Secure realization of Lightweight block cipher: A case study using GIFT

Varsha Satheesh[1], and Dillibabu Shanmugam[2]

[1]SSN College of Engineering, Anna University
[2]Society for Electronic Transactions and Security (SETS)

December 18, 2018

Strategy and Synergy for Security

outline

- GIFT
- Vulnerability Analysis of GIFT
- Countermeasure: Threshold implementation (TI) and Implementation techniques
- Evaluation of Countermeasures

Outline

GIFT cipher

GIFT is lightweight block cipher, based on Substitution and permutation network, proposed at CHES2017 by *Subhadeep et al*.

- Smaller area, higher throughput, and faster key schedule
- Better resistance against classical crypt-analysis

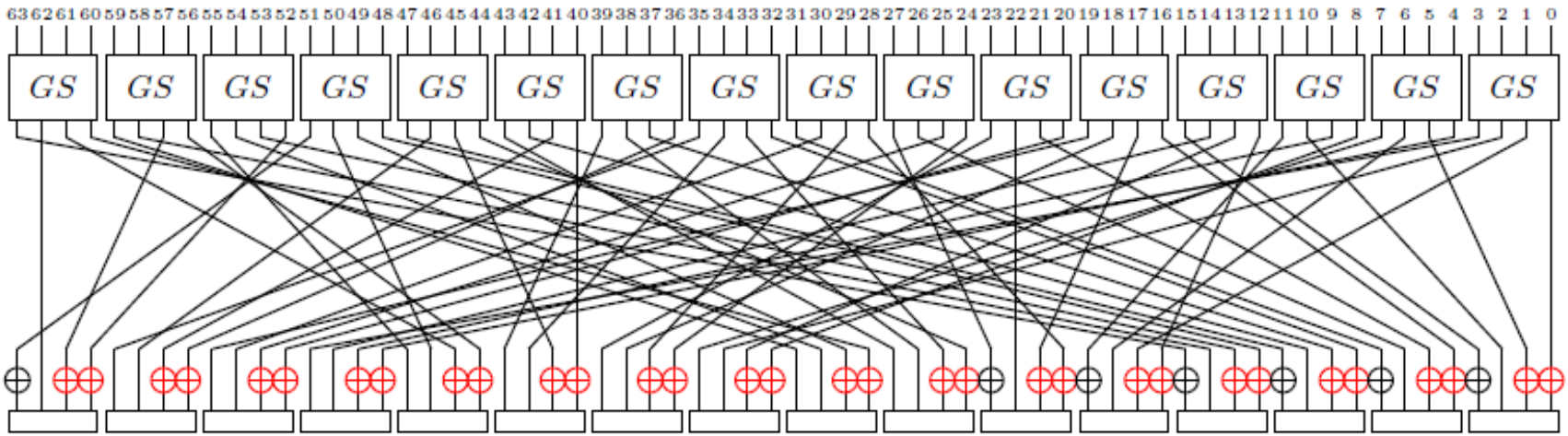
GIFT variants	Block Size(bits)	Keysize(bits)	Rounds
GIFT-64	64	128	28
GIFT-128	128	128	40

Round Function: consists of

- Substitution-box(Sbox)
- Permutation-Layer(PL) and
- AddRoundkey(ARK)

GIFT cipher

- Sbox: Replace 4bits with another 4bits value. Provides confusion property
- PL: Shuffles bits among themselves, provides diffusion property
- ARK: State value is Xored with round key and constant.



Outline

Implementation Vulnerability Analysis

Implementation attack(power analysis attacks) pose serious threat to crypto primitives

