Ligthweight Cryptography

Marc Beunardeau

April 28, 2015

Table of contents

Itroduction

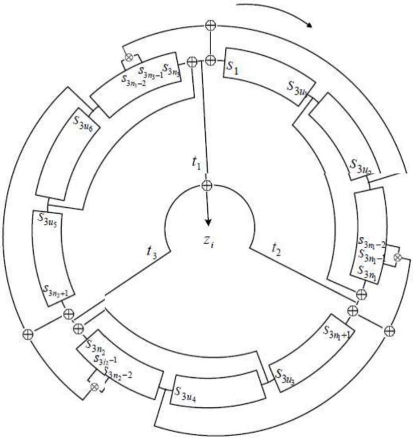
- Developpement of tiny devices (RFID,wireless sensors....)
- Need for new algorithms ($\neq$ AES)
- Pervasive environement (invasive attacks)

- Clock Cycles per encryption
- Memory
- Security
- Consumption

Generalities

- Introduced by Cannire and Preneel in 2005
- Stream cipher
- 1100 cycles for initialisation
- 1 cycle per bits
- 2 faults attack
- optimized for hardware

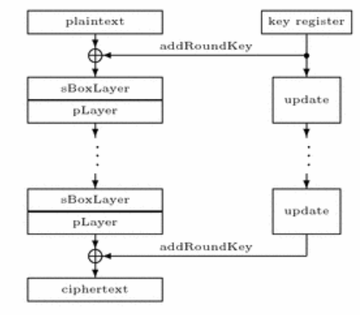
Structure


s_1
s_{3u_1}
s_{3u_2}
s_{3n_1-2}
s_{3n_1-1}
s_{3n_1}
s_{3n_1+1}
s_{3u_3}
s_{3u_4}
s_{3n_2}
s_{3n_2-2}
s_{3n_2+1}
s_{3u_5}
s_{3u_6}
s_{3n_3-2}
s_{3n_3-1}
s_{3n_3}
t_1
t_2
t_3
z_i

Generalities

- Bogdanov & Al in 2007
- SP-network
- 32 rounds
- 80, 128 bits keys, 64 bits block
- 32 cycles per block (hardware implementation)
- Cube attack : 2^{15} chosen plain text, 2^{32} encryption
- optimized for hardware

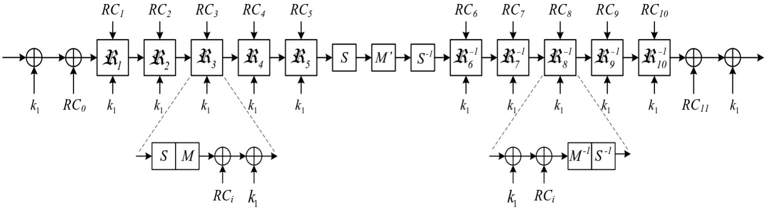
Structure


plaintext
key register
addRoundKey
sBoxLayer
pLayer
update
sBoxLayer
pLayer
update
addRoundKey
ciphertext

Generalities

- Introduced by
- SP-network
- Low latency
- Small aera when fully unrolled
- 128 bits key, 64 bits block
- 1 cycle per block (unrolled hardware implementation)
- α - reflection : $Dec_{(k_0||k_0'||k_1)}(.) = Enc_{(k_0'||k_0||k_1 \oplus \alpha)}(.)$
- 3-4 faults attack

Structure


RC_1
RC_2
RC_3
RC_4
RC_5
RC_6
RC_7
RC_8
RC_9
RC_{10}
$\mathfrak{R}_1$
$\mathfrak{R}_2$
$\mathfrak{R}_3$
$\mathfrak{R}_4$
$\mathfrak{R}_5$
S
M'
S^{-1}
$\mathfrak{R}_6^{-1}$
$\mathfrak{R}_7^{-1}$
$\mathfrak{R}_8^{-1}$
$\mathfrak{R}_9^{-1}$
$\mathfrak{R}_{10}^{-1}$
k_1
RC_0
RC_{11}
S
M
RC_i
M^{-1}
S^{-1}

