Design of Efficient Cryptographically Robust Substitution Boxes

---*Search for an Efficient Secured Architecture*

Debdeep Mukhopadhyay, Assistant Professor
Dept of Computer Sc and Engg, IIT Madras

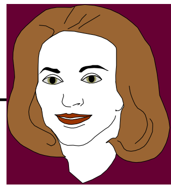
Outline of the Presentation

- What is an S-Box?
- Motivation to design S-Boxes
- Cellular Automata: A Finite State Machine
- Construction of an S Box
- Implementation of the proposed construction

Crypto

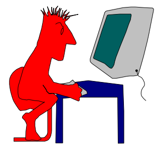
- **Cryptology** — The art and science of making and breaking "secret codes"
- **Cryptography** — making "secret codes"
- **Cryptanalysis** — breaking "secret codes"
- **Crypto** — all of the above (and more)

Goals of a Cryptosystem



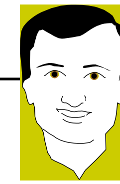
Alice

COMUNICATION CHANNEL



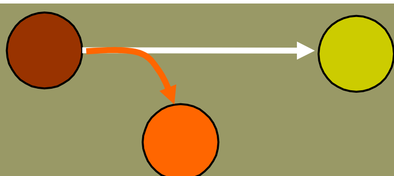
Mallory

Policy
- Confidentiality
- Integrity
- Authenticity

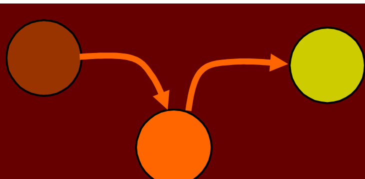


Bob

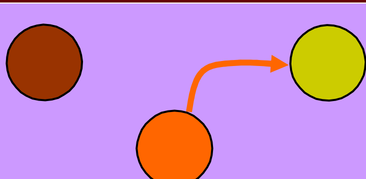
Security Attacks



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



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



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

Types of ciphers

- Symmetric Key Crypto:
  - Bob and Alice share the same key.
- Assymetric Key Crypto:
  - Alice encrypts with a public key
  - Bob decrypts with a secret key (private key)

Types of symmetric key algorithms

- Block Ciphers: Manipulates blocks of data. Say 128 bits at a time.
- Stream Ciphers: Manipulates streams of data, typically one bit at a time.
- We, shall be concentrating on

BLOCK CIPHERS…

Substitution and Transposition

- Substitution example
  - A B C D E F G …
  - C D E F G H I …
- Transposition example
  - HERE_IS_A_MESSAGE

    H E S _ S G

    E _ _ M S E

    R I A E A _

Simple Substitution

- Plaintext: fourscoreandsevenyearsago
- Key:

Plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

- Ciphertext:
  IRXUVFRUHDAGVHYHABHDUVDIR
- Shift by 3 is "Caesar's cipher"

Block Ciphers

(Iterated) Block Cipher

- Plaintext and ciphertext consists of fixed sized blocks
- Ciphertext obtained from plaintext by iterating a **round function**
- Input to round function consists of key and the output of previous round
- Usually implementation friendly. Gives a high throughput.

Feistel Cipher

- **Feistel cipher** refers to a type of block cipher design, not a specific cipher
- Split plaintext block into left and right halves: Plaintext = (L_0, R_0)
- For each round $i=1,2,\ldots,n$, compute

  $L_i = R_{i-1}$

  $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$

  where F is **round function** and K_i is **subkey**
- Ciphertext = (L_n, R_n)

Feistel Cipher

- Decryption: Ciphertext = (L_n, R_n)
- For each round $i = n, n-1, \ldots, 1$, compute

