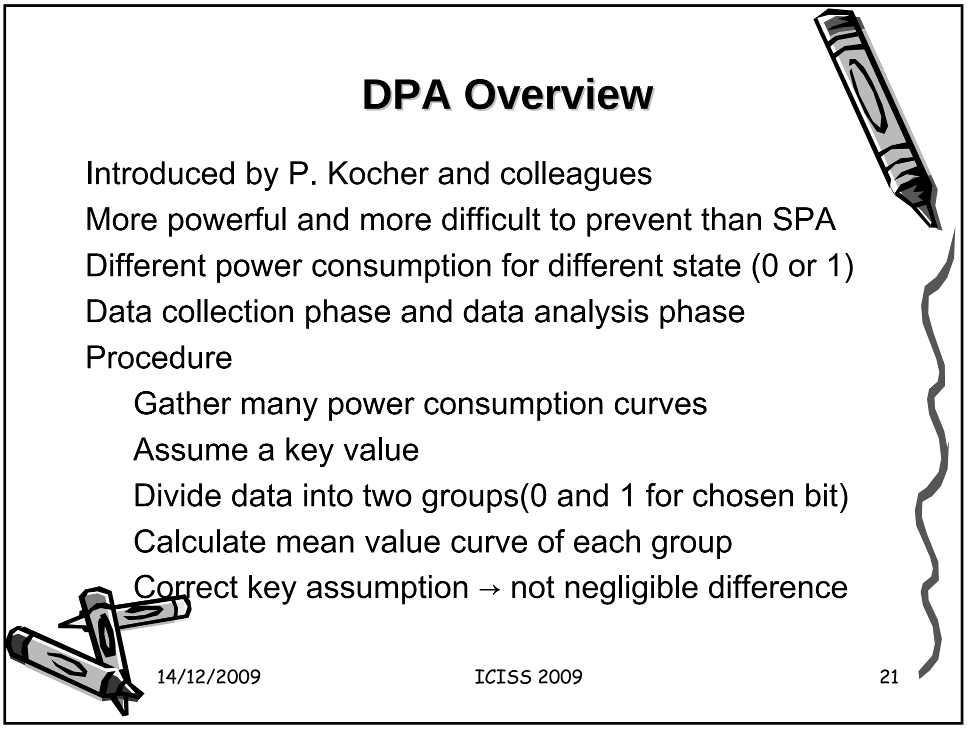


Power Traces for DES



The 28 bit key registers C and D are rotated once in round 2, while twice in round 3. These conditional branches depending on the key bits leak critical information.

Differential Power Analysis (DPA)



DPA Overview

Introduced by P. Kocher and colleagues

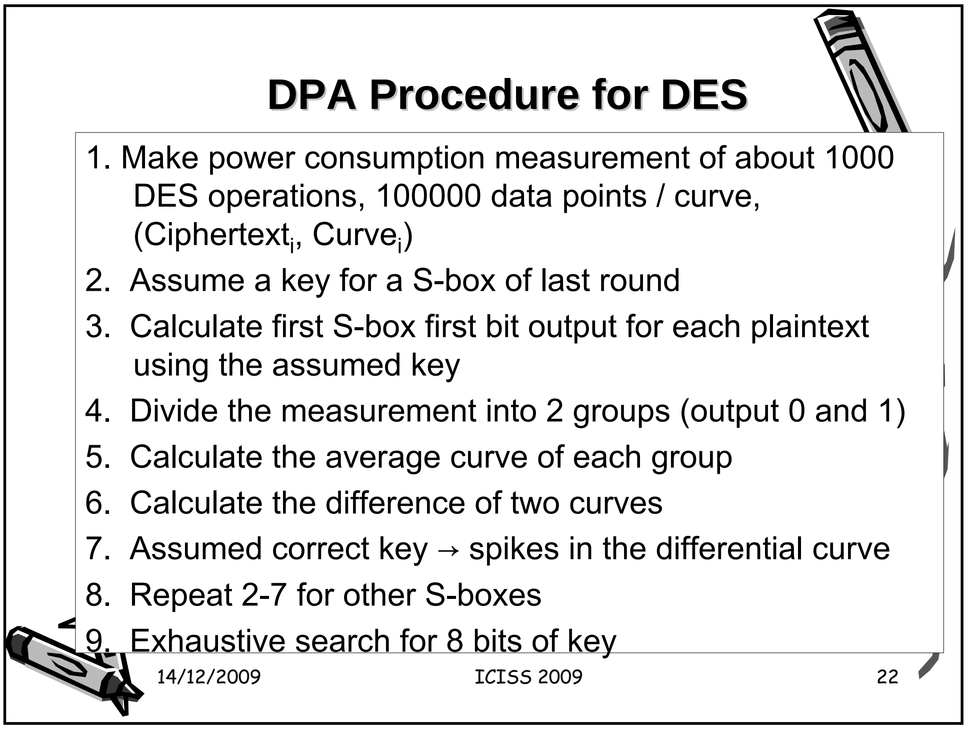
More powerful and more difficult to prevent than SPA

Different power consumption for different state (0 or 1)

Data collection phase and data analysis phase

Procedure

- Gather many power consumption curves
- Assume a key value
- Divide data into two groups(0 and 1 for chosen bit)
- Calculate mean value curve of each group
- Correct key assumption → not negligible difference



DPA Procedure for DES

1. Make power consumption measurement of about 1000 DES operations, 100000 data points / curve, (Ciphertext$_i$, Curve$_i$)
2. Assume a key for a S-box of last round
3. Calculate first S-box first bit output for each plaintext using the assumed key
4. Divide the measurement into 2 groups (output 0 and 1)
5. Calculate the average curve of each group
6. Calculate the difference of two curves
7. Assumed correct key → spikes in the differential curve
8. Repeat 2-7 for other S-boxes
9. Exhaustive search for 8 bits of key

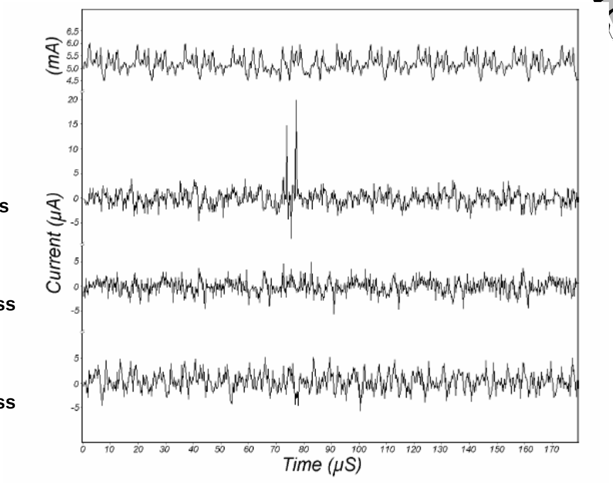
DPA Result Example

Average Power Consumption

Power Consumption Differential Curve With Correct Key Guess

Power Consumption Differential Curve With Incorrect Key Guess

Power Consumption Differential Curve With Incorrect Key Guess



DPA in details

- DPA selection function : D(C,b,K_s) is defined as computing the value of the
  - b^{th} output bit, depending upon
    - C: Ciphertext
    - K_S is the guessed key (6 bits) for the S-Box
- **Note: If K_s is incorrect evaluating D(…) gives the correct bit in half of the cases for each of the ciphertexts.**

DPA in details

- Attacker obtains m encryption operations and capture power traces, $T_{1..m}[1..k]$, with k sample points each.
- An attacker records the m ciphertexts
- No knowledge of the plaintext is required

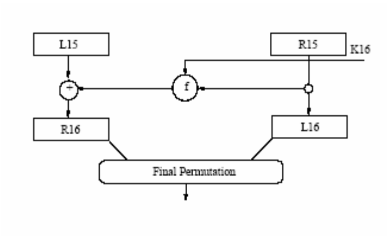
Attacker's Power Board

Sample Points

C I P H E R T E X T S

T[1][1]	T[1][2]		T[1][k]
T[2][1]	T[2][2]		T[2][k]
		. . . .	
T[m][1]	T[m][2]		T[2][k]

The Selection Function D



$$f(R_{15}, K_{16}) = P(S(E(R_{15} \oplus K_{16})))$$

- Attacker knows L16, hence R15
- Attacker knows R16
- Guess K16 (6 bits)
- Compute output of f
- Compute the bth bit of L15
- If K_{16} is wrongly guessed, then the computed value b matches with the correct result half of the time

DPA in details

- Attacker now computes a k-sample differential trace $\Delta_D[1..k]$ by finding the difference between the average of the traces for which D(…) is one and the average for which D(…) is zero.

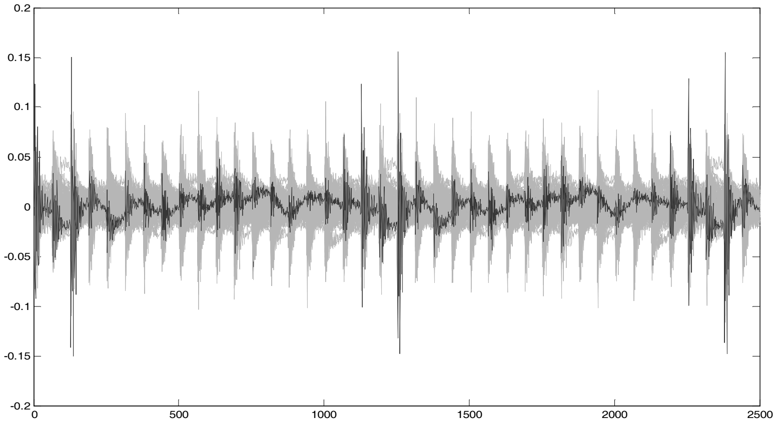
$$\Delta_D = \frac{\sum_{i=1}^{m} D(C_i, b, K_s) T_i[j]}{\sum_{i=1}^{m} D(C_i, b, K_s)} - \frac{\sum_{i=1}^{m} (1 - D(C_i, b, K_s)) T_i[j]}{\sum_{i=1}^{m} (1 - D(C_i, b, K_s))}$$

Principle: If K_s is wrongly guessed, D behaves like a random guess. Thus for a large number of sample points, ΔD[1..k] tends to zero. But if its correct, the differential will be non-zero and show spikes when D is correlated with the value being processed.

