Scan based Attack on Hardware Implementations of Data Encryption Standard

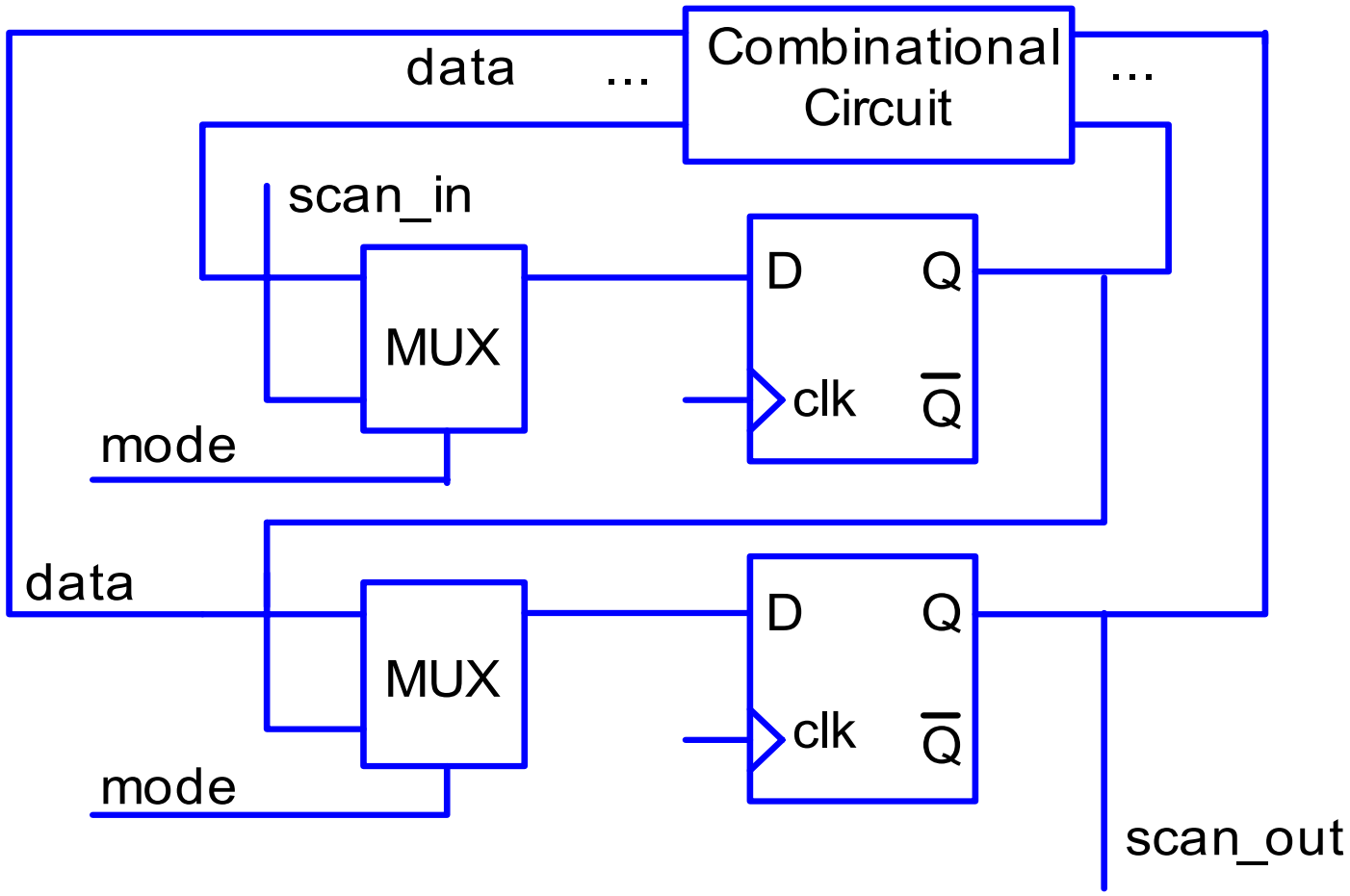
Bo Yang, Kaijie Wu and Ramesh Karri
ECE Department
Polytechnic University, Brooklyn, NY USA

