Received: 2 June 2020 | Revised: 28 September 2020 | Accepted: 22 October 2020

DOI: 10.4218/etrij.2020-0175

ORIGINAL ARTICLE

Power analysis attack resilient block cipher implementation based on 1-of-4 data encoding

Shanthi Rekha Shanmugham | **Saravanan Paramasivam**

Department of ECE, PSG College of Technology, Tamil Nadu, India

Correspondence
Saravanan Paramasivam, Department of ECE, PSG College of Technology, Tamil Nadu, India.
Email: dpsaravanan@gmail.com

Funding information
Visvesvaraya PhD Scheme of Ministry of Electronics and Information Technology (MeitY), Government of India, Grant/Award Number: MEITY-PHD-1905; Special Manpower Development Programme for Chips to System Design, Department of Electronics and Information Technology (DeitY), Government of India

Side-channel attacks pose an inevitable challenge to the implementation of cryptographic algorithms, and it is important to mitigate them. This work identifies a novel data encoding technique based on 1-of-4 codes to resist differential power analysis attacks, which is the most investigated category of side-channel attacks. The four code words of the 1-of-4 codes, namely (0001, 0010, 1000, and 0100), are split into two sets: set-0 and set-1. Using a select signal, the data processed in hardware is switched between the two encoding sets alternately such that the Hamming weight and Hamming distance are equalized. As a case study, the proposed technique is validated for the NIST standard AES-128 cipher. The proposed technique resists differential power analysis performed using statistical methods, namely correlation, mutual information, difference of means, and Welch's t-test based on the Hamming weight and distance models. The experimental results show that the proposed countermeasure has an area overhead of 2.3× with no performance degradation comparatively.

KEYWORDS
Data encoding, differential power analysis, hamming weight/distance equalization, hiding technique, register transfer level countermeasure

1 | INTRODUCTION

With the recent evolution of powerful digital technologies like the Cloud, big data, Internet of Things, and artificial intelligence, a huge amount of data is processed and communicated throughout the world. This leads to a greater concern about the security of the data used, so cryptography is unavoidable to ensure privacy and trust for the users. The hardware and software implementations of cryptographic algorithms leak potential information regarding the operands and their internal processing through various side-channels such as power consumption, electromagnetic radiations, and execution time. Therefore, side-channel attacks have evolved as a powerful hazard to the implementation of cryptographic algorithms in hardware/software platforms and must be alleviated. Specifically, leveraging the power consumption for secret-key extraction is described in the literature under the name of differential power analysis (DPA) attacks. Non-profiled DPA attacks can be evaluated based on two prominent hypothetical models, namely the Hamming weight (HW) and Hamming distance (HD) power models.

The objective of this work is to develop a generic power balancing technique using data encoding to address both HW and HD-based DPA that is applicable to any vulnerable hardware implementation with a reasonable area overhead.

The proposed technique was evaluated for NIST certified Advanced Encryption Standard (AES) [1] with 128 bits of data and key length. The number of encryption/decryption rounds is 10, with each round comprising the following: (i) Shift Rows—implemented using wiring. (ii) Substitution Box (S-box)—implemented using memory-based look-up tables (LUTs) or composite field architecture (CFA)-based gate-level realization. CFA-based realization is preferable for ease of implementation since it can be realized using only XOR and AND gates with a low area. (iii) Mix Columns—implemented using XOR and AND gates to realize Galois field multipliers. (iv) Add Round Key—implemented using XOR gates. Overall, the datapath of the gate-level implementation of AES can be realized using AND and XOR gates, whereas the control path can be realized using suitable multiplexers and registers. The round keys are assumed to be available in memory.

This paper is organized as follows. Section 1 introduces the DPA attacks, AES block cipher, and mentions the objective. Section 2 reviews the related literature. Section 3 explains the proposed work in detail. Section 4 describes the implementation of the proposed work for the AES cipher followed by its performance comparison. Section 5 presents the security analysis results.

2 | RELATED WORK

Various methods have been proposed to counteract the DPA attacks, predominantly in the form of data masking [2–5] and hiding [6–11]. The masking approach in software and hardware obscures the data processed using random numbers and resists HW and HD-based DPA attacks. Threshold implementation (TI) [12,13], a well-established masking technique, splits the input data into two or more shares, and this number is determined by the order of DPA security. TI works successfully when three properties, namely correctness, uniformity, and non-completeness, are satisfied. Achieving uniformity requires fresh randomness; hence, a single 8-bit S-box calculation requires a minimum of 32 bits [14] and a maximum of 64 bits [15] of randomness, apart from the initial random numbers required for input and key masking. Thus, it is not suitable for applications where random number generators cannot be easily integrated, especially in a resource-constrained environment like the Internet of Things [16].

The hiding approach is well defined by the dual-rail pre-charge logic (DPL) at the gate level, in which the data are processed in two phases, namely pre-charge and evaluation. In this method, the data is pre-charged to (0, 0) and evaluated either to (0, 1) for a logic "0" or (1, 0) for a logic "1." Thereby, the number of transitions is always constant, thwarting HW and HD-based DPA. Many variants of the DPL have been developed, for example, wave dynamic differential logic (WDDL) [7,8] and balanced DPL (BCDL) [6]. However, the very large scale integration (VLSI) implementation of DPL at the gate level is tedious, because it requires a good understanding of the vulnerable circuit architecture that makes it susceptible to more design defects. Power balancing circuits [17,18] developed at the algorithmic level add extra logic cells to equalize power consumption, following the same concept as that of DPL logic. The approaches reported in [17,18] have very large area overheads of 6.5× and 8×, respectively.

Yet, another category of hiding countermeasures is data encoding logic-based power equalization for hardware and software implementations. In hardware implementations, 1-of-2 and 1-of-4 based adder cells are developed at transistor level in [10], where the resistance is shown using simulation traces against HW-based DPA. In addition, a recent work combines HW equalization logic with the masking technique to show resistance against HW-based DPA [11]. In software implementations, the work quoted in [19] was the first to adopt the DPL concept of encoding the actual data along with its complementary form to achieve HW equalization. This idea was further developed by Chen and colleagues in [20], in which he suggested two encoding techniques to address both HW and HD-based DPA. The first technique was to append the complementary data along with the original data and the second was to use another encoding format that is specific to the PRINCE cipher. However, the re-ordering layer used to change between the two encoding formats showed first-order chosen plain-text attack (CPA) leakage [20]. A 3-of-6 code-based data encoding countermeasure was proposed for AES cipher in [21]. Both approaches [20,21] have been analyzed in [22] and were found to have prominent leakage to bit-based correlation attacks. Pour and colleagues in [23] proposed 2-of-4 encoding for use in three sets for HW and HD equalization for a SIMON cipher implementation on the microprocessor. The authors mentioned that they have not considered the status flag of the registers, which might result in leakage, and there is no explanation of how the data were switched between the three sets.

To summarize, though the masking countermeasure shows resistance to HW and HD-based DPA attacks, it requires a huge amount of randomness [12–15] and is specific to cryptographic implementations [2–5]. Among the hiding countermeasures, DPL technique [6–11] has a high implementation complexity, whereas the algorithmic power balancing technique [17,18] has a high area overhead. Data encoding techniques provide a good scope for achieving power equalization. Table 1 presents a summary of the proposed work and the existing ones in literature.