DPA in details

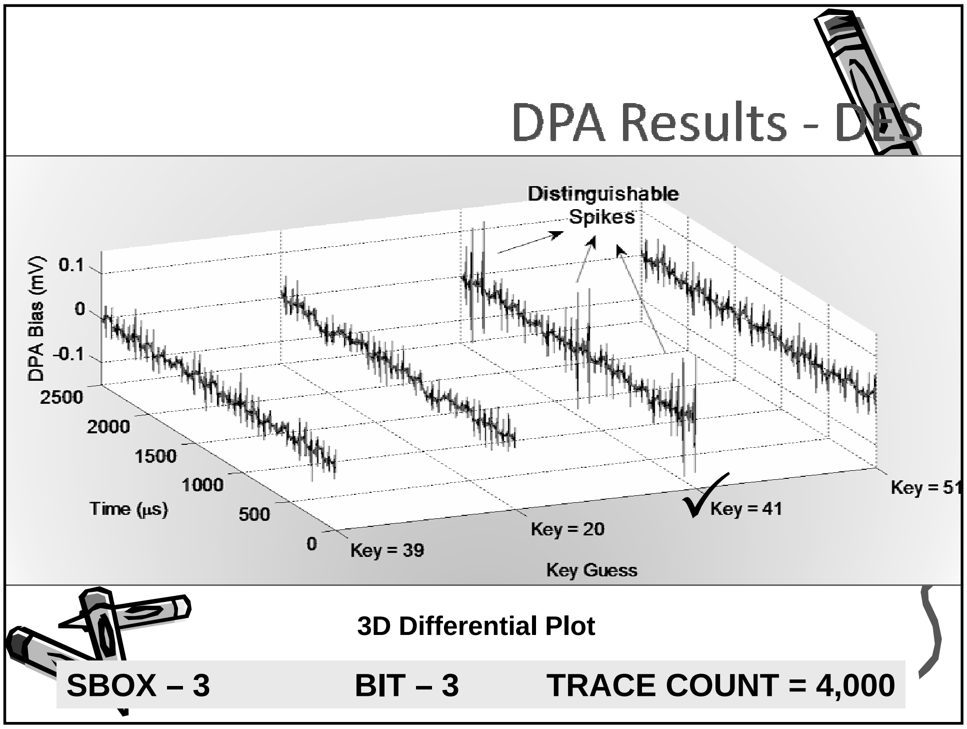
- The correct value of K_s can thus be identified from the spikes
- After computing the 48 bits, one can perform brute force attack on the remaining 8 bits in the keying material.
- Note that noise, measurement errors etc have no effect on this method (as they also are uncorrelated to the data being processed--- just like the wrong guess)…

DPA Results - DES



2D Differential Plot

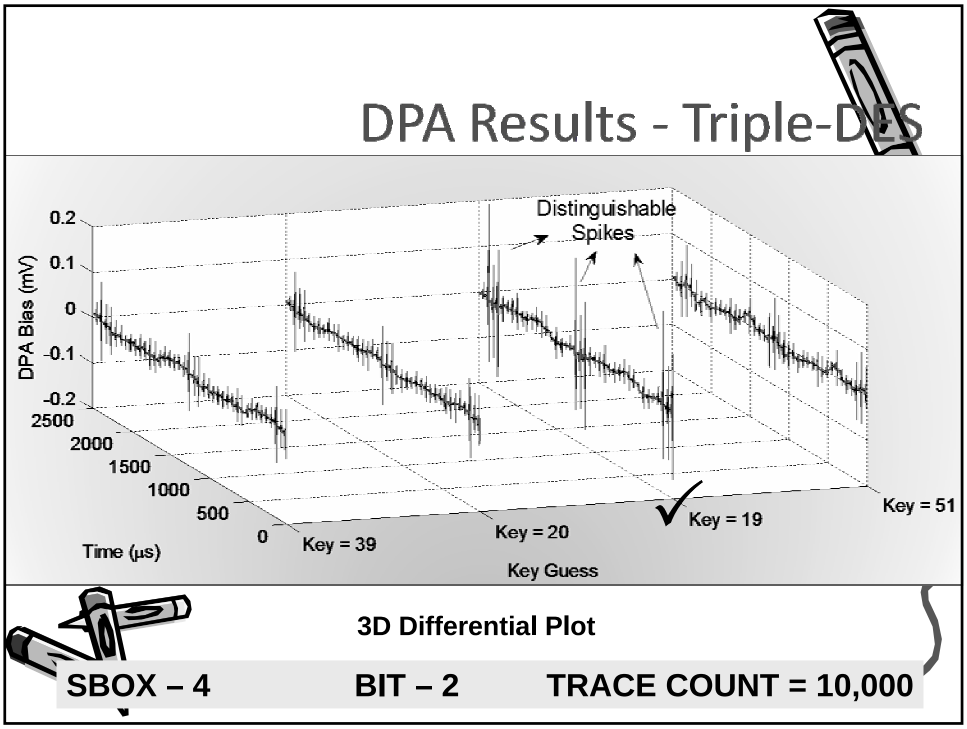
SBOX – 3 BIT – 3 TRACE COUNT = 4,000



DPA Results - DES

3D Differential Plot

SBOX – 3 BIT – 3 TRACE COUNT = 4,000

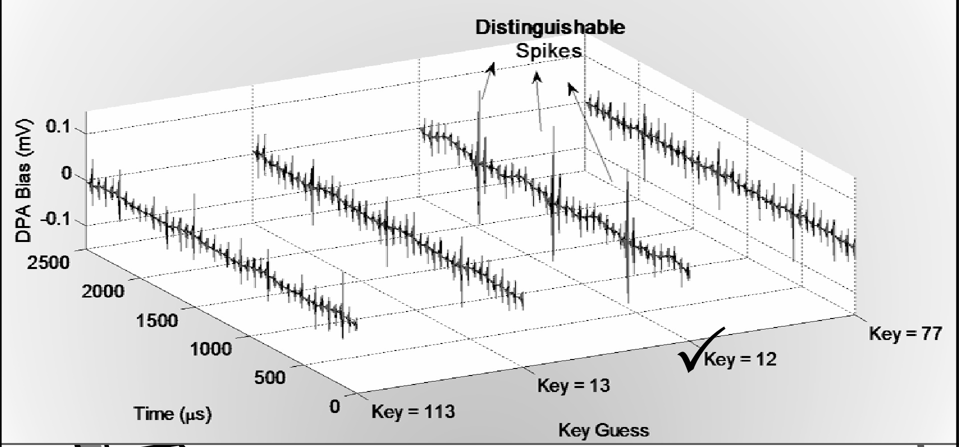


DPA Results - Triple-DES

3D Differential Plot

SBOX – 4 BIT – 2 TRACE COUNT = 10,000

DPA Results - AES



3D Differential Plot

SBOX – 11　　BIT – 8　　TRACE COUNT = 15,000

Countering DPA

- Two broad *approaches* are taken
  - Make the power consumption of the device independent of the data processed
    - Detached power supplies
    - Logic styles with a data independent power consumption
    - Noise generators
    - Insertion of random delays
  - Methods are costly and not in tune with normal CAD methodologies

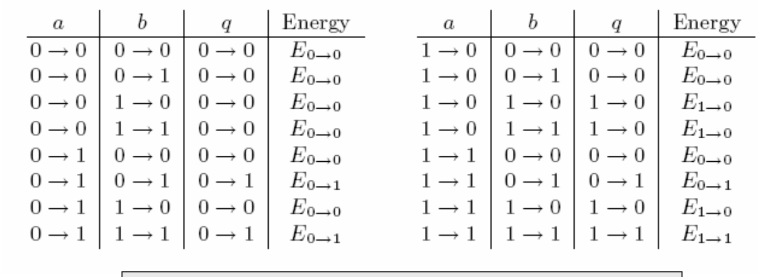
Countering DPA

- *Second Approach* is to randomize the intermediate results
- Based on the principle that the power consumption of the device processing randomized data is uncorrelated to the actual intermediate results
- Masking: Can be applied at the algorithm level or at the gate level

Gate Level Masking

- No wires stores a value that is correlated to an intermediate result of the algorithm.
- Process of converting an unmasked digital circuit to a masked version can be automated

Why are normal gates susceptible to DPA?



a	b	q	Energy
$0 \rightarrow 0$	$0 \rightarrow 0$	$0 \rightarrow 0$	$E_{0\rightarrow 0}$
$0 \rightarrow 0$	$0 \rightarrow 1$	$0 \rightarrow 0$	$E_{0\rightarrow 0}$
$0 \rightarrow 0$	$1 \rightarrow 0$	$0 \rightarrow 0$	$E_{0\rightarrow 0}$
$0 \rightarrow 0$	$1 \rightarrow 1$	$0 \rightarrow 0$	$E_{0\rightarrow 0}$
$0 \rightarrow 1$	$0 \rightarrow 0$	$0 \rightarrow 0$	$E_{0\rightarrow 0}$
$0 \rightarrow 1$	$0 \rightarrow 1$	$0 \rightarrow 1$	$E_{0\rightarrow 1}$
$0 \rightarrow 1$	$1 \rightarrow 0$	$0 \rightarrow 0$	$E_{0\rightarrow 0}$
$0 \rightarrow 1$	$1 \rightarrow 1$	$0 \rightarrow 1$	$E_{0\rightarrow 1}$

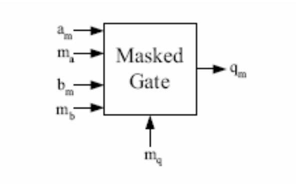
a	b	q	Energy
$1 \rightarrow 0$	$0 \rightarrow 0$	$0 \rightarrow 0$	$E_{0\rightarrow 0}$
$1 \rightarrow 0$	$0 \rightarrow 1$	$0 \rightarrow 0$	$E_{0\rightarrow 0}$
$1 \rightarrow 0$	$1 \rightarrow 0$	$1 \rightarrow 0$	$E_{1\rightarrow 0}$
$1 \rightarrow 0$	$1 \rightarrow 1$	$1 \rightarrow 0$	$E_{1\rightarrow 0}$
$1 \rightarrow 1$	$0 \rightarrow 0$	$0 \rightarrow 0$	$E_{0\rightarrow 0}$
$1 \rightarrow 1$	$0 \rightarrow 1$	$0 \rightarrow 1$	$E_{0\rightarrow 1}$
$1 \rightarrow 1$	$1 \rightarrow 0$	$1 \rightarrow 0$	$E_{1\rightarrow 0}$
$1 \rightarrow 1$	$1 \rightarrow 1$	$1 \rightarrow 1$	$E_{1\rightarrow 1}$

Normal And gate, q= a & b

Why are normal gates susceptible to DPA?