Presented by Nikhil Joshi, Polytechnic U, NY

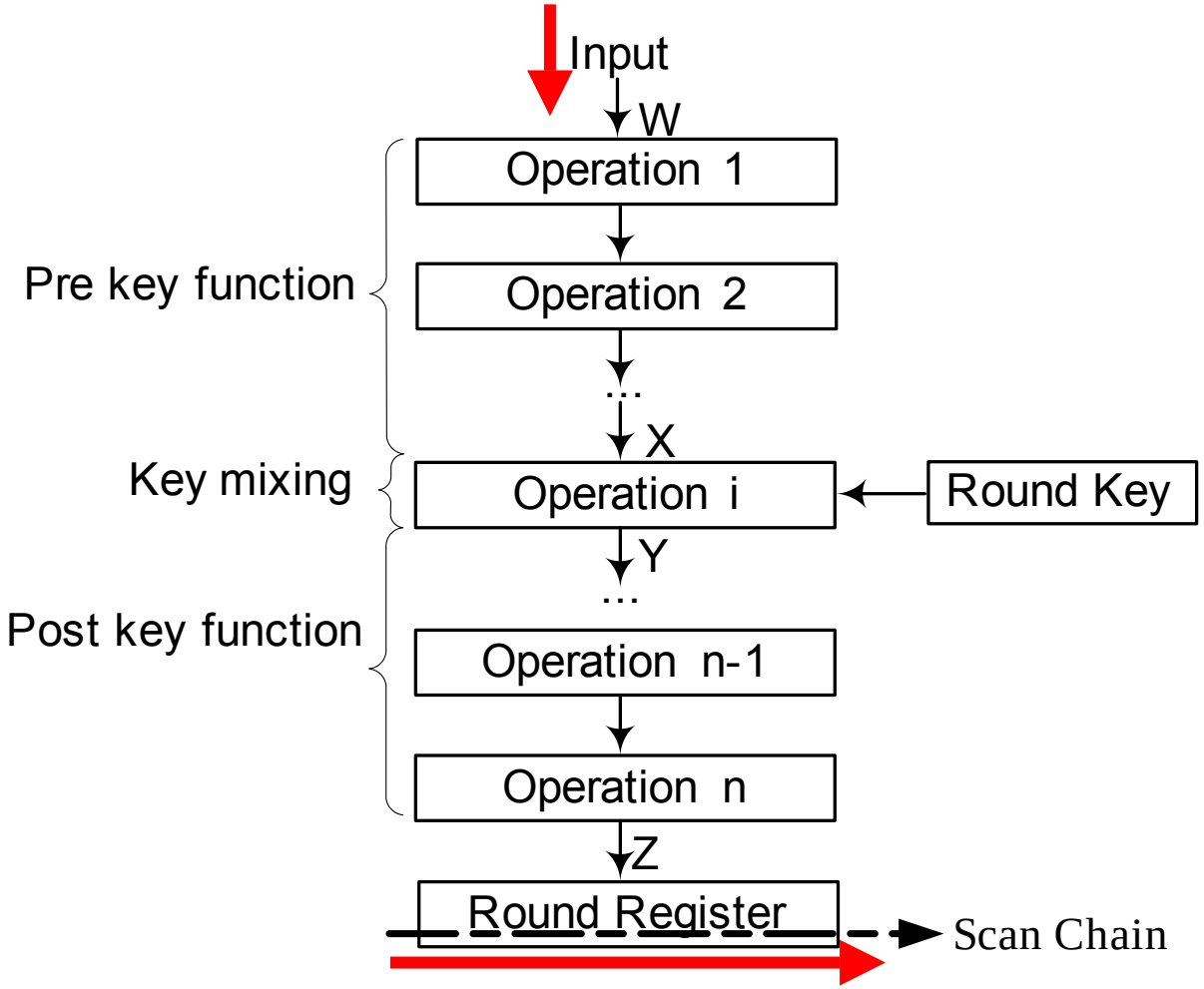
Cryptographic hardware

- More and more cryptographic algorithms have been implemented in Application Specific Integrated Circuit (ASIC) to provide high throughput.
- Any ASIC has to be tested after fabrication to validate its function.

