Flexible and high-throughput structures of Camellia block cipher for security of the Internet of Things

Bahram Rashidi

Department of Electrical Engineering, Ayatollah Boroujerdi University, Boroujerd, Iran

Correspondence
Bahram Rashidi, Department of Electrical Engineering, Ayatollah Boroujerdi University, Boroujerd, 69199-69411, Iran.
Email: b.rashidi@abr.ac.ir

Abstract
The advancements in wireless communication have created exponential growth in the Internet of Things (IoT) systems. Security and privacy of the IoT systems are critical challenges in many data-sensitive applications. Herein, high-throughput and flexible hardware implementations of the Camellia block cipher for IoT applications are presented. In the proposed structures, sub-blocks of the ciphers are implemented based on optimised circuits. The proposed structures for Camellia are designed and shared for implementing the encryption process and generating some intermediate key values in the two separate times. The most complex block in these ciphers is the substitution box (S-box). The S-boxes are implemented based on area-optimised logic circuits. The Camellia S-boxes consist of a field inversion over \( F_{2^8} \) and two affine transformations over \( F_2 \). The inversion operation is implemented over the composite field \( F_{(2^3)^2} \) instead of an inversion over \( F_{2^8} \) which is an important factor to reduce area consumption. A large number of gates, in the structure, have been implemented by 2-input NAND and 2-input NOR gates to reduce delay and area. Also, the flexible structure for Camellia that can do various configurations of this cipher to support variable key sizes 128, 192 and 256 bits was proposed. Implementation results of the proposed architectures in 180 nm CMOS technology for different key sizes are achieved. The results show improvements in terms of execution time, throughput and throughput/area compared to the other related works.

1 | INTRODUCTION

The increasing number of services provided by the Internet has generated a huge increase in the number of connected devices. There are many Internet of Things (IoT) applications such as e-health, e-commerce, smart home, smart city, smart hospital, etc. [1]. The IoT is a network used to interconnect embedded devices, such as sensors, which can generate, communicate and share data [2]. In the IoT systems, we have many new security and privacy challenges [1,3]. With the exponential growth in IoT systems, security and privacy issues have emerged as critical challenges in many sensitive applications such as e-health and e-commerce. For example, the IoT in the e-health improve the quality of healthcare services as well as reduce the healthcare cost. The healthcare field is remodelling by the IoT to improve the social benefits by offering continuous monitoring of patients and services as well as updating healthcare medical records [4]. The healthcare data provide real sensitive information that should be protected [5]. Therefore, there is an urgent need to address these challenges. In recent years, many cryptographic algorithms have been used to provide the security of transmitted data in the IoT [6]. Cryptographic algorithms proposed for IoT applications can be categorised into the symmetric and asymmetric (public key) algorithms. Public key algorithms are time-consuming. Therefore, most of the applications in IoT networks use symmetric algorithms to provide security of the transmitted information [7,8]. Lightweight block ciphers as one of the most important symmetric algorithms are used to providing the security of IoT communication. These ciphers are easy to implement, use fewer computational resources with low overhead.

Many lightweight block ciphers have been proposed to reduce the costs of hardware consumption [9]. Block ciphers are used for data protection in the cryptosystems as a good
candidate for resource-constrained cryptographic applications. These cryptographic primitives have been the important area of cryptographic researches [9]. Camellia [10] is suitable block cipher for hardware implementations in the embedded cryptographic systems. Camellia [10,11] cipher has 128 bits data block size and 128, 192 and 256 bits key lengths. This block cipher was jointly developed by Nippon Telegraph and Telephone Corporation (NTT) and Mitsubishi Electric Corporation (Mitsubishi) to provide flexibility and security in cryptographic applications [10]. The Camellia block cipher is standardized in the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) (ISO/IEC 18,033) [12]. It has been recognized by CRYPTREC (Cryptography Research and Evaluation Committee) [13] and NESSIE (New European Schemes for Signatures, Integrity, and Encryption) [14] projects as a cryptographic algorithm for use on the Internet. The security and robustness of Camellia are investigated in [10,15,16] by testing resistance against different attacks such as Linear attack, Differential attack, Boomerang attack, Higher-order differential attack, Interpolation attack, Impossible differential attack, Improved zero-correlation attack, Square/Collision attack, and Cache trace attack.

Different ASIC implementations of the Camellia algorithm have been reported in [10,13–24]. In [18] compact and high-speed hardware2 architectures for the 128-bit AES and Camellia block ciphers are presented. The composite field arithmetic is used to reduce the size of the S-Boxes in this work. In [19] error detection approaches for the Camellia block cipher taking into account its linear and non-linear sub-blocks are proposed. In [20] a loop architecture based on one round function block is presented. The S-boxes are implemented based on both lookup table (LUT) and composite field arithmetic.

The flexibility, throughput, execution time and area consumption are the main design dimensions (challenges) of secure digital systems. A good design should consider the trade-offs between these dimensions. The flexibility is the ability to the reconfiguration of the system parameters such as key size. It is even more important especially for applications with a diverse set of requirements [28]. IoT applications are an outstanding example of this scenario. The level of security required by IoT devices and their supporting solutions will vary depending on the specific functions they are performing. A common practice of flexibility is to implement adaptive security for IoT applications such as smart control system based on a collection of the sensors for example, temperature sensor, humidity sensor, light sensor, presser sensor, heart sensor, flow meter sensor, etc. In the flexible structure, the key size varies in an acceptable range depending on the trust level of the sensors. An encryption process with a lower level of security (shorter key) allows high-speed computations in an application with less important information. On the other hand, the sensitive data within the network such as healthcare IoT applications is encrypted with a higher security level (longer key). The IoT systems are constrained in terms of execution time and computational resources; the problem is compounded when dealing with multimedia contents. Therefore, a flexible cryptographic structure can be reconfiguration based on the requirement with the best performance. The previous works have focused less on improving the design parameters, especially flexibility. Herein, we proposed high throughput and flexible hardware structures for Camellia block cipher for secure hardware IoT applications. Due to flexibility, they can be used for different types of IoT applications with multiple levels of security strengths. A unified circuit with resources sharing is implemented for Camellia cipher to compute the encryption process and some intermediate key values. The proposed flexible structures can do various configurations of Camellia to support variable key sizes (128, 192 and 256 bits). Implementation results of the proposed structures in 180 nm CMOS technology for different key sizes are achieved. The results show that the proposed structures have high-throughput and low computation time with acceptable hardware consumption compared to related works. The contributions of this study are as follows:

- A unified circuit with resources sharing is implemented for Camellia cipher to compute the encryption process and some intermediate key values.
- The substitution boxes (S-boxes) in the Camellia cipher are implemented based on area-optimised circuits.
- The number of logic gates and critical path delay of field multiplier and field inversion over $\mathbb{F}_{2^2}$ are reduced by using the simplification of logic terms. Also, a large number of gates, in these structures, have been implemented by the two-input NAND and NOR gates to reduce delay and area.
- The proposed flexible structures can do various configurations of Camellia to support variable key sizes (128, 192 and 256 bits).