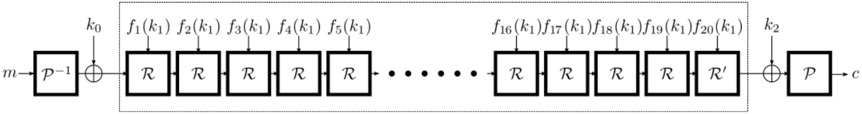
Generalities

- Introduced by Albrecht & Al in 2014
- SPN block cipher with focus on linear layer
- 64 bits blocks
- 128bits key
- 20 rounds

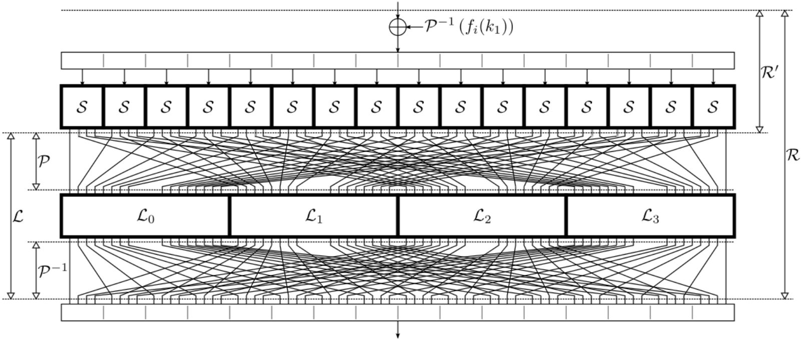
Performances

- 68 cycles per block
- 138 bytes of flash memory (943 flash + 33 S-RAM bytes, and 575 cycles for AES)

Structure


m
$\mathcal{P}^{-1}$
k_0
$f_1(k_1)$ $f_2(k_1)$ $f_3(k_1)$ $f_4(k_1)$ $f_5(k_1)$
$f_{16}(k_1)$ $f_{17}(k_1)$ $f_{18}(k_1)$ $f_{19}(k_1)$ $f_{20}(k_1)$
$\mathcal{R}$ $\mathcal{R}$ $\mathcal{R}$ $\mathcal{R}$ $\mathcal{R}$
$\mathcal{R}$ $\mathcal{R}$ $\mathcal{R}$ $\mathcal{R}$ $\mathcal{R}'$
k_2
$\mathcal{P}$
c

A Round of PRIDE



Key Scheduling

- $k = k_0 || k_1$
- $k_1 = k_{1_0} || k_{1_1} || k_{1_2} || k_{1_3} || k_{1_4} || k_{1_5} || k_{1_6} || k_{1_7}$
- $f_i(k_1) =$
  $k_{1_0} || g_i^{(0)}(k_{1_1}) || k_{1_2} || g_i^{(1)}(k_{1_3}) || k_{1_4} || g_i^{(2)}(k_{1_5}) || k_{1_6} || g_i^{(3)}(k_{1_7})$
- $g_i^j(x) = x + i \times C_j \mod 256$

S-boxes

- Involution
- Differential : $1/4$
- Linear : $1/2$
- 20 Clock cycles

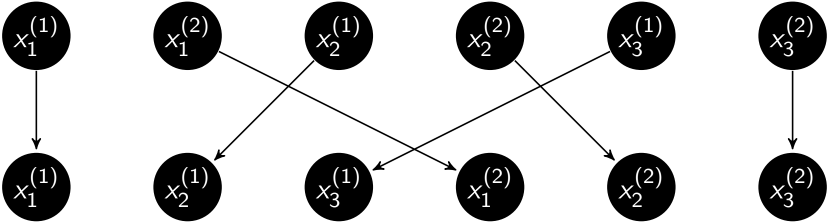
Interleaving

$$P^n_{b_1,\ldots b_k} : (\mathbb{F}_2^{b_1} \times \mathbb{F}_2^{b_2} \times \ldots \mathbb{F}_2^{b_k})^n \longrightarrow (\mathbb{F}_2^{b_1})^n \times (\mathbb{F}_2^{b_2})^n \ldots \times (\mathbb{F}_2^{b_k})^n$$

$$(x_1, \ldots, x_n) \longrightarrow ((x_1^{(1)}, \ldots, x_n^{(1)}), \ldots, (x_1^{(k)}, \ldots, x_n^{(k)}))$$

where $x_i = (x_i^{(1)}, \ldots, x_i^{(k)})$ with $x_i^{(j)} \in \mathbb{F}_2^{b_j}$