  $R_{i-1} = L_i$

  $L_{i-1} = R_i \oplus F(R_{i-1}, K_i)$

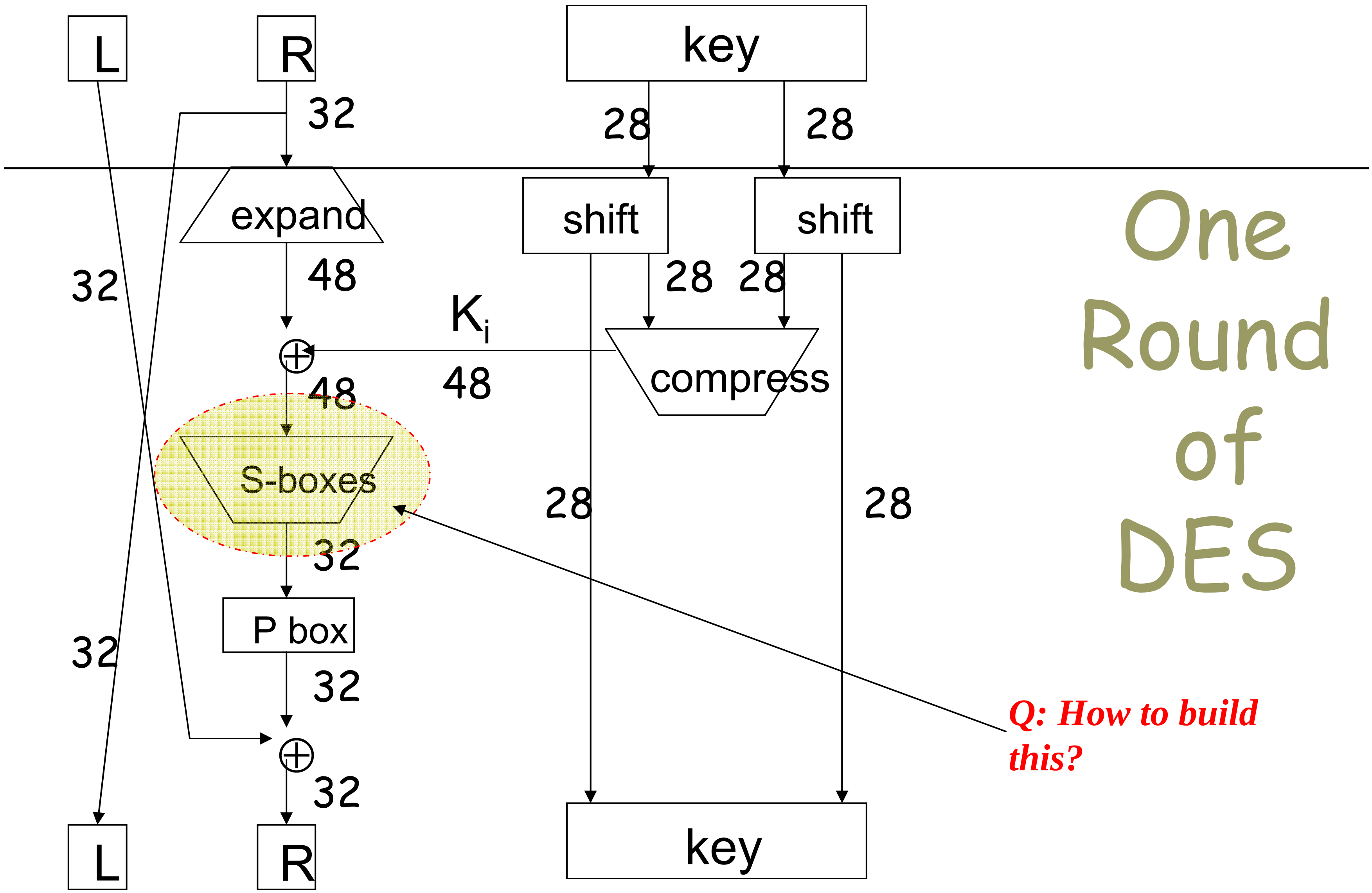
  where F is round function and K_i is subkey
- Plaintext = (L_0, R_0)
- Formula "works" for any function F
- But only secure for certain functions F

Data Encryption Standard

- DES developed in 1970's
- Based on IBM Lucifer cipher
- U.S. government standard
- DES development was controversial
  - NSA was secretly involved
  - Design process not open
  - Key length was reduced
  - Subtle changes to Lucifer algorithm

DES Numerology

- DES is a Feistel 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



One Round of DES

Q: How to build this?