- Attacker measures large number of power traces
- Splits the traces into two groups when q=0 and when q=1 at the end
- of the clock cycles.
- The expected means are not in general equal, leading to DPA attacks
- (as there are spikes in the differential trace)
- Here, means of the energies of the groups are:
- $E(q=0)=(3E_{1->0}+9E_{0->0})/12;$ $E(q=1)=(3E_{0->1}+E_{1->1})/4$
- Since, $E(q=0)\neq E(q=1)$, Hence, DPA attack is possible

Masked And Gate



$$a_m = a \oplus m_a$$
$$b_m = b \oplus m_b$$
$$q_m = q \oplus m_q$$
$$q = f(a,b)$$
$$q_m = \hat{f}(a_m, m_a, b_m, m_b, m_q)$$

Masked And Gate

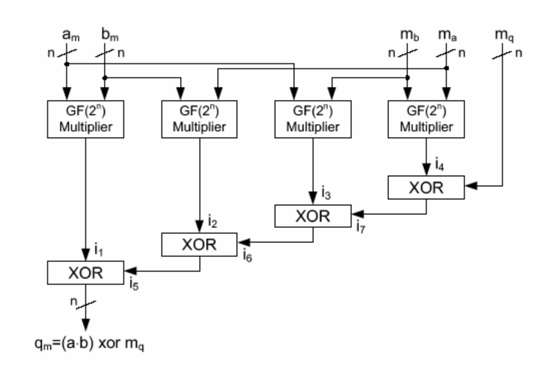
q_m=(a.b)xor m_q=($a_m$xor m_a).(b_m xor m_b) xor m_q
=(a_m.b_m) (xor (b_m.m_a) (xor (m_b.a_m) (xor ((m_a.m_b) xor m_q))))

Masked And Gate

- There are 4^5=1024 possible input transmissions that can occur.
- It turns out that the expected value of the energy required for the
- processing of q=0 and q=1 are identical.
- Thus protected against DPA, under the assumption that the CMOS

  gates switch only once in one clock cycles.
- But we know there are glitches, and so the output of gates swing a number of times before reaching a steady state. Hence... the argument continues.

Masked Multiplier



Same Principle may be applied for multiplier circuits.

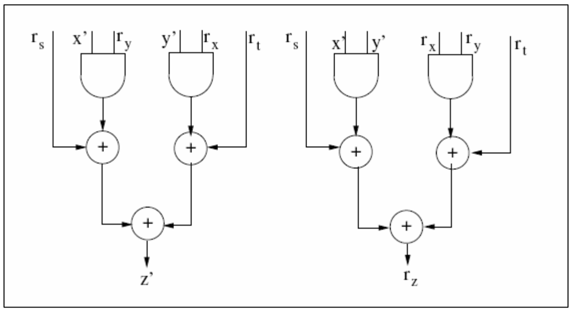
But Masked Circuits are not safe

- Transitions, $T(a_m)$, $T(m_a)$, $T(b_m)$, $T(m_b)$ does not leak
- Correlations, $\rho(T(i_j),a)= \rho(T(i_j),b)= \rho(T(i_j),c)=0$, for $j=1$ to 4.
- So xor gates leak information about unmasked values
- Reason is that the xor gates does not change output when both the inputs change value simultaneously or within a small time

But Masked Circuits are not safe

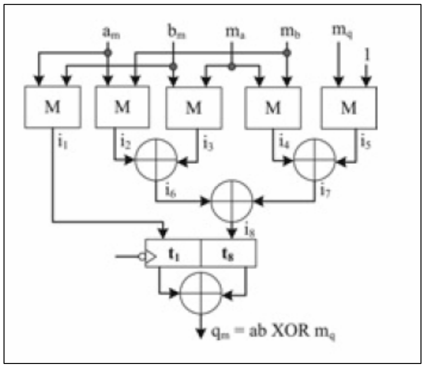
- Thus the power consumption of the xor gates depend on the time of arrival of the signals i_1 to i_4.
- These time delays are related to the unmasked values
- Thus the masked circuits are still vulnerable to DPA, because of delays in circuits.

Data dependent glitch free circuit



K. Kumar, D. Mukhopadhyay, D.RoyChowdhury, "Design of a Differential Power Analysis Resistant Masked AES S-Box", INDOCRYPT 2007

Pipe-lined AES S-Box



- **M. Alam, S.Ghosh, M.J. Mohan, D. Mukhopadhyay, D.R. Chowdhury, I.S.Gupta, "Effect of Glitches against masked AES S-Box Implementation and countermeasures", IET Security, Vol 3(1),, 2009**

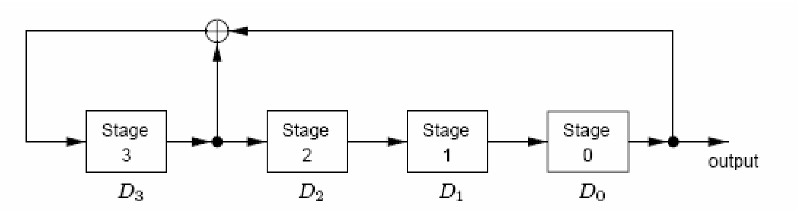
Stream Ciphers

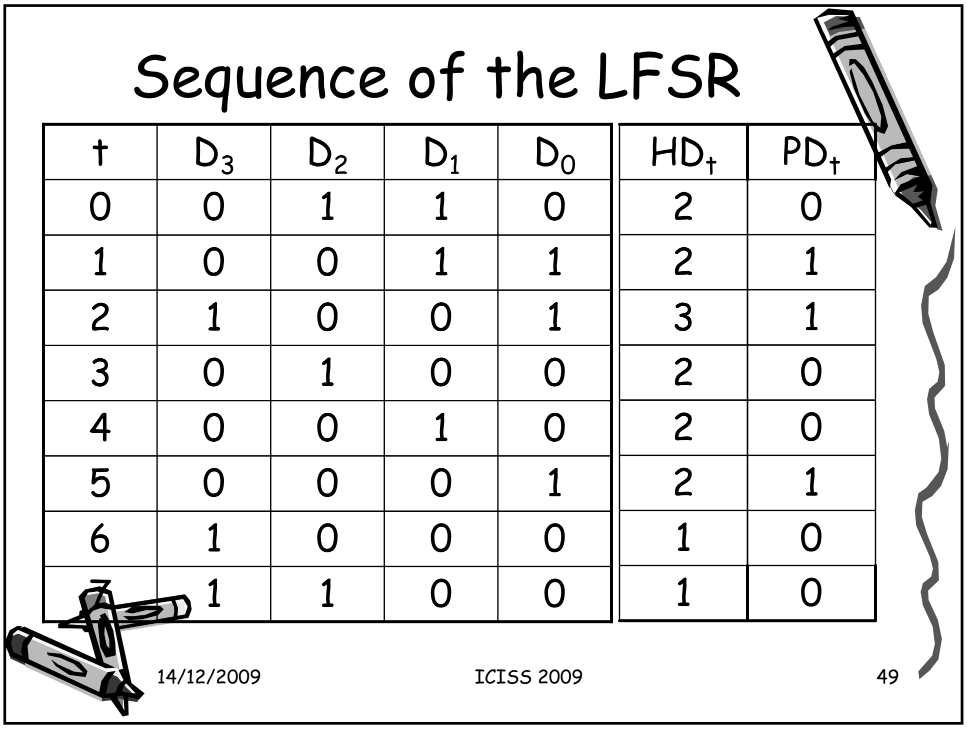
- LFSR (Linear Feedback Shift Registers) are used as building blocks for stream ciphers
- LFSRs are susceptible to power based SCAs
- An n-bit LFSR can be completely determined by making O(n) power measurements.
  - neither the primitive polynomial nor the value of n be known to the attacker.

S. Burman, D. Mukhopadhyay, V. Kamakoti, "LFSR Based Stream Ciphers are Vulnerable to Power Attacks", INDOCRYPT 2007

Example

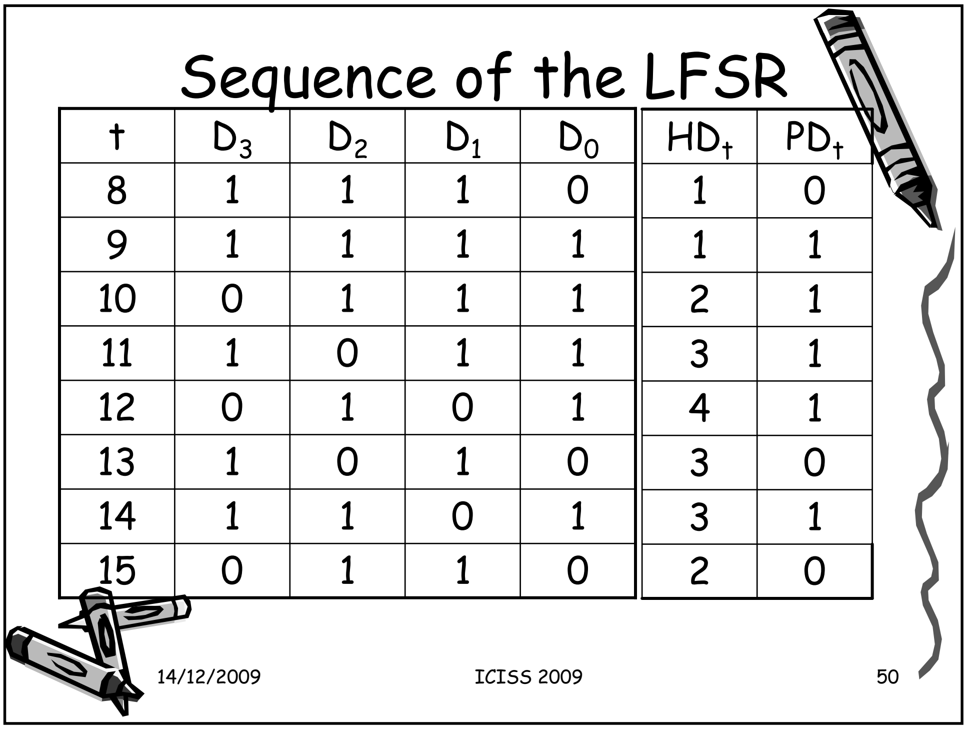
- Consider the LFSR <4,1+D+D^4>





Sequence of the LFSR

t	D_3	D_2	D_1	D_0	HD_t	PD_t
0	0	1	1	0	2	0
1	0	0	1	1	2	1
2	1	0	0	1	3	1
3	0	1	0	0	2	0
4	0	0	1	0	2	0
5	0	0	0	1	2	1
6	1	0	0	0	1	0
7	1	1	0	0	1	0