Secure algorithm $\neq$ secure implementation

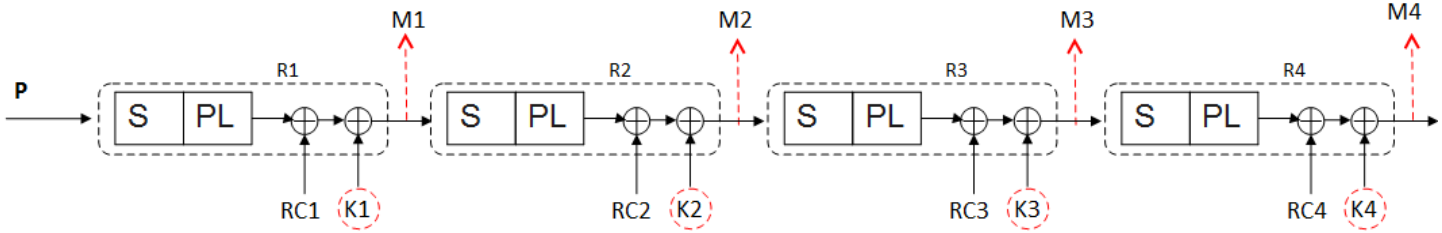
Need for Analysis

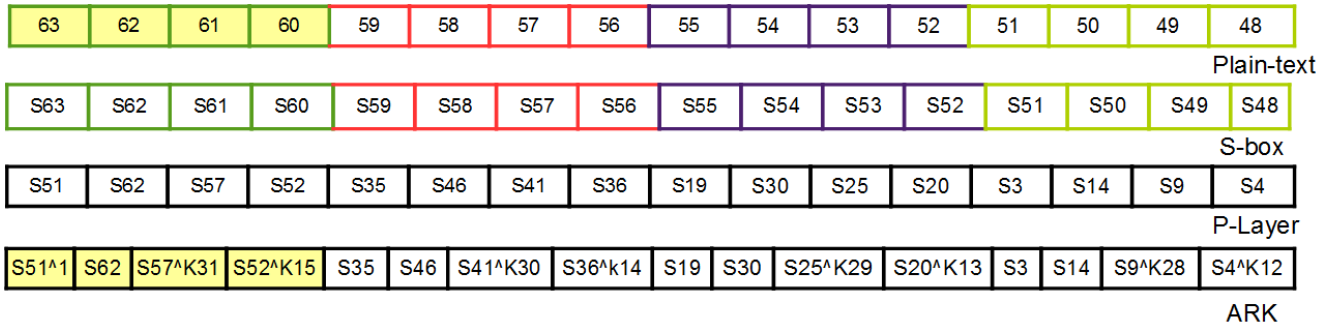
- To identify weak components
- Know the attack complexity
- To arrive efficient countermeasure for the component.
- Helpful for new components design

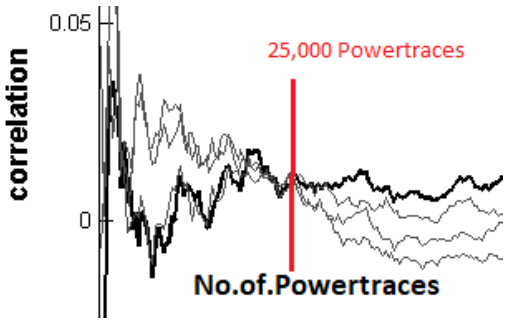
Differential Power Analysis steps:

- Identify Point of Interest
- Capture power consumption
- Hypothetical intermediate value
- Convert Hypothetical intermediate value to power model
- Correlate with capture power consumption to guess the secret key

Attack on GIFT Rolled implementation







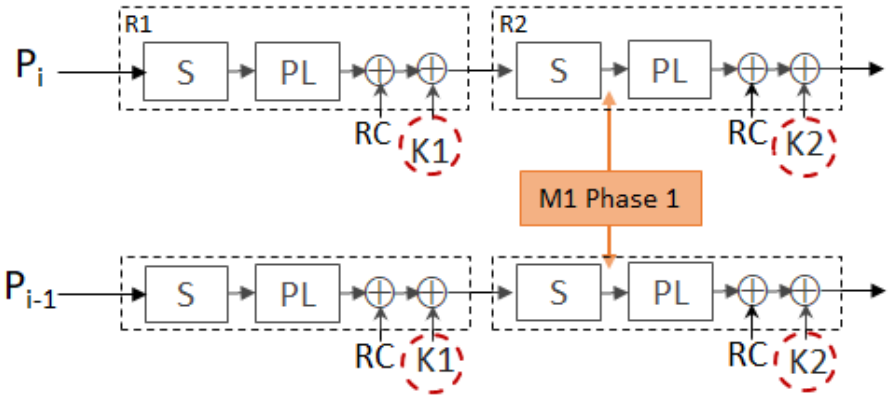
$HD = HW[(P63, P62, P61, P60) \oplus (1 \oplus S51, S62, S57 \oplus K31, S52 \oplus K15)]$

Similarly process for remaining key bits in this round, & subsequent rounds to guess complete 128 key bits. Overall attack complexity is $2^2 * 16 * 4 = 2^8$

Attack on Unrolled implementation

GIFT Unrolled Implementation PoI: (Non)Linear function

No register is used, power model is arrived between intermediate states of two encryption.



Phase 1: second round sbox

Hyp intermediate value: $M1^i = S(PL(S(P^i_j)) \oplus K1_{j,t} \oplus RC1_j)$

Where i and j represents as j^{th} nibble of i^{th} encryption.

Hyp Power Consumption : $P1^i_{hyp} = HD(M1^i_j, M1^{i-1}_j)$

This reveals two key bits. Similarly, 30 key bits are revealed in this phase. Subsequently, second round keys are revealed in third round sbox and so on. Overall attack complexity is $2^2 * 16 * 4 = 2^8$

Outline

Countermeasure

Break the correlation between power leakage and hypothetical power model
Non-linear function(Sbox) is predominantly vulnerable point for SCA.

