Decomposed S-Boxes and DPA Attacks: A Quantitative Case Study using PRINCE

Ravikumar Selvam, Dillibabu Shanmugam, Suganya Annadurai, Jothi Rangasamy
Society for Electronic Transactions and Security, India.
Email: {ravikumar, dillibabu, asuganya, jothiram}@setsindia.net

Abstract—**Lightweight ciphers become indispensable and inevitable in the ubiquitous smart devices. However, the security of ciphers is often subverted by various types of attacks, especially, implementation attacks such as side-channel attacks. These attacks emphasise the necessity of providing efficient countermeasures. In this paper, our contribution is threefold: First, we observe and resolve the inaccuracy in the well-known and widely used formula for estimation of the number of gate equivalents (GE) in shared implementation. Then we present the first quantitative study on the efficacy of Transparency Order (TO) of decomposed S-Boxes in thwarting a side-channel attack. Using PRINCE S-Box we observe that TO-based decomposed implementation has better DPA resistivity than the naive implementation. To benchmark the DPA resistivity of TO(decomposed S-Box) implementation we arrive at an efficient threshold implementation of PRINCE, which itself merits to be an interesting contribution.**

I. Introduction

Usage of smart electronic devices in day to day life is rapidly growing and almost unavoidable. Smart electronic devices are resource constraints having less memory, low power and limited computation capability. Since, lightweight ciphers require only minimal resources, they are identified to provide compact solutions to achieve security goals to protect such devices. The theoretical proofs of security for cryptographic algorithms give us some confidence; however may not be sufficient to protect against real-world attacks.

As the smart devices are portable and easily accessible to the attacker, these devices are shown to be prone to implementation attacks in a rapid phase. Side-channel analysis attacks exploit the information leakage through physical medium to reveal the secret key of the device. In particular, Differential Power Analysis (DPA) attack is considered as more effective form of side-channel attack that reveals the key with high probability [1]. These attacks brought the attention to develop effective and efficient countermeasures. One such countermeasure, which is widely studied and provably secure, is Threshold Implementation (TI) of block ciphers.

In TI, a non-linear component of block cipher is decomposed into secret shares and the method was first proposed by Nikova et al [2], [3]. Efficient way of realising TI for lightweight cipher was presented in [4]. Threshold Implementation of 4-bit S-Boxes is proposed in [5]. The degree of S-Box, d, decides the number of secret shares, which must be greater than d. In TI the higher degree S-Box is decomposed into smaller degree S-Boxes so that the number of shares needed get reduced, which in turn reduces the area requirement for TI. At the same time, the decomposed S-Boxes must satisfy the TI properties, such as correctness, non-completeness, and uniformity. Indeed not all decompositions need to satisfy the uniformity property, in such cases re-sharing is used to achieve it. However, if non-uniform decomposition is used then the design is vulnerable to attacks as shown in [6], [7].

Leander et al. proposed sixteen optimal 4-bit S-Boxes [8] for lightweight block ciphers. It is stated that only eight out of sixteen are suitable for PRINCE in [9]. Then G_{13} (also represented as C_{231}) class of S-Box is taken for PRINCE implementation based on lexicographical order. S-Boxes are predominantly the point of interest for DPA, due to bit flip occurs randomly in circuit. Theoretically this nature has been studied and defined a metric, Transparency order to evaluate the resistivity of the S-Box against DPA attack by Prouff et al [10]. Further, this metric is studied and explored in many papers [11]–[16]. In [11], it is found that G_{13} class of S-Box has high TO, and it is vulnerable to power analysis attack. Subsequently, differential power analysis attack on PRINCE is demonstrated in [17].

It is stated in [14] that the small reduction in the TO, will increase the trace requirement $2.5-3$ times to mount successful DPA attack. However this prediction on TO is not explored with practical evaluation so far.

Our Contributions

In this paper, we analyse DPA resistance of a decomposed 4-bit S-Box. Our contributions are as follows:

- We first observe the inaccuracy in the well-known and widely used formula proposed by Axel et al. for weighted sum estimation of shared function. In particular their formula leads to an incorrect result when used to compute the weighted sum for the shared implementation of a boolean function. We present a revised formula to produce accurate results in shared implementation.
- We then present an area-efficient TI of PRINCE block cipher, before adopting for resource constrained devices. For this, we use two-level decomposition of PRINCE S-Box that falls in class G_{13}. The chosen decomposition, which satisfy all TI properties is taken and optimized further for implementation.
- We finally study the transparency order for S-Box decomposition and its influence in DPA attack. Our practical evaluations on PRINCE S-Box show that the number of traces required for DPA attack on naive S-Box is

increased significantly for the case of S-Box decomposition with lower transparency order. This implies that the TO-decomposed S-Box implementation is superior over the naive implementation. However benchmarking TO-decomposed S-Box implementation against TI implementation reveals that TI implementation should provide a much better security than any (even unrolled) unprotected implementation. Nevertheless, our experiments show that low transparency order implementation of decomposed S-Box may be considered as an intermediate countermeasure between the naive and TI implementations and the lesser TO means the better immunity against DPA attacks.