Herein, Section 2 recalls the Camellia block cipher and the proposed hardware structures are described. Section 4 shows a comparison between our works and related works. Finally, Section 5 gives conclusions.

2 | DESCRIPTION OF THE CAMPELLIA BLOCK CIPHER AND PROPOSED HARDWARE STRUCTURE

The Camellia cryptographic algorithm is a block cipher that can process 128 bits data blocks, using 128, 192 and 256 bits secret keys. In this case, we have a flexible key size. It has better efficiency in both software and hardware implementations when compared to other ciphers [10]. The Camellia cipher uses an 18-round Feistel architecture for 128-bit keys and a 24-round Feistel architecture for 192-and 256-bit keys. Two main parts of the Camellia are the data processing part and key scheduling part. Algorithms 1 and 2 show the encryption procedure of Camellia for 128-bit and 192/256-bit keys, respectively. In the Camellia cipher, $FL$ and $FL^{-1}$ functions are inserted every six rounds in the algorithms.
Algorithm 1 Camellia Encryption Algorithm for 128-bit keys

**Input:** 128-bit plaintext $P = P_l||P_R$, whitening keys $kw_1$ to $kw_4$, round keys $k_1$ to $k_{18}$ and subkeys $kl_1$ to $kl_4$.

**Output:** 128-bit ciphertext $C$

1. $L_0 = kw_1 \oplus P_l$, $R_0 = kw_2 \oplus P_R$;
2. For $i$ from 1 to 6 do
   3. $L_i = R_{i-1} \oplus F(L_{i-1}, k_i)$, $R_i = L_{i-1}$;
   4. End For;
5. $L_6 = FL(L_0, kl_1)$, $R_6 = FL^{-1}(R_0, kl_2)$;
6. For $i$ from 7 to 12 do
   7. $L_i = R_{i-1} \oplus F(L_{i-1}, k_i)$, $R_i = L_{i-1}$;
   8. End For;
9. $L_{12} = FL(L_{12}, kl_3)$, $R_{12} = FL^{-1}(R_{12}, kl_4)$;
10. For $i$ from 13 to 18 do
11. $L_i = R_{i-1} \oplus F(L_{i-1}, k_i)$, $R_i = L_{i-1}$;
12. End For;
13. $C_L = kw_3 \oplus R_{24}$, $C_R = kw_4 \oplus L_{24}$;
14. Return $C = C_L||C_R$.

In each round, we have a $F$-function and a 64-bit XOR operation.

2.1 | $F$ function

The $F$-function is the most important block in the Camellia cipher computations. The structure and computations of $F$-function are shown in Figure 1. The $F$-function uses a Substitution-Permutation Network (SPN) structure. As seen from this figure, the function is constructed based on an 8-bit S-boxes $S_1$, $S_2$, $S_3$ and $S_4$ (S-function layer) and sixteenth 8-bit bit-wise XOR operations (P-function layer). The S-function part is a non-linear layer and the P-function part is a linear layer.

2.1.1 | $S$-boxes $S_1$, $S_2$, $S_3$ and $S_4$

The S-function layer consists of eight S-boxes, and four different S-boxes, $S_1$, $S_2$, $S_3$ and $S_4$ are used. All of them are equal to the field inversion operation over finite field $F_{2^8}$. The data of $S_2$, $S_3$ and $S_4$ can be generated from the S-box $S_1$. The 8-bit S-box $S_1$, for input $x$ and output $y_1$, is defined as follows:

$$S_1 : y_1 = S_1(x) = b((g(f(b(x5⊕x)))⊕x66)\lambda$$

Where $x$ denotes left cyclic shift of $x$ by $j$-bit and $\oplus$ is bit-wise XOR. The $f$ and $b$ functions can be presented based on the following computations:

$$f : b_7 = (a_6 ⊕ a_2), b_6 = (a_7 ⊕ a_4), b_5 = (a_3 ⊕ a_5 ⊕ a_0), b_4 = (a_5 ⊕ a_0), b_3 = (a_4 ⊕ a_1), b_2 = (a_6 ⊕ a_3), b_1 = (a_7 ⊕ a_0), b_0 = (a_4 ⊕ a_4)$$

$$b : b_7 = (a_6 ⊕ a_5 ⊕ a_2), b_6 = (a_0 ⊕ a_2), b_5 = (a_4 ⊕ a_1), b_4 = (a_6 ⊕ a_0), b_3 = (a_5 ⊕ a_1), b_2 = (a_7 ⊕ a_0), b_1 = (a_7 ⊕ a_1), b_0 = (a_5 ⊕ a_2)$$

The hardware resources for implementing the $F$-function are presented based on the following irreducible polynomials:

$$F_{2^8} : f_2(z) = z^8 + z^6 + z^5 + z^3 + 1$$

$$F_{2^{2^8}} : q(z) = z^2 + z + \lambda (\lambda = \omega^{2^8})$$

where $\omega$ is a root of $f_2(z)$. The following shows the proposed hardware structure of the field inversion over the composite field $F_{2^{2^8}}$. Let $n_1, n_0 \in F_{2^8}$, the inversion
FIGURE 1 Camellia encryption structures for 128-bit keys (a), 192/256-bit keys (b)
$m_1\beta + m_0 = (n_1\beta + n_0)^{-1}$ $(m_1, m_0 \in \mathbb{F}_2)$ can be calculated as follows:

- $m_1 = n_1\Delta^{-1}$,  
- $m_0 = (n_1 + n_0)\Delta^{-1}$,

where $\Delta = (n_1 + n_0)n_0 + \lambda n_1^2$ and vector representation of $\lambda$ is equal to $(1, 0, 0, 1)$ [10].