DES S-box

- 8 “substitution boxes” or S-boxes
- Each S-box maps 6 bits to 4 bits
- S-box number 1

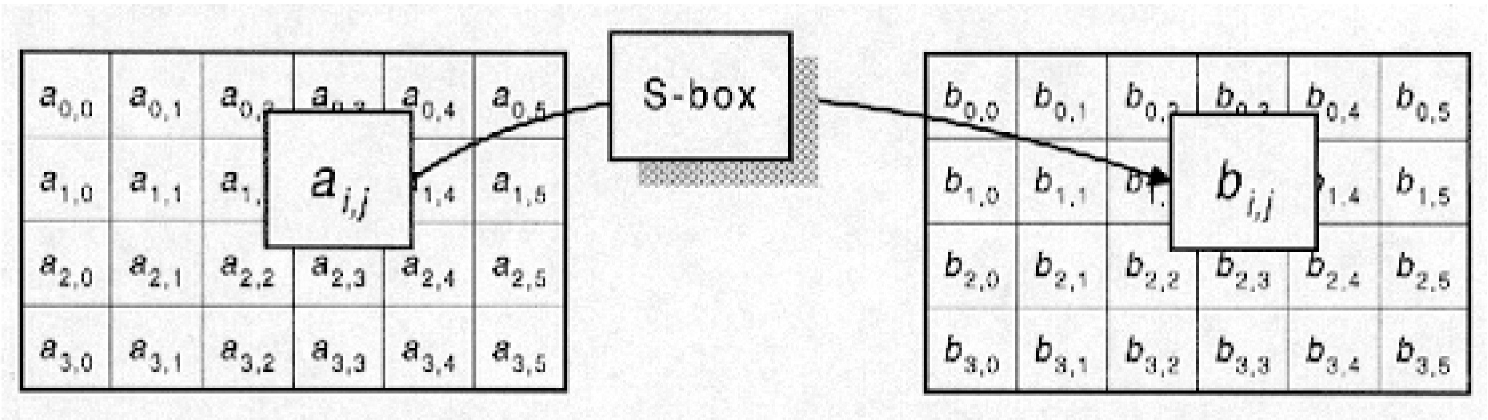
input bits (0,5)
↓
input bits (1,2,3,4)

```
   | 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
------------------------------------------------------------------------------------
00 | 1110 0100 1101 0001 0010 1111 1011 1000 0011 1010 0110 1100 0101 1001 0000 0111
01 | 0000 1111 0111 0100 1110 0010 1101 0001 1010 0110 1100 1011 1001 0101 0011 1000
10 | 0100 0001 1110 1000 1101 0110 0010 1011 1111 1100 1001 0111 0011 1010 0101 0000
11 | 1111 1100 1000 0010 0100 1001 0001 0111 0101 1011 0011 1110 1010 0000 0110 1101
```

What is the design principle?