II. On Estimation Formula for Weighted Sum

To find the efficient decomposition, in terms of number of gate count (W_{sum}) for shared implementation, Axel et.al [4] proposed a formula to estimate a weighted sum of shared function. We found that the proposed formula have an inconsistency in the result which differs from the actual gate count for the given shared function. This has been illustrated below with sample function (1). For simplicity, we have taken 1-bit inputs w, x, y, and z to compute F. We also defined shares for the function F as f_1, f_2, and f_3.

$$F = 1 + x + y + w + xz \qquad (1)$$

$$f_1 = 1 + x_2 + y_2 + w_2 + x_2z_2 + x_2z_3 + x_3z_2$$

$$f_2 = x_3 + y_3 + w_3 + x_3z_3 + x_3z_1 + x_1z_3$$

$$f_3 = x_1 + y_1 + w_1 + x_1z_1 + x_1z_2 + x_2z_1$$

Following Axel et.al, the XOR and AND gates are given the weightage of 2 and 1 GE respectively. With 16 XOR and 9 AND operations, the (manual) weighted sum calculation of the shared function results in 41 GE, whereas Axel et.al formula (2) outputs the estimated weighted sum as 47 GE. For fixing this inconsistency, we revised the Axel et al formula as in (3) .

$$W_{sum} = (2 \times C) + (6 \times L) + (27 \times Q) \qquad (2)$$

$$W_{sum} = 2 \times ((3 \times C) - 2) + 6 \times (L + Q - 1) + 21 \times Q \quad (3)$$

where,
C is number of Constant
L is number of Linear Co-efficient
Q is number of Quadratic Co-efficient.
We calculated the weighted sum for the same function using our formula and obtained the result as 41 GE, which matches with the actual value of the shared function. The Table I presents the result obtained manually, using our formula and from Axel et al formula. In the following sections, we use our formula to choose the efficient decomposition of S-box.

III. Threshold Implementation of PRINCE S-Box

The PRINCE family proposed eight classes of S-Box for their design, in which the author chose to use affine equivalent of eighth S-Box, as given in [9]. We studied the characteristics of that S-Box and its inverse using the TI Tool [18].

TABLE I
Estimation of weighted sum1

Function	Parameters			Weighted Sum		
	C	L	Q	Manual	Axel formula	Our formula
F=1+x+y+w+xz	1	3	1	41	47	41

TABLE II
S-Box and Inverse S-Box of PRINCE

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	B	F	3	2	A	C	9	1	6	7	8	0	E	5	D	4
$S^{-1}(x)$	B	7	3	2	F	D	8	9	A	6	4	0	5	E	C	1

The S-Box of PRINCE is one of the eight golden S-Boxes proposed in [19], and it falls under the Class 231 with algebraic degree 3 and presented in Table II. We have two choices for the implementation of TI: a) to implement with 5 or 4 shares satisfying all the properties; b) to implement the decomposed S-Box to reduce the number of required shares. In the first case, the implementation requires 5 or 4 times more area than the unprotected implementation. In the second case, if the S-Box is decomposed into lower degree functions, say quadratic functions, then the TI requires 3 shares to implement, which minimizes the required area. But the first level decomposition of PRINCE S-Box yields, one cubic function and one quadratic function. This decomposition requires at least 4 shares, which does not have significant gain in the area requirement. Therefore to reduce the area requirement further, subsequent level of decomposition on cubic functions yields two quadratic functions as shown in Figure 1. Finally, the PRINCE S-Box is decomposed into three quadratic functions.

Using TI tool [18], PRINCE S-Box is decomposed into 304 solutions in the first level of decomposition and 2576 solutions after the second level of decomposition. To construct a secure shared implementation, three TI properties, are to be fulfilled [3]. We have taken first 644 solutions out of 2576 for our analysis. Though all 644 solutions satisfy, correctness and non-completeness properties; only 40 solutions satisfy uniformity properties of TI. The other solutions require either

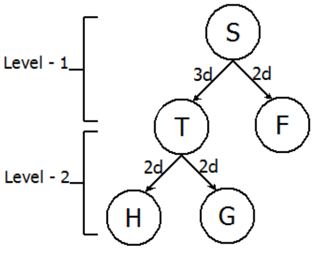

Fig. 1. Decomposition Approach

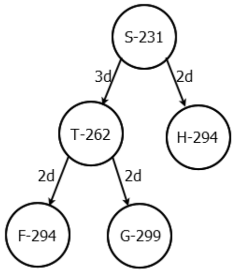

Fig. 2. PRINCE S-Box Decomposition

TABLE III
S-BOX DECOMPOSITION

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$F(x)$	0	A	2	8	1	3	B	9	E	5	D	6	F	C	4	7
$G(x)$	E	4	0	A	2	8	C	6	9	7	5	B	D	3	1	F
$H(x)$	3	6	D	8	A	F	4	1	7	2	C	9	0	5	B	E