Figure 3 (a) shows the structure of field inversion operation over composite field $\mathbb{F}_{2^4}$: This operation is constructed based on four sub-blocks consisting of multiplication by constant $\lambda$, four field multiplications, one field inversion, one field squaring, and two field additions. The sub-blocks multiplication, inversion, and squaring are defined over $\mathbb{F}_{2^2}$ with primitive polynomial $f_2(z) = z^2 + z + 1$. The multiplication and inversion, squaring operations over $\mathbb{F}_{2^2}$ are implemented based on [30,31]. But in the proposed implementation we applied further optimization in the structures of multiplication and inversion over $\mathbb{F}_{2^2}$ to reduce critical path delay and area consumption. The squaring of element $K = (k_3k_2k_1k_0)_2$, $S = (s_3s_2s_1s_0)_2 = K^2$, in field $\mathbb{F}_{2^2}$ with irreducible polynomial $f(x) = x^2 + x + 1$ is equal to $s_3 = k_3$, $s_2 = k_3 \oplus k_1$, $s_1 = k_2$, $s_0 = k_2 \oplus k_0$. Also, the formula for multiplication with constant $L = (l_3l_2l_1l_0)_2 = Y \times \lambda$ in $\mathbb{F}_{2^2}$ are as $l_3 = y_3$, $l_2 = y_2$, $l_1 = y_1$, $l_0 = y_0$. The proposed merged formula of squaring and multiplication with constant $\lambda$, $H = (A^2) \times \lambda$ we have:

- $m_3 = s_0 \oplus a_2 \oplus a_0$
- $m_2 = s_3 = a_3$
- $m_1 = s_2 = a_3 \oplus a_1$
- $m_0 = s_1 \oplus s_0 = a_3 \oplus a_2 \oplus a_0 = a_0$

The low-cost structure of merged squaring and multiplication with constant $\lambda$ in the proposed S-box is shown in Figure 3 (b). In this case, the computations of squaring and multiplication by constant $\lambda$ are implemented by only two XOR logic gates with CPD equals $T_X$, where $T_X$ is gate delay of the 2-input XOR gate.

Figure 3 (c) shows the proposed efficient hardware structure of field inversion over $\mathbb{F}_{2^4}$. After further simplifications on $l_0$, $l_1$, $l_2$, and $l_3$ terms, the hardware consumption, and critical path delay are equal to 8 2-input XOR, 3 2-input XNOR gates, 11 2-input NAND gates, 1 2-input NOR gate and three NOT gates, and $2T_X + T_{XN} + T_{NOR}$, respectively. A large number of gates, in the structure, have been implemented by 2-input NAND and 2-input NOR gates to reduce delay and area.

Table 1 shows the hardware results of the proposed structure of 8-bit S-box $S_1$ and other related works. As seen
from the table, the proposed 8-bit S-box $S_1$ has acceptable hardware resources and timing characteristics compared to other 8-bit S-boxes.

2.2 | $FL$ and $FL^{-1}$ functions

In the Camellia cipher, two functions $FL$ and $FL^{-1}$ are used in the data processing part. These logical functions are inserted every six rounds. Let $kl_i = kl_{iL} || kl_{iR}$ be an intermediate key and also $X = X_L || X_R$ and $Y = Y_L || Y_R$ be 128-bit input and output, respectively. Therefore, two functions $FL$ and $FL^{-1}$ are defined as follows:

$FL : Y_L = X_L \oplus (X_R \oplus (X_L \land kl_{iL}) \lll 1) \lor kl_{iR}$,

$Y_R = X_R \oplus ((X_L \land kl_{iL}) \lll 1))$.

$FL^{-1} : Y_L = X_L \oplus (X_R \lor kl_{iR})$, $Y_R = X_R \oplus ((X_L \oplus (X_R \lor kl_{iR})) \land kl_{iL}) \lll 1$.

where, $\land$ and $\lor$ are bit-wise AND and OR operations, respectively. Figure 2 (b) and (c) shows the structures of two functions $FL$ and $FL^{-1}$, respectively.

2.3 | The proposed hardware structures for 128-bit, 192-bit and 256-bit keys

In this section, we present the hardware structures for implementation of the Camellia block cipher. Figure 4 shows the proposed block diagram of Camellia block cipher. The structure is designed based on the merged hardware for computing the data processing and $K_A$ and $K_B$ variables, where the $K_A$ and
TABLE 1 Hardware results of the proposed structure of 8-bit S-box $S_1$ and other related works

| Works     | # AND (or OR) | # NAND (or NOR) | # XOR (or XNOR) | # NOT | CPD                  |
|-----------|---------------|-----------------|-----------------|-------|----------------------|
| Ref. [33] $S_0$ | 127           | —               | 6               | 16    | $3T_X + 4T_A + 4T_O$ |
| Ref. [33] $S_1$ | 57            | 3               | 83              | 5     | $9T_X + 3T_A + T_N$  |
| Ref. [34]   | —             | 41              | 68              | 1     | —                    |
| Ref. [35] Case 1 | 58           | —               | 89              | 9     | $15T_X + 4T_A + 2T_N$ |
| Ref. [35] Case 7 | 35           | —               | 113             | 9     | $19T_X + 4T_A + 2T_N$ |
| Ref. [35] Case 13 | 58          | —               | 94              | 9     | $14T_X + 4T_A + 2T_N$ |
| Ref. [36] PN  | 58            | —               | 98              | 9     | $17T_X + 4T_A + 2T_N$ |
| Ref. [36] NN  | 58            | —               | 100             | 9     | $16T_X + 4T_A + 2T_N$ |
| TW         | —             | 59              | 88              | 12    | $8T_X + T_XN + 3T_{NA} + T_N$ |

TW: This work; Normal-Normal (NN); Polynomial-Normal (PN); $T_A$, $T_{NA}$, $T_X$, $T_{XN}$, $T_{OD}$, $T_{NO}$, $T_N$ denote the time delay of a 2-input AND gate, 2-input NAND gate, 2-input XOR gate, 2-input XNOR gate, 2-input OR gate, 2-input NOR gate, and NOT gate, respectively.

FIGURE 4 Proposed block diagram of Camellia block cipher

$K_B$ are two modified keys which are used in key scheduling part (these variables are discussed in the next subsection). The proposed structure of the Camellia block cipher is shown in Figure 5 for 192/256-bit keys. The three 2-to-1 multiplexers with control signal $Sel_3$ are used for configuration of two modes of data processing and generating the $K_A$ and $K_B$ variables. In this case, if the control signal $Sel_3$ is equal to ‘1’ the structures are configured for computing the $K_A$ and $K_B$ variables and for case $Sel_3 = 0$ we have the configuration of the structures for data processing part computations. Two control signals $Sel_1$ and $Sel_2$ are used for control of the encryption process. In the structure, the control signal $Sel_2$ is set to ‘1’ (in the first clock cycles) and values $k_{w_1} \oplus P_L$, $k_{w_2} \oplus P_R$, subkeys $k_1$ to $k_n$ and $k_{l_1}$, $k_{l_2}$ are applied to the structure. In other clock cycles, the control signal $Sel_2$ is set to ‘0’ and other subkeys are used for computation. This structure computes $n$ rounds of computations at each clock cycle, where $1 \leq n \leq 6$. Therefore, generating the ciphertext $C$, after computing the $K_A$ and $K_B$, required to $r/n+1$ clock cycles for both 128-bit and 192/256-bit keys.

2.4 Key schedules of Camellia block cipher and the proposed hardware structures

The key schedule part generates 64-bit whitening keys $k_{w_i}$ ($t = 1, 2, 3, 4$), round keys $k_u$ ($u = 1, 2, \ldots, r$) for round functions and $k_{l_i}$ ($i = 1, 2, \ldots, r/3 - 2$) for $FL$ and $FL'$ functions from the main key $K$, where $r$ is the number of rounds. In the key schedule part of Camellia block cipher, two 128-bit variables $K_L = K_{LL}||K_{LR}$ and $K_R = K_{RL}||K_{RR}$ are introduced, which are defined as follows:

For 128-bit main key $K \rightarrow K_L = K$, $K_R = 0$.

