FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Improving Safety of an Automotive AES-GCM Core and its Impact on Side-Channel Protection

Dany José Correia Teixeira

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor at FEUP: Prof. José Carlos Alves

Second Supervisor at FEUP: Prof. Manuel Cândido Duarte dos Santos

Supervisor at Synopsys: Eng. Pedro Costa

July 28, 2020

Resumo

O incremento do número de componentes eletrónicos e o correspondente aumento do fluxo de dados no setor automóvel levou a uma preocupação crescente com a garantia de segurança dos sistemas eletrónicos, especialmente em sistemas críticos cuja violação seja passível de colocar em causa a integridade do sistema e a segurança das pessoas. A utilização de sistemas que implementam o Advanced Encryption Standard (AES) foi vista como uma solução para este problema, impedindo o acesso indevido aos dados dos veículos, através da sua encriptação.

O algoritmo AES não possui atualmente nenhuma vulnerabilidade efetiva, mas o mesmo não acontece com as suas implementações, as quais estão sujeitas a ataques ditos side-channel, onde informações que resultam da operação destas implementações são exploradas na tentativa de descobrir os dados encriptados. A aplicação de núcleos IP no setor automóvel requer que estes estejam em conformidade com a norma ISO-26262 de forma a garantir que a sua operação não compromete a segurança do veículo e dos ocupantes. Este cumprimento implica alterações na arquitetura dos sistemas que podem influenciar as características de operação que são normalmente exploradas em ataques para obter informação que eventualmente permita ganhar conhecimento sobre os dados encriptados. Assim, o desenvolvimento das componentes de segurança, na perspetiva da segurança informática da informação e no que se refere à segurança de operação do veículo e dos seus ocupantes, que são ainda consideradas como componentes independentes, podem na verdade estar relacionadas, já que as melhorias introduzidas para incrementar a resiliência a falhas e consequentemente a integridade de operação dos sistemas, podem aumentar a fragilidade do sistema a ataques que comprometam a segurança informática dos dados.

O presente trabalho tem como objetivo desenvolver uma arquitetura capaz de atingir as métricas para o nível mais alto de certificação em segurança de acordo com a norma ISO-26262 (certificação ASIL-D), a partir de uma arquitetura já existente, e comparar as duas arquiteturas em termos de vulnerabilidade a ataques ditos side-channel que exploram o seu consumo de potência dinâmica. Os resultados demonstram que para a arquitetura ASIL-D a identificação de pontos de interesse e de dados relevantes no consumo de potência é mais evidente, o que sugere existir uma maior vulnerabilidade da arquitetura desenvolvida a ataques desenvolvidos por esse processo.

Abstract

The increase in electronic components and the corresponding increment in the data flow among electronic systems in automotive applications made security one of the main concerns in this sector. The use of IP cores that implement the Advanced Encryption Standard (AES) was seen as a solution to this problem, preventing improper access to vehicle data, through its encryption.

The AES algorithm does not currently have any effective vulnerability, but the same does not happen with its implementations, which are subject to side-channel attacks, where information that results from the operation of these implementations is exploited in an attempt to discover the encrypted data. The application of IP cores in the automotive sector requires the compliance of the implementations with the ISO-26262 standard in order to ensure that their operation does not compromise the vehicle's safety. For this compliance it is required changes in the core architecture that can influence the characteristics of operation that are normally exploited in attacks. Thus, the development of safety and security components in the automotive sector, which are still considered as independent processes, may be related since safety improvements may cause changes in the system's vulnerability to attacks that can compromise its security.

This work aims to develop an architecture capable of reaching the metrics for the highest level of safety certification (ASIL-D), based on an existing architecture, and compare the two architectures in terms of vulnerability to side-channel attacks that exploit their dynamic power consumption. The results show that for the ASIL-D architecture, the identification of points of interest and relevant data on the power consumption traces is more evident, which suggests greater effectiveness of the attacks performed in this architecture.

Acknowledgements

I would like to express my gratitude to my supervisor at FEUP, Professor José Carlos Alves, for all his interest, ideas and suggestions throughout this entire project and for always being available to help me. I am also grateful to Professor Manuel Cândido dos Santos for introducing me this opportunity and for the assistance he provided to make this project possible.

I would like to thank my supervisor at Synopsys, Eng. Pedro Costa, for his continued guidance, advice and help during the development of this dissertation. I would also like to thank Eng. Hugo Santos for all the support he provided me and for always being available to give me a hand. To both, Eng. Pedro Costa and Eng. Hugo Santos, I express my gratitude for helping me in this introduction to the FuSa world.

Finally, a special thanks to Inês Honrado for being there for me at all good and less good moments throughout these years, always willing to share the thoughts and kindness of an incredible person she is. To all my friends, colleagues and family, my sincere thanks. I deeply believe that every person in our life influences our development and, without them, I would not be the person I am.

Dany José Correia Teixeira

"A ship in harbor is safe, but that is not what ships are built for."

John A. Shedd

Contents

List of Figures

List of Tables

Abbreviations

AAD	Additional Authenticated Data
AES	Advanced Encryption Standard
ASIC	Application-Specific Integrated Circuit
ASIL	Automotive Safety Integrity Level
CBC	Cipher Block Chaining Mode
CFB	Cipher Feedback Mode
CTR	Counter Mode
DES	Data Encryption Standard
DPA	Differential Power Analysis
ECB	Electronic CodeBook Mode
ECU	Electronic Control Unit
EDA	Electronic Design Automation
FIPS	Federal Information Processing Standard
FIT	Failure In Time
FMEDA	Failure Modes, Effects, and Diagnostic Analysis
FPGA	Field-Programmable Gate Array
GCM	Galois Counter Mode
GF	Galois Field
HARA	Hazard Analysis and Risk Assessment
HSM	Hardware Secure Module
IBM	International Business Machines
IEC	International Electrotechnical Commission
IP	Intellectual Property
ISO	International Organization for Standardization
IV	Initialization Vector
MAC	Message Authentication Code
MCU	Micro-controller Unit
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OFB	Output Feedback Mode
RTL	Register Transfer Level
SDF	Standard Delay Format
SEooC	Safety Element out of Context
SHE	Secure Hardware Extension
SPA	Simple Power Analysis
TCL	Tool Command Language
UVM	Universal Verification Methodology
VCD	Value Change Dump

Chapter 1

Introduction

1.1 Context

The technology evolution implied a considerable increase in the information exchanged between systems. As a result, the need to preserve the integrity, availability and confidentiality of systems has become not only a priority for governments and their military purposes but all businesses, organisations and the general population, leading to the creation of multiple encryption systems by private organisations. These systems were found to have considerably different characteristics between them, making their usage and interoperability a problem. In order to solve this, the National Institute of Standards and Technology (NIST) (at that time called National Bureau of Standards) started, in 1973, a public request for proposals of cryptography algorithms that could be used to create a cryptography standard. An algorithm, based on Lucifer cypher, was proposed by IBM and analysed and adjusted by the National Security Agency (NSA), resulting in a federal standard that was used for over 20 years, the Data Encryption Standard (DES) [1]. The DES was, however, only empirically secure and the progressive increase in the computational processing power of the technology evidenced some vulnerabilities in the standard, as shown in [2].

The NIST started then, in 1997, a process to select the most suitable algorithm for a new standard that could solve the flaws of the previous one and establish a new reference in the security domain. The selection criteria were not only the algorithm's security but also its performance, efficiency, ease of implementation (in hardware, software and firmware) and flexibility [3]. The algorithm that was selected to establish the new Federal Information Processing Standard (FIPS) was Rijndael, and the standard was named the Advanced Encryption Standard (AES).

Despite the apparent absence of vulnerabilities in the algorithm, the same does not seem to happen with the respective implementations. The security of the implementations has been largely exploited through side-channel attacks, where the implementation characteristics such as the power consumption or the electromagnetic emission are analysed, in an attempt to extract hints that allow the identification of the data being processed. In the case of the AES, these attacks focus in reveal the key which will allow the decryption of all data that has been encrypted using that key. Since implementations of the AES are used in the automotive industry to protect sensitive data

that flows through communications and, consequently, to protect against attacks critical safety elements, these weaknesses in security can also create vulnerabilities in the safety domain.

The automotive sector defines the ISO-26262 as the leading standard for granting a qualitative assessment of the risk of hazardous operational events that may induce safety failures, expressed in the form of an Automotive Safety Integrity Level (ASIL), which reveals if the component meets the imposed safety requirements.

1.2 Motivation

The automotive industry is one of the sectors where the developed systems require compliance with specific safety metrics to ensure that in case of a system malfunction or failure, the people and the environment involved do not suffer an increase in their level of risk. This awareness of the importance of safety for the automotive industry is well described by ISO-26262 standard, which defines all the regulations and recommendations for the development of a safe system along all the development phases, including the definition of risk classes that are attributed to components according to the exposure, controllability and severity of the risks. The first edition of this standard that dates from 2011 has, however, a considerable flaw in the approach of security concerns and its implications in the safety domain. With the increase in the information that is necessary to store and transfer to support all the features present in modern vehicles, concerns about the unauthorised access and modification of this data are gaining a relevant consideration. Proof of this is that a revision to the ISO-26262 standard published in 2018, already considers some references to cyber security and a new part only about the application of the standard to semiconductors.

The AES has been used in the last few years as one of the data encryption mechanisms in the automotive industry to protect sensitive data, usually in the form of hardware implementations. Due to the complexity of the operations executed in each data decryption and considering the processing power of today's computers, brute force attacks, which are normally the simplest way to break a cryptographic algorithm, would take billions of years, even considering millions of computers working in parallel, to test the 2^{128} possible key combinations for a key of 128 bits, making this method impracticable. Thus, the algorithm itself still provides a substantial level of protection for the system. The flaws arise only in the hardware that implements the algorithm, either through debug mechanisms, which exist in current integrated circuits due to their complexity (and which open the possibility for fault injection attacks) or by analysis of the physical parameters of the core, which can provide information about the encryption key.

The implementations for the automotive industry present, therefore, two main limitations in the security domain. On the one hand, a standard that defines in detail the safety demands for a safety-critical system but that does not fully address the relationship between a safe system and a secure system and, on the other hand, the existence of exposure to attacks that these systems present, due to their implementation properties, and which can compromise them. Those two limitations do not seem although to be completely independent, as shown in [4], with the possibility to interconnect them, analysing the actual existence of exposure to attacks in the IP core and the impact that the

safety measures have in the security field. The existence of a relationship between both domains can create the possibility of using the safety mechanisms (which are mandatory in this type of systems) to prevent or, at least, help to mitigate the risk of those attacks to an acceptable level, without compromising the systems' safety integrity level.

1.3 Objectives

Synopsys currently has an IP core which implements the AES in the Galois Counter Mode (GCM) of operation, used in automotive applications but without any safety integrity level (ASIL) certification according to ISO-26262. The first objective of this dissertation is to implement safety measures in the current design in order to meet the metrics for an ASIL-D certification, following the specifications of the ISO-26262 standard. Then, the objective is to analyse both systems (with and without safety mechanisms) in terms of exposure to power consumption based side-channel attacks. A comparison between both systems will provide an assessment of the impact of the safety mechanisms on security, from where conclusions between the safety and security domains may be extracted. These conclusions can provide useful information for future works that require safety enhancements so that they are developed without increase the security vulnerabilities of the design.