Scan based test

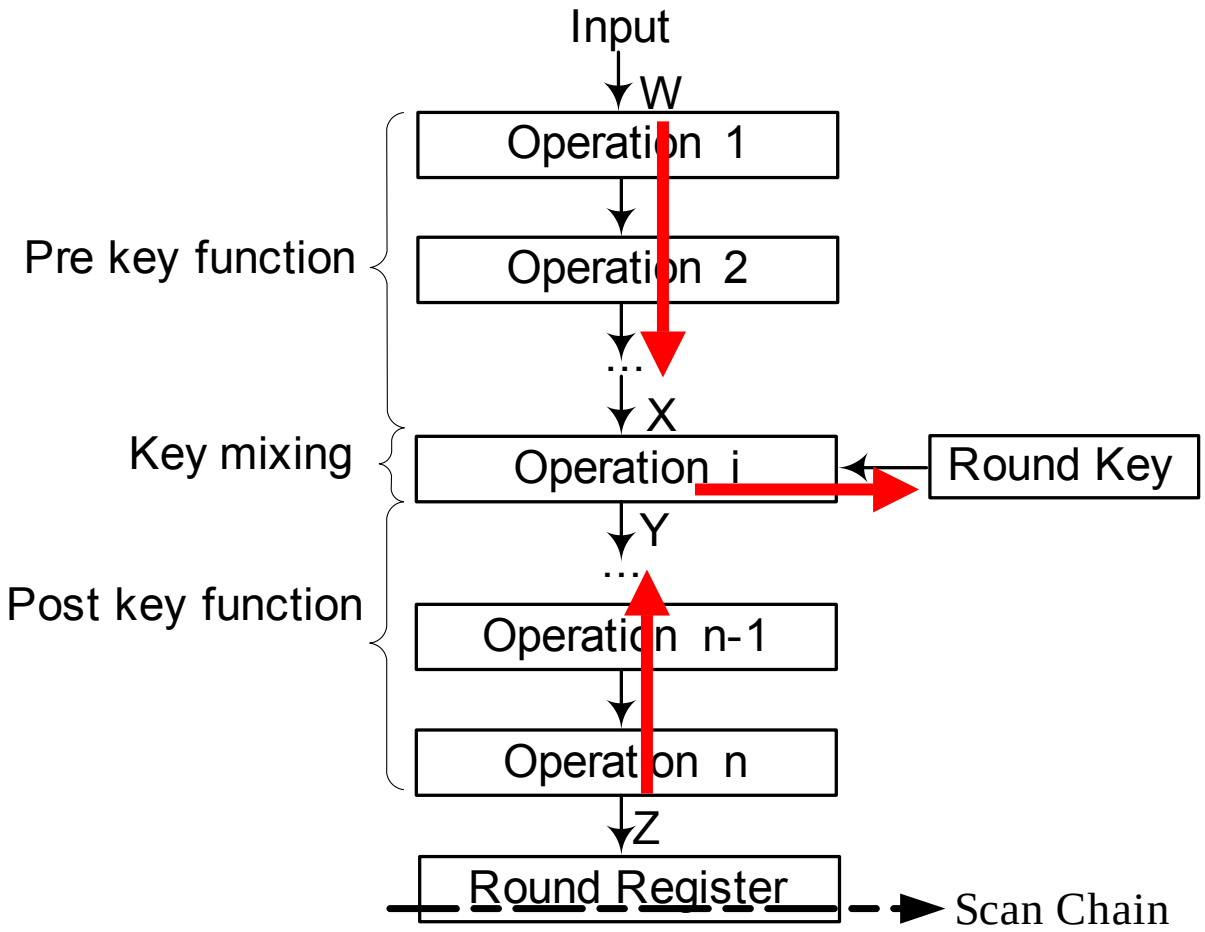


How to mount a scan based attack?



- X = Pre key function (W) where W is the input to this round
- Y = Key Mixing (X, Round key)
- Z = Post key function (Y) where Z is the round output
- What can we do?
- We can apply different inputs
- We can scan out the value in round register

How to mount a scan based attack?

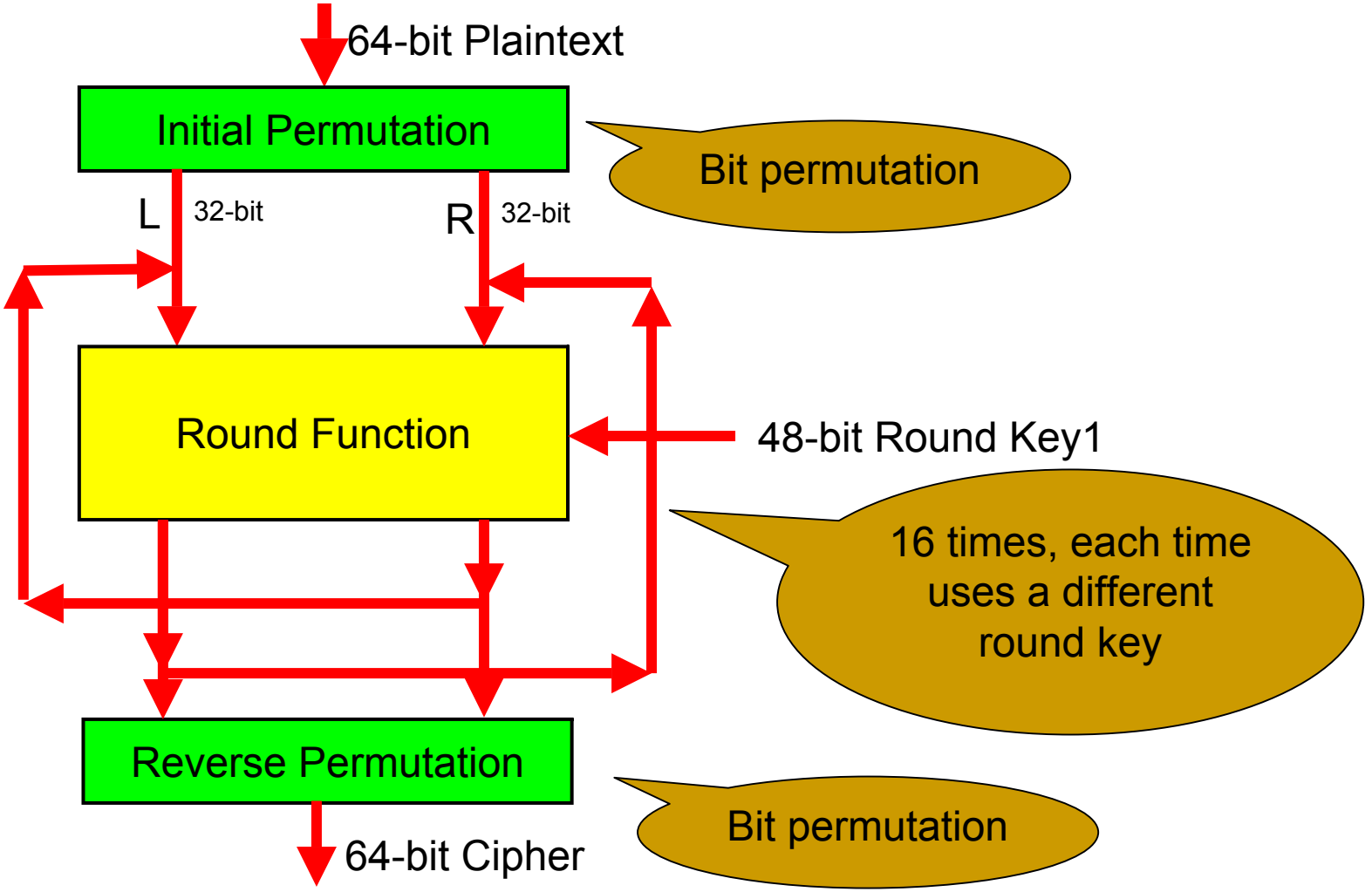


- Can we calculate X from W
- It is easy, because the algorithm is public
- Calculate Y from Z
- It is not easy, because the post key function may be not a bijective function
- Round key can be determined by solving Key mixing function