For 192-bit main key $K \rightarrow K_L||K_{RL} = K$, $K_{RR} = (K_{RL})'$.

For 256-bit main key $K \rightarrow K_L||K_{LR} = K$.

Two 128-bit variables $K_A$ and $K_B$ are generated based on $K_L$ and $K_R$, where for the 128-bit key we use only $K_A$ and for 192/256-bit keys, both $K_A$ and $K_B$ are used for key schedule part. The structure for generating the $K_A$ and $K_B$ is presented in work [11] (Figure 8). In this figure the 64-bit constants $\Sigma_i$ ($i = 1, 2, \ldots, 6$) are used as keys in the structure (Feistel network). These constants are presented in [10]. The $K_A$ and $\langle K_A$ and $K_B \rangle$ variables, for 128-bit and 192/256-bit keys, are computed in five and seven clock cycles, respectively. The control signals $L_1$, $L_2$ and $L_3$ are used for control of the 64-bit constants $\Sigma_i$, $1 \leq i \leq 6$ and intermediate results. In the first round block, multiplexers with control signal $L_1$, $L_2$, $L_3$ and 128-bit XOR operation are used for implementation of $K_A$ and $\langle K_A$ and $K_B \rangle$ computations. In the proposed structures, we first compute the variables $K_A$ and $K_B$ then the key scheduling and the data processing are started. Control signals of the proposed structure for computing the $K_A$, $K_B$ variables in the case (192/256-bit keys and $n = 2$, where $n$ is the number of computed rounds at each clock cycle and $1 \leq n \leq 6$) are presented in Table 2.

The 64-bit subkeys $k_{w_i}$ ($1 \leq t < 4$), $k_u$ ($1 \leq u < r$), and $k_{l_i}$ ($0 \leq i < r/3 - 2$) are generated by rotating $K_L$, $K_R$, $K_A$, and $K_B$ and taking the left-half or right-half of them. The computations of 64-bit subkeys for 128-bit and 192/256-bit keys are
shown in Table 3. The proposed structures (for case \( n = 2 \)) to generation of round keys for case 192/256-bit main key is shown in Figure 6. The number of clock cycles for this case \( (n = 2) \) are equal to 9 and 12 for 128-bit and 192/256-bit main keys, respectively. The control signals \( \text{Sel}_1, \text{Sel}_2, d[1 : 0] \) and \( c[1 : 0] \) are used for control and application of the generated subkeys into the data processing part. The subkeys are stored into registers \( \text{Reg}_1 \) and \( \text{Reg}_2 \) based on control signal \( \text{Sel}_1, \text{Sel}_2 \). The register \( \text{Reg}_1 \) is used for storing of the generated subkeys \( k_1, k_2, \ldots, k \), based on rotate blocks. Also, the subkeys \( kl_1, kl_2, \ldots, kw_4 \) are stored into the register \( \text{Reg}_2 \). The proposed structures generate the subkeys \( (k_1, k_2, k_3, k_4), (k_5, k_6, k_7, k_8), \ldots, (k_{23}, k_{24}, k_{25}, k_{26}) \) at the first, second, \ldots, and last clock cycles, respectively. The rotate blocks, in the structures, are implemented by wired cyclic shift without extra hardware.

Table 4 shows the control signals of the proposed structure, \( n = 2 \), for case 192/256-bit key at the begin of encryption process (Figure 6 as key scheduling part and Figure 5, for \( n = 2 \), as data processing part). For example, in the first clock cycle, we have computation of the first two rounds of data processing. In this case, for computation of these operations, the control signals \( \text{Sel}_1, \text{Sel}_2, c[1 : 0], d[1 : 0] \), \( \text{Sel}_1 \) and \( \text{Sel}_2 \) are equal to “01”, 0, 0, ‘0’ and ‘1’, respectively. Also, the computations in the next clock cycles are implemented by the proposed structure based on Table 4. Therefore, the ciphertext is generated with latency 13 clock cycles.

### 2.5 The proposed flexible structures of Camellia block cipher

In this section, we present a flexible structure for implementation of the Camellia block cipher. As mentioned before, the Camellia block cipher has three key sizes consisting of 128-bit, 192-bit, and 256-bit. The flexible architecture is used to implementing the Camellia cipher with different security levels. The proposed structure supports the key sizes 128, 192 and 256. Figure 7 shows the proposed flexible structure for implementation of the Camellia cipher for the case \( n = 2 \). The width of the registers and multiplexers in the encryption part and the round keys generator part are equal to 64-bit and 128-bit, respectively. The structure is configured based on the control signals \( \text{Sel}_1, \text{Sel}_2, d[1 : 0], c[1 : 0], L_1 \) to \( L_3, \text{Sel}_1 \) to \( \text{Sel}_3 \) and \( \text{Sel}_{128, 192, 256} \). In the first step of computations, the control signal \( \text{Sel}_2 \) is equal to ‘0’ and we have computing the \( K_A \) and \( K_B \) parameters based on data processing part \( (K_A \) and \( K_B \) parameters are generated in five and seven clock cycles for
TABLE 3  Subkeys for 128-bit and 192/256-bit keys