TABLE IV
INVERSE S-BOX DECOMPOSITION

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$F^{-1}(x)$	3	9	B	1	7	C	F	4	A	8	2	0	6	5	E	D
$G^{-1}(x)$	E	4	0	A	2	8	C	6	9	7	5	B	D	3	1	F
$H^{-1}(x)$	4	C	9	1	2	A	F	7	E	6	B	3	5	D	0	8

re-masking or virtual variable technique to make them satisfy the uniformity property.

In decomposed S-Box, we analysed 644 solutions using our weighted sum formula which is given in (3). A solution that does not satisfy the uniformity property may also be area efficient after re-masking. To make the process easy and efficient, we classify the solutions into two divisions. Solutions that satisfy the uniformity property and solutions that fail to satisfy the uniformity property of TI. Using (3), we calculated the weighted sum for all solutions of F, G and H functions. We identified the least weighted sum on both classifications separately and compared. The candidate of first category has the least weighted sum of 412 GE, which is quite lesser than the least weighted sum of 447 GE for the second category. Therefore, we chose the candidate with weighted sum 412 GE for hardware space efficient implementation of PRINCE S-Box whose classes are given in Figure 2 and its shares are given in Appendix.

Similarly for Inverse S-Box, 2-level decomposition was performed and 644 solutions of quadratic functions were obtained. The solutions were divided into two divisions and the efficient implementation of Inverse S-Box is also obtained using the same procedure. The Inverse S-Box has the least weighted sum of 354 GE. The decomposed functions F, G and H for the S-box are presented in Table III and decomposed functions F^{-1}, G^{-1} and H^{-1} are presented in Table IV.

A. Hardware Implementation Optimization and DPA experiments

We present round based implementation of PRINCE with TI countermeasure. The implementation is done using VerilogHDL. Round based TI takes 16 clock cycles to complete an encryption as 2 clock cycle for key whitening, 12 clock cycles for round function and 2 clock cycles for mid-layer. The architecture of round based implementation with TI is presented in Figure 3. Inputs M1 and M2 are the mask values, which are 64-bits each. Three registers are maintained to update the state for each round. The S-Box and its inverse were implemented with efficient decomposition as $H(G(F(x)))$ and $H^{-1}(G^{-1}(F^{-1}(\text{x})))$ respectively. An interesting observation is that S-Box and Inverse S-Box are decomposed with same G-function. To improve the area efficiency further, we shared the G function module for S-Box and its Inverse as shown in Figure 4.

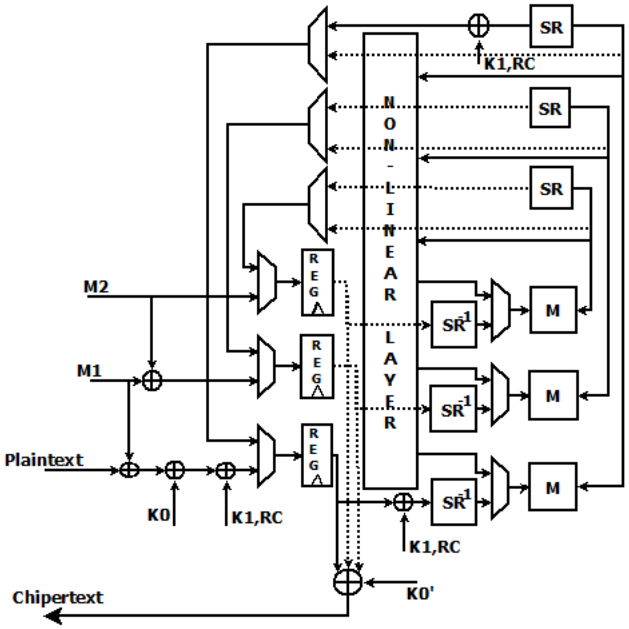

Fig. 3. Architecture of PRINCE Threshold implementation

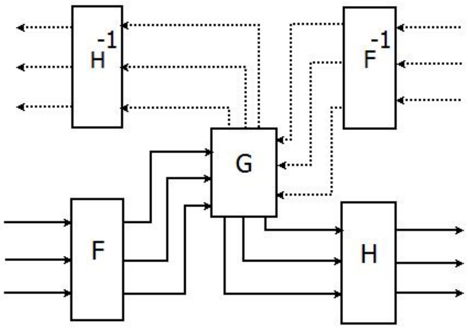

Fig. 4. Sharing of G-function between S-Box with TI and its Inverse

To evaluate the security of protected implementation, we realised TI of PRINCE in SASEBO G board in which the target FPGA device is Xilinx Virtex2Pro. Power measurements were taken for 300,000 encryptions and Pearson's correlation coefficient analysis were performed for the attack. Figure 5 shows that the correct key (plotted in black) is hidden with the other key hypothesis that are plotted in grey. It is understood that TI is secure against DPA attack. In [17], the DPA attack was successful with 30,000 encryptions. Whereas, the protected implementation is secure up to 300,000 encryptions, which is 10 times more secure than the unprotected implementation. Due to resource limitations the protected implementation is tested up to 300,000 encryptions. However, TI is believed to provide more security as mentioned in [3].