3 | PROPOSED WORK

3.1 | Attack scenario

DPA, in general, can be executed from two perspectives: CPA—when the attacker has knowledge of both the

TABLE 1 Summary of DPA countermeasures

Technique		Pros	Cons
Masking	Boolean and arithmetic masking [2–5]	Power consumption is independent of the sensitive data processed.	Leaks side-channel information through glitches. Vulnerable to higher order attacks.
	TI [12–15]	Does not leak information through glitches. Protected against probing adversary model.	Needs large randomness. Vulnerable to higher order attacks.
Hiding	DPL [6–11]	Transistor level technique that can be easily integrated with the standard design flow.	Requires complicated physical design process for achieving balanced routing.
	Software data encoding [19–23]	Algorithm level technique that can be applicable for any design.	Shows leakage, as illustrated in subsequent work.
	Proposed technique	Achieves HW and HD equalization at gate-level. Applicable for any design.	Additional area required for realizing the encoding functionality when compared with the unprotected implementation.

plain-text and cipher-text—and known cipher-text attack (KCA)—when the attacker has knowledge of the cipher-text only. Based on these perspectives, for the symmetric cryptographic cipher implementation, the attacker can form the hypothetical power consumption or leakage model based on the following: (i) The HW of the data processed—the output of the S-box in the case of symmetric cryptographic implementations. For a KCA scenario, the power can be formulated as the HW at the round register output, typically the final round. In the case of CPA, the model can be formed as the HW at the first round S-box output. The permutation layer is typically ignored. (ii) The HD between the data processed during two time intervals, which are calculated as the difference between two intermediate computation outputs. For a KCA scenario, the model can be computed as a) the HD between the final and pre-final round outputs and b) the HD between the cipher-text outputs of the k^{th} and $(k\text{-}1)^{th}$ encryptions. For a CPA case, the model is computed as the HD between the input and first round output.

It is important to consider the possible attack scenarios that depend on an adversary's capabilities without missing any loop-holes in the implementation. Hence, this work equalizes both the HW and HD of the data processed with respect to both KCA and CPA scenarios such that the implementation is resistant to DPA.

3.2 | Proposed 1-of-4 encoding

Data encoding techniques, namely *m*-of-*n* coding, were proposed for DPA resistance in hardware by [10,11]. These coding techniques, when chosen and designed appropriately, achieve balanced power consumption that can be leveraged against DPA. Among the various *m*-of-*n* encodings, 1-of-4 encoding is chosen in this work for equalizing HW and HD. The reason for the particular choice of this code is because this is one of the lowest weight choices with four equidistant code words that can be grouped into two sets. While the lower weight codes, namely 1-of-2, 1-of-3, and 2-of-3 cannot produce four code words, 2-of-4 codes produce six codewords that are not equidistant, namely 1100, 0110, 0011, 1001, 1010, and 0101. Another choice with properties similar to those of 1-of-4 codes is 3-of-4 codes, and its hardware implementation is complementary to the 1-of-4 realization. Hence, the XOR and AND gates used in the datapath must be replaced with XNOR and NAND gates, the area of which is larger. The other values of n > 4 are not analyzed since the size of the datapath increases linearly with *n*.

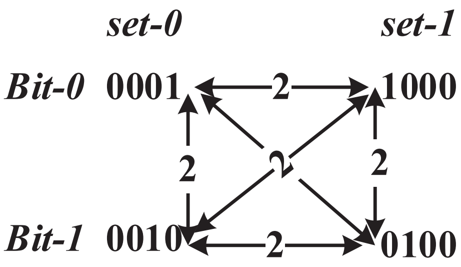


FIGURE 1 Proposed 1-of-4 data encoding concept

Using 1-of-4 encoding, the input Boolean data set {0, 1} can be mapped to four code words, namely 1000, 0100, 0010, and 0001. The code words are grouped in two sets, chosen by the "select" signal. In the first set (set 0)—select = 0: bit-0 is encoded as "0001" and bit-1 is encoded as "0010"; in the second set (set 1)—select = 1: bit-0 is encoded as "1000" and bit-1 is encoded as "0100," as shown in Figure 1. The reason for such a choice is so that the intra- and inter-HD between any encoding pair is equal (= 2) and the HW of each of the encoding patterns (= 1) is the same. This property helps achieve the required resistance against the leakage models. Let d be a data vector of length *n* whose value belongs to the Boolean data set {0, 1}. The proposed 1-of-4 encoding quadruples the size of the data vector d from *n*-bits to 4*n*-bits. The encoded data vector is denoted as d′.

The first step in implementing the countermeasure is to identify typical attack targets $v = \{v_{1, \ldots,} v_n\}$ of the given design with respect to both HW and HD leakage models. For AES, the vulnerable attack targets are an XOR of the plain-text with the secret-key and encryption/decryption round outputs. Because both CPA and KCA scenarios are considered, the attack targets for the HW-based leakage model (HWM) are v_1: XOR (plain-text, key) denoted as R0, and v_2 through v_{11}: round outputs (excluding the permutation layer, namely the Mix Columns), denoted as R1 through R10. For the HD-based model (HDM), the vulnerable attack targets are v_1 through v_{10}: the difference between the subsequent round outputs from R0 to R10. The countermeasure implementation should be such that it should eliminate the HW and HD based leakage for each v_i. This can be achieved with a suitable choice of encoding sets, which is explained as follows:

- If d′(t) represents the data processed at time t, encoding the data using set-0/set-1 ensures that $\mathrm{HW}(\mathrm{d}'\,(t)) = n$.
- If $\mathrm{d}'(v_i)$ denotes the data processed for an attack target v_i (for instance, the pre-final round operation of AES) and $\mathrm{d}'(v_j)$ represents the data processed for an attack target v_j (final round output of AES), $\mathrm{HD}(\mathrm{d}'(v_i), \mathrm{d}'(v_j)) = 2n$. This is achieved by encoding the data processed using set-0 and set-1 alternatively for each vulnerable computation, which is the round outputs in-case of AES, as shown in Figure 2. For a typical AES-128 bit scenario, the select signal and other control signal values of one complete encryption/decryption are shown in Figure 3.
- If d′(k-1) denotes the output cipher-text for a $(k-1)^{\mathrm{th}}$ encryption and d′(k) denotes the cipher-text output for a k^{th} encryption, then $\mathrm{HD}(\mathrm{d}'(k-1), \mathrm{d}'(k)) = 2n$. The proposed data encoding is implemented such that the sequence of selection of the encoding sets is inverted for subsequent encryptions. For the k^{th} encryption, R0's select signal takes the inverted value of the previously assigned at $k-1^{\mathrm{th}}$ encryption and is depicted in Figure 4.

By employing the proposed countermeasure, the actual data always appear in the encoded form of either of the two sets in the datapath. Hence, the proposed 1-of-4 data encoding technique equalizes both the HW and HD at every round computation output, which resists the attacks based on HW (i) and HD (ii)-a models described in Section 3.1. Moreover, the HD equalization is achieved between subsequent cipher-texts restricting model (ii)-b. Therefore, the proposed technique resists the leakage models from the CPA and KCA scenarios for all the vulnerable attack targets with respect to HW- and HD-based leakages.