Sequence of the LFSR

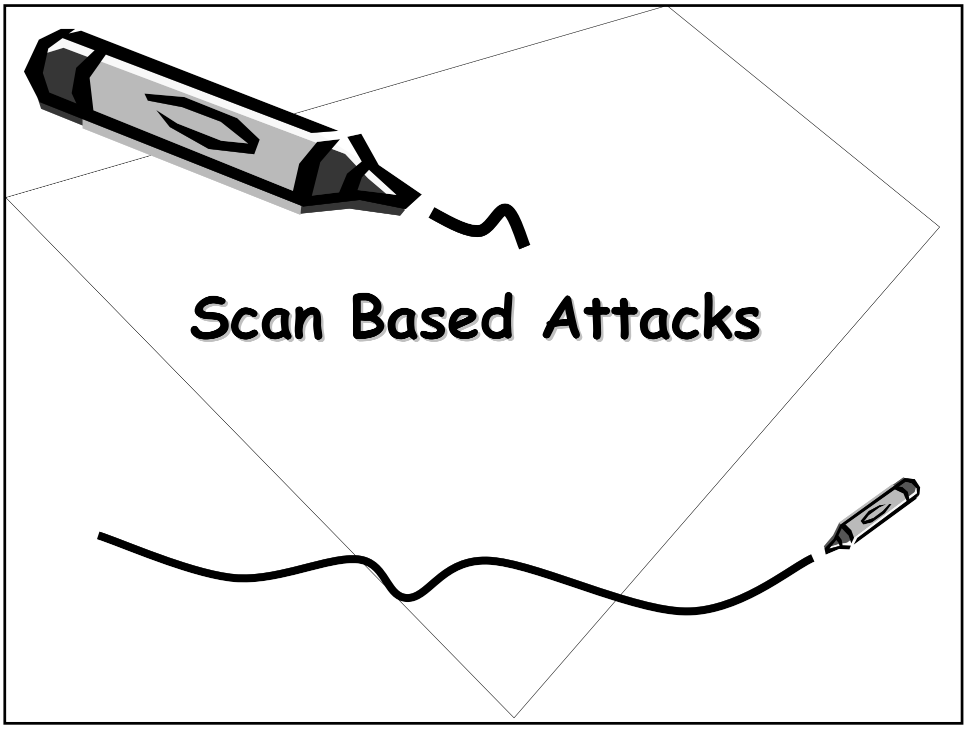
t	D_3	D_2	D_1	D_0	HD_t	PD_t
8	1	1	1	0	1	0
9	1	1	1	1	1	1
10	0	1	1	1	2	1
11	1	0	1	1	3	1
12	0	1	0	1	4	1
13	1	0	1	0	3	0
14	1	1	0	1	3	1
15	0	1	1	0	2	0

The Attack

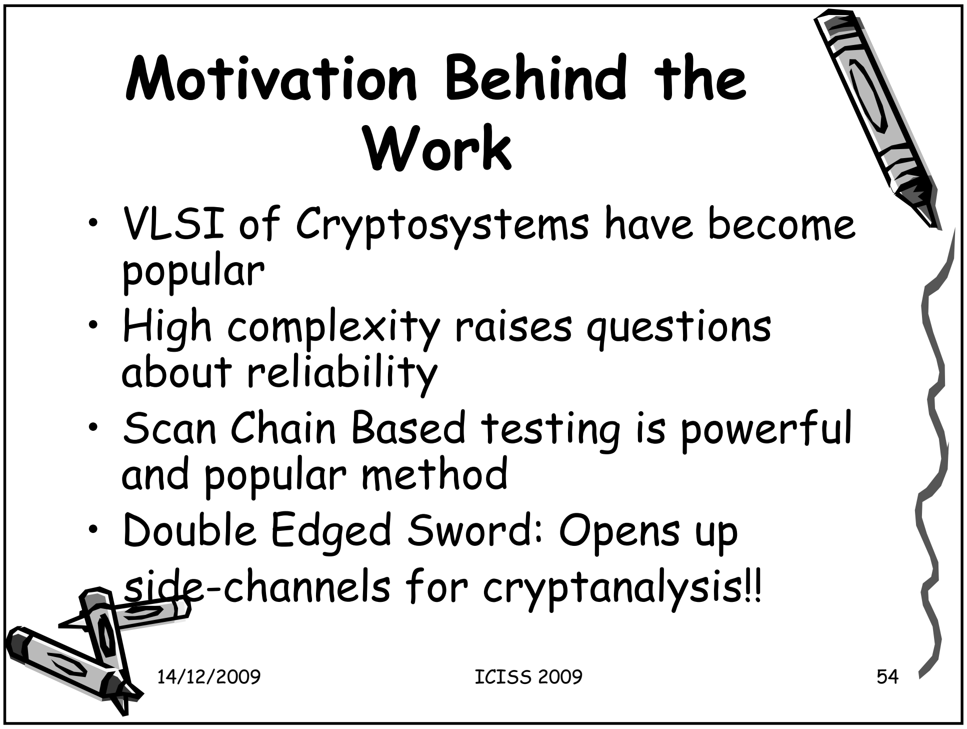
- Step 1: Measure Pow(0) (dynamic power) at t=0
- Step 2: For t=k, k≥1:
  - Measure Pow(k) (dynamic power)
  - PD'_{k-1}=1, if Pow(k-1)≠Pow(k), else 0
  - Input PD'_{k-1} to Berlekamp-Massey (BM). If BM terminates then exit, else repeat Step 2.

The Attack

- Result:
  - BM outputs the length of the LFSR, feedback polynomial, F and the connection polynomial realized by F.
  - The initial state can be ascertained by solving a system of linear equations from the previous knowledge.



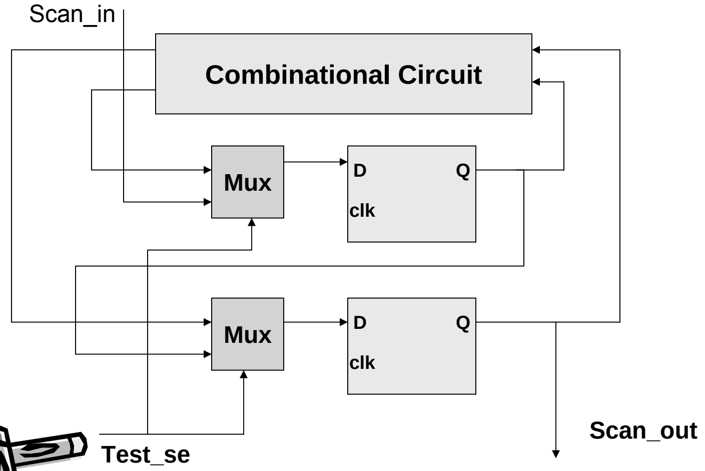
Scan Based Attacks



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 2004

  ATTACKED A BLOCK CIPHER
- D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury, and B. Bhattacharya, *"Cryptoscan: Secured Scan Chain Architecture",* ATS 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

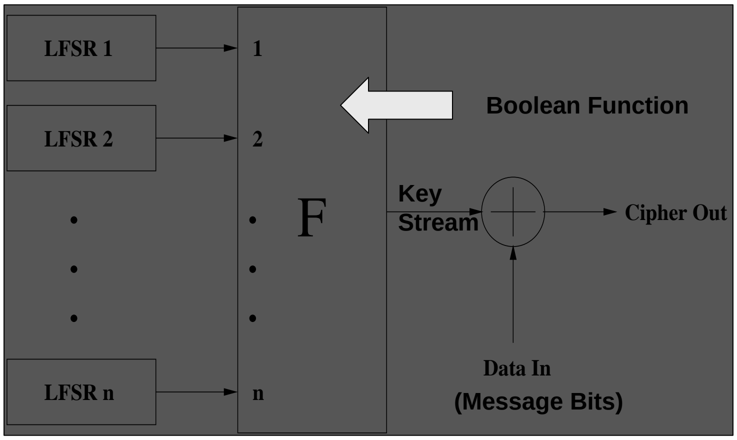
Overview of contemporary research

- Bo Yang, Kaijie Wu and R. Karri, “Secure scan:A Design-for-test Architecture for Crypto-chips”, DAC, 2005, pp. 135-140.
- G. Sengar, D. Mukhopadhyay and D.RoyChowdhury, “Secured Flipped Scan Chain Model for Crypto-architecture", IEEE Transactions on CAD, Nov 2007, Volume 26, Issue: 11 pp 2080-2084.
- Mukesh Agrawal, Sandip Karmakar, Dhiman Saha, Debdeep Mukhopadhyay, "Scan Based Side Channel Attacks on Stream Ciphers and Their Counter-Measures", INDOCRYPT 2008, pp. 226-238, LNCS.