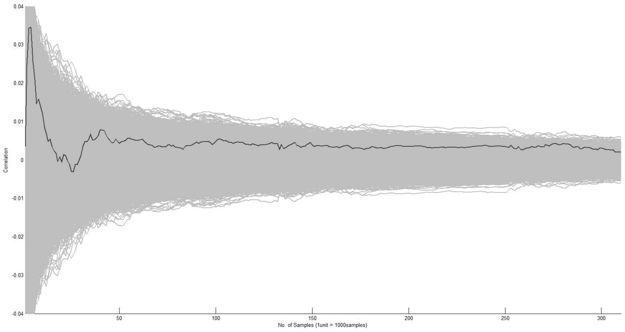

Fig. 5. Attack on Threshold Implementation

TABLE V
CASE 3 S-BOX DECOMPOSITION

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$F(x)$	7	5	0	1	C	A	E	B	F	D	8	9	4	2	6	3
$G(x)$	7	4	5	6	E	9	C	B	8	3	A	1	D	2	F	0
$H(x)$	6	1	7	0	2	5	4	3	8	F	C	B	D	A	E	9

IV. TRANSPARENCY ORDER AND DPA ATTACKS

In general, TO is calculated for naive S-Box to evaluate the DPA resistivity of any cipher. In this paper, we analyse the influence of TO in decomposed S-Box. Our first observation is, TO of naive S-Box is not same as the TO of decomposed S-Box. This observation motivates us to study the behaviour of TO value with respect to the DPA resistance of decomposed S-Box implementation. We use PRINCE for our case study.
The decomposition of PRINCE S-Box has many possible ways using cubic and quadratic functions. The first level decomposition, which comprises of a cubic and quadratic functions, are analysed initially. We noted that the TO of the first level decomposed functions and naive S-Box has negligible difference and may not be suitable for the analysis. Therefore, we analysed the second level decomposition of PRINCE S-Box. The second level decomposition comprises of three quadratic functions and has 2576 (644×4) possible solutions. All solutions are taken for analysis and sorted the solutions based on least TO. We also estimated area requirement in terms of weighted sum as discussed in Section II for the chosen decomposition.

We performed DPA attack on three types of implementations: $1)$ Naive S-Box implementation as shown in Table II with the TO of 3.4 $2)$ decomposed implementation of quadratic functions F, G and H as shown in Table III with TO of 2.933, 3.2 and 3.46 respectively $3)$ decomposed implementation of quadratic functions F, G and H as shown in Table V with TO of 2.933 each. Case 3 is the decomposition that has least TO among all solutions. To verify the impact of changing TO values, same experiment setup (SASEBO-G board) is retained for all experiments to neglect noise influence.

A. *Experimental Result*

1) **Naive S-Box implementation:** Naive S-Box of PRINCE cipher has TO 3.4 and area became 78. We plot the correlation values at $2.021\mu sec$ for different samples. The correct key bit is highlighted in black and others in grey. Figure 6 shows that after 30,000 encryptions the correlation coefficient for the correct key value 108 is ranked first with correlation value 0.038 on the hypothesis list of 2^{13}.

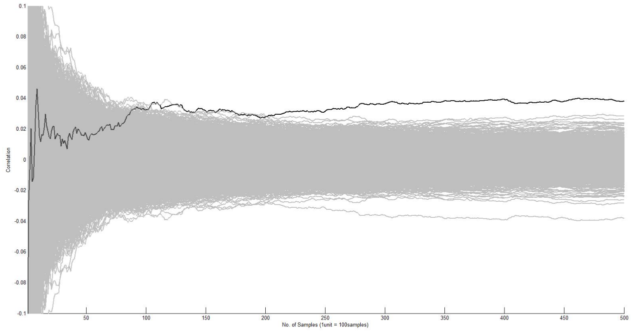

Fig. 6. Attack on naive S-Box

2) **Decomposed implementation with different TO values:** In this case, we had taken the decomposition used for TI for which the TO of decomposed functions are $(TO^F_{S_i} = 2.933, TO^G_{S_i} = 3.2, TO^H_{S_i} = 3.46)$ and the S-box area is measured as 72 GE. DPA attack on this implementation reveals secret key with 30,000 encryptions. In Figure 7, the correct key value 108 (in decimal) is uniquely distinguishable which is having the correlation value of 0.03 at $2.032\mu sec.$ This decomposed function did not have any impact on the resistivity against DPA attack. The reason could be $TO^H_{S_i} = 3.46$, which is being the highest among three functions TO value. The higher TO of H function may dominate the other functions. Subsequently, $TO^H_{S_i} = 3.4$ is same as $TO_{NS} = 3.4$. Therefore, number of traces required to attack did not vary.