Example $k = 2$, $n = 3$



Interleaving

- $G_i = [I|L_i^T]$ matrix generator of a $(2n, 2^n)$ code of minimal distance d_i over $\mathbb{F}_2$
- $L := P^{-1} \circ (L_1 \times L_2 \times L_3 \times L_4) \circ P$
- $[I|L^T]$ matrix generator of a $(2n, 2^n)$ code of minimal distance $mind_i$ over $\mathbb{F}_2^4$

frametitleFinding L_0

Principle

- Find differential characteristics :
- $\Delta X = X_1 \oplus X_2$ a constant
- $\Delta Y = Encr(X_1) \oplus Encr(X_2) = cst$ for a high number($>> 1/2^{|K|}$) of pair (X_1, X_2)
- Retrieve information on the key

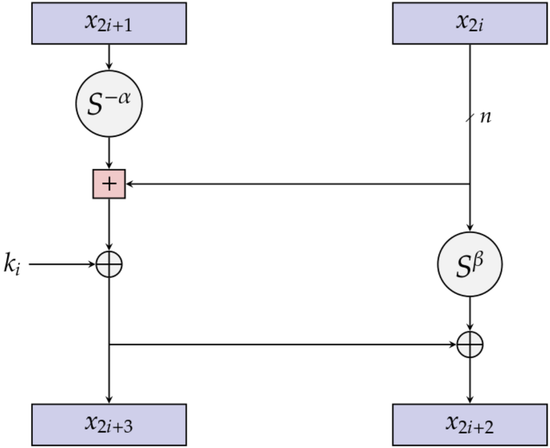
Generalities

- Introduced by Beaulieu & Al (NSA) in 2013
- Feistel network
- 48-128 bits blocks
- 96-256 bits key
- 22-34 rounds

Performances (64 bits block/128 bits key)

- 186 bytes of memory
- 150 cycles per block

Structure



Key Scheduling

- $K = (l_{m-2}||l_{m-1}||...||l_0||k_0)$
- $l_{i+m-1} = k_{i-1} + S^{-\alpha}(l_{i-1}) \oplus i$
- $k_i = S^{\beta}(k_{i-1}) \oplus l_{i+m-1}$
- $l_i, k_0 \in \mathbb{F}_2^n$
- $m \in \{2, 3, 4\}$

Principle

- Inject a fault in a chosen state of the computation
- Compare C and C^* the correct and faulty cipher texts
- Retrive information on the key

We control the position of the error (unrealistic)

- $x^T = (S^{-\alpha}(x^{T-1}) + y^{T-1}) \oplus k^{T-1}$
- $y^T = S^{\beta}(y^{T-1}) \oplus x^T$
- $k_j^T = (x^{T-1}_{j+\alpha \mod n} \oplus (y^T + x^T)_{j+\beta \mod n} \oplus c_j) \oplus x_j^T$
- $c_j = (x_{j-1-\alpha \mod n} \& y_{j-1}) | (c_{j-1} \& (x_{j-1-\alpha \mod n} | y_{j-1}))$

- $c_0 = 0$ is known
- Inject a fault in y_0^{T-1}
- Deduce x_α^{T-1} then k_0^{T-1}
- Inject a fault in higher bits of y^{T-1}

We don't control the position of the error

Locate the error

- $e = S^{-\beta}(x^{T^*} \oplus x^{T} \oplus y^{T^*} \oplus y^{T})$