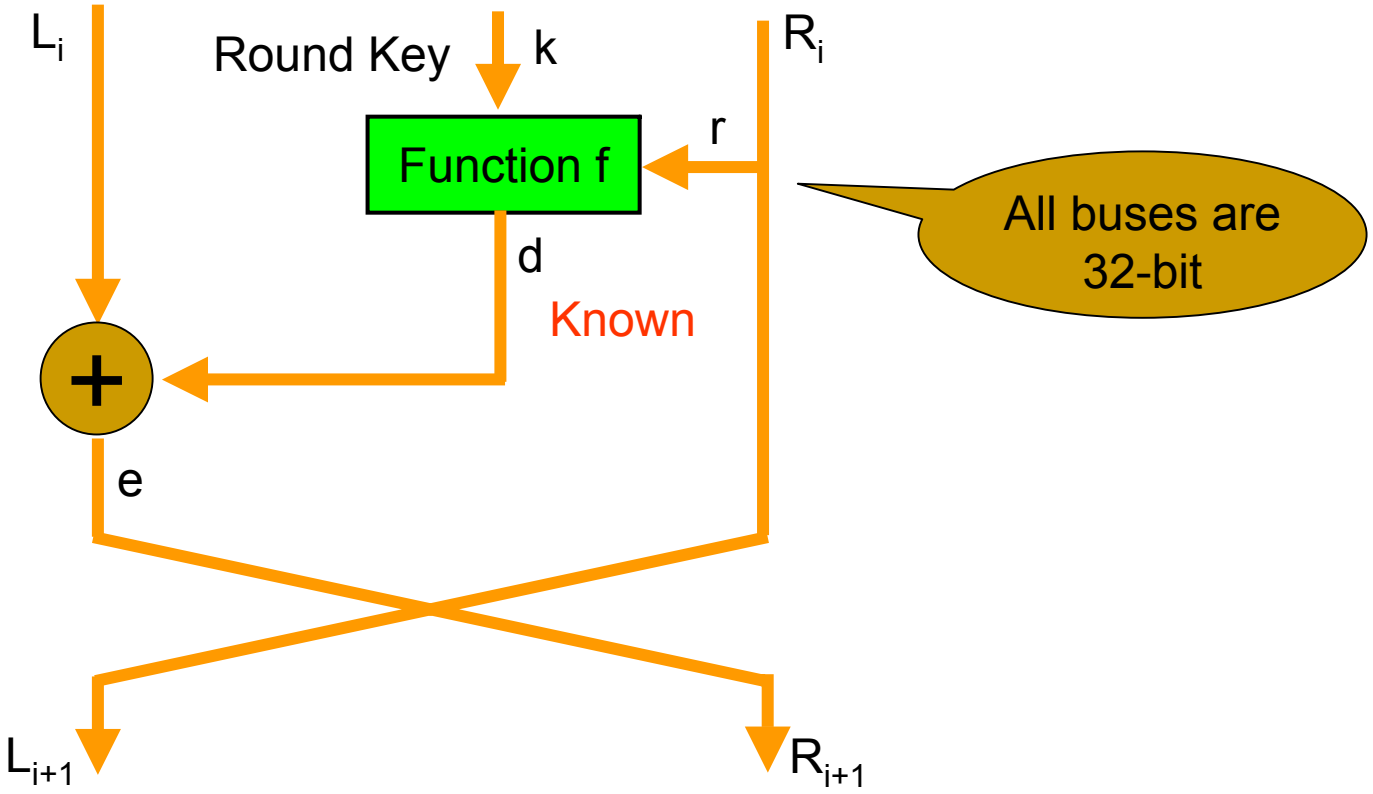
Data Encryption Standard

- The Data Encryption Standard (DES) is a symmetric encryption algorithm developed in the 1970s by IBM.
- DES encrypts 64-bit data blocks under the control of a 56-bit user key.
- DES decryption is the inverse of DES encryption and uses the same user key.
- Sixteen 48-bit round keys are generated from 56-bit user key by key schedule algorithm.