1.4 Thesis Structure

This introduction in chapter 1 provides a contextualisation about the development of the Advanced Encryption Standard (AES) and the problems that this security solution is facing with its adaptation for automotive safety applications.

Chapter 2 explains the principles of operation of the AES and chapter 3 details how AES implementations are exploited in an attempt to find the information being protected.

Chapter 4 describes the process to ensure functional safety in the automotive industry, according to the international standard ISO-26262, and chapter 5 describes how this process was applied in this work to develop an AES IP core compliant with the highest automotive safety integrity level.

Chapter 6 explains the approach used to compare the security vulnerabilities of the original configuration and the developed safety enhanced configuration. Chapter 7 presents the results of this comparison and a discussion of their meaning.

Chapter 8 presents the conclusions about the work developed and provides a brief discussion of possible future work.

Chapter 2

Advanced Encryption Standard

This chapter presents an introduction to the Advanced Encryption Standard (AES), where the cipher and inverse cipher processes are explained. Then, a detailed description of the Galois/Counter mode is provided along with a summary of the advantages of this operation mode. At the end of the chapter, some applications of the AES for the automotive industry are identified, which justify the relevance of the algorithm for this sector.

2.1 Introduction

The AES is, since 2001, a federal government standard established by the U.S. National Institute of Standards and Technology (NIST) for the encryption of electronic data. The cryptographic algorithm used in the standard was the result of a four-year selection process, which involved cooperation between the U.S. government and private organisations from the entire world to obtain a publicly disclosed algorithm, available in a royalty-free basis worldwide. The minimum restrictions imposed on candidates were the use of symmetric key cryptography as a block cipher, with support for 128 bits size blocks and key sizes of 128, 192 and 256 bits. From the selection came out five finalists: MARS, RC6^{TM}, Rijndael, Serpent and Twofish. All finalists are iterated block ciphers, which means that the same transformations are applied a given number of times, either for encryption or decryption, where each one of these iterations is called a round. The evaluation criteria for the finalists were mainly focused on three domains: the algorithm's security, computational cost and implementation characteristics. Security was the most relevant factor, and all five algorithms met the security needs established by NIST. In the cost domain, computational performance metrics such as the efficiency in terms of speed and memory usage were assessed. Finally, the third domain evaluated the flexibility, the suitability for hardware and software implementations and the simplicity of the implementations. The final decision was for algorithm Rijndael, which adopted the designation of AES algorithm [3].

The AES allows both encryption and decryption operations. In the encryption, a message (plaintext) is converted into encrypted code (ciphertext), using a secret encryption key. In the

decryption, the plaintext is recovered from the ciphertext using the same key that was used for the encryption.

2.2 Algorithm Specification

The operations performed in the AES algorithm use blocks of 128 bits, where the input data is arranged in a two-dimensional array of bytes called State. The AES was designed to allow different block sizes than 128 bits, and this difference is reflected in the number of columns of the State since this number is given by the block size divided by 32. The number of rows is independent of the block size and is equal to four. Thus, in the AES, the operations are performed in a 4x4 matrix, where $s[r,c]$ identifies each position of the array. The conversions from the input array to the State and from the State, after the operations, to the output array are given by equations 2.1 and 2.2, respectively, which were extracted from [5].

$$s[r,c] = in[r+4c] \qquad \text{for } 0 \leq r \leq 3 \text{ and } 0 \leq c \leq 3 \tag{2.1}$$

$$out[r+4c] = s[r,c] \qquad \text{for } 0 \leq r \leq 3 \text{ and } 0 \leq c \leq 3 \tag{2.2}$$

The number of rounds in the encryption and decryption is dependent on the key size. For the key sizes 128, 192, 256 the number of rounds is 10, 12 and 14, respectively. Each round, with the exception of the final round, is composed of four functions: AddRoundKey, SubBytes, ShiftRows and MixColumns. The final round does not have the MixColumns function, which permits symmetrical executions for the encryption and decryption. Thus, the decryption can be considered as the inverse process of the encryption, as shown in figure 2.1.

For the sake of clarity, we consider an application scenario where the encryption of the plain data is done by a system (the transmitter) to send securely data to another system (the receiver) that will decrypt the received secure data to gain access to the plain data.

2.2.1 Introduction to Galois fields

In the data encryption process, the AES algorithm interprets each byte as an element from a finite field, also known as Galois Field (GF), which corresponds to a set with a finite number of elements were the mathematical operations (additions, subtractions, inversions and multiplications) can be easily and effectively computed when the operands are represented in binary forms. In the AES, the finite field considers polynomials of degree seven and is denoted by GF(2^8), where each set contains 256 elements. The AES only uses the addition and multiplication operations between finite fields. However, these two operations are more complex and require more steps than the same operations in the euclidean space.

The addition in finite fields is an exclusive-OR (XOR) between the coefficients of the corresponding powers of the operands. Due to the nature of the operation, the result, which is represented with eight bits, is never affected by overflow events.

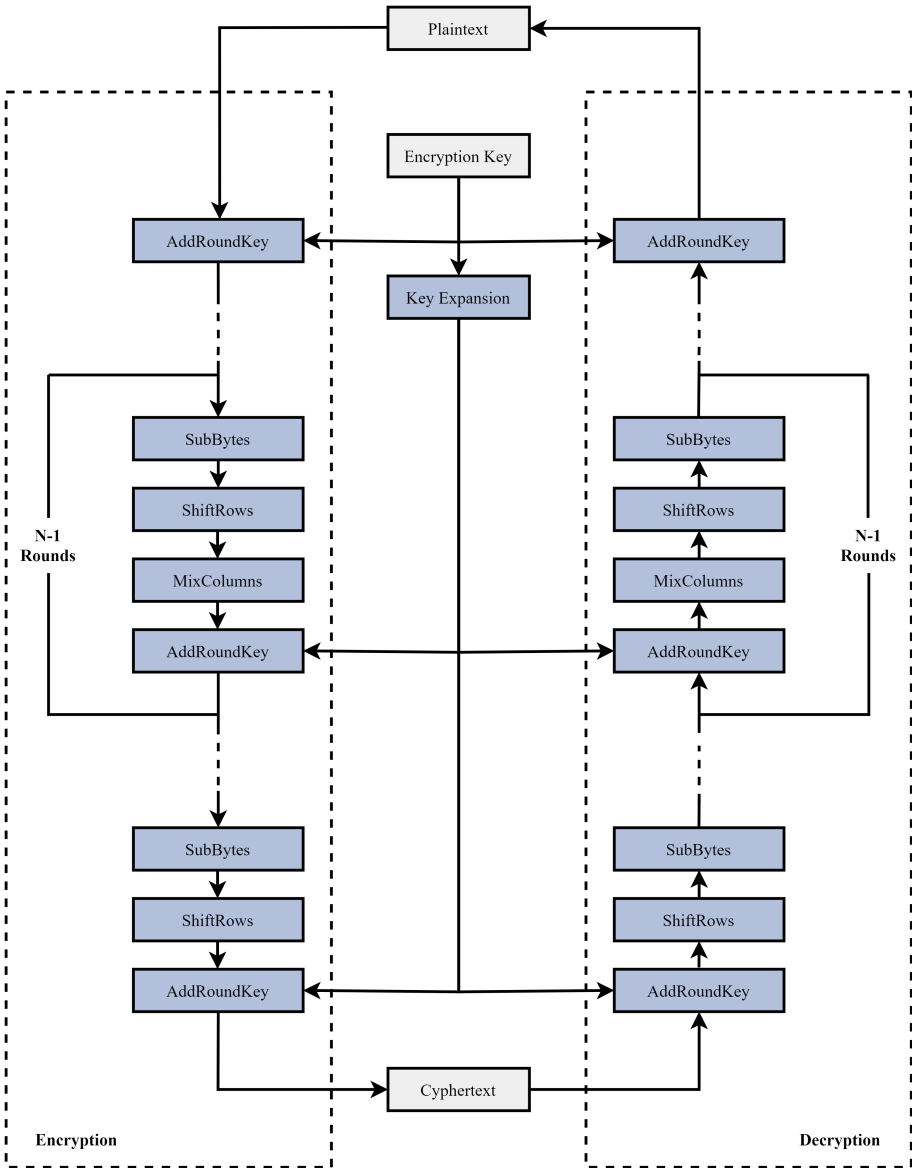

Figure 2.1: Encryption and decryption in the AES algorithm.

The multiplication in finite fields is more complex than the addition and the entire operation cannot be represented by a simple operator. The operation consists of the multiplication between the polynomials, modulo an irreducible polynomial of degree 8, which ensures that, as in the addition, the result is always a polynomial with a degree less than 8 and, therefore, can be represented using only one byte. Considering *a(x)* and *b(x)* as the operands and *m(x)* as the irreducible polynomial, the result *c(x)* can be determined by equation 2.3.

$$c(x) = (a(x) \cdot b(x)) \; mod(m(x)) \tag{2.3}$$

The irreducible polynomial *m(x)* is defined in the AES specification and is given by:

$$m(x) = x^8 + x^4 + x^3 + x + 1 \tag{2.4}$$

2.2.2 Cipher

Each AES round computed during the encryption phase consists of the four functions already mentioned, which describe three main transformations that are applied to the State matrix in each iteration:

- A key addition, in which a round key is XORed to the State in the function AddRoundKey.
- A byte substitution that performs a nonlinear transformation on each byte of the State using the function SubBytes.
- A bit diffusion of all State bits, first using the ShiftRows function and then the Mixcolumns function.

Each function processes a complete block of data (128 bits) at a time, but the functions are applied at a byte level, making the AES a byte-oriented cipher [6].

2.2.2.1 SubBytes

The SubBytes function uses a transformation table, called S-Box, to map and replace each byte of the State with a value in the table. This is a non-linear transformation and each one of the 256 possible inputs is one-to-one mapped with an output. The S-Box results from two steps: first from the multiplicative inverse in the finite field $GF(2^8)$ and then from an affine transformation in $GF(2)$. This affine transformation is expressed in equation 2.5, extracted from [5], where b_i is the i^{th} bit of the byte and c_i is the i^{th} bit of a fixed byte with value $\{63\}$, in hexadecimal. The b'_i is the new value that will replace b_i in the State.

$$b'_i = b_i \oplus b_{(i+4)mod8} \oplus b_{(i+5)mod8} \oplus b_{(i+6)mod8} \oplus b_{(i+7)mod8} \oplus c_i \tag{2.5}$$

The S-Box applied in the AES for the 16 bytes of the State is the same and is defined in the AES specification [5].

2.2.2.2 ShiftRows

The ShiftRows function applies a specific cyclic shift to each row of the State. This shift is equal to the index of the row minus one and cyclically moves the elements to the left. Thus, the positions on the first row remain the same (zero positions shift), the second row moves one position to the left and the third and fourth rows move two and three positions, respectively. This transformation is illustrated in Figure 2.2.

Figure 2.2: Representation of the ShiftRows transformation.