Motivation and Contributions

- Adopt existing countermeasure in efficient way
- Identify optimal S-box from implementation perspective

Countermeasure

- Created protected profiles by combining Threshold implementation (TI) and implementation techniques. It is believed to be secure against first order DPA.
- Trade-off factors (Area, Latency, Level of Security) need to be considered for resource constrained device, conventional and crypto-accelerators.

Threshold Implementation

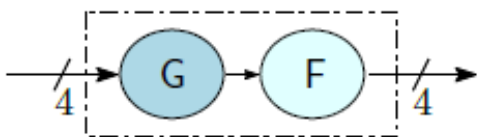
- TI works on sharing principle, proposed by *Nikova et al*
- No.of shares (S_n) is based on algebraic degree (d) of S-box, that is $S_n \geq d + 1$; GIFT Sbox has degree 3. $S_n \geq 3+1$; $S_n \geq 4$;
- Increases the circuit complexity and its area overhead

Decompose the S-box into smaller functions with lower degree

- Sbox is decomposed into two quadratic function, $S = A_1 * g * A * f * A_2$

Functions F,G has degree 2, that is S$\geq$ 2+1 has proposed by *Naina Gupta et al*

Unprotected S-Box



Decomposed

Threshold Implementation

Solutions need to satisfy TI properties for secure shared implementation

- Correctness
- Non-completeness
- Uniformity

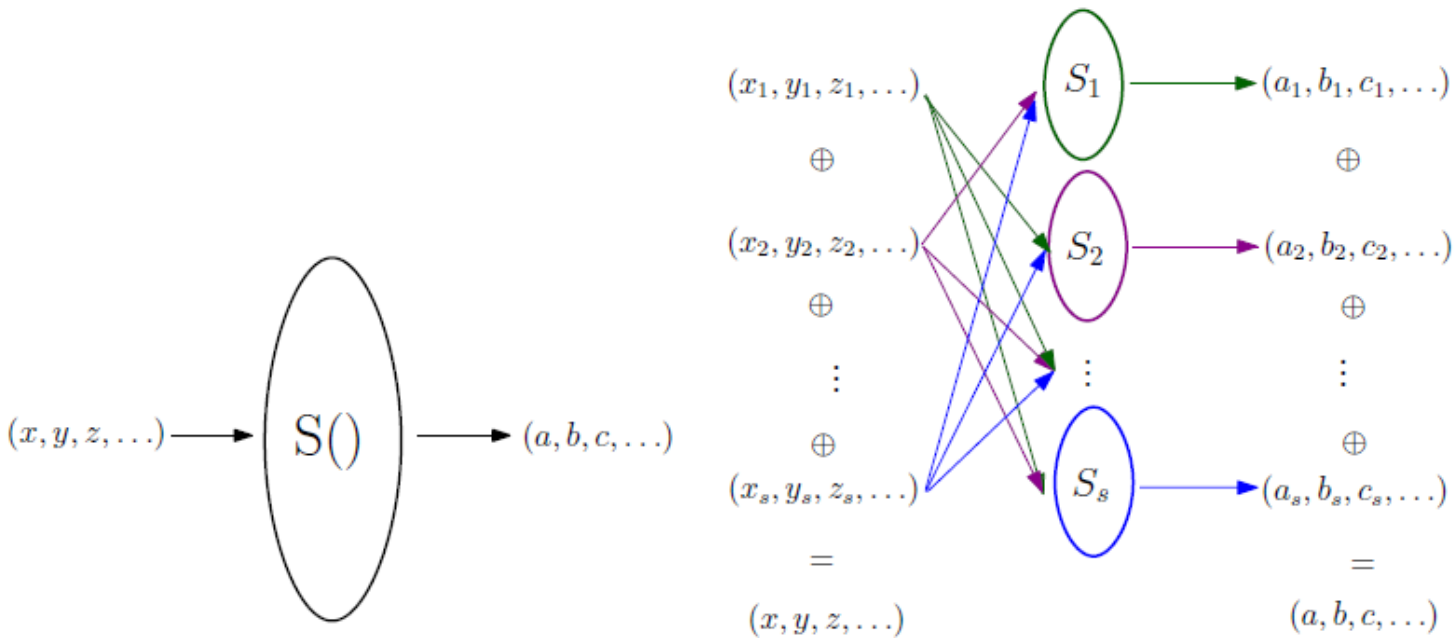


Figure: TI properties

Threshold Implementation

Example: y = f(x) = a AND b
a = 1; b = 1; a = (a_1, a_2, a_3); b = (b_1, b_2, b_3);
$a_1 = 1, a_2 = 1, a_3 = 1$;
$b_1 = 0, b_2 = 1, b_3 = 0$;