Encryption Algorithm

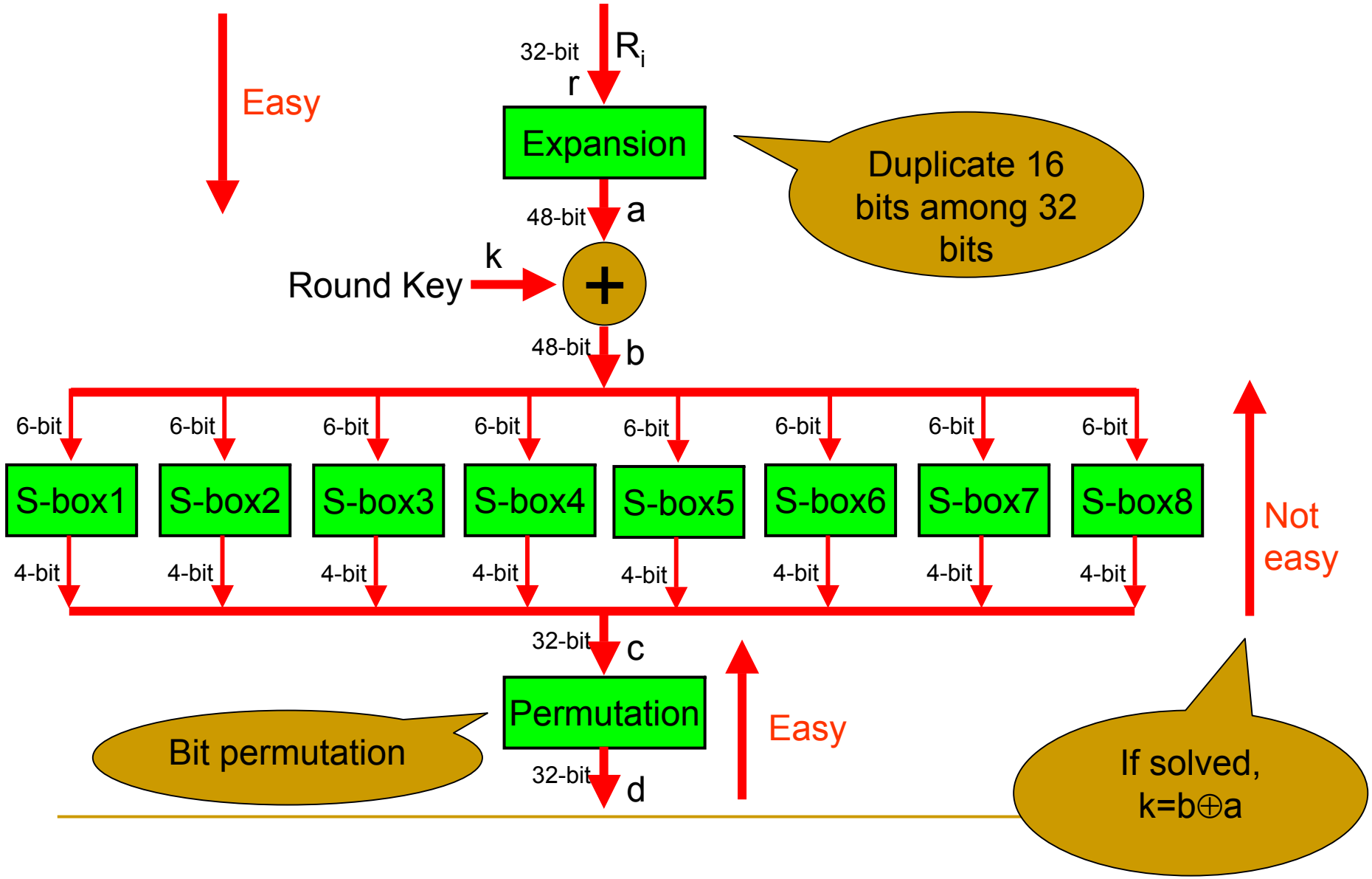


Round Function

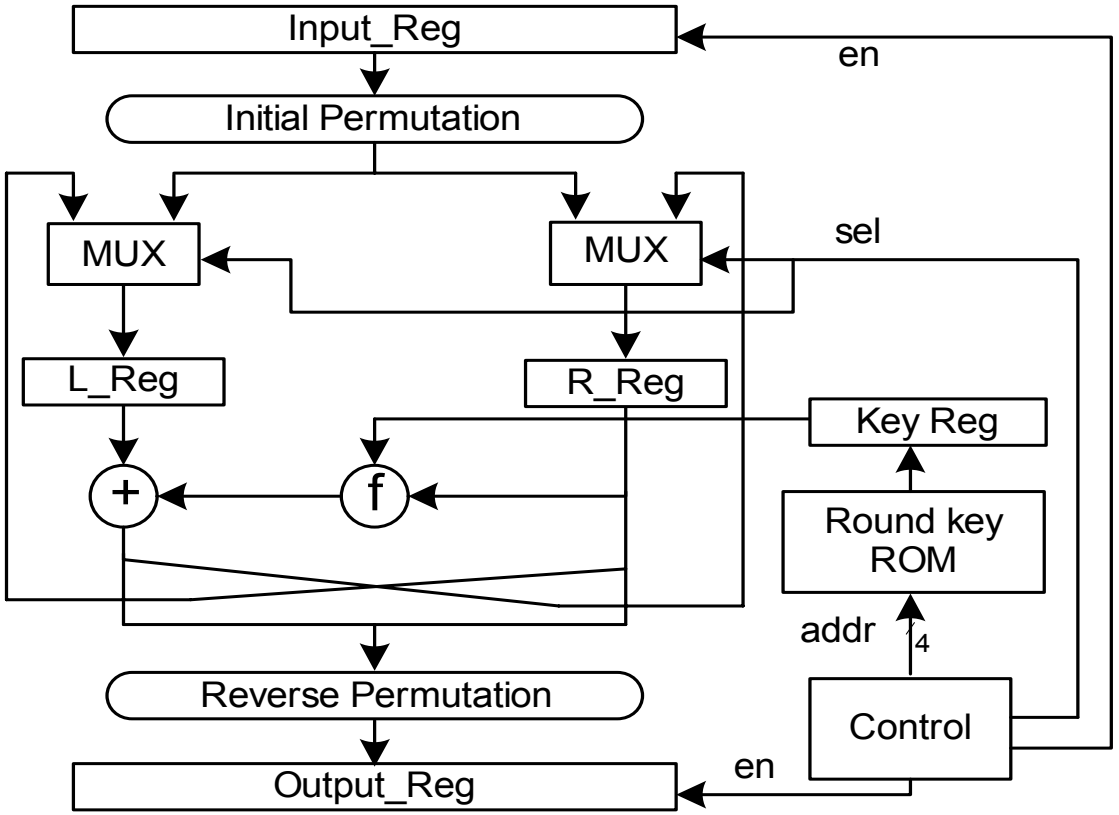


- If R_i, L_i and R_{i+1} are known, what will happen?
- If we can solve d=f(k,r), then k is retrieved.