Attackers can be smart…

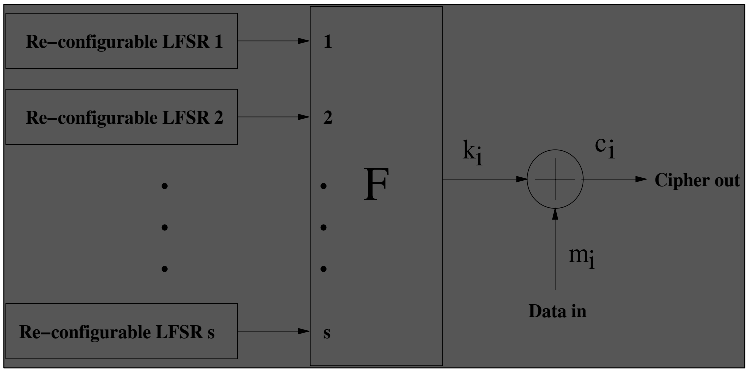
The Joy of Tech
by Nitrozac & Snaggy
MICROSOFT IS VERY SECURE. WE KNOW THERE ARE HACKERS OUT THERE, BUT WE ARE CONSTANTLY VIGILANT AND ANY ACTIVITY IS CAREFULLY MONITORED.
THAT'S GREAT TO KNOW. THANK YOU VERY MUCH, MR. GATES, SIR.
HEH, THAT WENT WELL. DID YOU GET HIS WALLET?
OH YA. LIKE CANDY FROM A BABY.
© 2000 Geek Culture™
joyoftech.com

Classical Structure of Stream Cipher



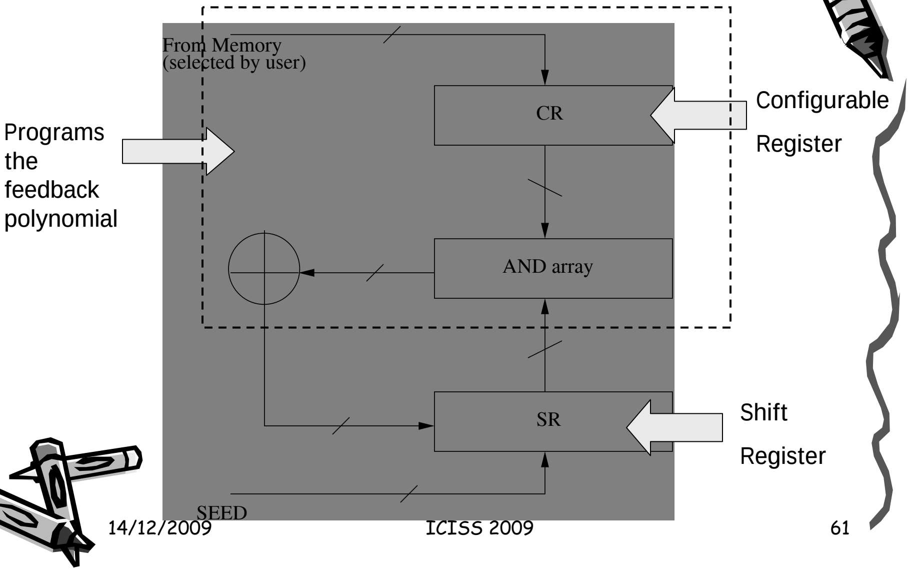
D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury and B. Bhattacharya, “CryptoScan: Secured Scan Chain Architecture”, ATS 2005

Hardware Implementation



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

Re-configurable LFSR



Programs the feedback polynomial

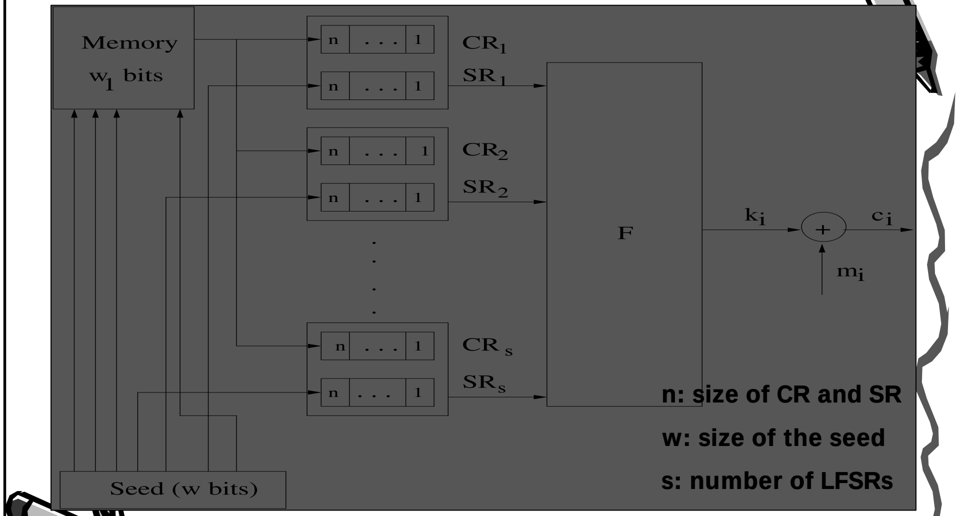
Configurable Register

Shift Register

Attacking the Stream Cipher Using Scan Chains

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

Attacking Environment



n: size of CR and SR

w: size of the seed

s: number of LFSRs

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

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))*

Ascertain the Structure of the Seed…

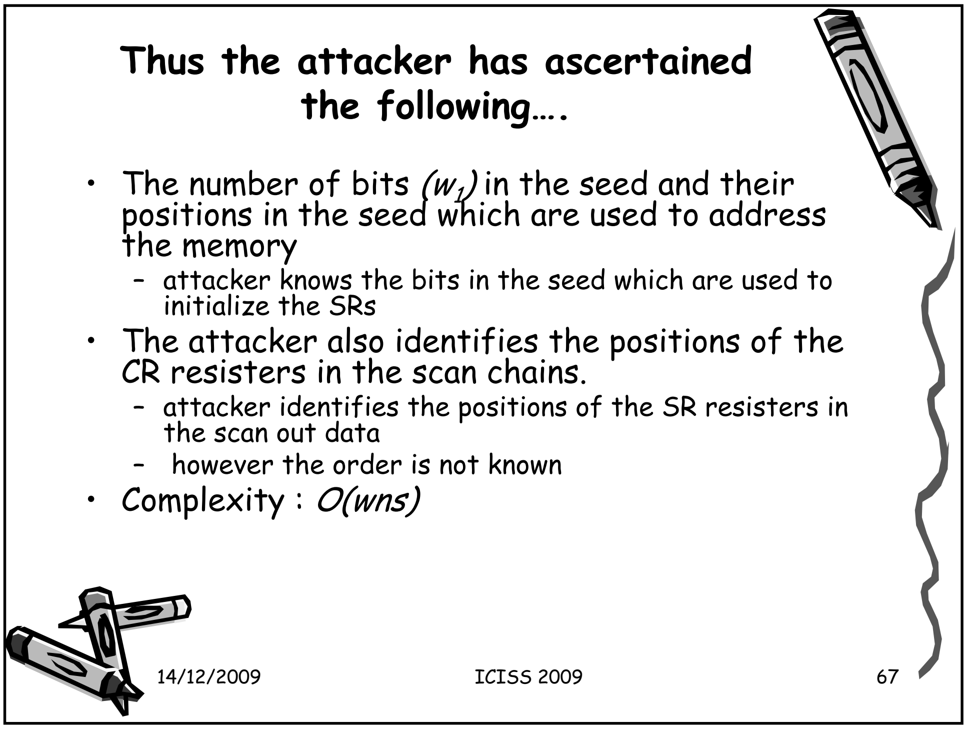
- Next, the attacker sets the first bit of seed to 1 and the rest to 0 and apply one clock cycle
- The bit with value 1 can go either to the memory or to the SRs
- Scan out the data in test mode.
- If the bit goes to the SR,