- Correctness: a = $(a_1 \oplus a_2 \oplus a_3)$; b = $(b_1 \oplus b_2 \oplus b_3)$;
  a = $(1 \oplus 1 \oplus 1) = 1$; b = $(0 \oplus 1 \oplus 0) = 1$;
- Non-completeness
  $f1(a_2, b_2, a_3, b_3) = a_2b_2 \oplus a_2b_3 \oplus a_3b_2 = 1.1 \oplus 1.0 \oplus 1.1 = 0$
  $f2(a_3, b_3, a_1, b_1) = a_3b_3 \oplus a_3b_1 \oplus a_1b_3 = 1.0 \oplus 1.0 \oplus 1.0 = 0$
  $f3(a_1, b_1, a_2, b_2) = a_1b_1 \oplus a_1b_2 \oplus a_2b_1 = 1.0 \oplus 1.1 \oplus 1.0 = 1$
- Uniformity Input(a,b) = 1.1 the output $f = f_1 \oplus f_2 \oplus f_3 = 1$ and the distribution of its shared output values $(f_1, f_2, f_3) \in \{001, 010, 100, 111\}$ has to be uniform. In other words, each possible shared output has to occur equally likely.

TI shares of Functions(F,G)

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
F(x)	4	d	f	7	1	a	2	8	5	c	e	6	0	b	3	9
G(x)	5	6	3	8	1	2	7	c	9	e	f	0	d	a	b	4
S(x)= G(F(x))	1	a	4	c	6	f	3	9	2	d	b	7	5	0	8	e

S(3) = G(F(3)) = G(7) = C

ANF form of F(a,b,c,d) [4df71a285ce60b39]

$F^1 = a + b + b * a + c + d$ $F^2 = b + c * a$

$F^3 = 1 + c$ $F^4 = a + b + c * b$

ANFs of the GIFT S-Box decomposition with TI 3-shares for function, F.

$F_1(a_2, b_2, c_2, d_2, a_3, b_3, c_3, d_3) = (f_{13}, f_{12}, f_{11}, f_{10})$

$f_{10} = a_2 + b_2 + b_2a_2 + a_3b_2 + b_3a_2 + c_2 + d_2$

$f_{11} = b_2 + c_2a_2 + c_3a_2 + a_3c_2$

$f_{12} = 1 + c_2$

$f_{13} = a_2 + b_2 + c_2b_2 + c_2b_3 + b_3c_2$

$F_2(a_3, x_3, y_3, z_3, a_1, x_1, y_1, z_1) = (f_{23}, f_{22}, f_{21}, f_{20})$

$f_{20} = a_3 + b_3 + b_3a_3 + a_1b_3 + b_1a_3 + c_3 + d_3$

$f_{21} = b_3 + c_3a_3 + c_1a_3 + a_1c_3$

$f_{22} = 1 + c_3$

$f_{23} = a_3 + b_3 + c_3b_3 + c_3b_1 + b_1c_3$

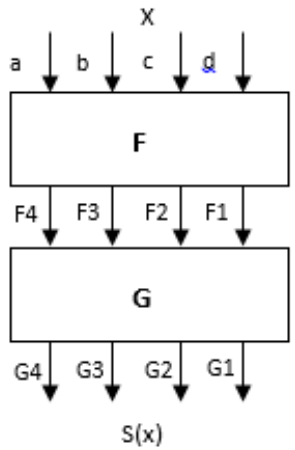
$F_3(a_2, b_2, c_2, d_2, a_1, b_1, c_1, d_1) = (f_{33}, f_{32}, f_{31}, f_{30})$