Function f


Easy
32-bit
R_i
r
Expansion
Duplicate 16 bits among 32 bits
48-bit
a
k
Round Key
+
48-bit
b
6-bit
S-box1
S-box2
S-box3
S-box4
S-box5
S-box6
S-box7
S-box8
4-bit
Not easy
32-bit
c
Permutation
Bit permutation
Easy
32-bit
d
If solved, k=b⊕a

Iterative DES architecture



- All 16 rounds use the same hardware
- If the L and R Register can be scanned out, then L_i and R_i are known. Then K_i will be retrieved

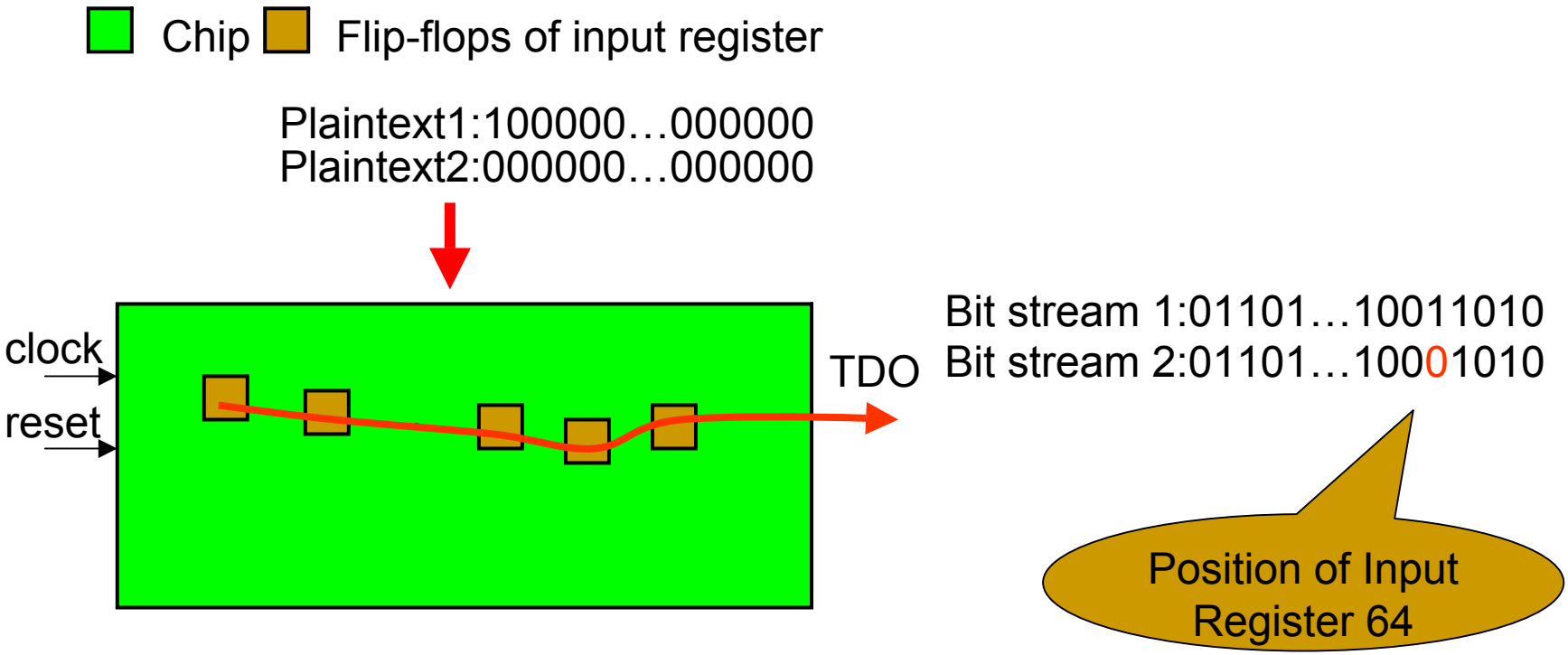
Two-step scan based attack

- The positions of flip-flops of L and R register should be determined in the scan chain. Then we can get the value of L and R register in the scanned out bit stream.
- Using L_0, R_0, L_1 and R_1 to discover Round Key1