AES Substitution

- Assume 192 bit block, 4x6 bytes



- ByteSub is AES's "S-box"
- Can be viewed as nonlinear (but invertible) composition of some math operations.
- ***What is the logic behind the construction? What is it based on?***

Design Issues and Modern Challenges

- We require large boolean functions : Typically operating on say 32 bits.
- Area required to implement
  - A Boolean function with n inputs –

    Exponential in n

- More complex if we require to generate more than one output simultaneously

Cryptographic Properties of boolean functions

- Balancedness
- Satisfy Strict Avalanche Criterion (SAC)
- High non-linearity
- High algebraic degree
  - ***Not only the component functions but also their linear combinations should have crypto merit.***
- Robustness against linear and differential attacks

Balancedness

- The truth-table of the boolean function has an equal number of 0's and 1's.
- XOR is a balanced function.
- AND is an unbalanced function.
- So, we prefer XOR…

Non-linearity

- What is a linear function?
- f is said to be linear wrt + if
  - f(x+y)=f(x)+f(y)

$$x = (x_1, x_2), y = (y_1, y_2), x \oplus y = ((x_1 \oplus x_2), (y_1 \oplus y_2))$$

$$Define,\ f(x) = x_1 \oplus x_2.$$

$$\begin{aligned} \therefore\ f(x \oplus y) &= f(x_1 \oplus x_2, y_1 \oplus y_2) \\ &= x_1 \oplus x_2 \oplus y_1 \oplus y_2 \\ &= f(x) \oplus f(y) \end{aligned}$$

So, XOR is a linear function. But we want non-linear functions.
So, we don't want XOR!

Computing Non-linearity.

x1	x2	x1x2	0	x1	x2	x1^x2
0	0	0	0	0	0	0
0	1	0	0	0	1	1
1	0	0	0	1	0	1
1	1	1	0	1	1	0

Non-linearity is the minimum distance from the truth tables of the linear equations.
Here it is 1. So, non-linearity of AND is 1.

We present a technique to generate such S Boxes…

…efficiently

Cellular Automata (CA)- A Quick Glance

- Mathematical model for self-organizing statistical systems
- Discrete lattice of cells (0 or 1)
- Cells evolve according to a rule depending on local neighbours
- We shall employ 3 neighbourhood structure:
  - **$q_i(t+1) = f(q_{i-1}(t), q_i(t), q_{i+1}(t))$, where f is a boolean function**
  - **We shall restrict f to be composed of only xor gates: Linear Cellular Automata**