$f_{30} = a_2 + b_2 + b_2a_2 + a_1b_2 + b_1a_2 + c_2 + d_2$

$f_{31} = b_2 + c_2a_2 + c_1a_2 + a_1c_2$

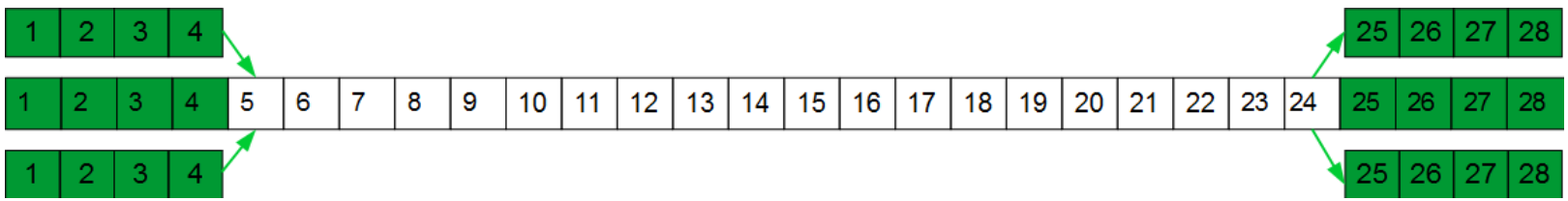
$f_{32} = 1 + c_2$

$f_{33} = a_2 + b_2 + c_2b_2 + c_2b_1 + b_1c_2$



Threshold Implementation and Implementation techniques

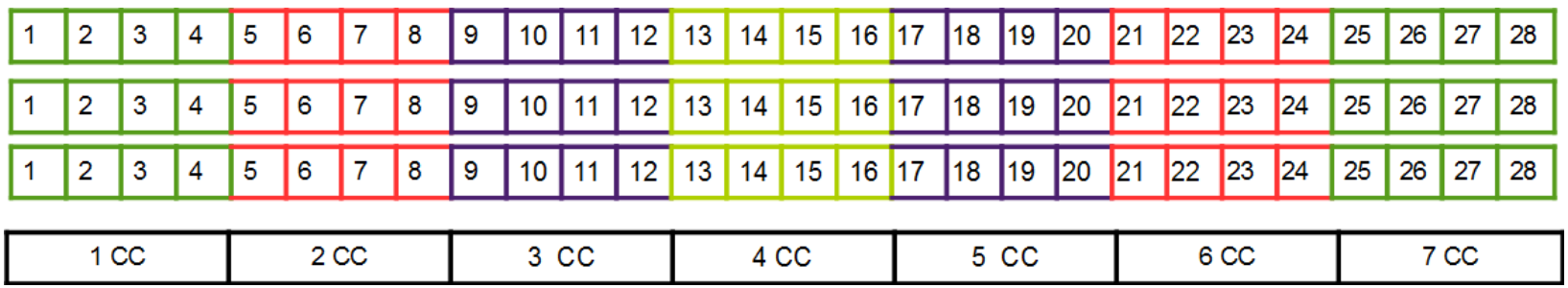
Efficient TI solution is taken for implementation.
First and last four rounds are vulnerable against attack. Therefore protecting these rounds using TI make sense, end-up Profile 1.
Profile 1: Round based implementation with TI on specific rounds



Attack resistant is increased

Profile 2

Round based TI is vulnerable against specific attack, say, static power SCA, explored by *AmirMoradi et al*
Therefore, different implementation styles are combined with TI and created profiles as follows.
Profile 2: Partially unrolled with TI.
Here, every four round is unrolled with TI

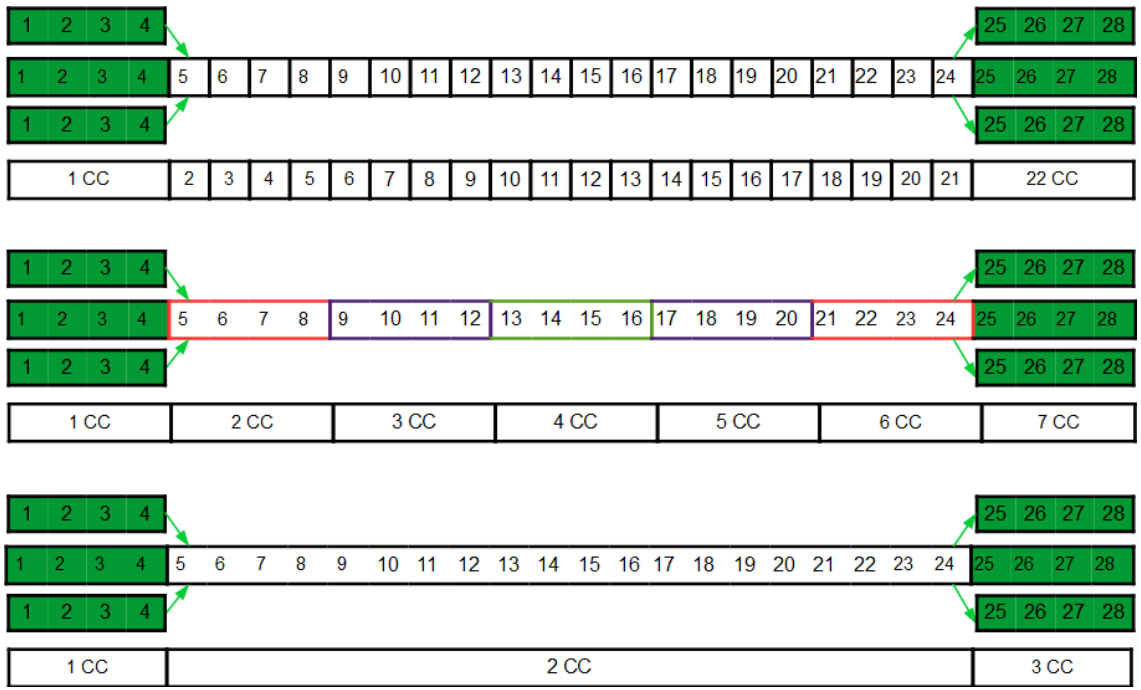


Though area is increased, latency is reduced and attack resistant is increased due to mask.