| Sub key | Value for 128-bit keys | Value for 192/256-bit keys | Sub key | Value for 128-bit keys | Value for 192/256-bit keys |
|---------|------------------------|-----------------------------|---------|------------------------|-----------------------------|
| kl1     | \(K_{LL}\)             | \(K_{LL}\)                  | \(k_j\) | (\(K_{LL} \lhd 45\))_L | (\(K_{LL} \lhd 30\))_L     |
| kl2     | \(K_{LR}\)             | \(K_{LR}\)                  | \(k_8\) | (\(K_{LR} \lhd 45\))_R | (\(K_{LR} \lhd 30\))_R     |
| k1 & kl3 | \(K_{A1}\)             | \(K_{BL}\)                  | \(k_9\) | (\(K_{A1} \lhd 15\))_L | (\(K_{A1} \lhd 45\))_L     |
| k2 & kl4 | \(K_{AR}\)             | \(K_{BR}\)                  | \(k_{10}\) | (\(K_{AR} \lhd 60\))_R | (\(K_{AR} \lhd 45\))_L     |
| kl5     | \((K_{d} \lhd 15)_L\)  | \((K_{d} \lhd 15)_L\)       | \(k_{11}\) | (\(K_{d} \lhd 60\))_L | (\(K_{d} \lhd 45\))_L     |
| kl6     | \((K_{d} \lhd 15)_R\)  | \((K_{d} \lhd 15)_R\)       | \(k_{12}\) | (\(K_{d} \lhd 60\))_R | (\(K_{d} \lhd 45\))_L     |
| kl7     | \((K_{d} \lhd 15)_L\)  | \((K_{d} \lhd 15)_L\)       | \(k_{13}\) | (\(K_{d} \lhd 30\))_L | (\(K_{d} \lhd 30\))_L     |
| kl8     | \((K_{d} \lhd 15)_R\)  | \((K_{d} \lhd 15)_R\)       | \(k_{14}\) | (\(K_{d} \lhd 30\))_R | (\(K_{d} \lhd 30\))_L     |
| kl9     | \((K_{d} \lhd 77)_L\)  | \((K_{d} \lhd 60)_L\)       | \(k_{15}\) | (\(K_{d} \lhd 77\))_L | (\(K_{d} \lhd 60\))_L     |
| kl10    | \((K_{d} \lhd 77)_R\)  | \((K_{d} \lhd 60)_R\)       | \(k_{16}\) | (\(K_{d} \lhd 77\))_R | (\(K_{d} \lhd 60\))_L     |
| kl11    | \((K_{d} \lhd 94)_L\)  | \((K_{d} \lhd 60)_L\)       | \(k_{17}\) | (\(K_{d} \lhd 94\))_L | (\(K_{d} \lhd 94\))_L     |
| kl12    | \((K_{d} \lhd 94)_R\)  | \((K_{d} \lhd 60)_R\)       | \(k_{18}\) | (\(K_{d} \lhd 94\))_R | (\(K_{d} \lhd 94\))_L     |
| kl13    | \((K_{d} \lhd 111)_L\) | \((K_{d} \lhd 77)_L\)       | \(k_{19}\) | (\(K_{d} \lhd 111\))_L | (\(K_{d} \lhd 77\))_L     |
| kl14    | \((K_{d} \lhd 111)_R\) | \((K_{d} \lhd 77)_R\)       | \(k_{20}\) | (\(K_{d} \lhd 111\))_R | (\(K_{d} \lhd 77\))_R     |
| kl15    | \((K_{d} \lhd 111)_L\) | \((K_{d} \lhd 111)_L\)      | \(k_{21}\) | (\(K_{d} \lhd 111\))_L | (\(K_{d} \lhd 111\))_L     |
| kl16    | \((K_{d} \lhd 111)_R\) | \((K_{d} \lhd 111)_R\)      | \(k_{22}\) | (\(K_{d} \lhd 111\))_R | (\(K_{d} \lhd 111\))_R     |

FIGURE 6  The proposed structures for generating the round keys \((n = 2)\) for case 192/256-bit keys

128-bit and 192/256-bit keys, respectively). Then, in the second step, after computing the \(K_d\) and \(K_f\) variables, this control signal is equal to ‘1’ and the computation of ciphertext \(C\) is started. Two registers \(R_{d1}\), \(R_{d2}\) and 2-to-1 multiplexers with control signal \(S_d\) and \(S_f\) are used for storing of the \(K_d\) and \(K_f\) parameters during the encryption process. The round keys and \(kl_i\) (1 \(\leq i \leq 4\) for 128-bit key and 1 \(\leq i \leq 4\) for the 192/256-bit key) and \(k_{w_i}\) (1 \(\leq j \leq 4\)) are generated and applied concurrently with the encryption process. The control signal \(Sel_{128,192/256}\) is used for selecting between cases 128-bit and 192/256-bit keys. For \(Sel_{128,192/256} = ‘1’\) the structure is configured for the 192/256-bit key otherwise it is configured for the 128-bit key computations. The hardware consumption and critical path delay of the proposed flexible structure of the Camellia cipher are presented in Section 4.

3 | SECURITY ANALYSIS OF THE PROPOSED STRUCTURES

The main focus of this work is the design and implementation of flexible and high-throughput hardware structures of the Camellia block cipher. However, we analyse the security of structures from a hardware point of view. Side-channel attacks
are the most threats to the security of cryptosystems. The sensitive data such as the main key can be recovered by these attacks. Side-channel attacks are non-invasive passive attacks that use certain physical information leaked during encryption operation. In this case, physical information such as power consumption [37], time delay [38], or electromagnetic radiation [39] are used to find the secret main key. The power analysis attack is a side-channel attack that interprets power consumption measurements during cryptographic operations. It can achieve information about a device's operation as well as the secret key based on a power trace.

In the proposed architectures, at each clock cycle, we have the computation of operations with the same hardware complexity. Therefore, the power consumption at each clock cycle is almost constant. This feature leads to a unified power trace in total clock cycles and the power consumption traces are independent of the secret key patterns. The timing attack is another important side-channel attack. The algorithm execution time in this attack is measured precisely [38]. If the algorithm computation time for different keys and plaintexts is different, this will lead to obtaining information about the bit-pattern of key by an attacker. Therefore, the hardware implementation of the algorithm should reduce dependence on the timing information. Each encryption operation in the proposed structures has a fixed computation time. The execution time is independent of the bit-pattern of key and the structures leak no information about this parameter. For example, the timing results for the proposed 128-bit and flexible structures of Camellia cipher are shown in Table 5. In this case, the encryption time for 50 separate plaintexts \( (P_1, P_2, \ldots, P_{50}) \) and main keys \( (K_1, K_2, \ldots, K_{50}) \) are measured. The encryption of the plaintexts \( (P_1, P_2, \ldots, P_{50}) \) takes the same time \( T_1 \) and \( T_2 \) for the 128-bit and flexible structures, respectively. Table 5 shows some samples. The execution time for each of the 50 measurements for the proposed 128-bit and flexible (for 128-bit key) structures is equal to 95.10 and 100.65 ns, respectively on 180 nm CMOS technology. As seen from this table, the execution time of the structures does not change when the plaintext or key being manipulated. Therefore, the internal computations of the block cipher algorithm are hidden.

4 | RESULTS AND COMPARISON

The ASIC implementation results of the proposed structures with other ASIC implementations of the Camellia block cipher are compared in this section. The ASIC results in the proposed structures are achieved by using Synopsys Design Compiler tool based on the library of standard cells with 180 nm CMOS technology. In the literature the area is measured based on \( \mu m^2 \) and the gate equivalents (GE), the \( \mu m^2 \) value depends on the