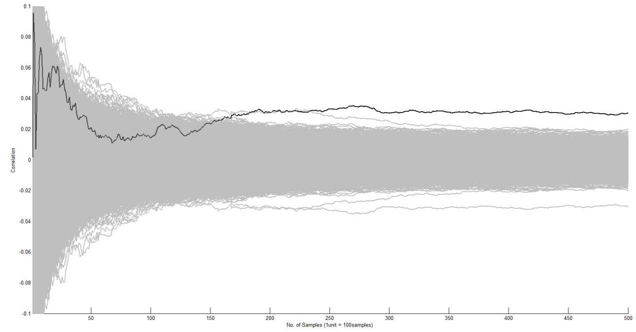

Fig. 7. Attack on Decomposed S-Box with different TO values

3) **Decomposed implementation with same TO values** The least TO from the decomposed functions ($TO^F_{S_i} = TO^G_{S_i} = TO^H_{S_i} = 2.933$) is taken for analysis and its area requirement is 87. When DPA is explored for this decomposition, the cipher requires 2,50,000 traces to reveal 85 percentage of secret key. Figure 8 shows highest correlation value of 0.01 at $2.102\mu sec$ for the key value 108. From this, we observe that the decomposed solution achieve eight times better security than the naive implementation in terms of DPA resistance. Hence, TO based decomposed implementation may be considered as an implementation strategy to resist DPA to certain extent.

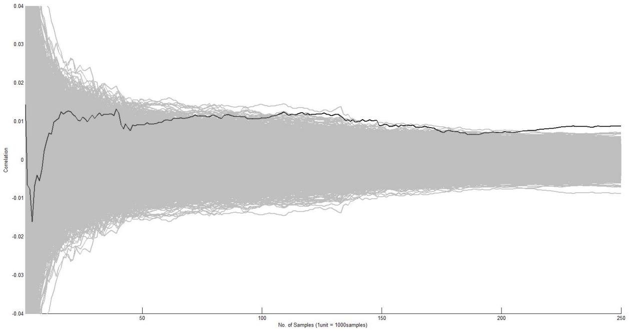

Fig. 8. Attack on Decomposed S-Box with same TO values)

We practically evaluate the impact of TO value on the DPA resistivity of S-Box, which has not been verified so far. From the experimental results we observe that TO-based decomposed implementation seems to provide better trade-off between naive and TI-based approaches. That is, TO-based decomposed implementation is superior to naive implementation but is inferior to TI-based approach. Experimentally we verified that there is an inverse relationship between TO value and the DPA resistance; that is, when the TO value increases, DPA resistance decreases.

B. Comparative study of PRINCE S-Box results for constrained device

Normally, constrained devices has low area, limited computational capability and low power consumption. Though device has many limitations, it is expected to provide the level of security as in conventional device. Such thing may not be practically realisable. But, word trade-off gives two choices for users depending on application requirement.

- Select the specific parameter (in this case security)
- Parameter affordability (level of the security)

In this paper, security affordability is studied and its metric are tabulated in Table VI. Three kinds of security affordability were discussed, such as naive, decomposed S-Box (least TO) and decomposed TI (least area) of PRINCE. Even though decomposed TI has better security when compared to other implementations, it comes at the cost of area. Therefore, by far decomposed S-Box implementation achieves better trade-off, i.e. small increase in area, say about 10 GE in weighted sum, achieves eight times better security when compared to naive S-Box implementation.

TABLE VI
COMPARATIVE STUDY

Metrics	Naive	TO	TI
No. of encryptions for DPA attack	30,000	250,000	> 300,000
Area of S-box in GE	78	87	412

V. CONCLUSION

Protecting lightweight ciphers from side-channel attack is seen to be a mammoth task. In this paper, we observed and corrected the inaccuracy in the widely-accepted formula for estimating gate equivalents for shared implementation. Then we presented the first quantitative study on the efficacy of Transparency Order (TO) of decomposed functions of S-Boxes and its effectiveness in thwarting a specific side-channel attack, namely DPA. Using PRINCE S-Box we observed that TO-based S-Box decomposition may be considered as an intermediate countermeasure since TO-based decomposed implementation provides better DPA immunity than the naive implementation but not as strong as DPA immunity that can be achieved via the TI method. For this we arrived at an efficient threshold implementation (TI) for PRINCE block cipher using two-level decompositions, which itself is an interesting contribution.