  no of ones = *s.wt(m(0))+1*

  else no of ones = *s.wt(m(p))*

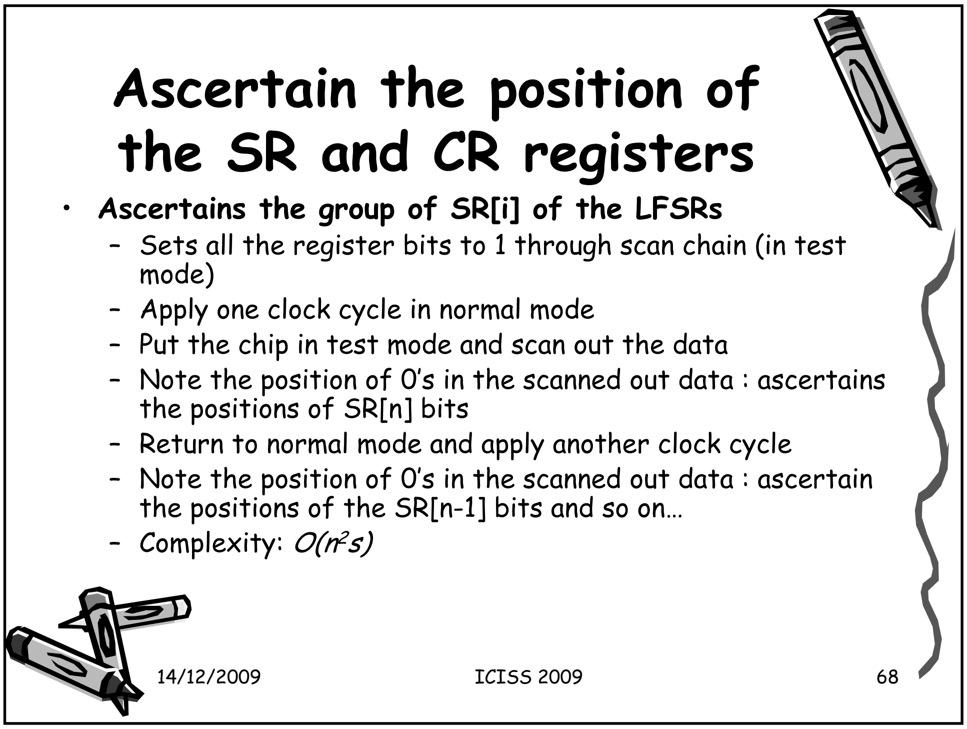
  Not Equal

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



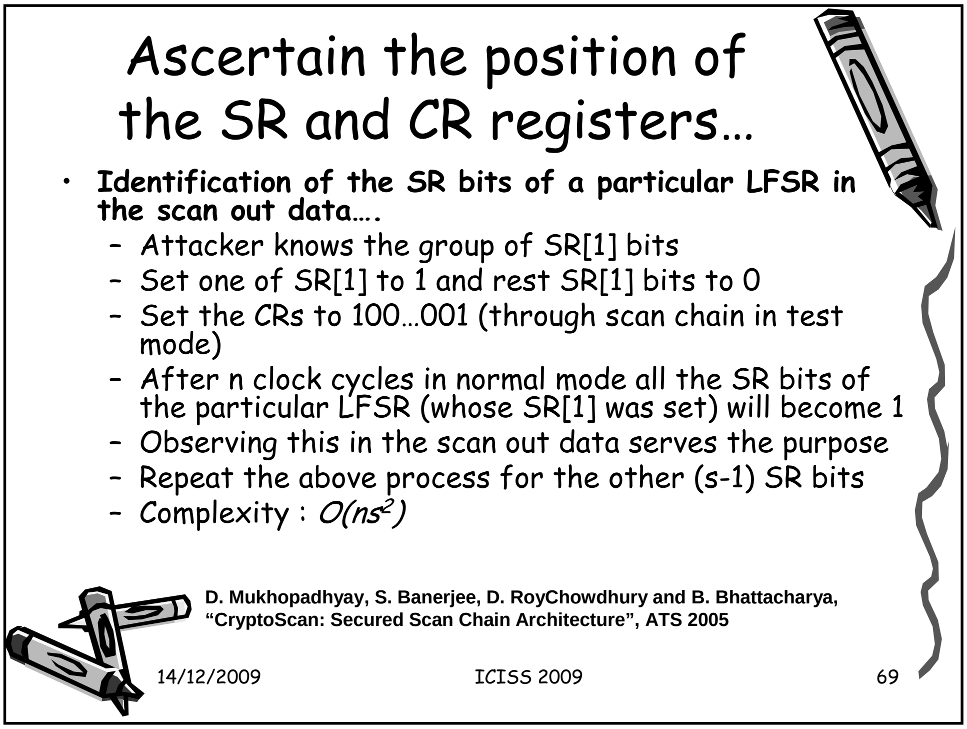
Thus the attacker has ascertained the following….

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



Ascertain the position of the SR and CR registers

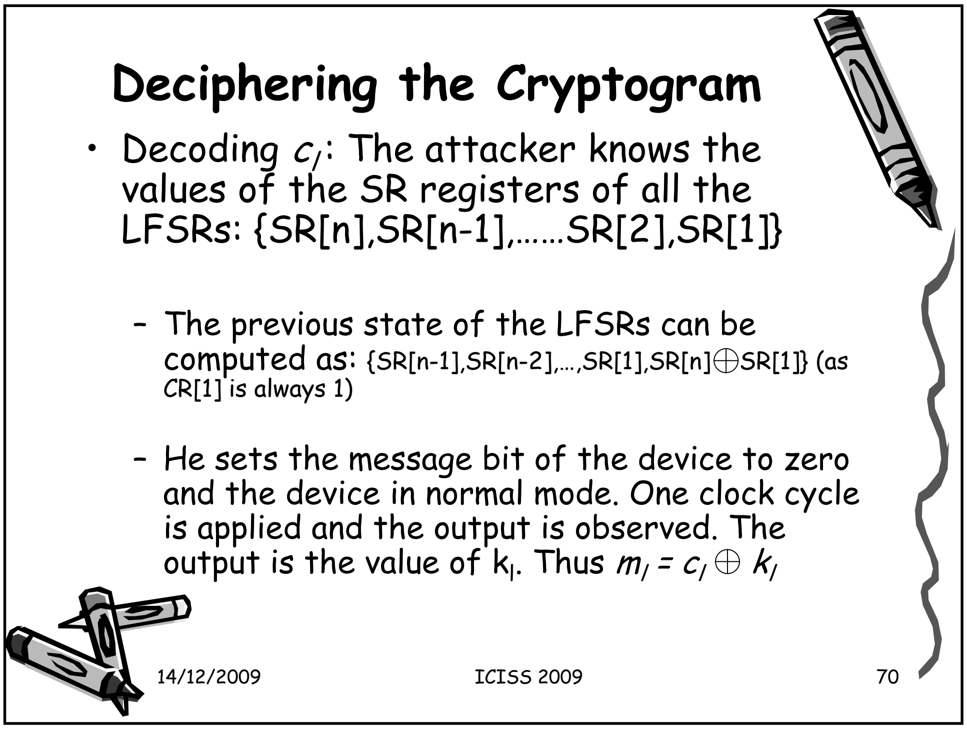
- **Ascertains the group of SR[i] of the LFSRs**
  - Sets all the register bits to 1 through scan chain (in test mode)
  - Apply one clock cycle in normal mode
  - Put the chip in test mode and scan out the data
  - Note the position of 0’s in the scanned out data : ascertains the positions of SR[n]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



Ascertain the position of the SR and CR registers…

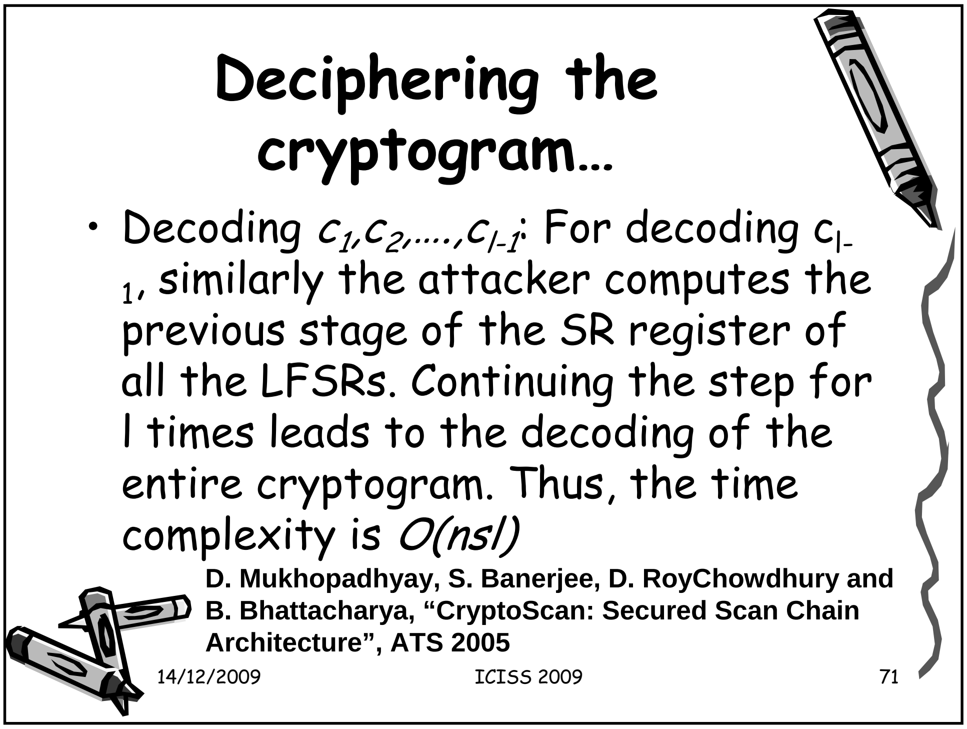
- **Identification of the SR bits of a particular LFSR in the scan out data….**
  - Attacker knows the group of SR[1] bits
  - Set one of SR[1] to 1 and rest SR[1] bits to 0
  - Set the CRs to 100…001 (through scan chain in test mode)
  - After n clock cycles in normal mode all the SR bits of the particular 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

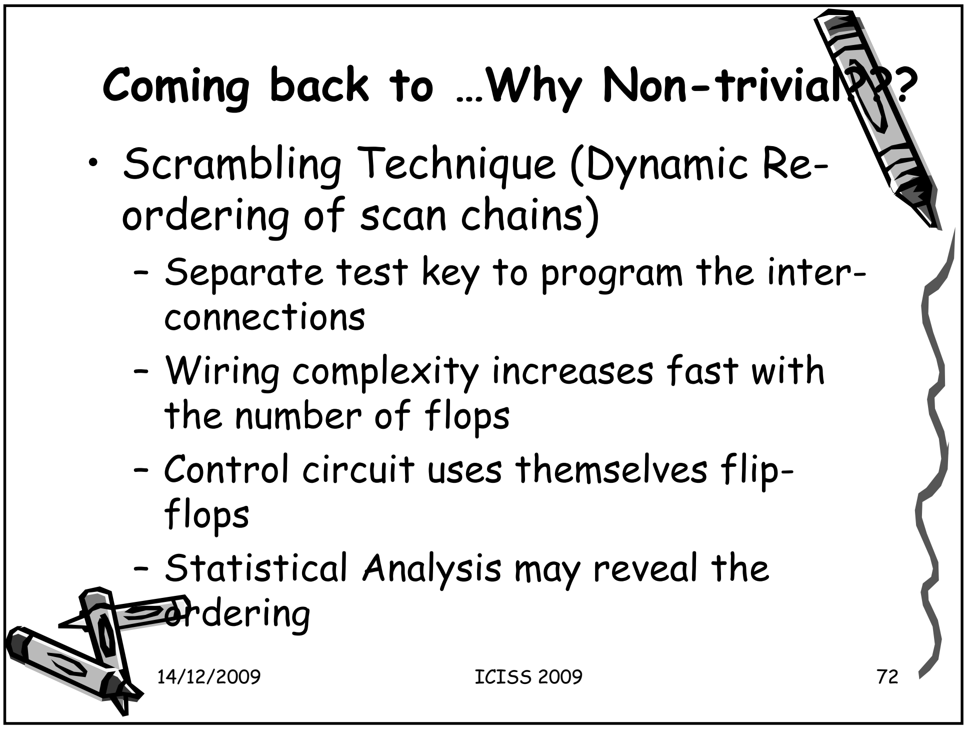
- Decoding c_l: The attacker knows the values of the SR registers of all the LFSRs: {SR[n],SR[n-1],……SR[2],SR[1]}
  - The previous state of the LFSRs can be computed as: {SR[n-1],SR[n-2],…,SR[1],SR[n]$\oplus$SR[1]} (as CR[1] is always 1)
  - He sets the message bit of the device to zero and the device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