| #Clock Cycles | \( S_{c1}, S_{c2} \) | \( c[1:0] \) | \( d[1:0] \) | \( Sel_1 \) | \( Sel_2 \) | \( Sel_3 \) |
|---------------|-----------------|-------------|-------------|----------|----------|----------|
| 1             | 01              | 0           | 0           | 0        | 1        | 1        |
| 2             | 10              | 0           | 0           | 0        | 0        | 1        |
| 3             | 00              | 0           | 0           | 0        | 0        | 1        |
| 4             | 01              | 0           | 1           | 0        | 0        | 1        |
| 5             | 10              | 1           | 1           | 0        | 0        | 1        |
| 6             | 00              | 1           | 1           | 0        | 0        | 1        |
| 7             | 01              | 1           | 1           | 1        | 0        | 1        |
| 8             | 10              | 2           | 2           | 0        | 0        | 1        |
| 9             | 00              | 2           | 2           | 0        | 0        | 1        |
| 10            | 01              | 2           | 2           | 1        | 0        | 1        |
| 11            | 10              | 3           | 3           | 0        | 0        | 1        |
| 12            | 00              | 3           | 3           | 0        | 0        | 1        |
| 13            | 00              | 3           | 3           | 0        | 0        | 1        |
fabrication technology. To compare the area requirements independently it is common to state the area based on GE. One GE is equivalent to the area of a 2-input NAND gate with the lowest driving strength of the corresponding technology. In other words, this metric represents the amount of consumed area normalized to the area of one 2-input NAND gate. The performance and results of the designs are evaluated in terms of critical path delay, number of clock cycles, area, computation time, throughput, throughput/area and power consumption. The power consumption for a frequency of 100 KHz is measured.

The results of the proposed works are achieved based on different key sizes (128, 192 and 256 bits) and two cases \( n = 1 \) and \( n = 2 \). Results of the proposed implementations and related works for Camellia cipher are shown in Table 6. The optimization in works [18, 20] is based on area and speed. In [18] compact and high-speed hardware architectures for the 128-bit block cipher Camellia are presented. The composite field arithmetic is used to reduce the area consumption of the S-boxes in this work. In [20] a loop architecture based on one round function block is presented. The S-boxes in [20] are implemented by two methods LUT and composite field arithmetic. As given in the table, the proposed structures consume acceptable hardware resources with reasonable timing characteristics compared to the other architectures. We also get improvements in terms of area, throughput, and throughput/area. When \( n \geq 3 \), the critical path delay, hardware resources and execution time of the circuit become longer than those of the circuit for cases \( n = 1, 2 \). Also, the throughput and throughput/area are reduced for cases \( n \geq 3 \).

Results of the proposed implementation for flexible Camellia cipher are shown in Table 7. This structure supports three main key sizes of 128, 192 and 256-bit, and it is based on the 2-round Feistel network. The architecture provides a versatile implementation that enables adaptive security level using a variable key size. The flexible structure is a suitable candidate for a broad range of applications with multiple levels of security. As given in Table 7, the proposed structure consumes an acceptable area with low critical path delays. Also, Table 8 shows the results of proposed 128-bit key structure of Camellia cipher and other lightweight ciphers with 128-bit key. It should be noted that the HIGHT, PRESENT, and LED block cipher have the 64-bit block size (plaintext) with the 128-bit key size.

The main differences between the proposed Camellia structures and other related works are as follows:

1. To computing the encryption process and some intermediate key values a unified circuit with resources sharing is implemented for Camellia cipher.
2. The 8-bit S-boxes in the Camellia cipher are implemented based on area-optimised circuits.
3. The efficient circuits for field multiplier and inversion over \( \mathbb{F}_2^5 \) in the S-box are presented. A large number of gates, in these structures, have been implemented by the 2-input NAND and NOR gates to reduce delay and area.

| Plaintext | Main key | Cipher text | Time for 128-bit structure (T1) | Time for flexible structure (T2) |
|-----------|----------|-------------|-------------------------------|-------------------------------|
| P1        | 00000000000000000000000000000001 | FEDCBA9876543210 | 100.65 ns | 100.65 ns |
| P2        | 40000000000000000000000000000000 | 00000000000000000000000000000000 | 95.10 ns | 95.10 ns |
| P3        | 80000000000000000000000000000000 | 00000000000000000000000000000000 | 95.10 ns | 95.10 ns |
| P4        | 12000000000000000000000000000000 | 00000000000000000000000000000000 | 95.10 ns | 95.10 ns |
| P5        | 16000000000000000000000000000000 | 00000000000000000000000000000000 | 95.10 ns | 95.10 ns |
| ⋮         | ⋮        | ⋮           | ⋮                             | ⋮                             |

TABLE 5 Some samples of hardware results of the proposed 128-bit and flexible (for 128-bit key) structures of Camellia cipher.
### TABLE 6  Results of the proposed implementations and other related works on Camellia cipher

| Works | Tech. (nm) | Area (GE) | #CC | CPD (ns) | Time (ns) | Thr. (Mbps) | Thr./Area (Mbps)/GE | Power ($\mu$w) |
|-------|------------|-----------|------|----------|-----------|-------------|---------------------|---------------|
| Ref. [19], 128-128 | 65 | 190,751 | -- | -- | 12.40 | 10,200 | 0.054 | 108.34 |
| Ref. [21], A, 128-128 | 130 | 14,918 | 22 | 8.79 | 193.39 | 661.18 | 0.044 | 16.41 |
| Ref. [21], S, 128-128 | 130 | 24,424 | 22 | 5.20 | 114.40 | 1118.89 | 0.046 | 26.87 |
| Ref. [23], LPCA, 128-128 | 180 | 3128 | 84 | 100 | 8400 | 15.24 | 0.0049 | 4.69 |
| Ref. [23], RCA, 128-128 | 180 | 3328 | 84 | 100 | 8400 | 15.24 | 0.0046 | 4.99 |
| Ref. [24], 128-128 | 130 | 4313 | 400 | 3.95 | 1580 | 81.01 | 0.0188 | 4.31 |
| $TW_n=1$, 128-128 | 180 | 10,446 | 24 | 4.15 | 99.60 | 1285.14 | 0.123 | 15.66 |
| $TW_n=1$, 128-192/256 | 180 | 12,430 | 32 | 4.15 | 132.80 | 963.86 | 0.078 | 18.64 |
| $TW_n=2$, 128-128 | 180 | 13,758 | 15 | 6.34 | 95.10 | 1345.95 | 0.098 | 20.63 |
| $TW_n=2$, 128-192/256 | 180 | 15,614 | 20 | 6.34 | 126.8 | 1009.46 | 0.065 | 23.42 |

TW: This work; CCs: Clock cycles; Thr.: Throughput; A: Area; S: Speed; 128-192/256: The structure supports both 192- and 256-bit keys. The power consumption is archived for a frequency of 100 KHz.

### TABLE 7  Results of the proposed flexible Camellia cipher based on 180 nm technology

| Area (GE) | #CCs (128, 192/256) | CPD (ns) | Time (ns) (128, 192/256) | Thr. (Mbps) (128, 192/256) | Thr./Area (Mbps)/GE (128, 192/256) | Power ($\mu$w) |
|-----------|---------------------|---------|-------------------------|-----------------------------|----------------------------------|---------------|
| 19,142 (15, 18) | 6.71 (100.65, 120.78) | (1271.73, 1059.78) | (0.066, 0.055) | 29.943 |

Note: The power consumption is archived for a frequency of 100 KHz.