APPENDIX

Listed below are the algebraic normal forms (ANFs) of the PRINCE non-linear function that implemented with TI countermeasure.

A. F and H function of S-Box decomposition with 3-share.

$F_1(w_2, x_2, y_2, z_2, w_3, x_3, y_3, z_3) = (f_{13}, f_{12}, f_{11}, f_{10})$
$f_{10} = x_2 + w_2y_2 + w_2y_3 + w_3y_2 + w_2z_2 + w_2z_3 + w_3z_2$
$f_{11} = z_2 + y_2 + w_2$
$f_{12} = w_2$
$f_{13} = z_2 + w_2 + x_2z_2 + x_2z_3 + x_3z_2 + x_2y_2 + x_2y_3 + x_3y_2$

$F_2(w_3, x_3, y_3, z_3, w_1, x_1, y_1, z_1) = (f_{23}, f_{22}, f_{21}, f_{20})$
$f_{20} = x_3 + w_3y_3 + w_3y_1 + w_1y_3 + w_3z_3 + w_3z_1 + w_1z_3$
$f_{21} = z_3 + y_3 + w_3$
$f_{22} = w_3$
$f_{23} = z_3 + w_3 + x_3z_3 + x_3z_1 + x_1z_3 + x_3y_3 + x_3y_1 + x_1y_3$

$F_3(w_1, x_1, y_1, z_1, w_2, x_2, y - 2, z_2) = (f_{33}, f_{32}, f_{31}, f_{30})$
$f_{30} = x_1 + w_1y_1 + w_1y_2 + w_2y_1 + w_1z_1 + w_1z_2 + w_2z_1$
$f_{31} = z_1 + y_1 + w_1$
$f_{32} = w_1$
$f_{33} = z_1 + w_1 + x_1z_1 + x_1z_2 + x_2z_1 + x_1y_1 + x_1y_2 + x_2y_1$

$H_1(w_2, x_2, y_2, z_2, w_3, x_3, y_3, z_3) = (h_{13}, h_{12}, h_{11}, h_{10})$
$h_{10} = 1 + z_2 + x_2 + w_2y_2 + w_2y_3 + w_3y_2$

$h_{11} = 1 + y_2 + w_2x_2 + w_2x_3 + w_3x_2$
$h_{12} = z_2+y_2+w_2+w_2y_2+w_2y_3+w_3y_2+w_2x_2+w_2x_3+w_3x_2$
$h_{13} = y_2 + x_2 + w_2x_2 + w_2x_3 + w_3x_2$

$H_2(w_3, x_3, y_3, z_3, w_1, x_1, y_1, z_1) = (h_{23}, h_{22}, h_{21}, h_{20})$
$h_{20} = z_3 + x_3 + w_3y_3 + w_3y_1 + w_1y_3$
$h_{21} = y_3 + w_3x_3 + w_3x_1 + w_1x_3$
$h_{22} = z_3+y_3+w_3+w_3y_3+w_3y_1+w_1y_3+w_3x_3+w_3x_1+w_1x_3$
$h_{23} = y_3 + x_3 + w_3x_3 + w_3x_1 + w_1x_3$

$H_3(w_1, x_1, y_1, z_1, w_2, x_2, y_2, z_2) = (h_{33}, h_{32}, h_{31}, h_{30})$
$h_{30} = z_1 + x_1 + w_1y_1 + w_1y_2 + w_2y_1$
$h_{31} = y_1 + w_1x_1 + w_1x_2 + w_2x_1$
$h_{32} = z_1+y_1+w_1+w_1y_1+w_1y_2+w_2y_1+w_1x_1+w_1x_2+w_2x_1$
$h_{33} = y_1 + x_1 + w_1x_1 + w_1x_2 + w_2x_1$

B. F^{-1} and H^{-1} function of inverse S-box decomposition with 3-share

$F_1^{-1}(w_2, x_2, y_2, z_2, w_3, x_3, y_3, z_3) = (f_{13}^{-1}, f_{12}^{-1} f_{11}^{-1}, f_{10}^{-1})$
$f_{10}^{-1} = 1 + w_2 + x_2z_2 + x_2z_3 + x_3z_2$
$f_{11}^{-1} = 1 + z_2$
$f_{12}^{-1} = x_2$
$f_{13}^{-1} = z_2+y_2+w_2+w_2z_2+w_2z_3+w_3z_2+w_2x_2+w_2x_3+w_3x_2$