Cellular Automata - Rules

Rule 150

$q = l \oplus s \oplus r$

l	*s*	*r*	*q*
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

150

Rule 90

$q = l \oplus r$

l	*s*	*r*	*q*
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

90

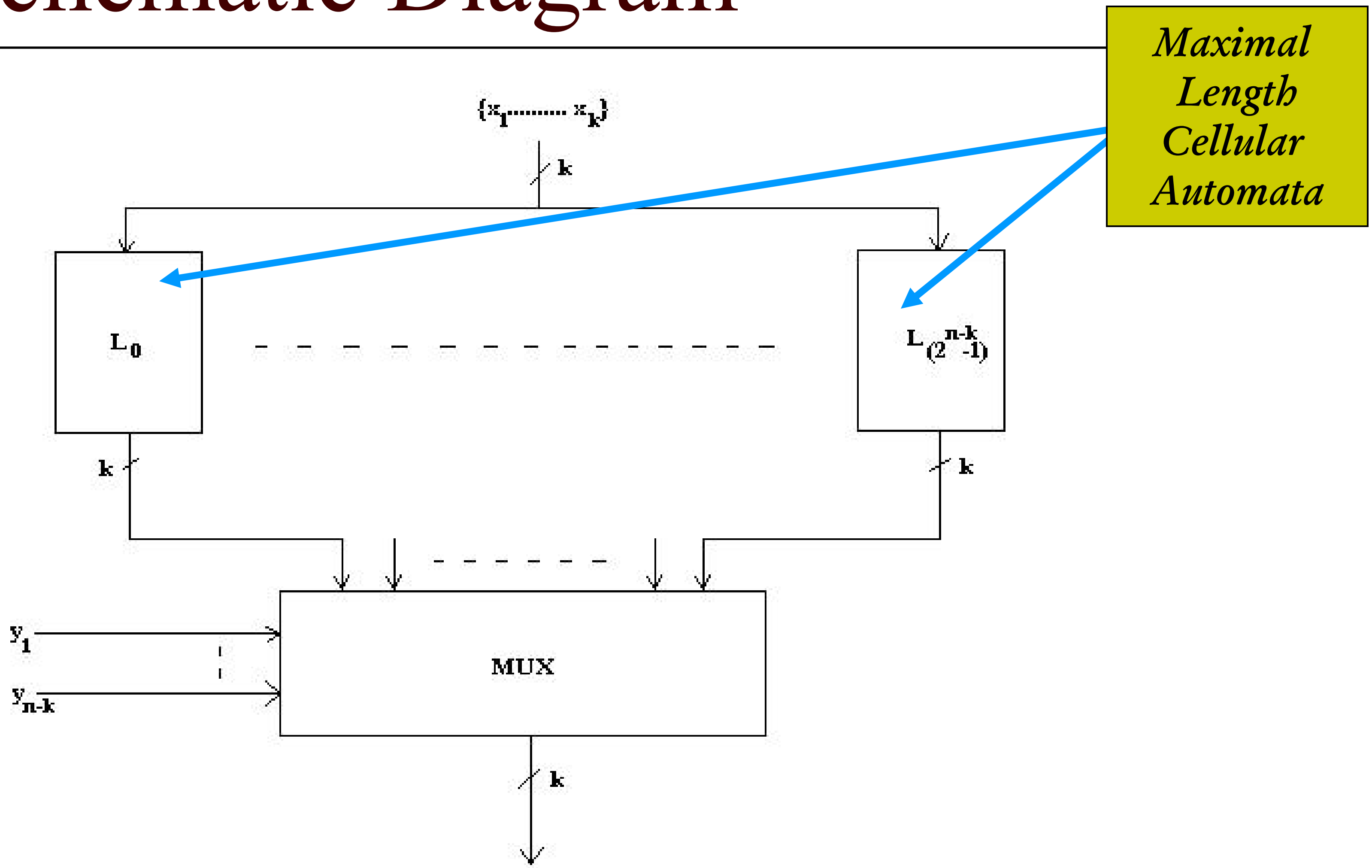
Evolution of Cellular Automata (CA)

- For a k-cell CA, Y = T (X) where
  - X = k-bit input to the CA
  - Y = k-bit output of the CA
  - T = characteristic matrix (k x k) of the CA
- Evolution goes like X, T (X), T^2 (X),……., T^{2^k-2} (X)
- A Group CA is one that forms cyclic group i.e. simply a cycle of length *l:*
  - $T^l(X)=X$
  - For group CA, $|T| = 1$
- Maximal length Group CA: All the non-zero states lie in a cyclic additive group
  - T^{2^k-1} (X) = X and so on….

Construction of $n \times k$ S-Boxes

- The n-bit input is split into two portions:
  - x of size k bits
  - y of size n-k bits
- $2^{(n-k)}$ k cell maximum length CA are used
  - Each CA transforms operates on x
  - Converts the k-bit input to a k-bit output
- Input, z = (y, x)
- Output, $Q(z) = \{ q_1(z), \ldots\ldots , q_k(z) \}$

A Schematic Diagram



Why $k > n/2$?

- Total distinct CA transformations available $= 2^k - 1$ (cycle length of a maximal length CA)
- Total CA required in the construction $= 2^{(n-k)}$
- Hence,

$$2^k - 1 > 2^{(n-k)}$$

$$\leftrightarrow 2^k > 2^{(n-k)}$$

$$\leftrightarrow k > n-k$$

$$\leftrightarrow k > n/2$$

Set of CA Transformations

- If characteristic matrix of the CA is Tk (k X k),
  - Set of transformations, S
    - { I, Tk, ... , Tk^{2k-2}}
- $Tk^{2k-1} = I$
- Properties of set S:
  1. All the transformations in the set S are distinct
  2. The set S is closed under addition modulo 2
  3. All the matrices are invertible
  4. The rows of any 2 elements in set S are pairwise distinct (follows from 2 and 3)