In this work, the targets identified are the XOR output of the input and secret key as well as the XOR output of the round outputs and output between subsequent encryptions. Hence, encoding sets are switched accordingly. Similarly, for other designs, the decision of when to switch between the encoding sets should be made based on the selected design and the identified vulnerable target based on the adversary's capabilities. Typically, the vulnerable target for block ciphers is the non-linear computation between the input data and secret key.

3.3 | Correctness

For any cryptographic function, f with n-bit plain-text represented as $\mathrm{PT}\{0,1\}^n$, key as $K\{0,1\}^n$, and output cipher-text represented as $\mathrm{CT}\ \{0,1\}^n$ is defined by (1).

$$\mathrm{CT}\{0,1\}^n = f(\mathrm{PT}\{0,1\}^n, K\{0,1\}^n) \tag{1}$$

Each of the n-bit inputs PT and K are encoded using the proposed 1-of-4 encoding set-0/set-1 into $4n$-bit word, represented by an encoding function Enc() described in (2).

$$f(\mathrm{Enc}(\mathrm{PT}\{0,1\}^{4n}), \mathrm{Enc}(\mathrm{Key}\{0,1\}^{4n})) = \mathrm{CT}\{0,1\}^{4n} \tag{2}$$

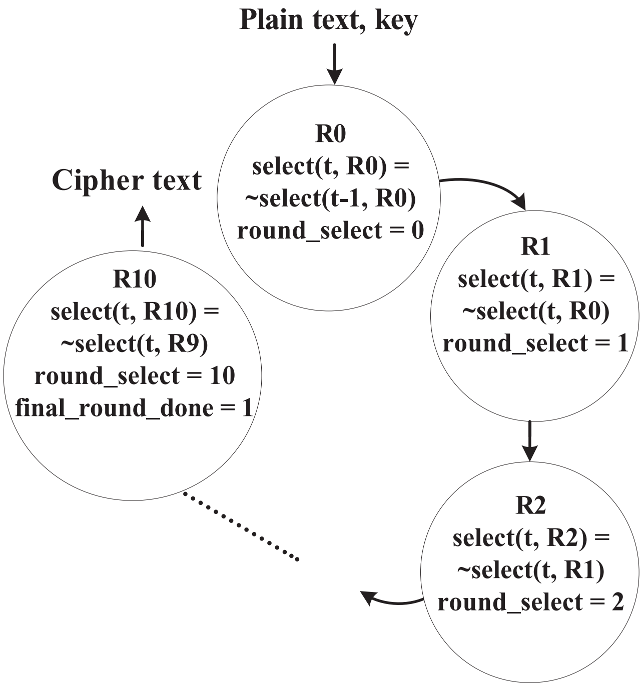


FIGURE 3 Control signals of a single AES encryption/decryption

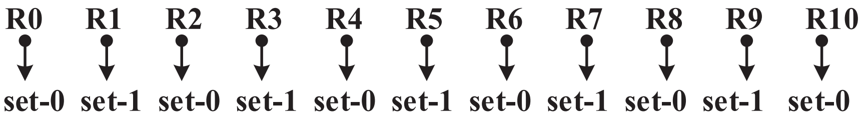


FIGURE 2 Sequence of encoding sets for a single AES encryption/decryption

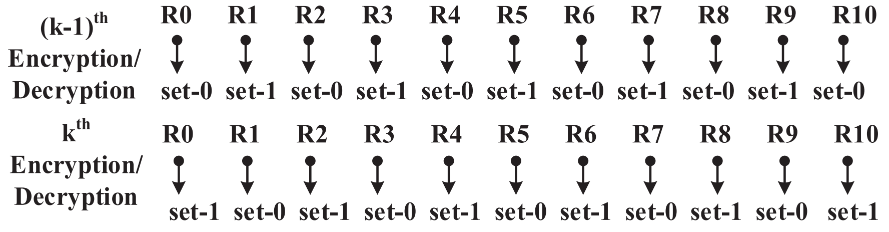


FIGURE 4 Sequence of encoding sets between two AES encryptions

TABLE 2 Energy of AND gate—Protected and unprotected

a	b	y_{up}	y_p	E_{up}	E_p
0 → 0	0 → 0	0 → 0	0001 → 1000	$E_{0→0}$	$E_{0→1} + E_{1→0} + 2.E_{0→0}$
0 → 0	0 → 1	0 → 0	0001 → 1000	$E_{0→0}$	$E_{0→1} + E_{1→0} + 2.E_{0→0}$
0 → 0	1 → 0	0 → 0	0001 → 1000	$E_{0→0}$	$E_{0→1} + E_{1→0} + 2.E_{0→0}$
0 → 0	1 → 1	0 → 0	0001 → 1000	$E_{0→0}$	$E_{0→1} + E_{1→0} + 2.E_{0→0}$
0 → 1	0 → 0	0 → 0	0001 → 1000	$E_{0→0}$	$E_{0→1} + E_{1→0} + 2.E_{0→0}$
0 → 1	0 → 1	0 → 1	0001 → 0100	$E_{0→1}$	$E_{0→1} + E_{1→0} + 2.E_{0→0}$
0 → 1	1 → 0	0 → 0	0001 → 1000	$E_{0→0}$	$E_{0→1} + E_{1→0} + 2.E_{0→0}$
0 → 1	1 → 1	0 → 1	0001 → 0100	$E_{0→1}$	$E_{0→1} + E_{1→0} + 2.E_{0→0}$
1 → 0	0 → 0	0 → 0	0001 → 1000	$E_{0→0}$	$E_{0→1} + E_{1→0} + 2.E_{0→0}$
1 → 0	0 → 1	0 → 0	0001 → 1000	$E_{0→0}$	$E_{0→1} + E_{1→0} + 2.E_{0→0}$
1 → 0	1 → 0	1 → 0	0010 → 1000	$E_{1→0}$	$E_{0→1} + E_{1→0} + 2.E_{0→0}$
1 → 0	1 → 1	1 → 0	0010 → 1000	$E_{1→0}$	$E_{0→1} + E_{1→0} + 2.E_{0→0}$
1 → 1	0 → 0	0 → 0	0001 → 1000	$E_{0→0}$	$E_{0→1} + E_{1→0} + 2.E_{0→0}$
1 → 1	0 → 1	0 → 1	0001 → 0100	$E_{0→1}$	$E_{0→1} + E_{1→0} + 2.E_{0→0}$
1 → 1	1 → 0	1 → 0	0010 → 1000	$E_{1→0}$	$E_{0→1} + E_{1→0} + 2.E_{0→0}$
1 → 1	1 → 1	1 → 1	0010 → 0100	$E_{1→1}$	$E_{0→1} + E_{1→0} + 2.E_{0→0}$

The proposed data encoding or concealing function Enc() is said to be correct since the decoded or revealed (Dec() function) output represents the output of the original function, as shown in (3).

$$\mathrm{Dec}(\mathrm{CT}\{0,1\}^{4n}) = \mathrm{CT}\{0,1\}^{n} \tag{3}$$

3.4 | Security proof

Table 2 shows the energy consumption of a 2-input AND gate. The signals y_{up} and E_{up} show the output transition and corresponding energy consumption of the unprotected gate. Based on Ref [24] the output energy that is directly proportional to the power consumption for an output state q at logic 0 and logic 1 is given as

$$E_{\mathrm{up}}(q=0) = \frac{9E_{0\to0} + 3E_{1\to0}}{12}, \tag{4}$$

$$E_{\mathrm{up}}(q=1) = \frac{3E_{0\to1} + E_{1\to1}}{4}. \tag{5}$$

From (4) and (5), it is observed that $E(q=0) \neq E(q=1)$ and DPA exploits the same to perform the attack. By applying the proposed technique, the output transitions of the modified AND gate are shown in Table 2 as y_{p}. In this case, it is assumed that at time t, data are encoded using set-0 and at time $t+1$, data are encoded using set-1. For the protected case, signals a and b represent the logic values of the input transitions. The output energy E_{p} is constant for all the input transitions. Similar expressions are obtained when set-0 and set-1 are swapped.