Profile 3

Profile 3: Partially unrolled with TI on specific rounds

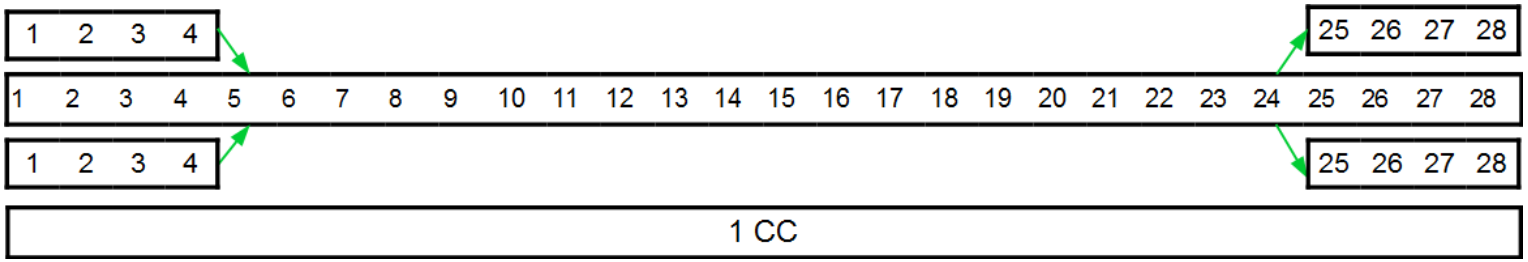
TI on specific rounds, i.e first and last four rounds are implemented in unrolled fashion. Then, based on the middle rounds realization, profile is subdivided into three category as follows for trade-off purpose.



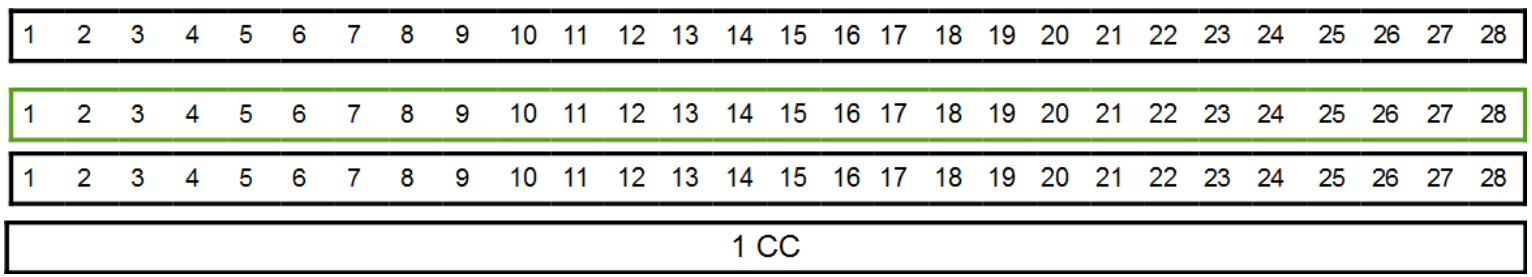
This kind of implementation are suitable for resource constrained and conventional crypto devices.