Deciphering the cryptogram…

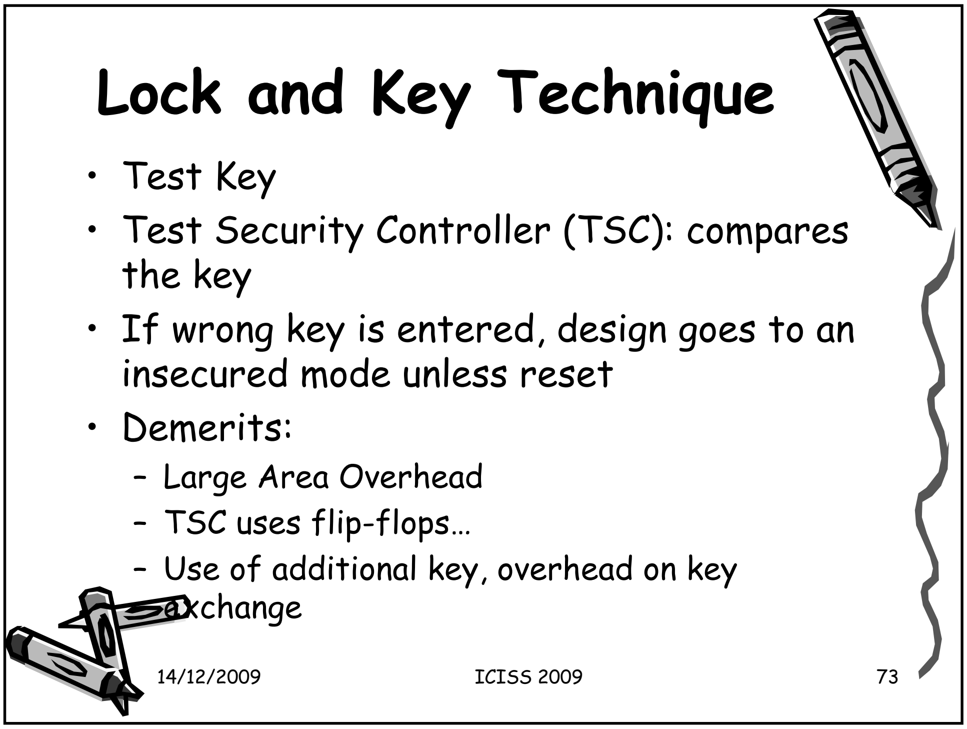
- Decoding $c_1,c_2,….,c_{l-1}$: For decoding c_{l-1}, similarly the attacker computes the previous stage of the SR register of all the LFSRs. Continuing the step for l times leads to the decoding of the entire cryptogram. Thus, the time complexity is $O(nsl)$

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



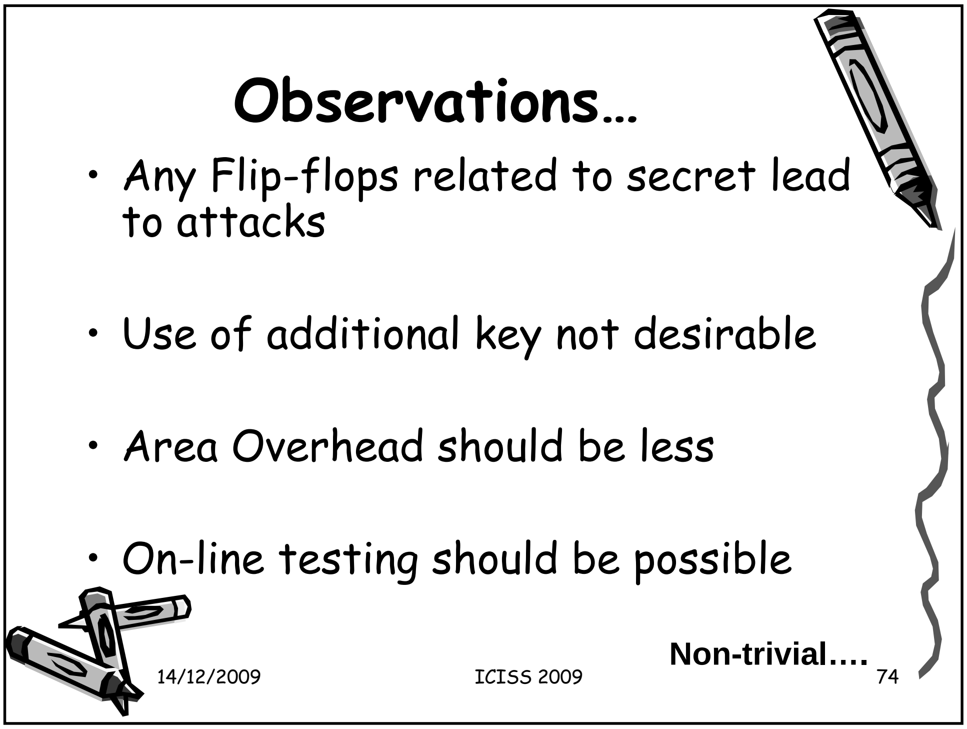
Coming back to …Why Non-trivial??

- 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



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…

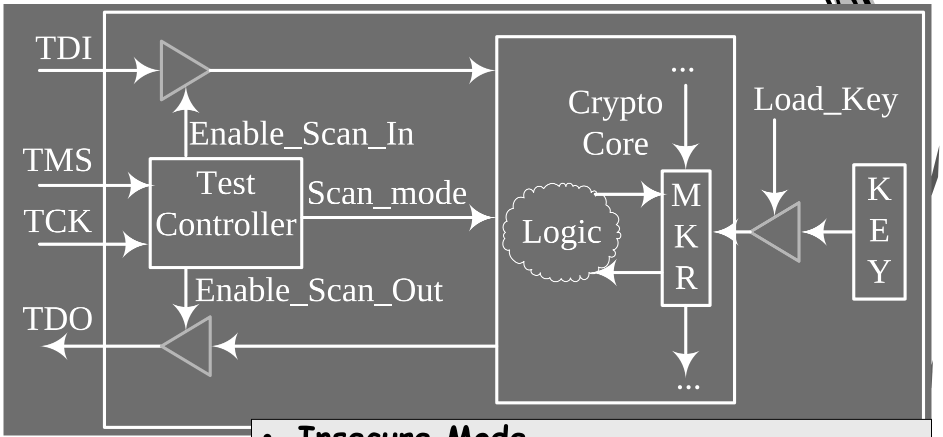
- Any Flip-flops related to secret lead to attacks
- Use of additional key not desirable
- Area Overhead should be less
- On-line testing should be possible

Non-trivial….

Secure Scan

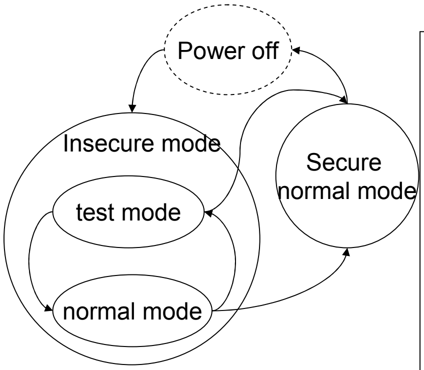
- Test and debug crypto chips using general scan based DFT
  - Information obtained from scan chains should not be useful in retrieving the secret key
- Two copies of the secret key
  - Secure key: hardwired or in secure memory
  - Mirror Key (MKR): used for testing
- Two modes of operation: Insecure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

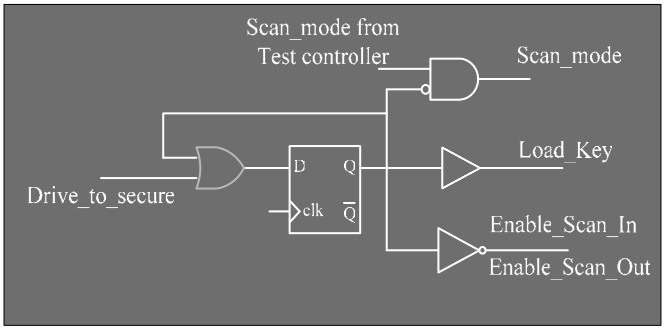
Secure Scan: State Diagram

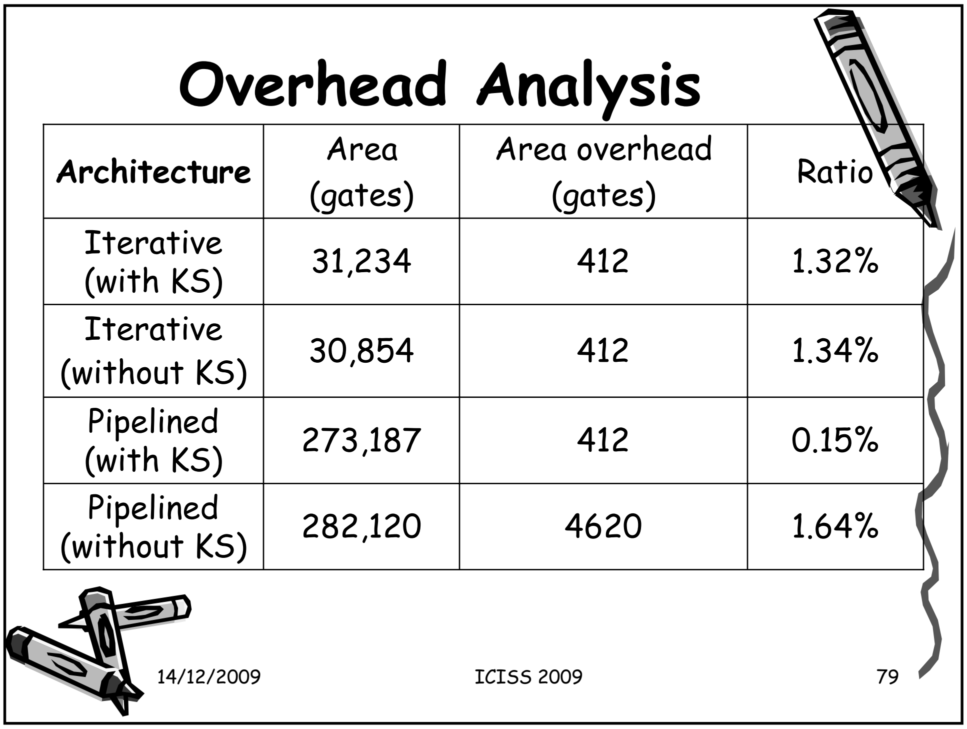


- **Enable Scan if Load_Key = ‘0’, Enable_Scan_In = ‘1’and Enable_Scan_Out = ‘1’**
- **Disable Scan if Load_Key = ‘1’, Enable_Scan_In = ‘0’and Enable_Scan_Out = ‘0’**