2.2.2.3 MixColumns

The MixColumns function operates over each column of the State, mixing each byte of the column with the other three bytes. Each column is converted in a polynomial and multiplied by a fixed 4x4 matrix, where each row is the polynomial given in 2.6 with a cyclic shift of three, two, one and zero positions to the left for the first, second, third and fourth row, respectively. Thus, the new values for each State column can be calculated using the equation in 2.7, extracted from [5].

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\} \tag{2.6}$$

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{for } 0 \leq c \leq 3 \tag{2.7}$$

2.2.2.4 AddRoundKey

The AddRoundKey function performs a bitwise XOR (which is the same as a Galois Field addition) between the State and a round key. Each round key is obtained from the original key in a process known as Key Expansion. This process generates a number of keys equal to the number of rounds plus one, where the first key is used at the beginning of the algorithm, before the initialisation of the first round. Each key expansion generates a 4-byte word (32 bits), denoted by $[w_i]$. These words are aligned in an array called key schedule. Considering that the algorithm requires four words of key data to XOR with each data block (that is also composed of four words), for N_r rounds, the dimensions of the key schedule is given by $4(N_r+1)$ rows per one column. The number of words in the key is denoted by N_k and, as said earlier, is equal to 4, 6 or 8, depending on whether the key size is 128, 192 or 256 bits, respectively.

The first words in the key schedule are the words of the original key, and so, the first key generated is the same as the original key. The generation of words for the remaining keys is given by one of the following cases:

1. If the word is in a position multiple of N_k, the result will be given by an XOR between the previous word modified ($w[i-1]$) and a round constant. The modification done to the word

[i-1] is a cyclic shift of one byte to the right followed by the application of the S-Box to the four bytes in the word. The round constant is given by $[x^{i-1},\{0,0\},\{0,0\},\{0,0\}]$, where x is equal to $\{02\}$ and x^{i-1} are powers of x.

2. Otherwise, the word $w[i]$ is simply the result of the XOR between the previous word ($w[i-1]$) and the word N_k positions before ($w[i-N_k]$).

2.2.3 Inverse cipher

The AES decryption is done with the inverse of the functions in the reverse order of the encryption. Thus, in the decryption, the ShiftRows, SubBytes, MixColumns functions are designated by InvShiftRows, InvSubBytes and InvMixColumns. The AddRoundKey function does not need an inverse version, as it applies only an XOR operation between two terms, and then, the function and its inverse are the same.

2.2.3.1 InvShiftRows

In the InvShiftRows, each row of the State is cyclically shifted in the opposite direction that was used during the ShiftRows function. Thus, the bytes in the first row remain in the same position and the bytes in the second, third and fourth rows shift one, two and three positions to the right, respectively.

2.2.3.2 InvSubBytes

The InvSubBytes apply the inverse of the S-Box that was used during the SubBytes function, to each byte. The inverse of the S-Box is generated using the inverse of the affine transformation stated above, followed by the multiplicative inverse in GF(2^8).

2.2.3.3 InvMixColumns

The InvMixColumns is also the inversion of the MixColumns function, where instead of the polynomial referred in 2.6, is used its inverse, described in 2.8.

$$a^{-1}(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\} \tag{2.8}$$

From this, results a different matrix from the one illustrated in 2.7, that gives rise to the expression in 2.9.

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{for } 0 \leq c \leq 3 \tag{2.9}$$

2.2.3.4 AddRoundKey

The operations of the AddRoundKey during the encryption and the decryption are similar with the exception for the key schedule, which in the decryption is in the reverse order of the one used during the encryption. This means that, before the decryption initialisation, it is necessary to compute all the *Nr+1* round keys and store them in memory. Then, the last computed key will be the first to be used.

2.3 Galois/Counter Mode of Operation

The AES, as a block cipher algorithm, has different modes of operation that were designed to complement the algorithm or adapt it for specific purposes. Initially, NIST proposed four modes of operation: the Electronic CodeBook (ECB) mode, the Cipher Block Chaining (CBC) mode, the Cipher Feedback (CFB) mode and the Output Feedback (OFB) mode. The Counter (CTR) mode was added later and received considerable interest in the industry due to its potential for applications in the domains of Network and IP Security [2]. These modes, however, only provide data confidentiality, preventing unauthorised access to the information, but do not guarantee authentication and data integrity. To establish a connection between both characteristics, another mode of operation called the Galois Counter Mode (GCM) has been proposed [7]. This algorithm results from an adaptation of the CTR mode, with the advantages of authenticating and certifying data integrity.

To perform the encryption, the GCM requires an additional set of inputs called counters, where each one of these counters must have a unique value for each plaintext block that is encrypted. Usually, only one initial random value for the counter is provided to the system and the other counters result from a sequential increment by one, for each new plaintext block [8].

The encryption process starts then with the increment of the initial counter and the resulting value is encrypted using the AES algorithm and XORed with the first block of plaintext. The subsequent blocks of plaintext follow the same procedure: before the XOR, the associated counters are incremented and then encrypted. In the decryption, the operations involved are the same but blocks of ciphertext substitute the plaintext blocks and the result of the operation are blocks of plaintext.

In the authentication, the objective is to protect another string known as Additional Authenticated Data (AAD), which is introduced in the system as an input and normally contains information about the plaintext and how it should be interpreted. The authentication requires a hash subkey (H) that results from the encryption of the zero block (a block composed by 128 zeros). The authentication is then executed through a chain of Galois field multiplications: for each block of plaintext, an intermediate authentication parameter g_i is generated from an XOR between the block of ciphertext from the encryption and the parameter g_{i-1} multiplied by the subkey H. The final result of this chain is the authentication tag (T), also called the Message Authentication Code (MAC). The authenticity of the data is verified in the decryption when the receiver computes the

authentication tag from the data received (*T'*) and compares it with the authentication tag from the encryption (*T*). If both match, the information was not modified in the transmission and the sender authenticity is guaranteed [6]. Figure 2.3 shows the block diagram of the GCM mode of operation, where the *CIPH* operation denotes the AES cipher, and the $mult_{gf}$ denotes a Galois field multiplication.

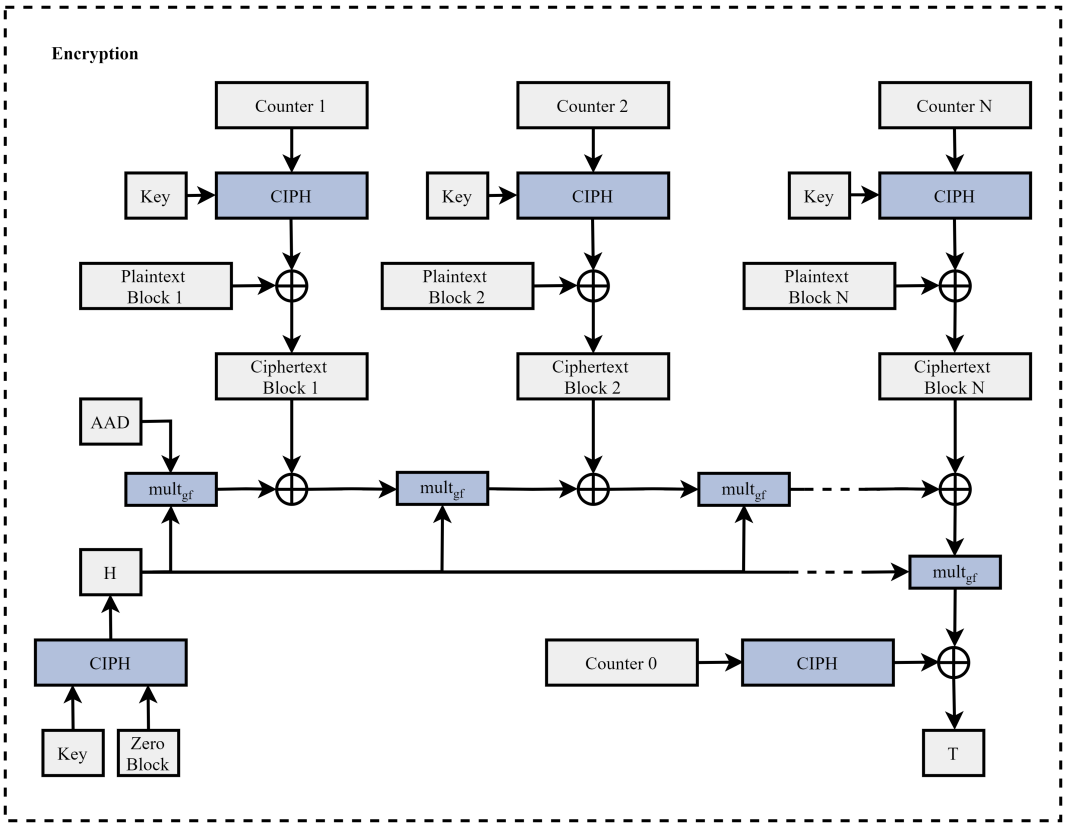

Figure 2.3: Galois Counter Mode (GCM).

The use in the GCM mode of the same encryption methodology as the CTR mode confers the GCM the same advantages that the CTR has when compared to the other modes of operation:

- A higher efficiency, since the encryption and decryption of data blocks is not dependent on the previous ones. This allows the computation in parallel of multiple blocks, which is particularly attractive for increasing the throughput in hardware implementations.
- The possibility for pre-processing, considering that the operation before each XOR is not dependent from the plaintext (for the encryption) or the ciphertext (for the decryption), allowing the computation of these values before the arrival of the plaintext. With pre-processing, the inverse blocks can be obtained from a simple XOR operation.
- The random access to each ciphertext (for the encryption) or plaintext (for the decryption) is possible and depends only on knowing the counter value that was used and the inverse block.

- The similarity between the encryption and the decryption that only differs in the type of input block (plaintext or ciphertext), allowing the use of the same logic for encryption and decryption.

2.4 Automotive Applications

In today's world, automobiles became tremendously essential in our quotidian, which caused the automotive industry to evolve in the way of transforming the traditional and purely mechanical vehicles, in environments capable of fulfilling their purpose of locomotion but with more comfort, entertainment, safety and with the ability to automate and integrate some of our daily tasks. This improvement was possible due to the technological innovation that allowed the integration of a large number of embedded systems called Electronic Control Units (ECU) in the vehicles and which, through their interconnection, made possible all the features that the modern vehicles provide us. The drawback of these improvements was the additional security risks that came from the inclusion and communication of multiple electronic devices, which led to the application of the AES to the automotive industry.

This increase in security awareness tends to continue to grow, as several developments (which require data protection) are still expected in the automotive market, as stated in [9], and which include:

- The usage of reprogrammable ECUs which implies the protection of the reprogramming process to avoid unintended access and, consequently, improper operation of these devices.
- The evolution of autonomous driving which implies the wireless communication of the cars with the involving environments and other cars, requiring strong protection of these communications.
- The evolution of the vehicles' infotainment systems which will probably evolve towards the remote access to contents and which will require updates to continue working according to the needs of those contents.
- The evolution of remote monitoring systems for vehicles such as highway tolls, tachographs and emergency systems that need to be protected to avoid non-legal uses.
- The advance of electronic systems for localisation of vehicles in case of theft, for example, which will imply strict measures of privacy and confidentiality.