Profile 4 and 5

Profile 4: Unrolled with TI on specific rounds



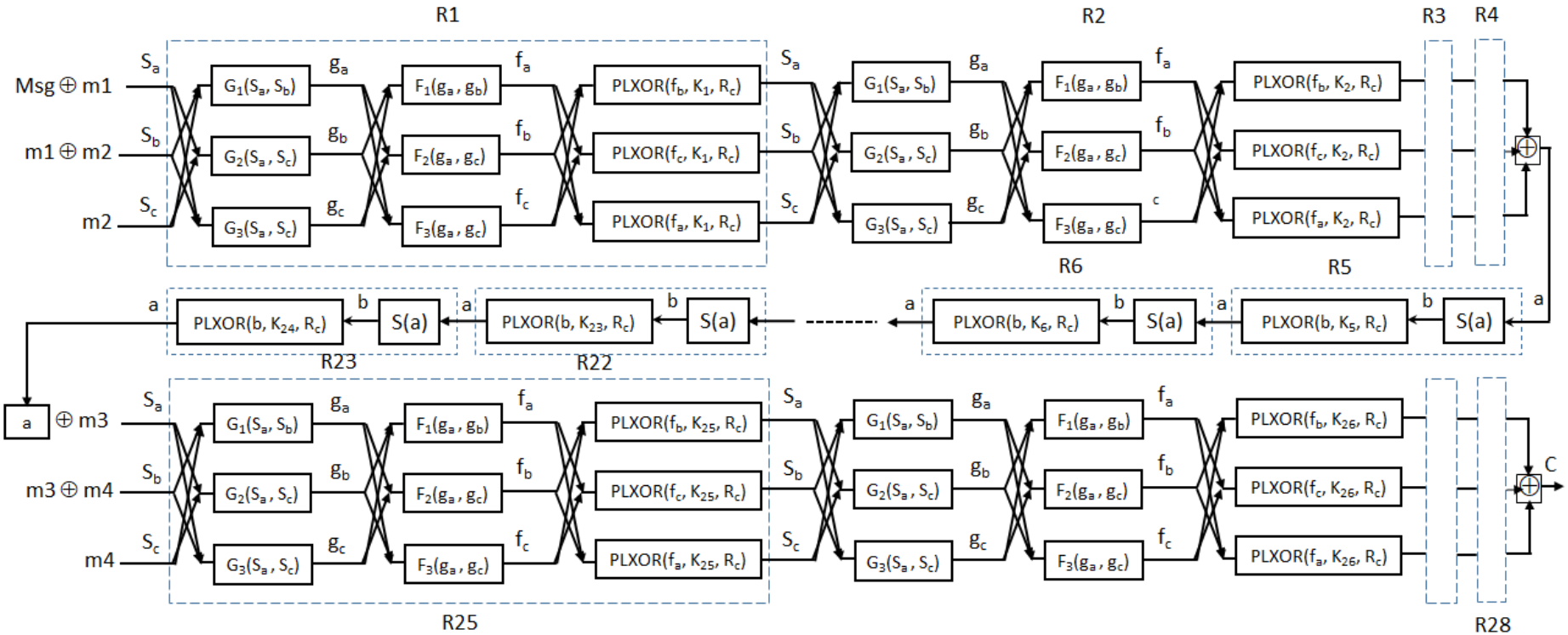
Profile 5: Unrolled with TI



Level of security increased

Unrolled with TI on specific rounds

Architecture of GIFT unrolled with TI on specific rounds

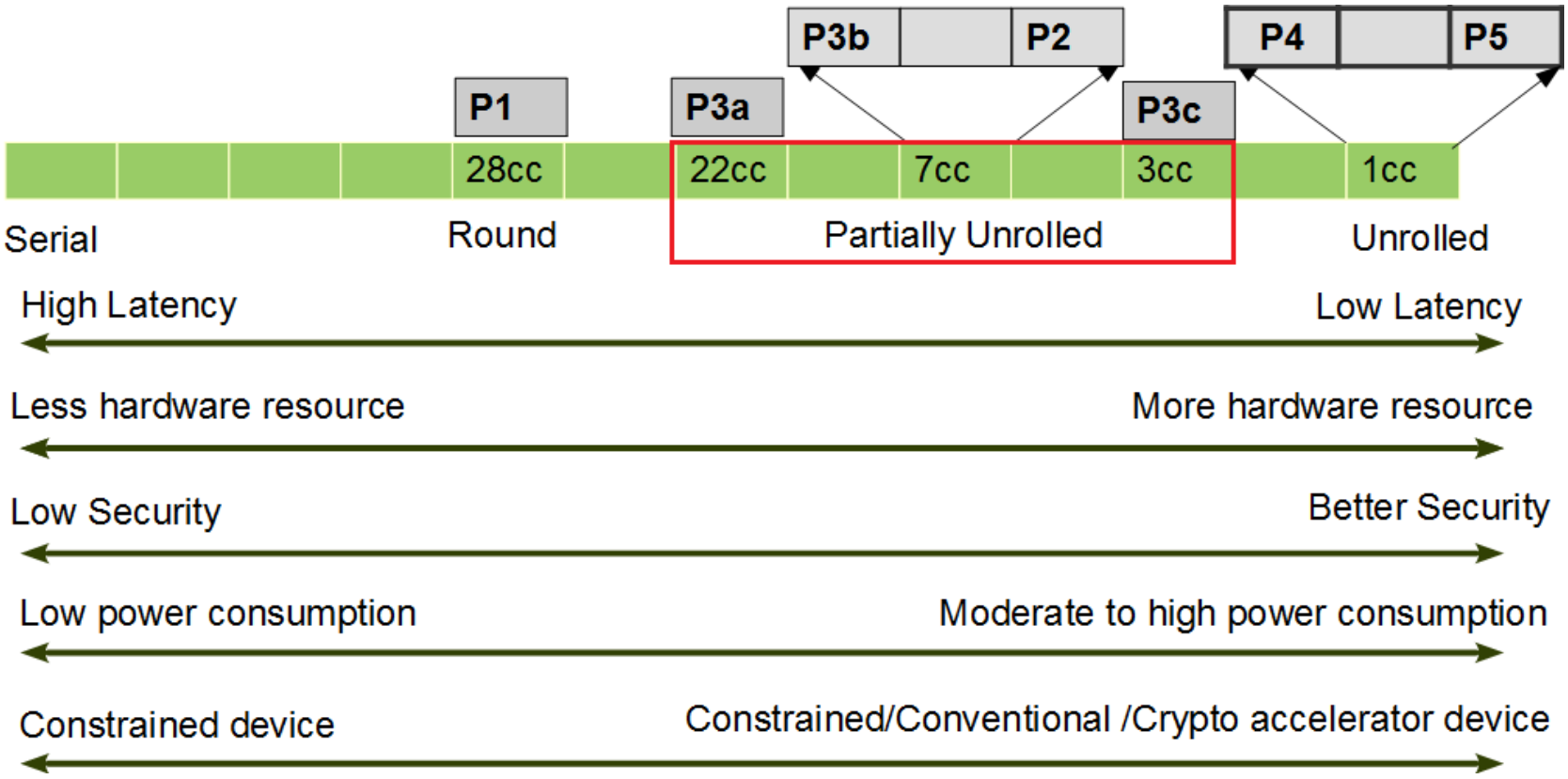


Trade-off factors

- Silicon Area (slices in FPGA) : Circuit realization should be compact.
- Throughput : Output bits/sec, Should be high.
- Power : Consumption, Should be optimal.
- Latency : Execution time, Should be low.
- Security : Attack resistance, should be high.

Implementation Styles	3 Share TI	3 Share TI for First and last four rounds	Naive
Round based(28cc)	3X	3X	X
Partially Unrolled(7cc)	9X	8X	2X
Unrolled(1cc)	18X	15X	6X

Profiles: Trade-off factors



Outline

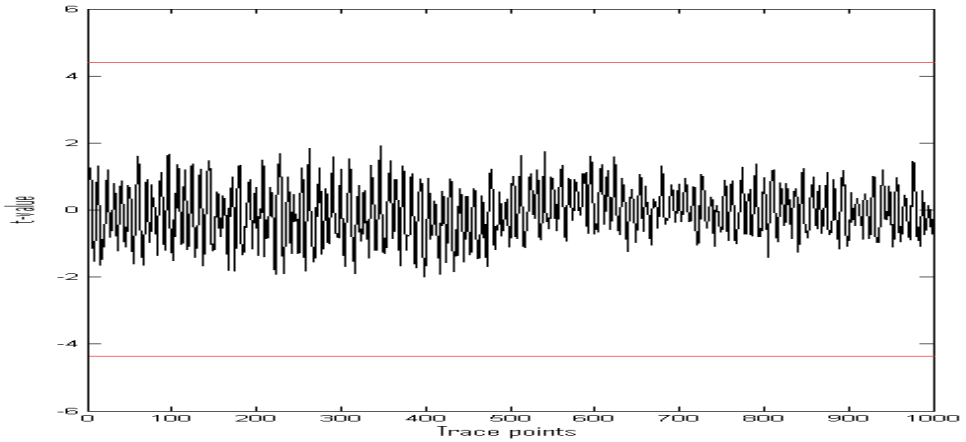
Standards : FIPS140-2/3 and Common Criteria

1. Effectiveness of test: Reproducible & reasonable indicators of resistance
2. Ease & cost-effective: validation should not require excess amount, time and skill.

- FIPS140-2/3
  - Conformance-style testing: Detects the presence of any leakage, independent of attack methodologies and leakage models
  - Test Vector Leakage Assessment(TVLA)
- Common Criteria
  - Evaluation-style testing: Evaluating the system against all state-of-the-art attack strategies.
  - Testing methodology tedious, costly and limited by the testing expertise available at the hand.

Test Vector Leakage Assessment(TVLA)

- TVLA uses well known Welch's t-test. It was proposed as a PASS/FAIL test. If t-value crosses the predefined threshold (proposed as +4.5 to -4.5), then it will leak potential information; otherwise not.
- TVLA on Partial unrolled with TI on specific rounds



By this conformance test, the countermeasure is secure upto 1 million traces.

Summary

- Protected profiles achieve better trade-off (Area, Latency, Level of security) required for resource constrained applications
- Partially unrolled with TI will be effective countermeasure, when the cipher has many rounds of operation.

References

- Subhadeep Banik et al for GIFT
- Gupta et al for GIFT TI, https://eprint.iacr.org/2017/1040.pdf
- Nikova et al, for TI
- Becker et al, for TVLA

Thank You for your Listening