Further, when the energy equations are substituted similarly as in (4) and (5),

$$\begin{aligned} E_{\mathrm{p}}(q=0) &= \frac{9(E_{0\to1}+E_{1\to0}+2E_{0\to0})+3(E_{0\to1}+E_{1\to0}+2E_{0\to0})}{12} \\ &= \frac{12E_{0\to1}+12E_{1\to0}+24E_{0\to0}}{12} \\ &= E_{0\to1}+E_{1\to0}+2E_{0\to0,} \end{aligned} \tag{6}$$

$$\begin{aligned} E_{\mathrm{p}}(q=1) &= \frac{3(E_{0\to1}+E_{1\to0}+2E_{0\to0})+(E_{0\to1}+E_{1\to0}+2E_{0\to0})}{4} \\ &= \frac{4E_{0\to1}+4E_{1\to0}+8E_{0\to0}}{4} \\ &= E_{0\to1}+E_{1\to0}+2E_{0\to0}. \end{aligned} \tag{7}$$

From (6) and (7), it is found that $E(q=0) = E(q=1)$ and DPA attacks do not work based on both HW- and HD-based models. Similarly, the expressions can be derived for other gates. Therefore, any vulnerable cryptographic and non-cryptographic circuits can achieve equal power for all the input transitions, thereby thwarting DPA.

4 | IMPLEMENTATION

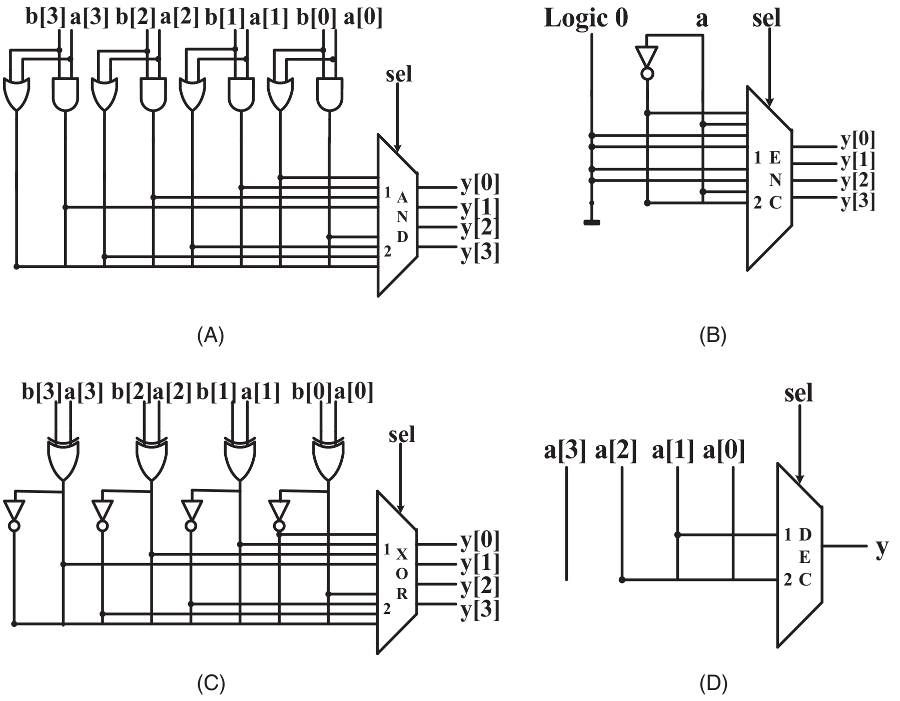
The implementation of the proposed technique requires the following components: (i) Encoders—This block encodes the incoming data to set-0 or set-1 code words based on the select input (abbreviated as "sel" in the figures). (ii) Decoders—This block decodes the actual cipher-text output after the final round based on the select input. (iii) Re-mappers—This block is essential to convert the encoded data from set-0 to set-1 and vice versa. The re-mappers can be implemented using the wiring technique and do not require extra hardware.

The proposed data encoding technique was evaluated for AES-128 cipher with a round and column folded S-box reuse based architecture proposed in [25]. Since the datapath is a gate-level architecture, the components required are (i) XOR gates, (ii) AND gates, (iii) Encoders, (iv) Decoders, and (iv) Re-mappers. The XOR gates and AND gates are modified to process two four-bit inputs and produce one four-bit output based on the select input. The select input is used to choose between two data encoding sets, set-0 (select = 0) and set-1 (select = 1). The modified implementation is based on the expressions given in Table 3. The hardware realization of the same using multiplexers is shown in Figure 5.

Figure 6 shows the implementation of the proposed countermeasure on the AES datapath architecture. It can be seen that the datapath of the proposed architecture is not altered; only the width is quadrupled. The required encoders, decoders, and re-mapping units are inserted at the appropriate places. The input plain-text and key are encoded at the start of each encryption. Re-mappers are employed to change the encoding set from one to another. Each round computation uses S-box, Shift Rows, Mix Columns, and Add

TABLE 3 Implementation of the proposed technique

Sel i/p	Bit 0	Bit 1	Modified XOR o/p	Modified AND o/p	Decoder	Encoder	Re-mapper
0	0001	0010	y[3] = a[3] ⊕ b[3] y[2] = a[2] ⊕ b[2] y[1] = a[1] ⊕ b[1] y[0] = ~(a[0] ⊕ b[0])	y[3] = a[3] & b[3] y[2] = a[2] & b[2] y[1] = a[1] & b[1] y[0] = a[0] | b[0]	y = a[1]	y[3] = 1′b0 y[2] = 1′b0 y[1] = a y[0] = ~a	y[3] = a[0] y[2] = a[1] y[1] = a[2] y[0] = a[3]
1	1000	0100	y[3] = ~(a[3] ⊕ b[3]) y[2] = a[2] ⊕ b[2] y[1] = a[1] ⊕ b[1] y[0] = a[0] ⊕ b[0]	y[3] = a[3] | b[3] y[2] = a[2] & b[2] y[1] = a[1] & b[1] y[0] = a[0] & b[0]	y = a[2]	y[3] = ~a y[2] = a y[1] = 1′b0 y[0] = 1′b0	y[3] = a[0] y[2] = a[1] y[1] = a[2] y[0] = a[3]



(A)

(B)

(C)

(D)

FIGURE 5 Hardware realization of components using 2 × 1 4-bit multiplexers: (A) AND gate, (B) Encoder, (C) XOR gate, and (D) Decoder. The "sel" signal acts as the multiplexer's selection input