Mathematical Formulation

- Linear transformations can be represented as kxk matrices:

$$Li = \begin{pmatrix} li1 \\ \ldots \\ lik \end{pmatrix}, 0 \le i \le 2^{n-k} - 1$$

- Mathematically, the output k-bit vector Q(z) is

$$Q(z) = \oplus_{\sigma=0}^{2^{n-k}-1} D_\sigma(y) L_\sigma(x)$$
$$D_\sigma(y) = (i_1 \oplus y_1)(i_2 \oplus y_2)\ldots(i_{n-k} \oplus y_{n-k}),$$
$$\sigma = (i_1 i_2 \ldots i_{n-k}), y = (y_1 y_2 \ldots y_{n-k})$$

Cryptographic Properties

- For each component function $q_i(z)$
  - Non – linearity is at least $2^{n-1} - 2^{k-1}$, $k>n/2$
  - It is balanced
    - ***Same is true for any non-zero linear combinations***
  - Algebraic degree is $(n-k+1)$
  - Mapping $Q(z) = \{ q_1(z), \ldots\ldots, q_k(z) \}$ is regular from V_n to V_k
- Number of mappings generated is $\boxed{P_{2^{n-k}}^{2^k-1}}$

Strict Avalanche Criterion

- Boolean function f on Vn satisfies SAC iff

  $f(x) \oplus f(x \oplus \alpha)$ is balanced for all $\alpha \in Vn$
- Original construction Q(z) does not satisfy SAC
- For z' = Wz,
  - Q(Wz) satisfies SAC
  - W is a non-degenerate n x n matrix with entries from GF(2)

$$W = \begin{pmatrix} I_{n-k} & 0 \\ D_{kXn-k} & I_k \end{pmatrix}; D = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

VLSI Design of the Architecture

- Input y denotes the CA to be selected
  - NB: All the CA are the same machine in different states of evolution (the clock cycles are different)
  - y determines the number of cycles, s, the CA is to be applied
  - A mapping, *g*, from y to s is required=> $Q(z)=T^{g(y)}(x)$
    - (Alternate expression of the construction)
- Domain of g is V_{n-k}, while range is Vk
- One to many mapping (as, k>n/2)
  - *No deterministic hardware possible*