### TABLE 8  Results of the proposed 128-bit key structure of Camellia cipher and other lightweight ciphers with 128-bit key

| Works | Tech. (nm) | Area (GE) | #CCs | CPD (ns) | Time (ns) | Thr. (Mbps) | Thr./Area (Mbps)/GE |
|-------|------------|-----------|------|----------|-----------|-------------|---------------------|
| Ref. [40] HIGHT | 350 | 2608 | 34 | 8 | 272.34 | 235 | 0.09 |
| Ref. [41] HIGHT, SK, UF = 1 | 180 | 2605 | 32 | 1.596 | 51.07 | 1253 | 0.481 |
| Ref. [41] HIGHT, SK, UF = 2 | 180 | 3550 | 16 | 2.322 | 37.15 | 1723 | 0.485 |
| Ref. [41] HIGHT, SK, UF = 3 | 180 | 4493 | 10 | 3.013 | 30.13 | 2124 | 0.473 |
| Ref. [41] HIGHT, SK, UF = 4 | 180 | 5441 | 8 | 4.541 | 36.33 | 1762 | 0.324 |
| Ref. [42] PRESENT | 180 | 23,006 | 1 | 38.10 | 38.10 | 1961 | 0.068 |
| Ref. [41] PRESENT, UF = 1 | 180 | 2306 | 32 | 1.02 | 32.64 | 1961 | 0.851 |
| Ref. [41] PRESENT, UF = 2 | 180 | 3386 | 16 | 1.699 | 27.18 | 2354 | 0.695 |
| Ref. [41] PRESENT, UF = 3 | 180 | 4466 | 11 | 2.248 | 24.73 | 2588 | 0.58 |
| Ref. [41] PRESENT, UF = 4 | 180 | 5546 | 8 | 2.987 | 23.89 | 2678 | 0.483 |
| Ref. [41] PRESENT, UF = 5 | 180 | 6626 | 7 | 3.784 | 26.49 | 2416 | 0.365 |
| Ref. [44] CLEFIA | 65 | 9423 | 18 | 1.65 | 29.63 | 4310 | 0.457 |
| Ref. [33] CLEFIA | 180 | 8667 | 15 | 3.08 | 46.20 | 2771 | 0.320 |
| Ref. [42] SIMON | 130 | 23,584 | 1 | 41.70 | 44.76 | 1430 | 0.061 |
| Ref. [43] SIMON | 130 | 23,584 | 1 | 41.70 | 44.76 | 1430 | 0.061 |
| Ref. [46] SIMON, iterative | 130 | 2342 | 70 | 1.597 | 111.79 | 1145 | 0.489 |
| Ref. [46] SIMON, key-agile pipe. | 130 | 146,287 | 1 | 1.196 | 1.196 | 106,961 | 0.731 |
| Ref. [46] SIMON, non-key-agile pipe. | 130 | 104,790 | 1 | 1.458 | 1.458 | 87,798 | 0.838 |
| Ref. [45] SIMON | 180 | 3771 | 68 | 0.53 | 35.84 | 3571.83 | 0.947 |
| Ref. [43] SPECK | 130 | 16,371 | 1 | 182.4 | 193.94 | 330 | 0.0202 |
4. The proposed flexible structure is useful for the realization of various configurations of Camellia to support variable key sizes (128, 192 and 256 bits).

5 | CONCLUSION

Security and privacy of the IoT systems are critical challenges in many data-sensitive applications. Efficient and flexible hardware structures of the Camellia block cipher are presented to provide the security of sensitive data in IoT communication. In the proposed structures, to reduce hardware consumption, a part of the key schedule (computation of the $K_A$ and $K_B$ variables) is implemented by the encryption part. The most complex blocks in the Camellia cipher are substitution boxes. The S-boxes are implemented based on an area-optimised field inversion over $\mathbb{F}_{2^8}$ and two affine transformations over $\mathbb{F}_{2^4}$, which are computed by XOR gates. To enhance area reduction, the proposed structure of field inversion is designed over the composite field $\mathbb{F}_{2^{2\cdot2^4}}$ with optimised field inversion and field multiplier over $\mathbb{F}_{2^4}$. The flexible structures that can do various configurations of Camellia cipher to support variable key sizes (128, 192 and 256 bits) are proposed. The implementation results show improvements in terms of area, throughput and throughput/area compared to the related studies.

REFERENCES

1. Azzori, L., Iera, A., Morabito, G.: The Internet of Things: A survey. Comput. Netw. 54(18), 2787–2805 (2010)
2. Rashidi, B.: Cryptographic algorithms. In: Authentication Technologies for Cloud Computing, IoT and Big Data, 1st ed. IET, London, pp. 67–105 (2019)
3. Ferrag, M.A. et al.: Authentication protocols for internet of things: A comprehensive survey. Secur. Commun. Netw. 2017, 1–41 (2017)
4. Rahmani, A.M. et al.: Smart e-Health Gateway: Bringing intelligence to Internet-of-Things based ubiquitous healthcare systems. In: Proc. 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 Jan 2015 pp. 826–834 (2015)
5. Zhang, G.H., Poon, C.C.Y., Zhang, T.Y.: Smart e-Health Gateway: Bringing intelligence to Internet-of-Things based ubiquitous healthcare systems, vol. 2011, pp. 1–8. ISRN Communications and Networking Hindawi Publishing Cairo, (2011)
6. Rahi Sfar, A. et al.: A roadmap for security challenges in the Internet of Things. Digit. Commun. Netw. 4, 118–137 (2018)
7. Noura, H. et al.: One round cipher algorithm for multimedia IoT devices. Multimed. Tools Appl. 77(14), 18383–18413 (2018)
8. Seo, H. et al.: Compact implementations of ARX-based block ciphers on IoT Processors. ACM Trans. Embed. Comput. Syst. 17(3), 1–16 (2018)
9. Hatzivasilis, G. et al.: A review of lightweight block ciphers. J. Cryptogr. Eng. 11(3), 141–184 (2018)
10. Aoki, K. et al.: Camellia: A 128-bit block cipher suitable for multiple platforms-design and analysis, Sony Corporation. In: Proc. Selected areas in Cryptography (SAC), pp. 39–56.springer (2001)
11. Aoki, K. et al.: Specification of Camellia, a 128-bit block cipher, Technical Report 1, Mitsubishi and NTT DoCoMo, Tokio, Japan (August 2001)
12. ISO/IEC 18033: information technology-security techniques-encryption algorithms (2015)
13. Information-technology promotion agency, Japan, CRYPTREC Report (2002)
14. NESSIE project announces final selection of crypto algorithms, IST-199-12324 (February 2003)
15. Liu, Z. et al.: Improved zero-correlation linear cryptanalysis of reduced-round Camellia under weak keys. IET Inf. Secur. 10(2), 95–103 (2016)
16. Liu, Y. et al.: Improved impossible differential attack on reduced version of Camellia with $FL/FL^{-1}$ functions. IET Inf. Secur. 10(6), 425–432 (2016)
17. ISO/IEC 29192-2: information technology-security techniques-lightweight Cryptography-Part 2: block ciphers (2012)
18. Satoh, A., Morioka, S.: Hardware-focused performance comparison for the standard block ciphers AES, Camellia, and triple-DES In: Proc. International Conference on Information Security, Lecture Notes in Computer Science, vol. 2851, Bristol, United Kingdom, pp. 252–266 (2003)
19. Mozaffari-Kermani, M., Azarderakhsh, R., Xie, J.: Error detection reliable architectures of Camellia block cipher applicable to different variants of its substitution boxes. In: Proc. IEEE Asian Hardware-Oriented Security and Trust (AsianHOST), Yilan, Taiwan, pp. 1–6 (2016)
20. Sugawara, T. et al.: ASIC performance comparison for the iso standard block ciphers. In: Proc. 2nd Joint Workshop on Information Security, Tokyo, Japan, pp. 485–498 (2007)
21. Satoh, A., Morioka, S.: Unified hardware architecture for 128-bit block ciphers AES and Camellia. In: Proc. 5th Cryptographic Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, vol. 2779, pp. 304–318 (2003)
22. Cheng, A., Heys, M.H.: Compact hardware implementation of the block cipher Camellia with concurrent error detection. In: Proc. Canadian Conference on Electrical and Computer Engineering, Vancouver, BC, Canada, pp. 1129–1132 (2007)
23. Gangadari, B.R.: Low power VLSI architectures for cryptographic algorithms, Ph.D. Thesis. Department of Electronics & Electrical Engineering, Indian Institute of Technology Guwahati (2018)
24. Kavun, E.B., Yalcin, T.: A pipelined Camellia architecture for compact hardware implementation. In: Proc. 21st IEEE International Conference on Application-Specific Systems, Architectures and Processors, Rennes, France, pp. 305–308 (2010)