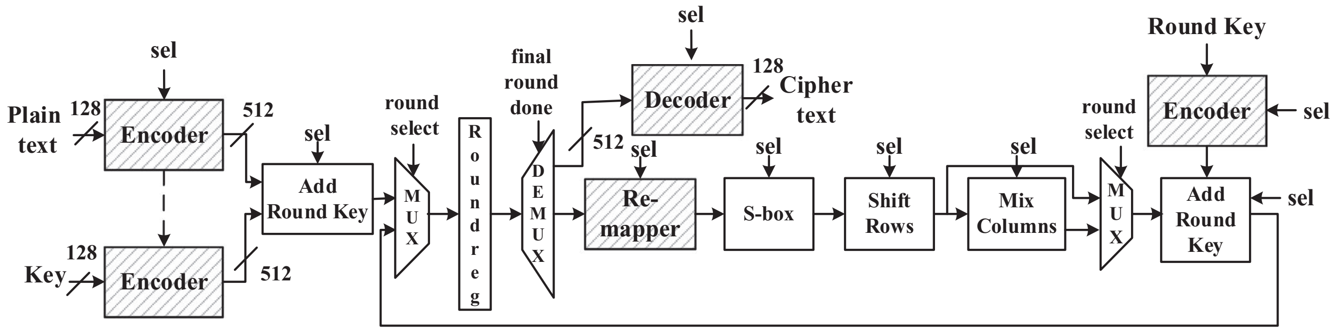


FIGURE 6 AES datapath with the proposed countermeasure

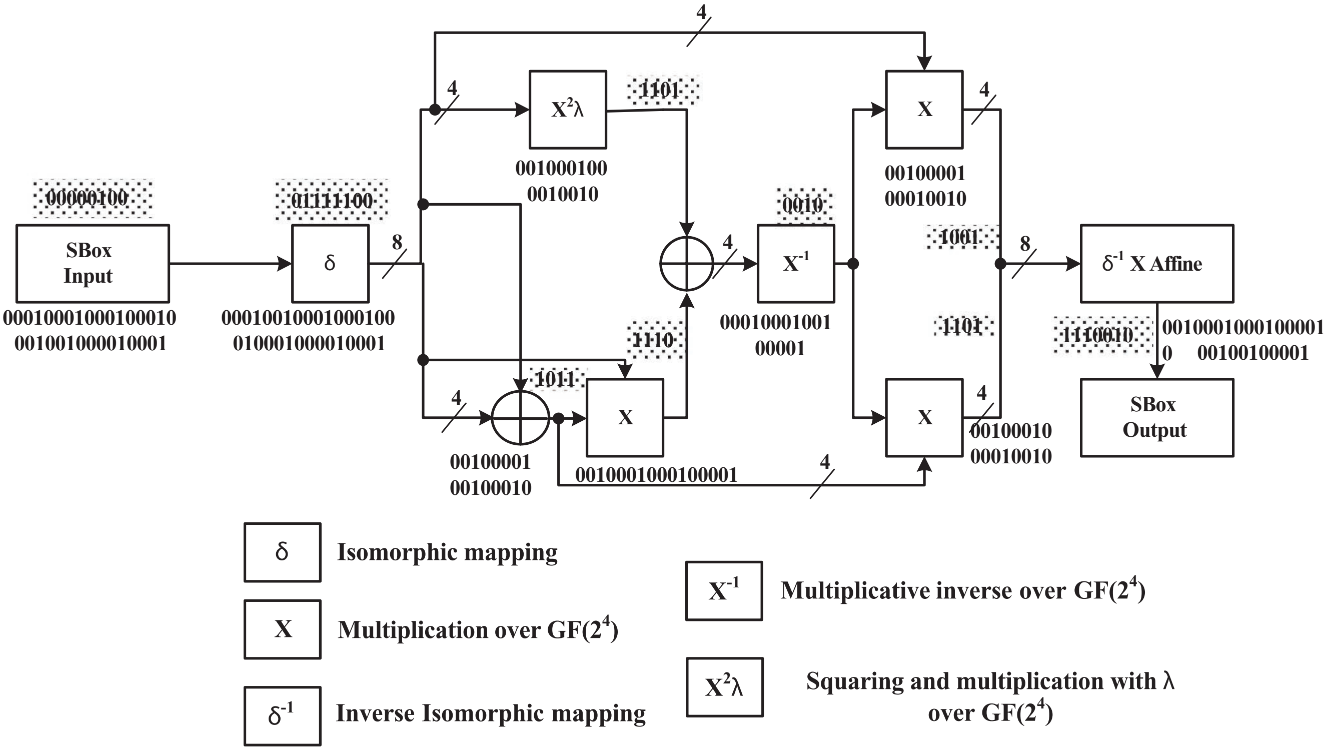


FIGURE 7 CFA based S-box architecture. The highlighted values are the actual data and the unhighlighted values are the encoded data using select signal "0"

Round Key blocks. Since a CFA-based S-box is chosen for our proposed implementation, the designed XOR and AND gates realize the functionality of the modified S-box. The input of the S-box appears in encoded form, the set of which is chosen at every round input. The CFA-based S-box performs the required computation and the output appears in a similar encoded form, as shown in Figure 7. Shift Rows is a wiring operation and does not need any hardware. The other blocks, namely Mix Columns and Add Round Key, also perform their functionality on the encoded data using the modified gates. For the 128 bit AES cipher, the datapath of the first nine rounds of encryption use all of the modified blocks while the tenth round skips the Mix Columns block. Finally, the encoded cipher-text is decoded to obtain the actual output. The select signal needs to be propagated throughout the datapath to choose between the two encoding sets.

The proposed technique is not restricted to a specific cryptographic architecture because it is generic and can be implemented for any gate-level design. Further, the proposed countermeasure implementation is straightforward and requires less design effort, unlike the full custom design flows like WDDL. Once the basic components of the architecture are designed, it is re-usable across multiple designs. Moreover, the proposed technique does not require any randomness for DPA resistance.

4.1 | Performance comparison

The protected and unprotected implementations are coded in Verilog hardware description language. Table 4 shows the performance comparison of the proposed architecture

TABLE 4 Performance comparison on Virtex-5 FPGA

		Area		Performance (Unprotected /Protected)		
C/m Category	**Design**	**Unprotected**	**Protected**	**Fmax (MHz)**	**T/P (Mbps)**	**Area over head**
Masking	Masking [6]	733 Slices 8 BRAMs	856 Slices 16 BRAMs	144.3/141.1	NA	L:1.16xM:2x
	TI[a] [12]	1890 LUTs 397 Regs	12 627 LUTs 1438 Regs	138.9/78.6	1778/503	L:6.7x
Hiding	WDDL[a] [7]	1890 LUTs 397 Regs	7292 LUTs 1160 Regs	138.9/56.3	1778/360	L:3.85x
	BCDL [6]	176 Slices 8 BRAMs	1128 Slices 16 BRAMs	258/283	NA	L:6.4xM:2x
	Ours	303 Slices 640 LUTs 743 Regs	692 Slices 1497 LUTs 1711 Regs	388.39	207.14	L:2.3x

Abbreviations: L, Logic, M, Memory.