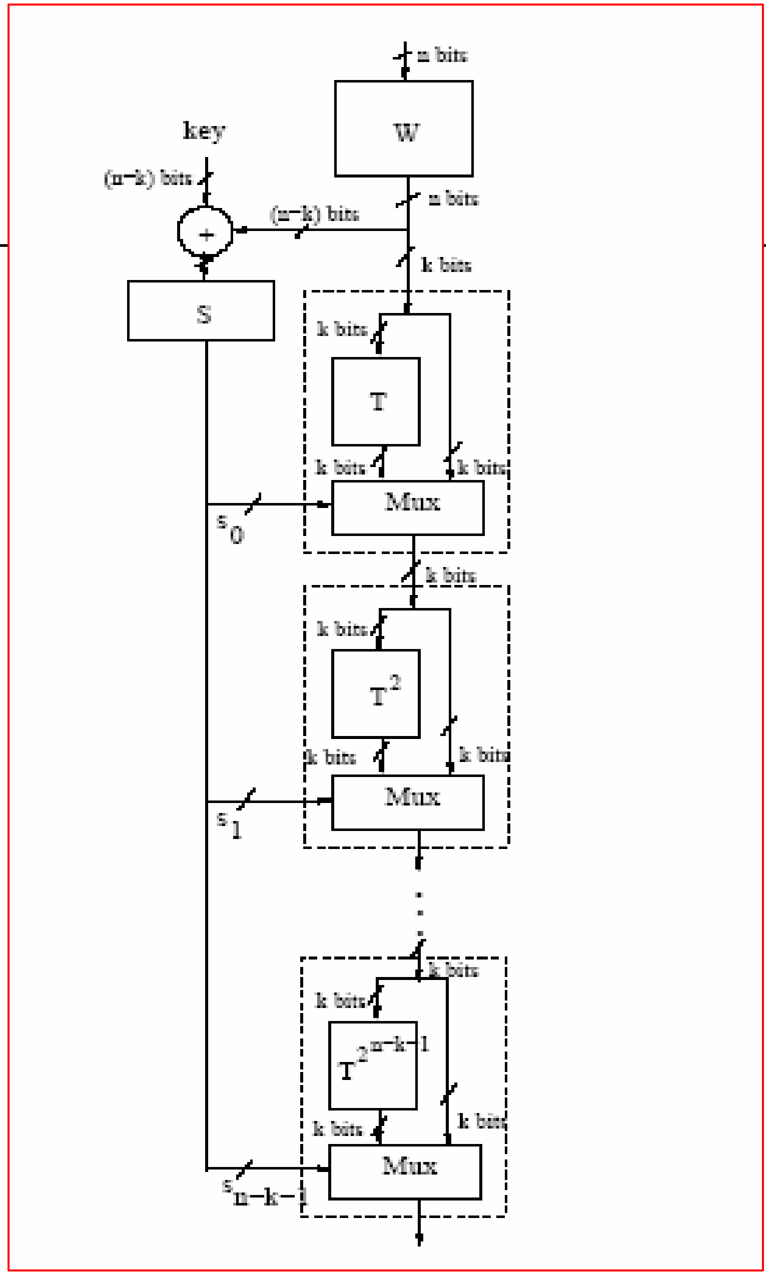
Restricted Design Architecture

- Restrict the clock cycles to $2^{(n-k)}$
- Mapping becomes (n-k) to (n-k)
- Permutation is done by using XORing with a secret k, s
- Value of s for a given y, will depend on the secret key, *key* of n-k bits
- Number of possible permutations 2^{n-k}
- Cryptographic properties remain the same, as this is an equivalent representation.

Restricted Design Architecture

- Each CA is to be cycled s times i.e. T needs to be multiplied s times
- Square and multiply algorithm is used for better performance
- Output is obtained in O(n-k) time

Block Diagram



Hardware Complexity

- (n-k) flip-flops
- $O(n^2)$ 2 input XOR gates.
- 2 to 1 MUXes : k(n-k)
- Time Complexity : O(n-k)

Example : 8x5 mapping

- **n=8, k>4=5**
- **Choose a 5 cell maximal length CA with rule set {150, 150, 90, 90, 150}.**

$$T = \begin{matrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{matrix}$$

Compute Q(156), assume key=0

$$W = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, Z = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

z=156

$$Z' = WZ = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

y=4

x=3

$$Q(156) = T^4(3) = ((T^2)^2)(3)$$

$$= \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Q(156)=192

Cryptographic Properties

- Non-linearity is 112 which is very high (maximum for 8 variables 120)
- Degree of each function is 4
- All non-zero combinations are balanced and have non-linearity of 112.
- Robustness against Differential Cryptanalysis is 0.848, bias in the Linear Approximation Table is 16.
- Each boolean function satisfies SAC

Experimental Results

Dimension	XOR	MUX	Flip-Flop	Time (clk cycles)
8 x 5	26	15	3	3
10 x 6	54	24	4	4
16 x 9	208	63	7	7
24 x 13	691	141	11	11

Observation: Growth of the resources is polynomial with dimension

Some Key References

- Systematic Generation of cryptographically robust S Boxes, Jennifer Seberry, Xian Zhang, Yuliang Zheng, 1st conference on Computer and Comm Security, USA, 93.
- Perfect Non linear S Boxes, Kaisa Nyberg, 1998, Springer Verlag.

Small and compact designs survive…

Thank You
Questions?