$F_2^{-1}(w_3, x_3, y_3, z_3, w_1, x_1, y_1, z_1) = (f_{23}^{-1}, f_{22}^{-1}, f_{21}^{-1}, f_{20}^{-1})$
$f_{20}^{-1} = w_3 + x_3z_3 + x_3z_1 + x_1z_3$
$f_{21}^{-1} = z_3$
$f_{22}^{-1} = x_3$
$f_{23}^{-1} = z_3+y_3+w_3+w_3z_3+w_3z_1+w_1z_3+w_3x_3+w_3x_1+w_1x_3$

$F_3^{-1}(w_1, x_1, y_1, z_1, w_2, x_2, y_2, z_2) = (f_{33}^{-1}, f_{32}^{-1}, f_{31}^{-1}, f_{30}^{-1})$
$f_{30}^{-1} = w_1 + x_1z_1 + x_1z_2 + x_2z_1$
$f_{31}^{-1} = z_1$
$f_{32}^{-1} = x_1$
$f_{33}^{-1} = z_1+y_1+w_1+w_1z_1+w_1z_2+w_2z_1+w_1x_1+w_1x_2+w_2x_1$

$H_1^{-1}(w_2, x_2, y_2, z_2, w_3, x_3, y_3, z_3) = (h_{13}^{-1}, h_{12}^{-1}, h_{11}^{-1}, h_{10}^{-1})$
$h_{10}^{-1} = y_2 + w_2x_2 + w_2x_3 + w_3x_2$
$h_{11}^{-1} = x_2 + w_2$
$h_{12}^{-1} = 1 + y_2 + x_2 + w_2x_2 + w_2x_3 + w_3x_2$
$h_{13}^{-1} = z_2+y_2+w_2+w_2x_2+w_2x_3+w_3x_2+w_2y_2+w_2y_3+w_3y_2$

$H_2^{-1}(w_3, x_3, y_3, z_3, w_1, x_1, y_1, z_1) = (h_{23}^{-1}, h_{22}^{-1}, h_{21}^{-1}, h_{20}^{-1})$
$h_{20}^{-1} = y_3 + w_3x_3 + w_3x_1 + w_1x_3$
$h_{21}^{-1} = x_3 + w_3$
$h_{22}^{-1} = y_3 + x_3 + w_3x_3 + w_3x_1 + w_1x_3$
$h_{23}^{-1} = z_3+y_3+w_3+w_3x_3+w_3x_1+w_1x_3+w_3y_3+w_3y_1+w_1y_3$

$H_3^{-1}(w_1, x_1, y_1, z_1, w_2, x_2, y_2, z_2) = (h_{33}^{-1}, h_{32}^{-1}, h_{31}^{-1}, h_{30}^{-1})$
$h_{30}^{-1} = y_1 + w_1x_1 + w_1x_2 + w_2x_1$
$h_{31}^{-1} = x_1 + w_1$
$h_{32}^{-1} = y_1 + x_1 + w_1x_1 + w_1x_2 + w_2x_1$
$h_{33}^{-1} = z_1+y_1+w_1+w_1x_1+w_1x_2+w_2x_1+w_1y_1+w_1y_2+w_2y_1$

C. Common G function of both S-box and inverse S-box decomposition with 3-share

$G_1(w_2, x_2, y_2, z_2, w_3, x_3, y_3, z_3) = (g_{13}, g_{12}, g_{11}, g_{10})$
$g_{10} = w_2$
$g_{11} = 1 + z_2 + y_2 + w_2 + w_2y_2 + w_2y_3 + w_3y_2$
$g_{12} = 1 + x_2 + y_2 + w_2 + w_2z_2 + w_2z_3 + w_3z_2$
$g_{13} = 1 + z_2 + y_2 + x_2 + w_2x_2 + w_2x_3 + w_3x_2$

$G_2(w_3, x_3, y_3, z_3, w_1, x_1, y_1, z_1) = (g_{23}, g_{22}, g_{21}, g_{20})$
$g_{20} = w_3$
$g_{21} = z_3 + y_3 + w_3 + w_3y_3 + w_3y_1 + w_1y_3$
$g_{22} = x_3 + y_3 + w_3 + w_3z_3 + w_3z_1 + w_1z_3$
$g_{23} = z_3 + y_3 + x_3 + w_3x_3 + w_3x_1 + w_1x_3$

$G_3(w_1, x_1, y_1, z_1, w_2, x_2, y_2, z_2) = (g_{33}, g_{32}, g_{31}, g_{30})$
$g_{30} = w_1$
$g_{31} = z_1 + y_1 + w_1 + w_1y_1 + w_1y_2 + w_2y_1$
$g_{32} = x_1 + y_1 + w_1 + w_1z_1 + w_1z_2 + w_2z_1$
$g_{33} = z_1 + y_1 + x_1 + w_1x_1 + w_1x_2 + w_2x_1$

REFERENCES

[1] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *CRYPTO'99*, M. Wiener, Ed. Springer, 1999, pp. 388–397.