[a] FPGA results of TI [12] and WDDL [7] works reported in [4] is quoted.

with the existing state-of-the-art countermeasures. For comparison purposes, the design was synthesized on an Virtex-5 xc5vlx50 field programmable gate array (FPGA) using Xilinx ISE 14.7 and the functionality was verified using NIST certified test vectors in ModelSim Altera 6.4a.

There is no change in the critical path delay between the protected and unprotected designs since there is no alteration in the datapath. Hence, the throughput remains the same. There is about a 2.3× increase in area in the protected design, and this can be attributed to the increase in the width of the datapath from n to $4n$. Accordingly, the number of registers and logic components quadruples, leading to the area increase. Even though there is about a 2.3× increase, the developed countermeasure on the AES occupies the least area when compared with the other techniques in the literature except for the masking work in [6]. In the reported masking implementation [6], there is a memory overhead of 2× along with a logic overhead of 1.16×. The proposed countermeasure does not use any block random access memories (BRAMs), and hence, the design is portable to an application-specific integrated circuit (ASIC) realization.

For the proposed technique, the area overhead is a compromise for obtaining the DPA security benefit. For any DPA countermeasure, an increase in the area when compared to an unprotected implementation is inevitable.

5 | SECURITY ANALYSIS

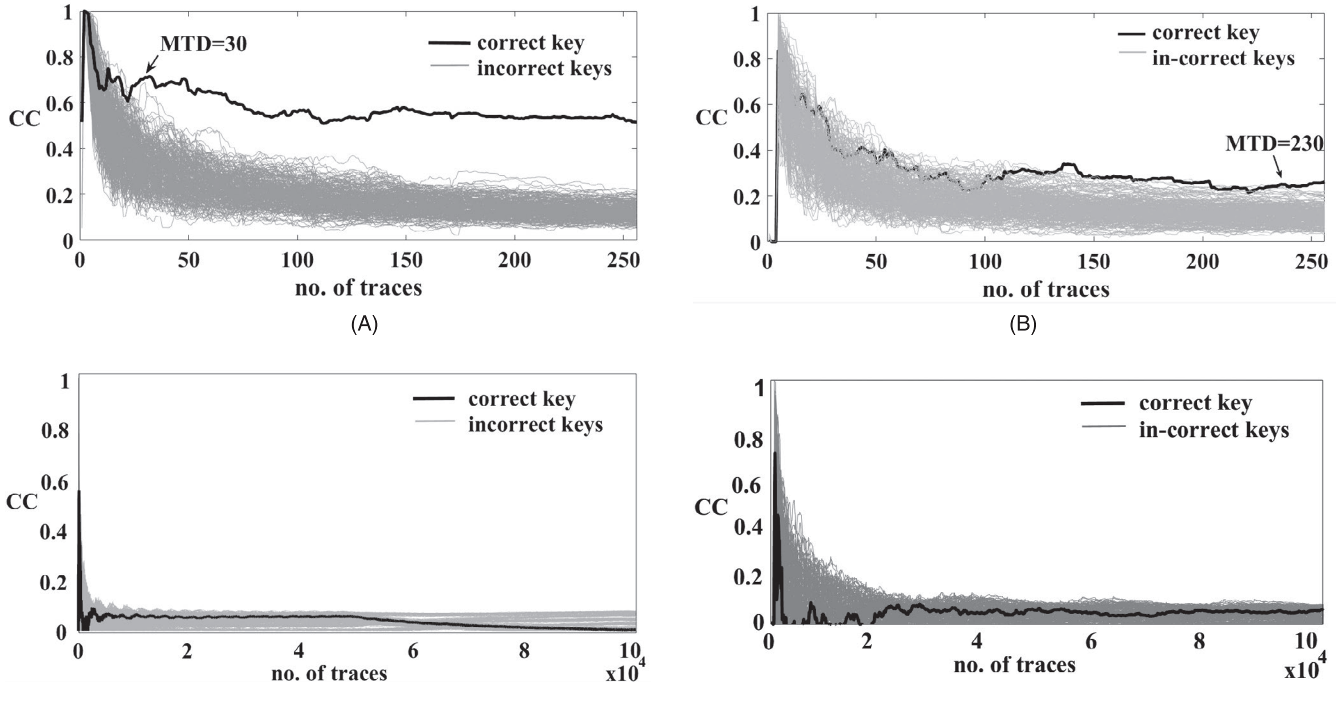
The proposed countermeasure technique can resist HWM- and HDM-based correlation power analysis attacks (XCPA), mutual information analysis (MIA), specific test vector leakage assessment (TVLA), and bit-model based DPA attacks in the KCA and CPA scenarios. In this section, the DPA resistance of the CPA scenario is shown. In a similar way, the KCA scenario can be proved. A total of 100 000 traces were recorded for the unprotected and proposed protected design from the SAKURA-G FPGA board connected with a computer and Keysight MSO 3104 T. The traces were preprocessed using Python, and the attack scripts were executed in MATLAB.

5.1 | XCPA

The XCPA attack calculates Pearson's correlation coefficient (CC) to estimate the linear relationship between the leakage model and the actual power consumption. Figure 8A,B show the minimum time to disclosure (MTD) plots of the unprotected design attacked using HWM and HDM, respectively. MTD estimates the minimum number of traces to extract the correct key as 30/230 on an unprotected implementation using HWM/HDM, respectively. Figure 8C,D show the MTD of the protected design. For the protected implementation, the correct key has not been revealed as of 100 000 traces because the HW and HD of the protected implementation are constant and do not vary with the input.

5.2 | DPA

A DPA attack works on bit models. On the calculated S-box output, a particular bit is chosen and allotted as the target bit. In this work, the most significant bit (MSB) of the AES S-box output bit is selected. DPA works on the assumption that a bit 0 takes a different power than bit 1. In the proposed technique, approximately the same power is



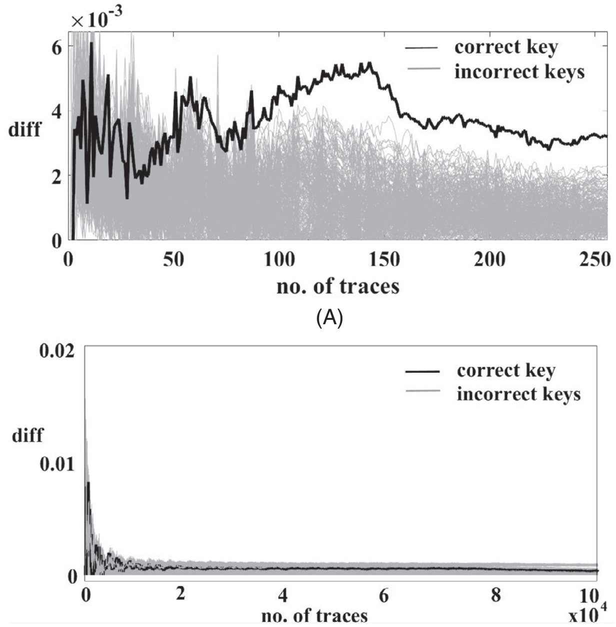
(A)

(B)

(C)

(D)

FIGURE 8 CC—MTD plot: (A) unprotected—HWM, (B) unprotected—HDM, (C) protected—HWM, and (D) protected—HDM



(A)

(B)

FIGURE 9 DPA plot: (A) unprotected and (B) protected

consumed for each input, which makes it resistant to DPA. While the unprotected implementation reveals the correct key's difference (diff) peak value as 0.47 with less than 256 traces, as shown in Figure 9A the protected implementation has a negligible difference value for all the key bytes in the range of 0 to 0.02 for 100 000 traces, as shown in Figure 9B.