In the literature there are already multiple approaches for some of these concerns that use the AES as a solution for data protection: in [10], a hardware implementation of AES is suggested for message encryption in real-time for on-board networking automotive systems; in [11] the AES algorithm is used to secure the exchange of multimedia messages (such as voice commands) between vehicles, using communication infrastructures that are located aside the roads (Road-Side units) and in [12] a remote keyless system for vehicles is proposed using the AES to secure the

communication between the key fob (handled by the driver) and the radio device located in the vehicle, which receives commands and distributes instructions for other components.

The problem of security has also been addressed by the automakers and the automotive engineering community, and diverse specifications, standards and guidelines have been proposed over the time, normally in the form of hardware security modules that were intended to be included in vehicles. Of all the proposals, two were considered as having good potential: the Secure Hardware Extension (SHE) specification [13] and the E-safety Vehicle Intrusion proTected Application (EVITA) project [14]. In both, the AES algorithm was suggested to ensure the privacy, integrity and authenticity of data.

Thus, the evolution of the automotive industry increasingly requires data protection in vehicles to prevent unauthorised access to data. Moreover, since the AES is currently the most recommended standard for this purpose, it is likely that its use will be more and more frequent in the future.

Chapter 3

Side-Channel Attacks

In this chapter, it is introduced the concept of side-channel attacks and explained how implementations are vulnerable to this type of attacks. A detailed description of side-channel attacks based on power consumption is also presented, as well as the attack methodologies that are normally used to extract information from the power consumption.

3.1 Introduction

The main purpose of cryptography is to maintain the data confidentially from third parties who do not have the right to access it. However, cryptography is an area that requires constant improvement since a system that is secure at this time, may not be in the future. The development of a cryptographic algorithm takes into account the vulnerabilities of the current algorithms and predicts some flaws that may arise from the evolution of the techniques and processing of computers but, at some point, the algorithms and respective implementations begin to evidence vulnerabilities that can be exploited in attacks. This implies that, over time, the algorithms need to be revised or, if not possible, replaced by new and more secure algorithms. In the case of the AES, its proposal came from the need to address existing vulnerabilities in the DES and therefore, it was developed so that obvious and simple attacks such as using brute force to test all the possible key combinations were not feasible [15].

The attention to flaws that exist in the extensive and rigorous processes of selection and refinement of the algorithms is not possible, however, in their implementations. The implementation of an algorithm can be done by diverse entities, using diverse techniques that can comply with the official specification of the algorithm, but which can be compromised by their operation. There are complementary standards such as the FIPS 140-3 [16] which define requirements and procedures that must be followed when implementing the algorithm, to reduce the obvious weaknesses that can compromise the security of the system. Nevertheless, it is not possible to cover and solve all possible flaws that may exist in the implementations and the solution for some of them may even enhance others. For this reason, most attacks do not focus on the vulnerabilities of the algorithm, but on the characteristics of the implementation, as it happens in the side-channel attacks.

The side-channel attacks are based on the leakage of physical information which results from the operation of the implementations and which is dependent on the input data, as illustrated in 3.1. This relation between the leakage information and the input data can be used to extrapolate the sensitive data of the system, such as the encryption key, giving access to all the encrypted data. Side-channel attacks typically focus on timing information, electromagnetic emanations, thermal radiation or power consumption patterns, since this is information that can be extracted externally from the device, using non-invasive methods, and without interfering with the device operation. These characteristics allow the execution of theses attacks without the need for expensive equipment and the possibility of harm to the device under attack [17].

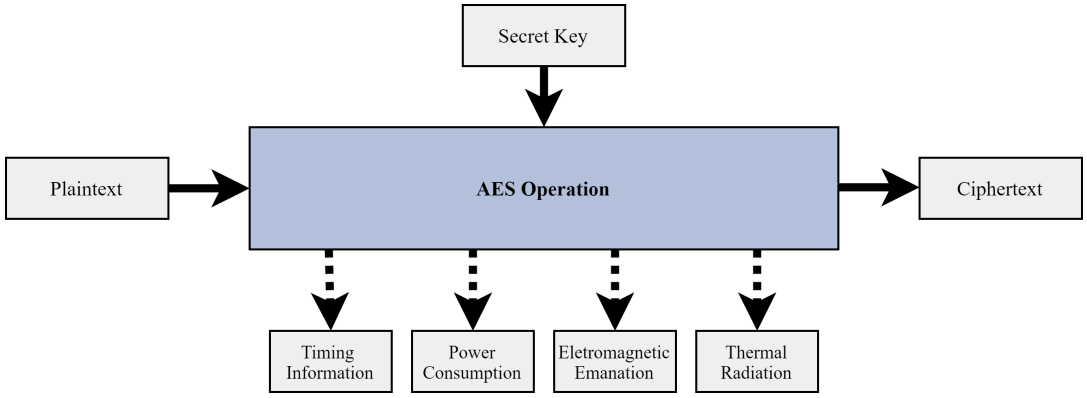

Figure 3.1: Side-channel leakage information.

3.2 Power Analysis

The power analysis takes advantage of the power consumption patterns that result from the operation of the device and its dependency on the instructions executed and the input values. From the power consumption traces of a device, it is possible to detect the different phases of encryption and from their variations is possible to extract information about the encryption key. The predominance of CMOS gates in digital circuits and their respective typical dynamic power consumption are the main justification for this variability in power traces [17].

The low complexity and low cost to execute these attacks, associated with its high potential of success when compared with other side-channel attacks, made these attacks the main focus of investigation leading to the development of multiple methodologies to extract information from the power traces. The most common attacks are the Simple Power Analysis (SPA), the Differential Power Analysis (DPA), the Correlation Power Analysis (CPA) and the template attacks.

3.2.1 Power Consumption in CMOS Devices

The power consumption in CMOS devices can be divided in two main components: the static and the dynamic power consumption. The static power consumption corresponds to leakage currents that flow in the device even when it is not switching. These leakage currents are essentially due to

subthreshold leakage and gate leakage currents. The dynamic power consumption comprehends the energy consumed charging and discharging the circuit capacitances during the switching activity and the small short-circuit currents that flow through the low impedance paths that are created between the supply and the ground when switching [18]. The equation 3.1 specifies the components of the total power consumption which includes the contribution of the three types of dissipation described. In the switching component, the C_L is the load capacitance of the circuit, f_{clk} is the clock frequency and the $\alpha_{0\to1}$ is a factor associated with the node transition activity. In the short-circuit component the I_{SC} is the average short-circuit current during a transition and in the leakage component $I_{leakage}$ is the leakage current.

$$P_{total} = P_{switching} + P_{short\text{-}circuit} + P_{leakage} = \alpha_{0\to1} \cdot C_L \cdot {V_{dd}}^2 \cdot f_{clk} + I_{sc} \cdot v_{dd} + I_{leakage} \cdot V_{dd} \quad (3.1)$$

Each one of these components of the power consumption in CMOS devices will then be reflected in the power traces. The power traces can also be divided into components, which are: an operation-dependent component (P_{op}), a data-dependent component (P_{data}), a constant component ($P_{const.}$) and a noise component (P_{noise}). The $P_{switching}$ and $P_{short\text{-}circuit}$ of CMOS devices can be associated with the P_{op} and P_{data} and the $P_{leakage}$ can be associated with the P_{const}. Thus, since only the P_{op} and P_{data} can provide valuable information for power analysis attacks, only the components of $P_{switching}$ and $P_{short\text{-}circuit}$ can expose sensitive information of the implementations.

3.2.2 Power Analysis Attacks

3.2.2.1 Simple Power Analysis

The Simple Power Analysis, as the name suggests, is a simple way to extract valuable information from the power traces through their direct interpretation or simple manipulation techniques as comparing pairs of power traces. Power consumption measurements that do not present high levels of noise, often provide power traces where characteristics of the device, the algorithm structure, data-dependent power variations and other operation characteristics are easily identified and which can be used to infer sensitive information of the system [19].

The SPA attacks are really complex to perform through a black box evaluation, that is, situations in which the internal implementation and algorithm operation are unknown. However, for situations where an attacker has detailed knowledge of the algorithm, it is possible, for example, to detect all phases of the encryption and extract the hamming weight of the computed data values [20]. Even in cases where the SPA is not successful and is not capable of completely reveal the expected information, they can still be used to facilitate the execution or help in preparation for other attacks, allowing the identification of the relevant power consumption samples and the points in time that are conducive to the application of attacks. As an example, in [21], a SPA attack is used to considerably reduce the number of keys that needs to be considered in a brute-force search to find out the secret encryption key.

3.2.2.2 Differential Power Analysis

The Differential Power Analysis is a more effective and common attack than the SPA since due to the use of statistical analyses it allows the extraction of information about the encryption key even in very noisy environments or in cases where the SPA does not evidence any relevant characteristic in the power traces. The DPA comprehends the same two phases of attack than SPA (data collection and data analyses), but instead of focusing only in one power trace, it usually requires the collection of multiple power traces.

To execute a DPA attack, it is necessary to understand in detail the complete operation of the algorithm or, at least, the phase of encryption where the attack will be executed, to identify a relationship between the secret encryption key and the collected power traces. A possible solution for this is to explore the known data for which we have access (the input data) and, from the knowledge we have from the algorithm, make predictions about the expected data that is directly associated with the power traces. This expected data is the basis for the manipulation of the power traces for DPA, and so it is normally called target data. This relation is shown in Figure 3.2.

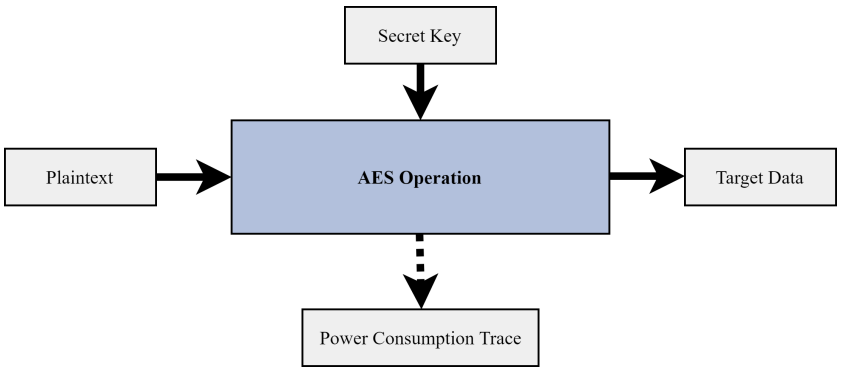

Figure 3.2: Target Data DPA Attack.

Since an attacker is only capable of modifying the plaintexts for the encryption and has no control over the encryption key, the calculation of the target bits is normally based on key guesses. The target bits are used to divide the power traces collected from simulations with the known plaintexts in groups, where each group corresponds to a different value of the target. Since the target bits were calculated assuming a key guess, if the key guess is correct, the power traces inside the same group shall have the same target bits and so, shall have a common point in the power traces. If the key guess is wrong, the division of the power traces was made using an incorrect assumption and therefore, the power traces will probably have no common point. The use of multiple power traces for each group aims to eliminate variations in power traces due to non-relevant data, through an average of the power traces. So, for each group, the final result is one single averaged power trace. The subtraction of those averaged power traces from different groups shall demonstrate significant spikes due to the different common points inside each group, if the correct key guess was used. If the key guess was wrong, the difference between the averaged power traces would be a flat line because as there was nothing in common between the power traces, the variations were removed with the average.