Secure Scan: Test Controller

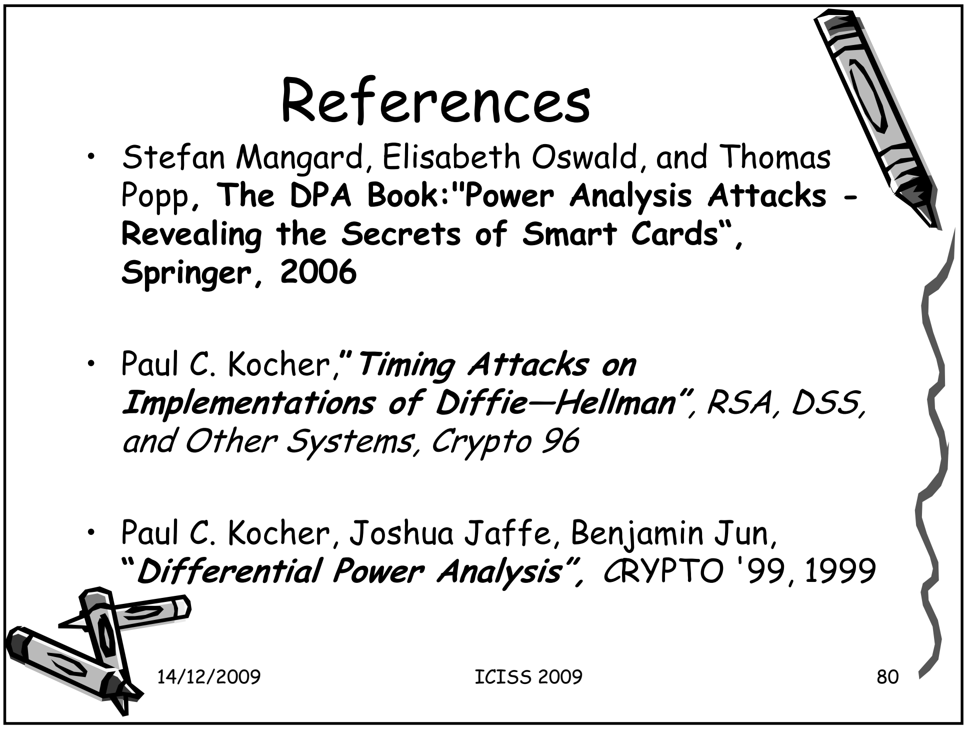
- Modify IEEE 1149.1 Test Controller
  - New instruction: Drive_to_secure
  - Three new output control signals
- Dedicated Secure Control Circuit





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%

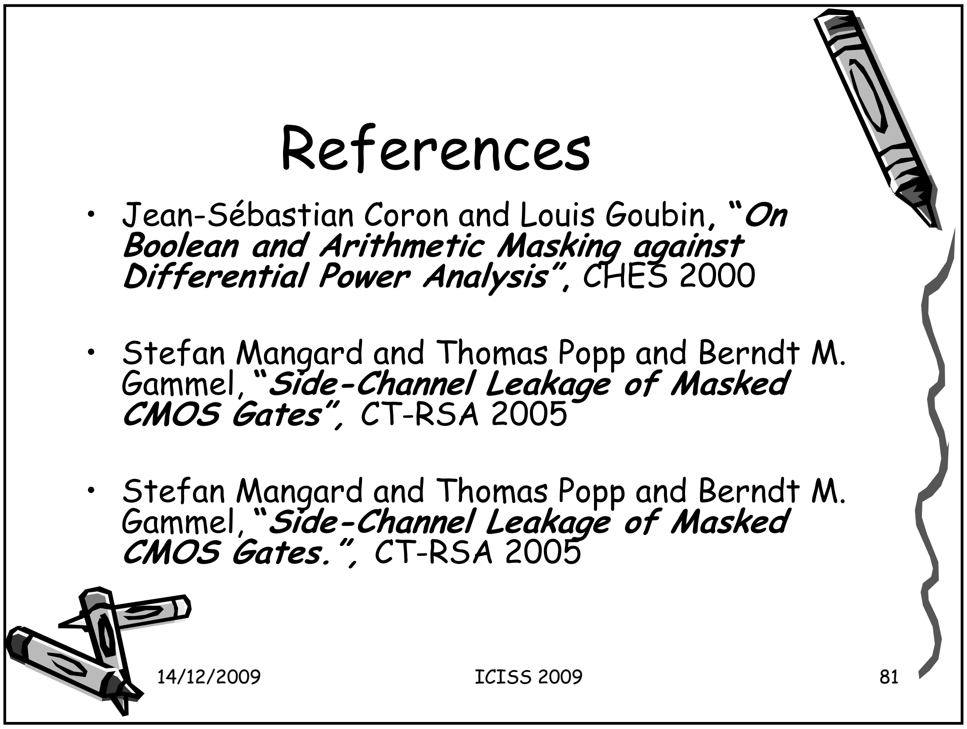


References

- Stefan Mangard, Elisabeth Oswald, and Thomas Popp, **The DPA Book:"Power Analysis Attacks - Revealing the Secrets of Smart Cards", Springer, 2006**

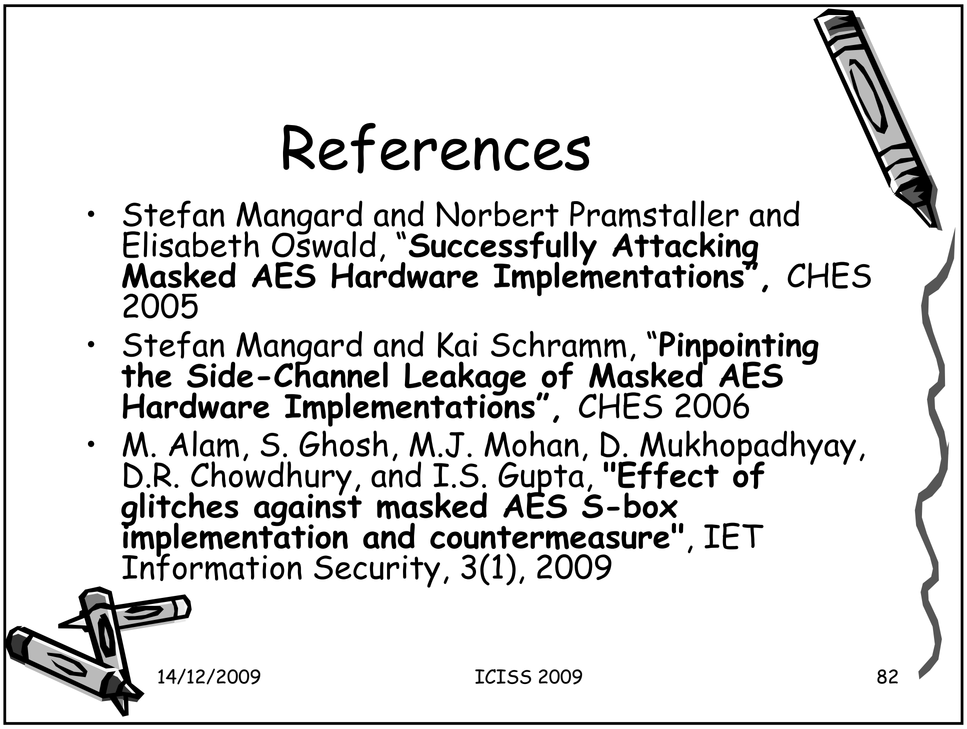
- Paul C. Kocher,"***Timing Attacks on Implementations of Diffie—Hellman"**, RSA, DSS, and Other Systems, Crypto 96*

- Paul C. Kocher, Joshua Jaffe, Benjamin Jun, "***Differential Power Analysis",*** CRYPTO '99, 1999



References

- Jean-Sébastian Coron and Louis Goubin, ***"On Boolean and Arithmetic Masking against Differential Power Analysis",*** CHES 2000
- Stefan Mangard and Thomas Popp and Berndt M. Gammel, ***"Side-Channel Leakage of Masked CMOS Gates",*** CT-RSA 2005
- Stefan Mangard and Thomas Popp and Berndt M. Gammel, ***"Side-Channel Leakage of Masked CMOS Gates.",*** CT-RSA 2005



References

- Stefan Mangard and Norbert Pramstaller and Elisabeth Oswald, "**Successfully Attacking Masked AES Hardware Implementations",** CHES 2005
- Stefan Mangard and Kai Schramm, "**Pinpointing the Side-Channel Leakage of Masked AES Hardware Implementations",** CHES 2006
- M. Alam, S. Ghosh, M.J. Mohan, D. Mukhopadhyay, D.R. Chowdhury, and I.S. Gupta, **"Effect of glitches against masked AES S-box implementation and countermeasure"**, IET Information Security, 3(1), 2009

References

- K. Kumar, D. Mukhopadhyay and D. RoyChowdhury, **"Design of a Differential Power Analysis Resistant AES S-Box"**, INDOCRYPT 2007, pp 373-383.
- S. Burman, D. Mukhopadhyay and V. Kamakoti, **"LFSR Based Stream Ciphers are vulnerable to Power Attacks",** INDOCRYPT 2007, pp 384-392.

References

- D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury and B. Bhattacharya, **,"CryptoScan: A Secured Scan Chain Architecture",** ATS 2005
- G. Sengar, D. Mukhopadhyay and D.RoyChowdhury, "Secured Flipped Scan Chain Model for Crypto-architecture", IEEE Transactions on CAD, 2007.
- Bo Yang, Kaijie Wu, Ramesh Karri: "**Secure scan: a design-for-test architecture for crypto chips",** DAC 2005