5.3 | MIA

The idea of using information theory metrics to quantify the dependence between the leakage model and the power consumption was proposed as MIA. The advantage of this method is that it relaxes the strict requirement of linear dependency between the measurements and predictions as in CPA and exploits more information than DPA. In an unprotected implementation, the mutual information (MI) calculation returns the correct key with 310/640 traces for HWM/HDM-based leakage variables, respectively, as shown in Figure 10A,B. Since the protected implementation equalizes the power consumption for both HWM and HDM, MIA was not successful in extracting the correct key as of 100 000 traces, as seen in Figure 10C,D, respectively.

5.4 | TVLA

The specific t test evaluation methodology classifies the power traces based on one particular bit (MSB in this work) of the value of the HW/HD in the correct key's S-box output. This work evaluates the same, and Figure 11A,B shows the t-values of the unprotected and protected

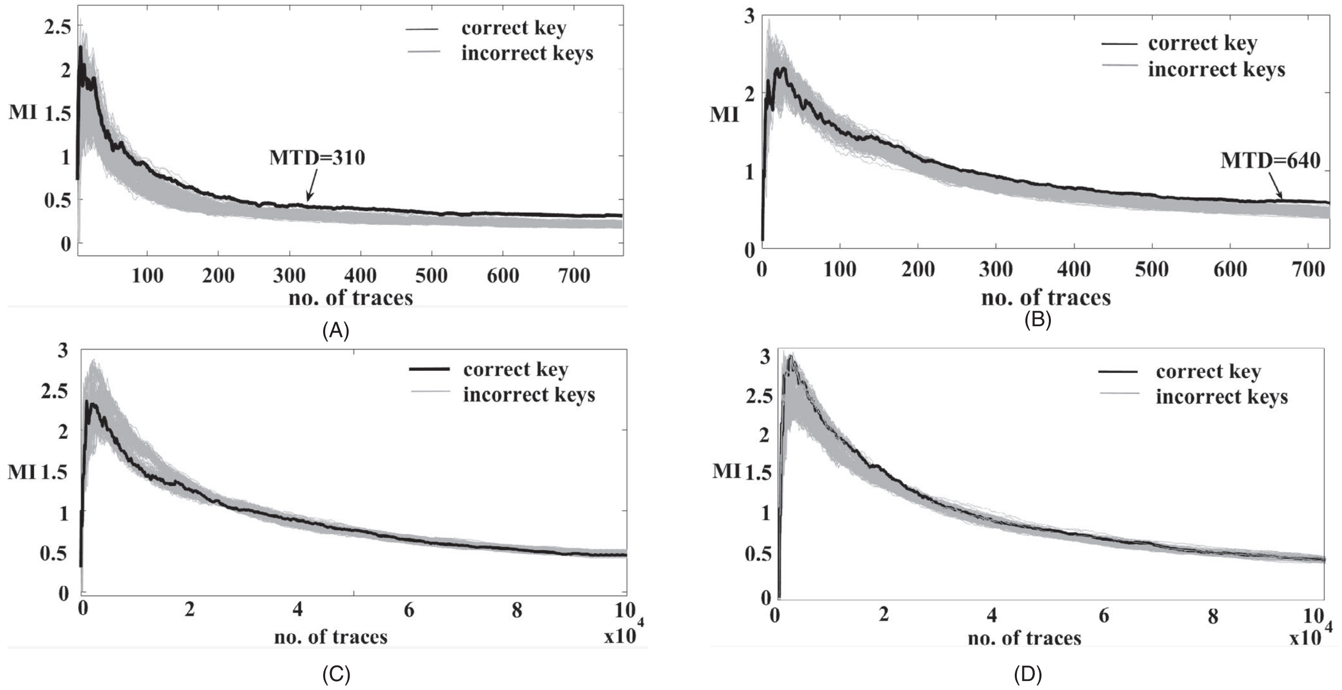


FIGURE 10 MIA plot: (A) unprotected—HWM, (B) unprotected—HDM, (C) protected—HWM, and (D) protected—HDM

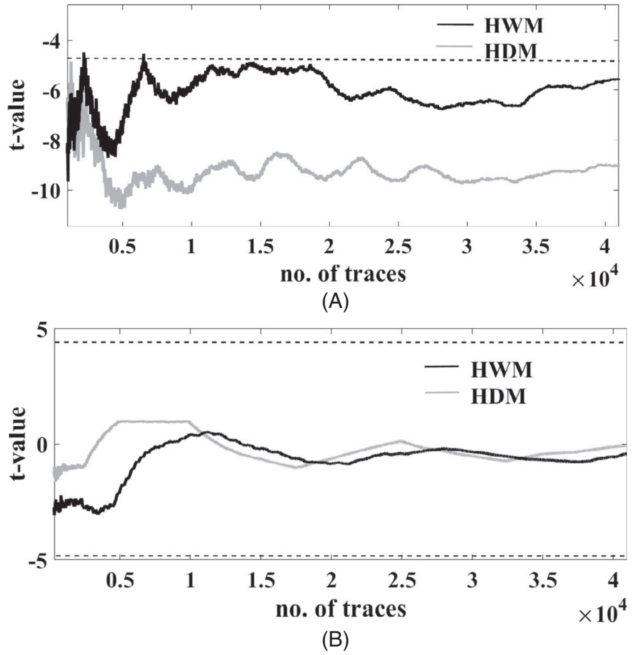


FIGURE 11 TVLA plot: (A) unprotected—HWM and HDM and (B) protected—HWM and HDM

implementations, respectively. It can be seen that whereas the t-value of the unprotected implementation exceeds the threshold, the protected implementation lies within the bounds of ± 4.5. Since the 100 000 traces are grouped into two bins based on the bit-model for t-value calculation, the number of traces halves in the x-axis of the plots in Figure 11A,B.

Hence, it can be observed that the protected implementation shows sufficient first-order resistance to DPA at 100 000 traces.

6 | CONCLUSION

This work explored a novel 1-of-4 data encoding method for resisting DPA attacks in hardware. The data processed in a block cipher are concealed using one of the two encoding sets alternatively between each round computation and consequent encryption/decryption. The blocks required for realizing the functionality of the proposed countermeasure, namely the encoder, decoder, and re-mappers were designed. They were further integrated with the datapath elements of the AES, namely the S-box, Mix Columns, and Add Round Key blocks, realized using modified XOR and AND gates. Thereby, the proposed technique achieves HW and HD equalization with a 2.3× area overhead on a Virtex-5 FPGA without any performance degradation. The DPA resistance was validated using 100 000 traces captured on the SAKURA-G platform.

The proposed technique is advantageous, because it requires less design effort, has zero randomness, and can be applied in any gate-level vulnerable hardware because the modified components are re-usable. The AES encryption circuit with the proposed countermeasure can be employed in any security-critical application in fields such as medicine, industry, and defense. Future work will analyze the security of the proposed encoding technique to other vulnerable non-cryptographic circuits such as machine learning (ML)-based implementations. Further, the security of the proposed technique for more than 100 000 traces will be evaluated.