The more power traces are used in the DPA attacks, the clearer the correct key guess will be. In [22] a successful DPA attack in an ASIC implementation without any protection against DPA required 25000 power traces but when the implementation had countermeasures to reduce the vulnerability to DPA attacks this number increased to 130000 power traces. Thus, although DPA attacks constitute a serious risk to the security of the system, execute them is not easy and requires a high investment in terms of time for attack optimisation and data extraction.

3.2.2.3 Correlation Power Analysis

The correlation power analysis is a variant of DPA that in some cases can exploit information in power traces with a higher efficiency than DPA but which also requires more detailed knowledge and comprehension about the implementation under attack, being essentially used in the white-box analysis where the device is well known. The CPA requires the development of a model of the device's power consumption, for a specific and small target execution sequence, which needs to be dependent from the values of intermediate results in the algorithm execution [19]. The model will then be used for comparisons with the power consumption patterns of the device under test. If the expected target for the model and the real device is the same, it is expected a high correlation between both power traces.

For hardware implementations, two power models are well known: the hamming distance power model [23] and the hamming weight power model [24]. The hamming distance model estimates the power consumption based on the transitions in a digital circuit, using the hamming distance between the value before and after the transition. Then, in this model, a transition from $0 \rightarrow 1$ is considered to consume the same amount of power as a transition from $1 \rightarrow 0$. In the Hamming weight model, the power consumption is calculated based on the number of bits that are set to one. The hamming distance model is normally associated with a better estimation of CMOS circuits but has the disadvantage of requiring a better knowledge of the implementation.

3.2.2.4 Template Attacks

The template attacks consist of the utilisation of an experimental device to create a set of templates that provides information about the expected power traces and noise characterisation of the device under test. Then, in a template attack, it is necessary that an attacker has access to two devices: the profiling device and the attacked device.

The execution of a template attack comprehends two main phases: the training phase and the attack phase. In the first, the profiling device is used to collect the leakage information from the encryption of random keys and plaintexts. For each key value, a model of power consumption is constructed using the set of power traces that had the same encryption key but different plaintexts. This model is composed by the average of the power traces and their covariance matrix. In the attack phase, the power traces from the encryption of multiple plaintexts in the device under attack are collected and using the Baye's rule are matched with the models created in the first phase [25].

One of the main advantages of the template attacks is that the time-consuming model building phase only needs to be completed once for a specific device. After that, the same templates can be used to execute multiple attacks on identical devices.

Chapter 4

Automotive Functional Safety

This chapter describes the vision of functional safety according to ISO-26262 and presents the different phases in the analysis and development of functional safety. Then, the safety levels required by some automotive systems are shown and it is explained how the compliance with these safety levels can affect the security of the system.

4.1 The ISO-26262 standard

The concept of functional safety is one of the bases of the automotive industry and one of the concerns that most contributes to the methodologies and approaches followed during the development phase. Its purpose is essentially to map the desired safety and integrity goals for the system into requirements that can be applied to the different architectural components in the main system. The ISO-26262 is the current international standard for safety compliance in the automotive industry.

The ISO-26262 standard, first released in 2011, resulted from an adaptation of the IEC-61508 series of standards with the scope of creating a specific safety standard for electrical and electronic systems within the road vehicles. The standard was developed as a guide to mitigate existing functional safety risks in the systems, proposing for that in [26]:

- A complete safety lifecycle suitable for the automotive sector, including support for the adaption of the different activities executed in each phase.
- A scale of automotive safety integrity levels (ASILs) based on a system risk assessment.
- Requirements to avoid unreasonable residual risk, using for it the different integrity levels defined.
- Requirements to cover the functional safety aspects of each phase of the lifecycle.
- Requirements for the relations between the different entities involved in the system lifecycle.

The ISO-26262 uses a risk-based approach to ensure the safety of the system throughout its lifecycle [27]. The standard addresses considerations about random hardware failures and systematic failures that are inherent to components and defines the approach to create protections that prevent the hazards from those failures.

4.1.1 Safety Lifecycle

The safety lifecycle defined in the ISO-26262 comprehends all the main phases of the product, from the concept and development phases to its service and decommissioning. Its objective is to define guidelines for the safety activities, ensuring their correct planning, coordination and monitoring throughout their progress. The lifecycle diagram defined by [26] in the part two is shown in Figure 4.1. The objectives of this project include the concept phase and part of the development phase.

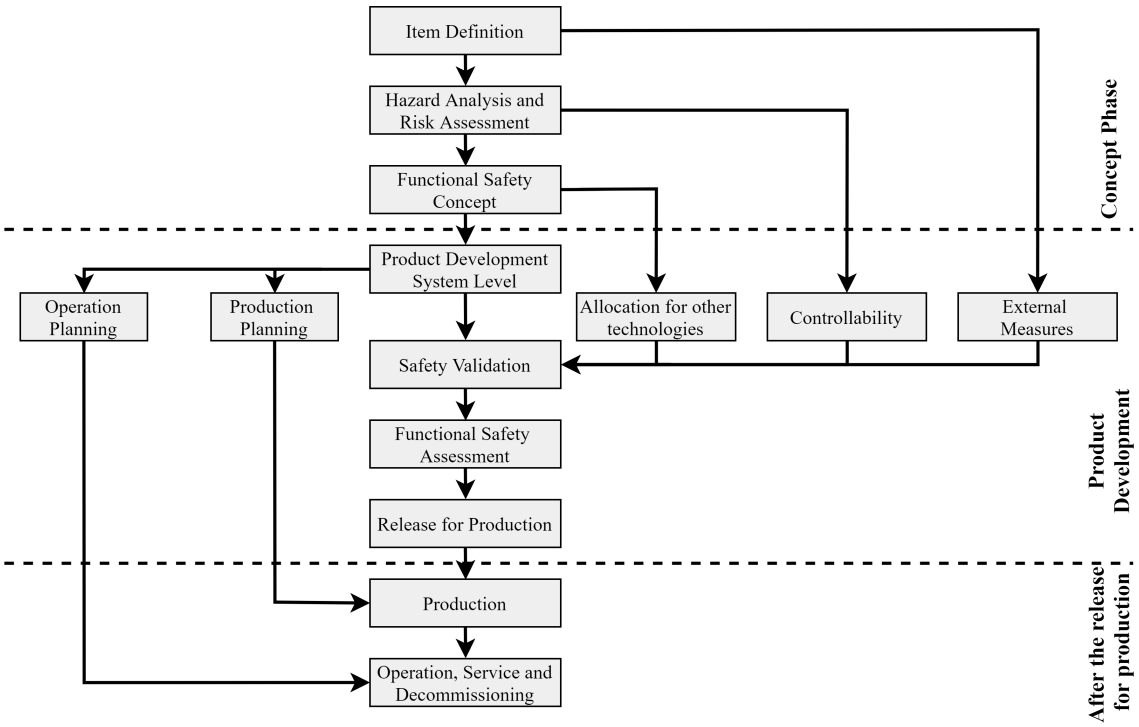

Figure 4.1: Safety Lifecycle.

4.1.2 Item Definition

The item definition in functional safety is one of the primary steps and corresponds to the development of a complete definition and description of the item to be developed, along with its dependencies and interactions with the environment for which it is intended. This is a fundamental step to properly understand the item and allow the subsequent phases of the lifecycle.

The definition of the item shall include non-functional requirements, legal requirements, and reference to other national and international standards that the item must comply with, considering the environment in which it is inserted. In this step, all the knowledge acquired from the behaviour of similar functions, systems or elements shall be identified, as well as assumptions on the behaviour and consequences of failure modes and hazards already known.

The definition of the item's interfaces and interactions with other elements shall consider the assumptions about the item's behaviour in the vehicle, the requirements imposed by other items and the allocation and distribution of functions in the involved systems. The operating scenarios shall be considered if they can impact the functionality of the item.

4.1.3 Hazard Analysis and Risk Assessment

The Hazard Analysis and Risk Assessment (HARA) intends to identify and classify hazardous events that can result from the malfunction of the item, using the item definition previously mentioned. The idea is to define safety goals for the system that act in the prevention and mitigation of hazardous events, avoiding unreasonable risks.

The HARA starts with an identification of the malfunctions that can result from the operational situations and operating modes, both for the cases where the vehicle is correctly handled or cases of improper use. Then, exploring all the possible combinations of malfunctions, situations and environmental conditions, it is necessary to derive the possible hazards at the vehicle level. Each hazard is classified considering three components: its severity (S), probability of exposure (E) and controllability (C). The quantification of these components comprehends the following logic:

- The severity can be quantified in four levels, from S0 to S3. The S0 level is used only when the hazard analysis and risk assessment proves that the resulting damage is limited to material goods and the S3 corresponds to injuries that may be fatal.

- The probability of exposure considers five levels from E0 (zero probability) to E5 (high probability), which are based on a rationale defined for each hazardous event, which considers a representative sample of operational situations that varies according to the target market.

- The controllability is an estimation of the control that a person related to a hazardous event has to prevent specific harm that arises as a consequence of that event and can be quantified in four levels (C0 to C3). This estimation assumes that the specified person is in a condition suitable for driving, including compliance with the legislation in force.

For each hazardous event, an Automotive Safety Integrity Level (ASIL) is determined according to the classifications of the measures mentioned above. The ISO-26262 considers four possible ASILs: ASIL A, ASIL B, ASIL C and ASIL D, where ASIL D corresponds to the highest safety integrity level and ASIL A to the lowest. Another class QM (Quality Management) is used for events where the safety requirements and risks can be managed using only quality processes and

therefore do not demand requirements for compliance with ISO 26262. Table 4.1 shows the ASIL corresponding to each combination of the classifications of the three measures.

	S1			**S2**			**S3**		
	C1	**C2**	**C3**	**C1**	**C2**	**C3**	**C1**	**C2**	**C3**
E1	QM	QM	QM	QM	QM	QM	QM	QM	A
E2	QM	QM	QM	QM	QM	A	QM	A	B
E3	QM	QM	A	QM	A	B	A	B	C
E4	QM	A	B	A	B	C	B	C	D

Table 4.1: ASIL determination based on the Probability of Exposure (E), Severity (S) and Controllability (C).

Finally, each hazard with an evaluated ASIL is associated with a safety goal. The safety goals are top-level safety requirements that are associated with functional objectives and should not be expressed using technical solutions. For hazards that result in similar safety goals, they shall be combined in only one safety goal. The ASIL of the combination of multiple safety goals will be the highest in the combination.

4.1.4 Safety Element out of Context (SEooC)

The development of elements for the automotive industry is not always direct to a specific vehicle or purpose. There are cases in which the development of an element intends to satisfy multiple applications or different customers. These generic elements are defined as Safety Elements out of Context (SEooC). Examples of elements developed as SEooC usually include hardware ICs and IPs such as microcontrollers, sensors, peripherals or even software components [26].

In these generic cases, it is not possible to perform a complete hazard analysis and risk assessment, as described in the previous section, since there is no knowledge about the hazards at the item level [28]. The solution is to develop the element based on two types of assumptions:

- Assumptions about the safety requirements for the element under development that assume possible use cases in which the element can be integrated. For this assumption it is necessary to consider the highest ASIL for which the element will be used, to ensure that it meets the imposed safety needs in all situations.