25. Wang, C., Hey, H.M.: An ultra compact block cipher for serialised architecture implementations. In: Proc. Canadian Conference on Electrical and Computer Engineering, St. John's, NL, Canada, pp. 1–6 (2009)

26. Rolfs, C. et al.: Ultra-lightweight implementations for smart devices-security for 1000 gate equivalents. In: Proc. International Conference on Ultra Advanced Systems, Springer, London, pp. 89–103 (2008)

27. Maene, P., Verbauwhede, I.: Single-cycle implementations of block ciphers. In: Proc. International Workshop on Lightweight Cryptography for Security and Privacy, vol. 9542, Bochum, Germany, pp. 131–147 (2015)

28. Gulcan, E., Aygu, A., Schaumont, P.: A flexible and compact hardware architecture for the SIMON block cipher. In: Proc. International Workshop on Lightweight Cryptography for Security and Privacy (LightSec), Istanbul, Turkey, pp. 34–50 (2015)

29. Rudra, A. et al.: Efficient Rijndael encryption implementation with composite field arithmetic. In: Proc. Cryptographic Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, vol. 2162, Paris, France, pp. 171–184 (2001)

30. Paar, C.: Efficient VLSI architectures for bit-parallel computation in Galois fields, Ph.D. Thesis. Institute for Experimental Mathematics, University of Essen, Essen, Germany (June 1994)

31. Rashidi, B., Sayedi, S.M., Rezaeian Farashahi, R.: Efficient implementation of bit-parallel fault tolerant polynomial basis multiplication and squaring over $GF(2^m)$. IET Comput. Digit. Tech. 10(1), 18–29 (2016)

32. Courtois, N.T., Hulme, D., Mourouzis, T.: Solving circuit optimisation problems in cryptography and cryptanalysis. In: Proc. The Fifth Workshop on Special-Purpose Hardware for Attacking Cryptographic Systems, Washington, DC, USA, pp. 179–191 (2012)

33. Rashidi, B.: Efficient and flexible hardware structures of the 128-bit CLEFIA block cipher. IET Comput. Digit. Tech. 14(2), 69–79 (2020)

34. Wei, Z. et al.: Scrutinising the tower field implementation of the $F_{2^m}$ inverter with applications to AES, Camellia, and SM4, Cryptography ePrint Archive, Report No. 2019/738 (2019)

35. Martinez-Herrera, A.F., Mex-Perera, J.C., Nolazco-Flores, J.A.: Some representations of the S-Box of Camellia in $GF(2^{128})$. In: Proc. 11th International Conference Cryptology and Network Security, Darmstadt, Germany, pp. 296–309 (2012)

36. Martinez-Herrera, A.F., Mex-Perera, C., Nolazco-Flores, J.: Merging the Camellia, SMS4 and AES S-boxes in a single S-box with composite bases. In: 16th International Conference Information Security, Lecture Notes in Computer Science, vol. 7807, Dallas, Texas, pp. 209–217 (2015)

37. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Proc. Advances in Cryptology, Berlin, Germany, pp. 388–397 (1999)

38. Kocher, P.C.: Timing attacks on implementations of Diffie–Hellman, RSA, DSS. In: Proc. Advances in Cryptology, Berlin, Germany, pp. 104–113 (1996)

39. Hayashi, Y.I. et al.: Analysis of electromagnetic information leakage from cryptographic devices with different physical structures. IEEE Trans. Electromagn. C. 55(3), 571–580 (2013)

40. Lim, Y.I. et al.: Implementation of HIGHT cryptic circuit for RFID tag. IEICE Electron. Express. 6(4), 180–186 (2009)

41. Rashidi, B.: Efficient and high-throughput application-specific integrated circuit implementations of HIGHT and PRESENT block ciphers. IET Circ. Devices Syst. 13(6), 731–740 (2019)

42. Maene, P., Verbauwhede, I.: PRESENT: an ultra lightweight block cipher. In: Proc. International Workshop on Lightweight Cryptography for Security and Privacy, vol. 9542, Bochum, Germany, pp. 131–147 (2015)

43. Rashidi, B.: High-throughput and flexible ASIC implementations of SIMON and SPECK lightweight block ciphers. Int. J. Circ. Theor. Appl. 47, 1–15 (2019)

44. Beauleau, R. et al.: Simon and Speck: block ciphers for the internet of things. In: Proc. NIST Lightweight Cryptography Workshop, pp. 1–15 (2015)

45. Schneider, T., Moradi, A., Gunesu, T.: PaTI-towards combined hardware countermeasures against side-channel and fault-injection attacks. In: Proc. The 2016 ACM Workshop on Theory of Implementation Security, Vienna, Austria, pp. 302–332 (2016)

How to cite this article: Rashidi B. Flexible and high-throughput structures of Camellia block cipher for security of the Internet of Things. IET Comput. Digit. Tech. 2021;15:171–184. https://doi.org/10.1049/cdt2.12025