ACKNOWLEDGMENT

This work was supported by the Visvesvaraya PhD Scheme of Ministry of Electronics and Information Technology (MeitY), Government of India (MEITY-PHD-1905). This work was also supported by Special Manpower Development Programme for Chips to System Design (SMDP-C2SD) project sponsored by the Department of Electronics and Information Technology (DeitY), Government of India.

ORCID

Saravanan Paramasivam https://orcid.org/0000-0003-0785-807X

REFERENCES

1. P. Fips, Advanced encryption standard (AES), National Institute of Standards and Technology, US Department of Commerce, Gaithersburg, MD, 2001.
2. F. Regazzoni, Y. Wang, and F.-X. Standaert, *FPGA implementations of the AES masked against power analysis attacks*, in Proc. Constr. Side-Channel Anal. Secur. Des. 2011, pp. 56–66.
3. J.-S. Coron, J. Großschädl, and P. Kumar Vadnala, *Secure conversion between boolean and arithmetic masking of any order*, in Proc. Int. Workshop Cryptogr. Hardware Embed. Syst. (Busan, South Korea), Sept. 2014, 188–205.
4. A.-T. Hoang and T. Fujino, *Intra-masking dual-rail memory on LUT implementation for SCA-resistant AES on FPGA*, ACM Trans, Reconfig. Tech. Syst. **7** (2014), 1–19.
5. M. Masoumi, *A highly efficient and secure hardware implementation of the advanced encryption standard*, J. Inf. Sec. App. **48** (2019), 102371.
6. S. Bhasin et al., *Exploiting FPGA block memories for protected cryptographic implementations*, ACM Trans. Reconfig. Tech. Sys. **8** (2015), 1–16.
7. K. Tiri and I. Verbauwhede, *A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation*, in Proc. Design, Autom. Test Europe Conf. Expo. (Paris, France), Feb. 2004, pp. 246–251.
8. X. Fang et al., *Balance power leakage to fight against side-channel analysis at gate level in FPGAs*, in Proc. IEEE Int. Conf. Application-specific Syst., Archit. Processors (ASAP), (Toronto, ON, Canada), July 2015, pp. 154–155.
9. D. Bellizia et al., *SC-DDPL: A novel standard-cell based approach for counteracting power analysis attacks in the presence of unbalanced routing*, IEEE Trans. Circuits Sys. Reg. Pap. **67** (2020), 2317–2330.
10. F. Burns et al, *Security evaluation of balanced 1-of-n circuits*, IEEE Trans. Very Large Scale Integ. Sys. **19** (2010), no. 11, 2135–2139.
11. X. Li et al., *Energy-efficient side-channel attack countermeasure with awareness and hybrid configuration based on it*, IEEE Trans. Very Large Scale Integ. Syst. **25** (2017), 3355–3368.
12. S. Nikova, C. Rechberger, and V. Rijmen, *Threshold implementations against side-channel attacks and glitches*, in Proc. Int. Conf. Inform. Commun. Sec. (Raleigh, NC, USA), Dec. 2006, pp. 529–545.
13. T. De Cnudde et al., *Masking AES with d+1 shares in hardware*, in Proc. Int. Conf. Cryptogr. Hardware Embed. Syst. (Santa Barbara, CA, USA), Aug. 2016, pp. 194–212.
14. B. Bilgin et al., *Tradeoffs for threshold implementations illustrated on AES*, IEEE Trans. Comput. Aid Des. Integr. Circ. Syst. **34** (2015), 1188–1200.
15. R. Ueno, N. Homma, and T. Aoki, *Toward more efficient DPA-resistant AES hardware architecture based on threshold implementation*, in Proc. Int. Workshop Constr. Side-Channel Anal. Secur. Des. (Paris, France), Apr. 2017, pp. 50–64.
16. A. Mosenia and N. K. Jha, *A comprehensive study of security of internet-of-things*, IEEE Trans. Em. Top. Comp. **5** (2016), 586–602.
17. D. Jayasinghe et al., *Quadseal: Quadruple algorithmic symmetrizing countermeasure against power based side-channel attacks*, in Proc. Int. Conf. Compilers, Archit. Synth. Embed. Syst. (CASES), (Amsterdam, Netherlands), Oct. 2015, pp. 21–30.
18. D. Jayasinghe, A. Ignjatovic, and S. Parameswaran, *NORA: Algorithmic balancing without pre-charge to thwart power analysis attacks*, in Proc. Int. Conf. VLSI Des. Embed. Sys. (Hyderabad, India), Jan. 2017, pp. 167–172.
19. P. Hoogvorst, J.-L. Danger, and G. Duc, *Software implementation of dual-rail representation*, in Proc. COSADE, (Darmstadt, Germany), 2011.
20. C. Chen et al., *Balanced encoding to mitigate power analysis: A case study*, in Proc. Int. Conf. Smart Card Res. Adv. Appl. (Montpellier, France), 2014, pp. 49–63.
21. V. Servant et al., *Study of a novel software constant weight implementation*, in Smart Card Research and Advanced Applications, Springer, vol. 8968, Paris, France, Mar. 2014, pp. 35–48.
22. Y.-S. Won et al., *Security of constant weight countermeasures*, ETRI J. **39** (2017), 417–427.
23. M. S. Pour and M. Salmasizadeh, *A new CPA resistant software implementation for symmetric ciphers with smoothed power consumption: SIMON case study*, ISC Int. J. Inform. Sec. **9** (2017), 119–130.
24. D. Mukhopadhyay and R. S. Chakraborty, Hardware security: Design, threats, and safeguards, CRC Press, Boca Raton, FL, 2014.
25. S. Shanthi Rekha and P. Saravanan, *Low-cost AES-128 implementation for edge devices in IoT applications*, J. Circuits, Syst. Comp. **28** (2019), 19500621–195006224.

AUTHOR BIOGRAPHIES

Shanthi Rekha Shanmugham received her BE degree in electronics and communication engineering (ECE) and ME degree in VLSI design from the PSG College of Technology, Coimbatore, Tamil Nadu, India, in 2010 and 2012, respectively. From 2012 to 2014, she was employed as physical verification engineer with QualComm India Private Limited, Bangalore, Karnataka, India. Since 2016, she has been a research scholar in the Visvesvaraya PhD Scheme at the Department of ECE at the PSG College of Technology. Her research interests include VLSI design, hardware security, and physical verification.

Saravanan Paramasivam received his BE degree in electrical and electronics engineering from Thiagarajar College of Engineering, Madurai, India, in 2007, his ME degree in VLSI design, and his PhD in hardware security from the PSG College of Technology, Coimbatore, India, and Anna University, Chennai, India, in 2009 and 2015, respectively. Since 2009, he has been with the Department of Electronics and Communication Engineering, PSG College of Technology, where he is now an associate professor. He has published more than 50 papers in various international and national journals and conferences. He is a member of the IETE, ISSS, and VLSI Society of India. His current research interests span hardware security, quantum computing and multi-scale modeling of nanoelectronic devices. His history also includes around five years of industrial experience.