- Assumptions about the external design context and environmental characteristics into which the element will be inserted. These assumptions must ensure that the SEooC is consistent with the requirements of the context in which it is used, at any level.

Since the development of a SEooC does not follow a straightforward approach of the ISO-26262 as a normal element and requires several assumptions, it will likely be necessary to tailor the applicable safety activities of the lifecycle. However, such tailoring can not be used to neglect any step of the lifecycle, as stated in part ten of the standard.

4.1.5 Safety Concepts

Once the safety goals for the system are defined, it is necessary to plan which safety measures will be implemented within the safety-related product to ensure the safeguard of the safety goals. To this end, the ISO-26262 refers the development of two different perspectives of safety concepts: the Functional Safety Concept and the Technical Safety Concept. The Functional Safety Concept intends to derive functional safety requirements from the safety goals and allocate them to the preliminary system architectural design, while the Technical Safety Concept specifies the technical safety requirements and their aggregation to the system architectural design, providing a rationale about why the safety requirements identified in the Functional Safety Concept are fulfilled with the technical decisions presented.

4.1.5.1 Functional Safety Concept

The Functional Safety Concept uses as inputs the item definition, the HARA and the system architectural design to derive at least one functional safety requirement from each safety goal, considering the system architectural design. Moreover, it serves as the basis for the Technical Safety Concept. The functional safety concept must be accompanied by a verification report which shall provide evidence about its coherence, compliance with the safety goals identified in previous phases and capacity to mitigate and avoid hazardous events.

The functional safety requirements shall take into consideration the eventual operating modes, fault tolerant time intervals, safe states, emergency operation time intervals or functional redundancies that may exist. This process of definition of requirements can be supported by available safety analyses such as Failure Mode and Effect Analysis (FMEA), Fault Tree Analysis (FTA) or HAZard and OPerability analysis (HAZOP) [26]. Any safety requirements defined according to ISO-26262 (and not only the functional safety requirements) shall be unambiguous, atomic (meaning it cannot be possible to divide them into more than one requirement at the considered level), consistent, feasible, achievable and especially they need to be verifiable.

The Functional Safety Concept requires a detail that a SEooC normally does not have. Then, in the development of SEooC, the Functional Safety Concept is usually one of the phases in the safety lifecycle that needs to be tailored. The complete Functional Safety Concept is typically the responsibility of other organisations in the supply chain, such as the tier1 suppliers who provide components for the original equipment manufacturer.

4.1.5.2 Technical Safety Concept

The Technical Safety Concept intends to define the technical safety requirements (which provide a technical perspective for the implementation of the functional safety requirements) and the respective system architectural design, ensuring the fault avoidance and safety integrity and operation aspects. The resulting system architectural design shall fulfil not only the allocated technical safety requirements but also the non-safety requirements [26].

The technical safety requirements shall consider all the safety-related dependencies and constraints of the elements, as well as all the external interfaces (if they exist) and available configurations options for the system operation. The technical safety requirements shall specify safety mechanisms that act in the detection, prevention or mitigation of failures that could lead to violation of safety goals.

Safety mechanisms are used to maintain the intended functionality of the system or conduct it to a safe state when a safe operation can no longer be ensured. In part five of [26] some examples of safety mechanisms are presented. However, the ISO-26262 does not impose any specific guideline in the selection of safety mechanisms. The only requirement is that the safety mechanism meets the needs that led to its implementation.

After the completion of the Technical Safety Concept, three work products shall be available for the subsequent phases of the development process: a technical safety requirement specification, a system architectural design specification and a verification report. This report must ensure that the system is capable of achieving the desired ASIL, that there is consistency between all the work products provided and that everything is in line with previous development phases.

4.1.6 Hardware Evaluation Metrics

The hardware evaluation is intended to evaluate the effectiveness of the architecture design against random hardware failures and complement it with an assessment of the residual risk of violation of safety goals. A safety-related hardware element in a system can be subject to four types of faults, which are defined in part one of [26] as:

- Single-point faults: faults in a hardware element that is not protected by any safety mechanism and which have a direct implication in the violation of a safety goal.
- Residual faults: faults in a portion of a hardware element where there is no safety mechanism implemented and which have a direct implication in the violation of a safety goal.
- Safe fault: a fault with a reduced effect on increasing the probability of violating a safety goal.
- Multiple-point fault: a fault that in combination with multiple independent faults can have a direct implication in the violation of a safety goal.
- Latent fault: a multiple-point fault that is not detected in a given time interval either by a safety mechanism or by the vehicle driver.

Each one of these faults corresponds to a specific failure rate (λ) denoted by λ_{SPF}, λ_{RF}, λ_{S}, λ_{MPF}, $\lambda_{\mathrm{MPF\,DP}}$ and $\lambda_{\mathrm{MPF\,L}}$, for single-point faults, residual faults, safe faults, multiple-point faults, detected multiple-point faults and latent faults, respectively. The total failure rate λ results then in the equation in 4.1.

$$\lambda = \lambda_{\mathrm{SPF}} + \lambda_{\mathrm{RF}} + \lambda_{\mathrm{MPF\,DP}} + \lambda_{\mathrm{MPF\,L}} + \lambda_{\mathrm{S}} = \lambda_{\mathrm{SPF}} + \lambda_{\mathrm{RF}} + \lambda_{\mathrm{MPF}} + \lambda_{\mathrm{S}} \tag{4.1}$$

4.1.6.1 Hardware Architecture Metrics

For the assessment of the safety architecture, the ISO-26262 defines two hardware-specific metrics: the single-point fault metric (SPFM) and the latent-fault metric (LFM). The SPFM is a measure of the robustness of the item to single-point and residual faults while the LFM is a measure of the robustness of the item to latent faults where the recognition of the fault does not happen before the violation of the safety goal, neither by safety mechanisms nor by the driver and nor by the design.

The comparison of the SPFM and LFM values to reference target values is required to validate each safety goal. The reference values may be derived from application of the metric to a similar well-trusted architecture design or from the table 4.2, which values were extracted from part five of [26]. These metrics are only applicable for ASILs B, C and D.

4.1.6.2 Random Hardware Failure Metric

The ISO-26262 also establishes that, in addition to the hardware architecture metrics, the overall residual risk of violation of each safety goal shall be calculated. For that, two alternatives are presented in the standard: a Probabilistic Global Metric for Random Hardware Failures (PMHF) and an evaluation of each cause of safety goal violation (EEC) considering each individual hardware part.

The most common is the PMHF, which takes into account the single-point faults, the dual-faults, the residual faults and the coverage of the safety mechanisms to quantify the maximum probability of violation of each safety goal. The resulting value needs to be within certain ranges to ensure that the system meets the expected ASIL. The standard provides a conservative reference for these ranges, which are presented in table 4.2. The values in the table for the PMHF metric are defined in terms of Failures in Time (FIT) which is the number of failures that can be expected in one billion (10^9) hours of system operation [27].

ASIL	SPFM	LFM	PMHF
B	$\geq 90\%$	$\geq 60\%$	< 100 FIT
C	$\geq 97\%$	$\geq 80\%$	< 100 FIT
D	$\geq 99\%$	$\geq 90\%$	< 10 FIT

Table 4.2: Reference target values defined by ISO 26262 for SPF, LF and PHMF metrics.

The PMHF as well as the hardware architecture metrics can be calculated through a Failure Modes Effects and Diagnostic Analysis (FMEDA).

4.2 Functional Safety and Security

As shown in section 2.4 there are several applications for the AES that intend to add security to the system, but which, for the automotive market, need to comply with the safety guidelines addressed

in 4.1. As the use of this type of system in the automotive sector is still starting, there is no public information about the common ASIL ratings for these devices. However, from [29] it is possible to extract some examples of systems that can be associated with AES implementations due to the need to protect the respective data flows. Some of these examples are shown in 4.3.

System	**Possible Failure**	**ASIL**
Instrument Cluster	Loss of Critical Data	B
Engine Management	Unwanted Acceleration	C/D
Radar Cruise Control	Inadvertent Braking	B
Anti-lock Braking	Unintended Full Power Braking	D
Electric Power Steering	Self-Steering	D

Table 4.3: Examples of ASIL ratings for different vehicle systems.

From the examples, we conclude that the AES is related to components that require high ASIL ratings. This means that the AES implementations themselves are safety-relevant components which will require high levels of protection to avoid hazardous events in any of the systems to which they are related. The major concern is that improving the safety of the system will require to add safety mechanisms to the implementation that, by increasing its consumption, may also increase the leakage of side-channel information and affect the vulnerability of the implementation to security attacks. The effect of safety enhancements on security has not been too explored until now and, therefore, it is not fully defined whether this relation between safety and security exists or not.

This concern is so important because if an IP core that intends to provide security for a system to protect critical safety elements in a vehicle has vulnerabilities, then they can be exploited in the attempt to create an abnormal behaviour of the system, such as a safety failure [30]. A safe system must be secure, but a secure system does not need to be safe.

Thus, if by improving the safety of a system, we increase the vulnerability of a security attack, we can be protecting some elements in terms of safety but creating safety concerns in other elements. Then, if the automotive industry wants to evolve in the direction of improving security in the vehicles, it is necessary first to fully understand this relationship between safety and security.

Chapter 5

AES-GCM Safety Improvement

In the previous chapter, it was discussed some concepts about functional safety and the phases in the process to achieve a given ASIL certification for the system, according to ISO-26262. In this chapter, it is described how the safety analysis in the IP core under study was conducted and which safety improvements were required to achieve the defined ASIL. The results of the system evaluation using the hardware safety metrics specified in the ISO-26262 are shown at the end of the chapter.

5.1 AES-GCM IP Core

The first step when performing a safety analysis is to fully describe the item under development. In this case, the development of the item was not done from scratch but using an existing IP core from Synopsys which is an implementation of the AES algorithm in the GCM operation mode. The development of these IP Cores is usually done in order to make them generic so that it is possible to use them in different applications and for various purposes. For that, they allow varied configurations and have other functionalities, besides the main ones, which make them adaptable. For this IP core, the complete list of functions is:

- The encryption and decryption of data according to the AES algorithm specification.
- The Message Authentication Code (MAC) computation according to the specification of the GCM mode of operation.
- The key expansion to create the round keys for the algorithm.
- The context switching to allow it to operate with different contexts at the same time.
- The context interleaving to allow the context switching in the middle of an encryption or decryption.
- A configuration functionality, to configure the operating characteristics to the needs of the context.

This IP Core receives the input data and the control commands for the operations through a specific interface which is controlled by an Ingress FIFO. The encryption keys are loaded into the system through an interface other than the one used for input data. The reason for this is that a normal user of the IP core is not expected to have access to this interface since the key load must be done only by the system developer so that it can be kept secret. The output of data from the system is controlled through an Egress FIFO. The block diagram of the original design is illustrated in Figure 5.1.