[2] P. Ning, S. Qing, and N. Li, Eds., *Threshold Implementations Against Side-Channel Attacks and Glitches*, ser. LNCS, vol. 4307. Springer, 2006.

[3] S. Nikova, V. Rijmen, and M. Schläffer, "In proc. international conference on information security and cryptology (icisc) 2008," P. J. Lee and J. H. Cheon, Eds. Springer Berlin Heidelberg, 2009, pp. 218–234.

[4] A. Poschmann, A. Moradi, K. Khoo, C.-W. Lim, H. Wang, and S. Ling, "Side-channel resistant crypto for less than 2,300 GE," *Journal of Cryptology*, vol. 24, no. 2, pp. 322–345, 2010.

[5] S. Kutzner, P. H. Nguyen, A. Poschmann, and H. Wang, "On 3-share threshold implementations for 4-bit s-boxes," in *Constructive Side-Channel Analysis and Secure Design: 4th International Workshop, COSADE 2013*, E. Prouff, Ed. Springer Berlin Heidelberg, 2013, pp. 99–113.

[6] B. Bilgin, "Threshold Implementations: As Countermeasure Against Higher-Order Differential Power Analysis," Ph.D. dissertation, KU Leuven and UTwente, 2015, pieter Hartel and Vincent Rijmen (promotors).

[7] P. Sasdrich, A. Moradi, and T. Gneysu, "Affine equivalence and its application to tightening threshold implementations," Cryptology ePrint Archive, Report 2015/749, 2015, http://eprint.iacr.org/.

[8] G. Leander and A. Poschmann, "On the classification of 4 bit s-boxes," in *Arithmetic of Finite Fields: First International Workshop, WAIFI 2007, Madrid, Spain, June 21-22, 2007. Proceedings*, C. Carlet and B. Sunar, Eds. Springer Berlin Heidelberg, 2007, pp. 159–176.

[9] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavun, M. Knezevic, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, P. Rombouts, S. S. Thomsen, and T. Yalçın, "Prince – a low-latency block cipher for pervasive computing applications," in *ASIACRYPT 2012*, X. Wang and K. Sako, Eds. Springer Berlin Heidelberg, 2012, pp. 208–225.

[10] E. Prouff, "Fast software encryption(fse)," H. Gilbert and H. Handschuh, Eds. Springer Berlin Heidelberg, 2005, pp. 424–441.

[11] S. Picek, B. Ege, K. Papagiannopoulos, L. Batina, and D. Jakobovic, "Optimality and beyond: The case of 4 * 4 s-boxes," in *2014 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2014*. IEEE Computer Society, 2014, pp. 80–83.

[12] C. Carlet, "On highly nonlinear s-boxes and their inability to thwart DPA attacks," in *Progress in Cryptology - INDOCRYPT 2005*, ser. LNCS, S. Maitra, C. E. V. Madhavan, and R. Venkatesan, Eds., vol. 3797. Springer, 2005, pp. 49–62.

[13] B. Mazumdar, D. Mukhopadhyay, and I. Sengupta, "Constrained search for a class of good bijective s-boxes with improved DPA resistivity," *IEEE Trans. Information Forensics and Security*, vol. 8, no. 12, pp. 2154–2163, 2013.

[14] S. Picek, B. Ege, L. Batina, D. Jakobovic, L. Chmielewski, and M. Golub, “On using genetic algorithms for intrinsic side-channel resistance: the case of AES s-box,” in *Proceedings of the First Workshop on Cryptography and Security in Computing Systems (CS2@HiPEAC) 2014*, J. Knoop, V. Salapura, I. Koren, and G. Pelosi, Eds. ACM, 2014, pp. 13–18.

[15] B. Mazumdar, D. Mukhopadhyay, and I. Sengupta, “Design and implementation of rotation symmetric s-boxes with high nonlinearity and high DPA resilience,” in *2013 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2013*. IEEE Computer Society, 2013, pp. 87–92.

[16] K. Chakraborthy, S. Sarkar, S. maitra, B. Mazumdar, D. Mukhopadhyay, and et al.., “Redefining the transparency order.” in *WCC2015 - 9th International Workshop on Coding and Crypography 2015*, 2015.

[17] R. Selvam, D. Shanmugam, and S. Annadurai, “Vulnerability analysis of prince and rectangle using cpa,” in *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security*, ser. CPSS ’15. ACM, 2015, pp. 81–87.

[18] Ti tools for the 3x3 and 4x4 s-boxes. [Online]. Available: http://homes.esat.kuleuven.be/~snikova/ti_tools.html,1999, Onlineaccessed2016-04-05

[19] M.-J. O. Saarinen, “Cryptographic analysis of all 4x4-bit s-boxes,” in *Selected Areas in Cryptography: 18th International Workshop, SAC 2011*, A. Miri and S. Vaudenay, Eds. Springer Berlin Heidelberg, 2012, pp. 118–133.