Some Assumptions

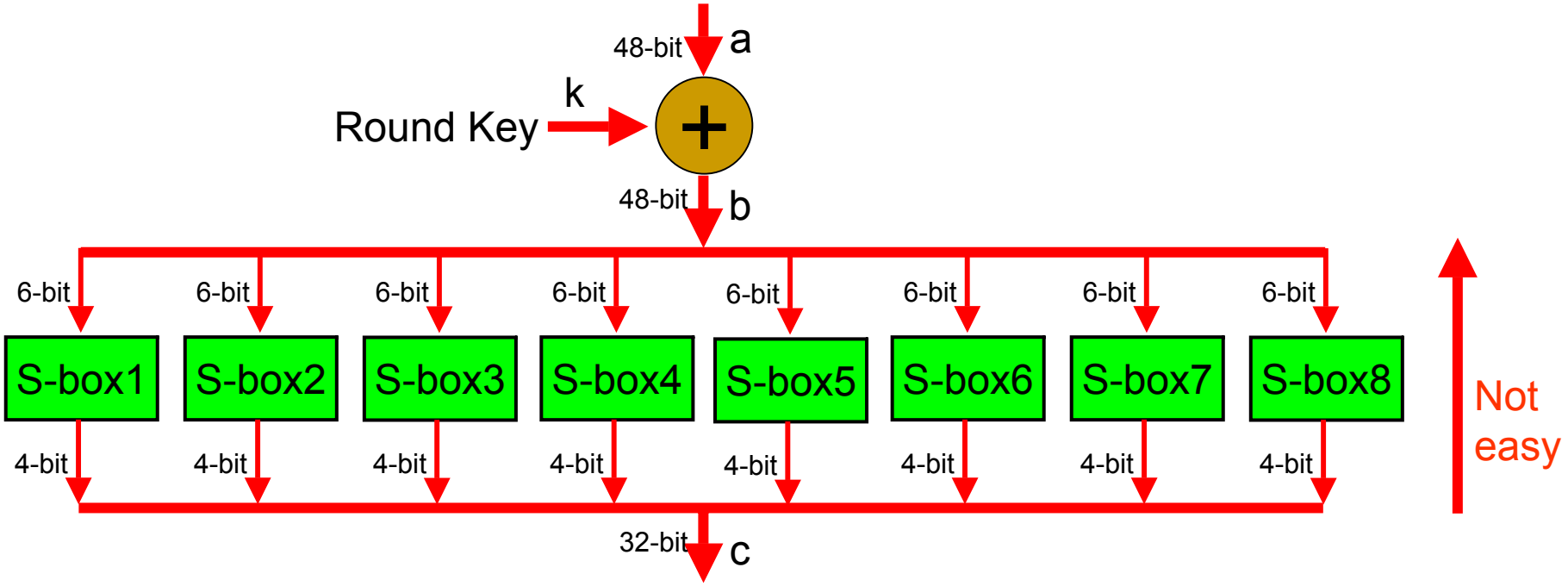
- The attack knows the algorithm (it is public)
- The attacker has access to high level timing diagrams provided by DES ASIC vendor
- Round keys are stored in a secure RAM/ROM
- The attacker has access to scan chains via the JTAG port
- Round key registers are not included in the scan chain; otherwise it will be easy to scan out the round key

Attack step 1: determine the scan chain structure



- Similarly, all the flip flops in Input register, L register and R register are determined.

Attack step 2: recover round key 1



- As we discussed, if we can figure out the input of S-box from the output of s-box, the round key can be recovered.
- Why is it not easy to determine?
- Each S-box compresses 6-bit input into 4-bit output, so it is not a bijective function.

Look into S-box structure: S1

Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

- Compresses 2^6 to 2^4. Each row has 16 different numbers ranging from 0 to 15
- Input is $b_{48\text{-}43}$, among which $b_{48}b_{43}$ is row address and $b_{47}b_{46}\,b_{45}b_{44}$ is column address
- For example, if $c_{32\text{-}29}$ is $(0100)_2$, $b_{48\text{-}43}$ can be either $(000010)_2$ or $(000111)_2$ or $(100000)_2$ or $(101001)_2$.

Apply the second input?

Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

- If $c_{32}c_{31}c_{30}c_{29}$ is $(0100)_2$, $b_{48}b_{47}b_{46}\,b_{45}b_{44}b_{43}$ can be either $(000010)_2$ or $(000111)_2$ or $(100000)_2$ or $(101001)_2$.
- Suppose we apply $b_{48}b_{47}b_{46}\,\overline{b_{45}}b_{44}b_{43}$
- the output will be 15 if $b_{48}b_{47}b_{46}\,b_{45}b_{44}b_{43}$ is $(000111)_2$
- the output will be 14 if $b_{48}b_{47}b_{46}\,b_{45}b_{44}b_{43}$ is $(100000)_2$
- the output will be 1 if $b_{48}b_{47}b_{46}\,b_{45}b_{44}b_{43}$ is $(000010)_2$ or $(101001)_2$
- We still can not determine the input according to the output

Apply three inputs

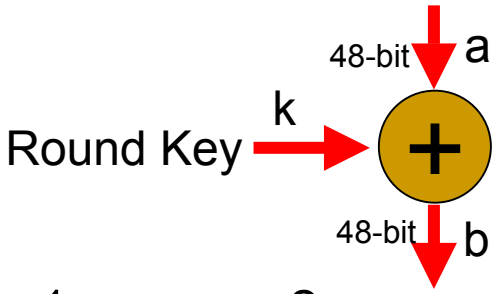
Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

We apply input $b_{48}b_{47}b_{46}\,b_{45}b_{44}b_{43}$, then $b_{48}b_{47}b_{46}\,\overline{b_{45}}b_{44}b_{43}$ and finally $b_{48}b_{47}b_{46}\,b_{45}b_{44}\overline{b_{43}}$

- If the output sequence is 4→15 →1, $b_{48}b_{47}b_{46}\,b_{45}b_{44}b_{43}$ is $(000111)_2$
- If the output sequence is 4→14 →15, $b_{48}b_{47}b_{46}\,b_{45}b_{44}b_{43}$ is$(100000)_2$
- If the output sequence is 4→1 →15, $b_{48}b_{47}b_{46}\,b_{45}b_{44}b_{43}$ is $(000010)_2$
- If the output sequence is 4→1 →13, $b_{48}b_{47}b_{46}\,b_{45}b_{44}b_{43}$ is $(101001)_2$
- Input is determined