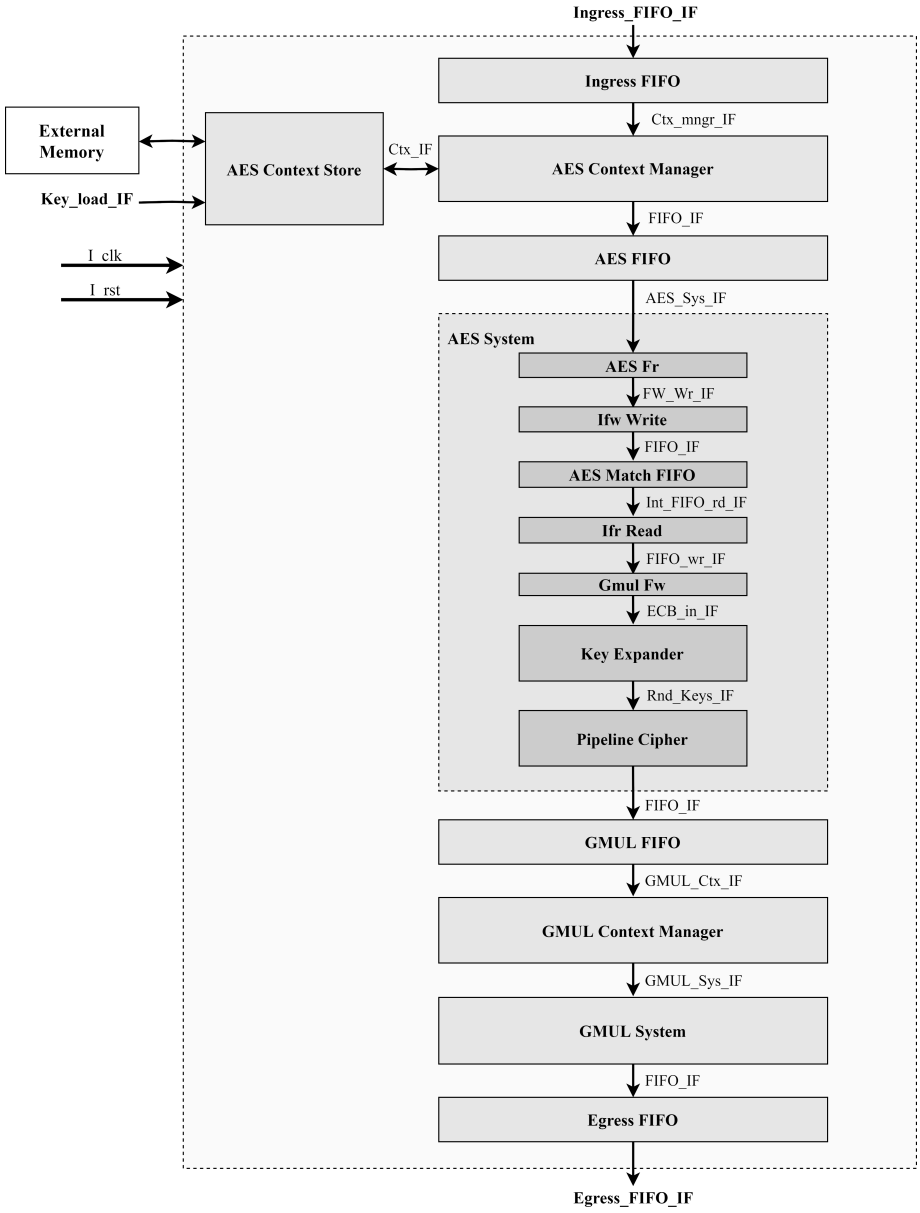

Figure 5.1: AES-GCM IP Core Block Diagram.

5.1.1 Part Allocation for each Function

The block diagram permits the identification of the purpose of each block and the association with the functions executed by the core. The decomposition and allocation of each part are necessary for the subsequent phases of the safety analysis to clearly identify how each part of the design can fail and what is the correct procedure to protect it. The function of the main blocks in the design is:

- AES System: is responsible for the entire process of encryption and decryption, and includes the following blocks:
  - Key Expander: makes the key expansion for the rounds of the algorithm.
  - Pipeline Cipher: contains the logic for the rounds in the encryption and decryption process.
- GMUL System: computes the MAC for data authentication.
- Context Managers and the Context Store: are related to the functionality of context switching and interleaving.
- The remaining blocks, such as the FIFOs, are essentially used to store and control the progression of data throughout the operations.

Although it is possible to isolate the purpose of each block, they cannot be considered as independent inside the system since these functions are related between them. As an example, the incorrect execution of the encryption function will affect the result of the MAC computation. Furthermore, the configuration functionality is common to all the blocks in the system.

5.1.2 Core Configuration

When operating with this IP core, there are some configurations which need to be defined in advance. The criteria to choose these configurations were based on the most common implementations used in the literature and the configurations that best suited the analysis environments for the subsequent phases.

The configuration of the operation was defined for keys with 128 bits, which means ten rounds of encryption in the AES algorithm. For each one of these rounds, one dedicated instance is used, although for area constrained implementations instance sharing can be used. The S-Box required for the encryption/decryption was implemented using a Look-Up Table (LUT) and it was only set up one Key Expander. In terms of the input data, the maximum allowed length for messages was 2^{14} bits and the maximum number of input contexts allowed was four.

5.2 SEooC Definition

The safety enhancement to which this IP core was subject was not targeted at any specific use case or application. The objective was to maintain it versatile so that it could integrate, for example,

the applications shown in 2.4 or others. Then, this element was considered as a Safety Element out of Context (SEooC) for safety analysis.

Considering the related environment of application of this IP core, it is evident the high safety requirements that it needs to satisfy to avoid hazardous events and the interfaces and dependencies with high critical safety external elements with which it is associated. Although for a SEooC a Hazard Analysis and Risk Assessment (HARA) does not apply because the exact application of the item is not defined, think about the three components of risk (probability of exposure, severity and controllability) helps in determining the ASIL rating for the element. Considering the most restrictive cases of application for the element, the severity of the hazardous events can be fatal (severity level three), the probability of exposure can be high (exposure level four) and the events can be uncontrollable for the driver (controllability level three). Thus, the proper rating for an AES implementation seems to be the highest certification in terms of safety: an ASIL-D.

An ASIL-D architecture is usually associated with a large area due to the substantial number of safety mechanisms that it implies. This means that an ASIL-D, the highest certification, is not always the best solution for all the systems. For cases where the area is a critical factor, and the application itself does not have high safety requirements, the option goes typically for lower ASIL ratings. In this case, the safety level of the system was considered to have a higher priority than the final area and, therefore, the latter was not considered as a restriction.

5.3 Functional Safety Concept

The objective of the Functional Safety Concept is to define the functional safety requirements to comply with the defined ASIL and establish the starting point for the Technical Safety Concept. Then the first step is to identify which of the functions of the core are safety-relevant and from them define the safety goals for the system. The functional safety requirements, which are the main objective of this concept, can then be derived from the safety goals.

5.3.1 Safety Related Functions

The designation of a safety-relevant function means that its operation or incorrect output can have a direct impact in the safety of the system. Then, the safety-related functions of the core are the encryption, decryption, MAC authentication, key expansion and context switching. The context interleaving was excluded from the safety analysis since its operation was not considered critical for safety purposes and was then deactivated in the ASIL design.

5.3.2 Safety Goals

The definition of the safety goals is related to the safety-related functions and in specific those which have a direct impact on the outputs of the system. These safety goals must be defined from a high-level view and in such a way that, if they are met, it can be ensured that the potential safety risk of the system is reduced to a tolerable level. The safety goals for this IP core are defined in

table 5.1, where the corresponding ASIL assigned was the same ASIL that was defined for the overall system. This means that all safety goals are equally relevant for achieving the safety level in the final project.

SG ID	SG Description	ASIL
SG 1	AES-GCM shall detect faults that lead to errors in the encryption or decryption functions	D
SG 2	AES-GCM shall detect faults that lead to errors in the MAC computation	D
SG 3	AES-GCM shall detect faults that lead to errors in the key expansion function	D
SG 4	AES-GCM shall detect faults that lead to errors in the configuration space	D

Table 5.1: Safety Goals (SG) for the AES-GCM ASIL-D.

5.3.3 Functional Safety Requirements

From the previous safety goals, it was possible to derive the respective functional safety requirements, considering a more detailed examination of the safety-related functions, the dependencies with other elements, the faults that may affect the design and the expected behaviour of the system in case of fault detection. At this level, it is not expected yet the identification of the technical solutions that will permit achieve these requirements. This means that similar functional safety requirements in the design can lead to different technical solutions. The Functional Safety Requirements are expressed in Table 5.2.

SG ID	FSR ID	FSR Description
SG 1	FSR 1.1	AES-GCM shall detect transient faults that lead to encryption or decryption errors
	FSR 1.2	AES-GCM shall detect permanent faults that lead to encryption or decryption errors
	FSR 1.3	AES-GCM shall protect the integrity of all FIFO buffers accesses related to the encryption/decryption operations
	FSR 1.4	AES-GCM shall report faults that lead to encryption or decryption errors
	FSR 1.5	AES-GCM shall provide a safe state control mechanism for the encryption and decryption operations
SG 2	FSR 2.1	AES-GCM shall detect transient faults that lead to errors in the MAC
	FSR 2.2	AES-GCM shall detect permanent faults that lead to errors in the MAC
	FSR 2.3	AES-GCM shall protect the integrity of all FIFO buffers accesses related to the MAC computation
	FSR 2.4	AES-GCM shall report faults that lead to errors in the MAC
	FSR 2.5	AES-GCM shall provide a safe state control mechanism for the MAC computation
SG 3	FSR 3.1	AES-GCM shall detect transient faults that lead to key expansion errors
	FSR 3.2	AES-GCM shall detect permanent faults that lead to key expansion errors
	FSR 3.3	AES-GCM shall report faults that lead to key expansion errors
	FSR 3.4	AES-GCM shall provide a safe state control mechanism for the key expansion
SG 4	FSR 4.1	AES-GCM shall protect the system to avoid un-intended modes of operation
	FSR 4.2	AES-GCM shall protect the mechanism of context switching

Table 5.2: Functional Safety Requirements (FSR) derived from the Safety Goals (SG) for the AES-GCM ASIL-D.

The FSRs 1.1, 1.2, 2.1, 2.2, 3.1 and 3.2 define which types of faults shall be considered for the violation of each safety goal. In this analysis, transient faults (which occur only once and disappear) and permanent faults (which once happen, stay until removed or corrected) were considered. The FSRs 1.3 and 2.3 aim to protect the data store of the most import results from the IP core that are the ciphertext in case of encryption, the plaintext in case of decryption and the MAC tags. The FSRs 1.4, 1.5, 2.4, 2.5, 3.3, 3.4 define that in case of faults that could lead to

the violation of the respective safety goals, these faults shall be reported and the system shall be brought to a safe state. Finally, the requirements linked with safety goal four, define the protection of the mechanisms associated with the configuration space.

5.4 Technical Safety Concept

The Technical Safety concept shall refine the Functional Safety Requirements into technical solutions at the architecture level, called safety mechanisms. For that, it is necessary to go into detail in the architecture and explore the possible manners in which an element can fail executing the intended behaviour, which corresponds to the identification of the failure modes.

The following subsections describe the identification of failure modes and the safety mechanisms adopted to cover them. The technical safety requirements, although identified, are not presented here, since they are only a formal description of the technical solutions used to satisfy each of the functional safety requirements. Instead, it is provided a direct association between the safety mechanisms and the blocks of the design in which they were implemented.