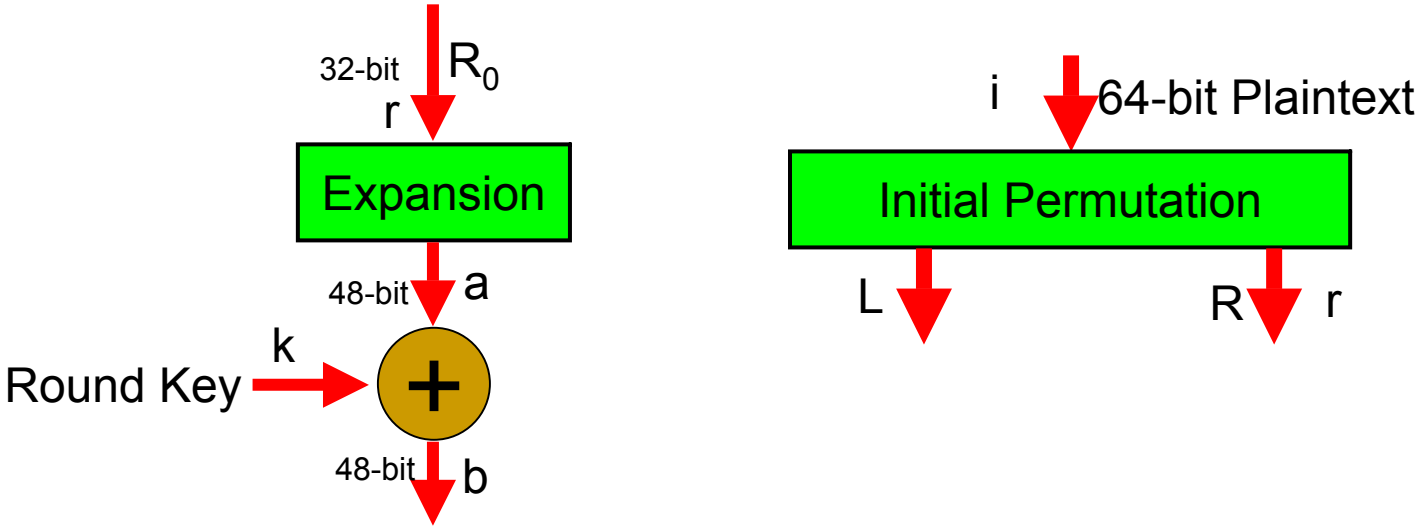
How to apply plaintexts?

- If we apply the three inputs:
  - $b_{48}b_{47}b_{46}\,\overline{b_{45}}b_{44}b_{43}$
  - $b_{48}b_{47}b_{46}\,\overline{b_{45}}b_{44}\overline{b_{43}}$
  - $b_{48}b_{47}b_{46}\,b_{45}b_{44}\overline{b_{43}}$



- According to the sequence of $c^1_{32\text{-}29}$, $c^2_{32\text{-}29}$ and $c^3_{32\text{-}29}$, we can determine $b_{48}b_{47}b_{46}\,b_{45}b_{44}b_{43}$. Then $k_{48}k_{47}k_{46}\,k_{45}k_{44}k_{43}$ can be calculated.
- Now the problem is: Can we apply three plaintexts to generate required input sequence of S1?

Trace back to plaintext



- Since b=a⊕k and k is unchanged, $\overline{a}_{45} \rightarrow \overline{b}_{45}$
- According to Expansion, $\overline{r}_{30} \rightarrow \overline{a}_{45}$
- According to Initial Permutation, $\overline{i}_{24} \rightarrow \overline{r}_{30}$
- Similarly, $\overline{i}_{40} \rightarrow \overline{r}_{43}$
- So we can control plaintext to apply required inputs to S1

Summary of attack step 2

- We can random pick up a plaintext i^1
- Switch its 24^{th} bit as i^2
- Switch its 40^{th} bit as i^3
- Calculate $c^1_{32\text{-}29}$, $c^2_{32\text{-}29}$ and $c^3_{32\text{-}29}$ from R_1 and L_1.
- Determine $b_{48}b_{47}b_{46}\,b_{45}b_{44}b_{43}$ from $c^1_{32\text{-}29}$, $c^2_{32\text{-}29}$ and $c^3_{32\text{-}29}$
- Calculate $a_{48}a_{47}a_{46}\,a_{45}a_{44}a_{43}$ from R_0
- Calculate $k_{48}k_{47}k_{46}\,k_{45}k_{44}k_{43}$ from $a_{48}a_{47}a_{46}\,a_{45}a_{44}a_{43}$ and $b_{48}b_{47}b_{46}\,b_{45}b_{44}b_{43}$
- Attend this method to other S-box, we can recover Round Key 1

Totally using 3 plaintext in attack step 2

- If we discover the Round Key 1 by attack S-box one by one, we need 24 plaintexts.
- By exhaustively simulating, we find we can use only 3 plaintexts that work for all 8 S-boxes simultaneously.
- For example:
  - i^1: $(0000000000000000)_{16}$
  - i^2: $(0000550000005500)_{16}$
  - i^3: $(5500400110000401)_{16}$.

Discover user key

- Similarly, we can discover Round Key 2 and Round Key 3.
- From these three round key, we can discover the user key by key schedule algorithm.

Conclusions

- We develop a two step attack to DES hardware implementations that use scan test
- First, we determine the positions of flip flops in the round register in the scan chain.
- By using the temporary round results, we can discover the corresponding round key.