5.4.1 Failure Modes

The identification of failure modes is normally done by dividing the system into parts and exploring the possible failures that can happen in each component. The degree of detail for analysing the implementation architecture is not specified by the standard. However, it should be done in a way to clearly identify the possible failures but without creating multiple similar failures for the same part that could be aggregated into only one. In our case, since it is possible to divide the architecture into blocks which have a well-defined functionality, the failure modes were defined at the block level. The complete list of failure modes is provided in the FMEDA in Appendix A.

The safety mechanisms shall actuate in this failure modes to cover the residual faults that may occur. However, not all failure modes in the design need to be covered by safety mechanisms, as long as the unprotected failures constitute a safety risk for the system that is tolerable and compatible with the safety metrics for the ASIL rating we are targeting for. Since we are targeting an ASIL-D, which requires 99% of coverage for the metric of single-point faults, all the failure modes associated with this metric were protected.

The safety mechanisms themselves can be subject to failure modes which are the cause for latent faults. The metric for latent faults defined by the standard for an ASIL-D is less restrictive (90%), which means that we had the possibility to not cover all failure modes that could lead to latent faults.

5.4.2 Safety Mechanisms

The definition of safety mechanisms for this IP core presented two constraints that considerably reduced the options of mechanisms that could be used. The first was the fact that this is an IP core for cryptography and so the input data is normally transformed in the blocks, turning it difficult

to verify if some error happened or if the output data is the expected without redundancy in the operations. The second is the high coverage metrics that are necessary to achieve for an ASIL-D. Each safety mechanism has associated a probability of diagnosis coverage in the component it is trying to protect (residual fault coverage) and a diagnosis coverage of failures that can happen in themselves (latent fault coverage). Since this analysis aims an ASIL-D, the safety mechanisms used need to have a high diagnosis coverage of residual faults (99%) and the majority needs to cover also latent faults (ideally 99% of diagnosis coverage).

Considering these points, the safety mechanisms to protect each block were defined. The main safety mechanism applied to the Key Expander, the Pipeline Cipher and the GMUL System was their operation in dual lock-step. These blocks contain the safety-critical combinational logic to compute the required mathematical operations of the algorithm, and so, their protection is determinant. The protection of the FIFOs, which do not make any changes to the data, could be done through the implementation of Error Correcting Codes (ECC) for data and address. In all blocks, the safety-critical registers were implemented as Safety Registers and the propagation of safety-relevant signals was done using dual-rail signals. For each one of the blocks it was implemented a Safety Controller and, at the top-level, it was implemented a Safety Monitor to manage all the safety activity of the core. The description of each one of these safety mechanisms is presented in the following subsections.

5.4.2.1 Dual Lock-step Operation

A dual lock-step operation consists in using two instances of the same block, connected to the same inputs and computing the same results in parallel or with a small difference in time (called time diversity). The objective is to compare the outputs of both instances and, in case of difference, trigger an alarm that indicates that there has been a transient or permanent fault in one of the instances.

The lock-step operation is capable of 99% of diagnostic coverage for failures in the block it is protecting. However, since the quantity of data to compare is normally huge, the size of the lock-step comparators is also considerable, which means that the probability of failure in the comparator, which can result in latent faults, cannot be ignored. To solve this problem, it was necessary to use a comparator capable of detect faults in itself. The decision was to use a totally self-checking comparator, that is, a comparator that indicates a wrong result in case the result between the two instances being compared is wrong, or in case a fault in the comparator (as example a "stuck at" fault) happens. The comparator used is capable of comparing two 2-bit signals and the result from the comparator also has two bits. The comparator logic diagram is shown in Figure 5.2, where (x_0, x_1) is a 2-bit signal from one instance and (y_0, y_1) is the 2-bit signal from the other instance in lock-step. To obtain a comparison match, it is necessary that one of the inputs is inverted in relation to the other, that is, that the value $x_0 = \overline{y_0}$ and $x_1 = \overline{y_1}$. This implies that before connecting to the comparator, the output of one of the instances in lock-step needs to be inverted.

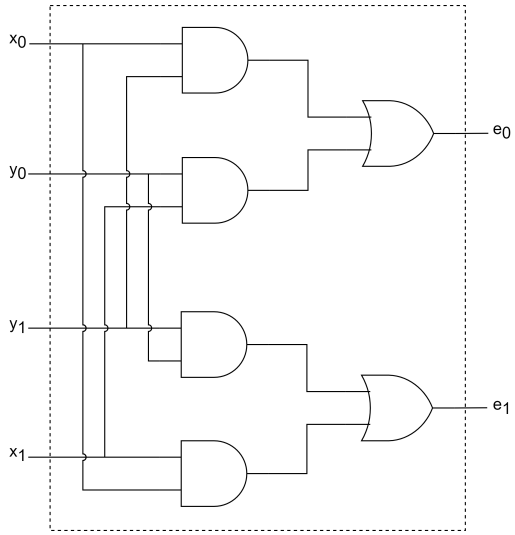

Figure 5.2: Totally Self-Checking 2-bit equality comparator.

The combination of values (e_0, e_1) is the result of the comparison. If the combination is (0,1) or (1,0), it means that the lock-step instances and the comparator are working properly. If the combination is (0,0) or (1,1) means that a fault in the lock-step, or in the comparator, occurred.

To compare signals with more than two bits, the 2-bit comparators can be connected in a kind of tree. Figure 5.3 shows an example of this type of connection for an 8-bit lock-step comparator. The diagnosis coverage for latent faults is then 99% with this comparator.

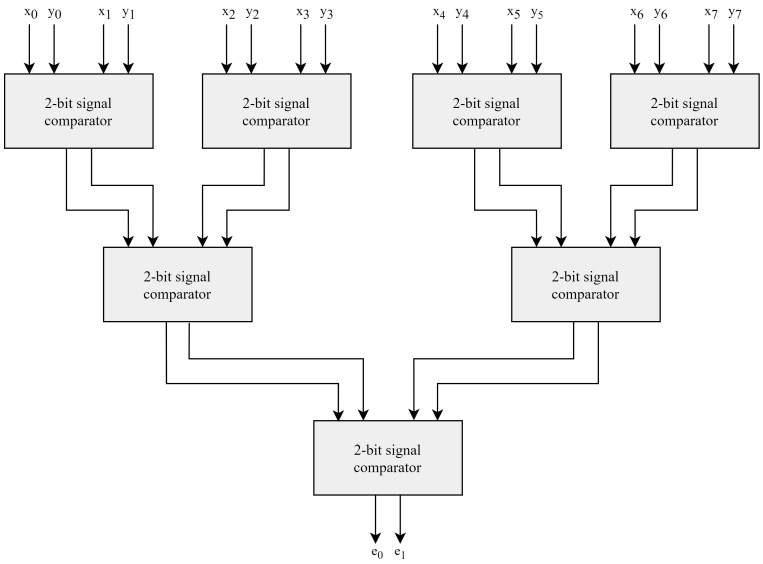

Figure 5.3: Totally Self-Checking 8-bit equality comparator.

5.4.2.2 Safety Registers

The safety registers use the concept of safety flip-flops, illustrated in Figure 5.4, to protect registers of any width. The idea consists of using a second flip-flop to store the data, where in this second flip-flop it is saved the inverted data value. At the output, the value of both instances is compared using an XOR, which indicates an error if some fault happened in the flip-flops (such as a bit-flip) or in the propagation of the signals. These characteristics confer this safety mechanism a diagnosis coverage of 99% to residual faults and latent faults.

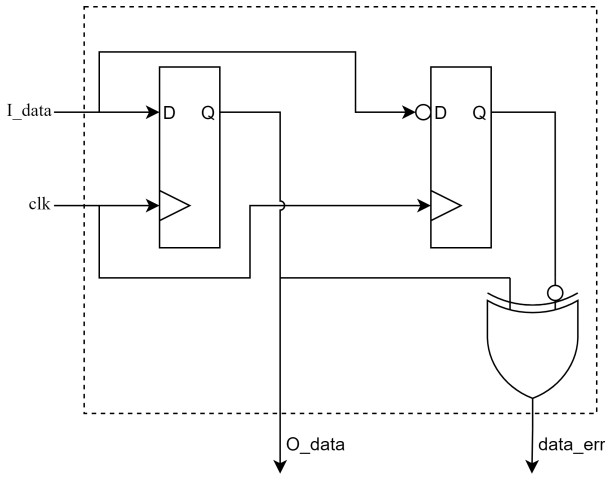

Figure 5.4: Concept of Safety Flip-Flop.

5.4.2.3 Error Correction Codes

The error correction codes implemented were Hamming codes with a distance of four which means that they had the capability of Single Error Correction and Double Error Detection (SECDED). However, this basic functionality does not have the capability of detect other failures that can happen in data storage such as when the code word to store, turns into all logic low (all-zero errors) or all logic high (all-one errors). Then, in addition to the basic SECDED functionality, it was implemented a variant of the algorithm which permits the detection of all-zero and all-one errors.

The implementation was composed by two modules: a SECDED code encoder, which receives the input data and the write address and computes the respective code, and a SECDED code decoder which receives the data at the output of the FIFO, the read address and the code calculated by the encoder and verifies the existence of some error. If the decoder detects single errors in the address or data, it will set the single error signal (or address signal if the error is in the address) but will also correct the data or address. If a double bit error is detected, the double error signal will be set, but the data will not be corrected. An extra signal in the decoder is provided, which is called unknown error. This signal permits the detection of odd bit errors (3 or more bits) in

the data (which is also an extra functionality compared with the default SECDED definition) or to detect the all-zero and all-one errors. The interfaces of the modules are illustrated in Figure 5.5.

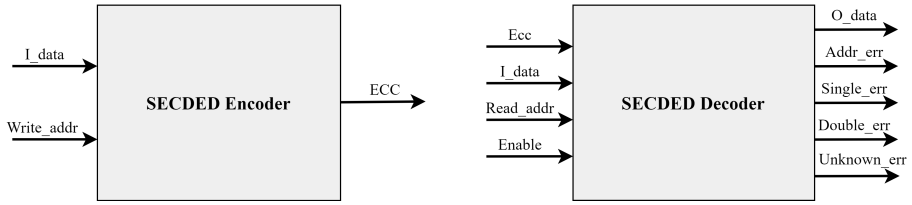

Figure 5.5: Module Interface of SECDED Encoder and Decoder Modules.

This safety mechanism allows a diagnosis coverage of 99% with respect to residual faults, but it does not give any protection to latent faults. This point was not considered as a high concern since the encoder and decoder modules are purely combinational and their sizes, when compared to the components that they are protecting, are considerably small. This means that existing latent faults are, in principle, minimal and will not have a high impact on the respective metric. If this assumption turned out to be wrong when calculating the metrics, it would be necessary to cover the latent faults of this safety mechanism.

5.4.2.4 Safety Controller

A safety controller was implemented for each block of the design with the objective of including all the safety logic for the respective block within that controller. One of the advantages of using this block is that it permits to estimate the area used for the safety logic and, from that, quantify the latent failures in that block.

For the blocks that were implemented in lock-step, the safety controller was also used to control the inputs for each instance of the lock-step. Although for this application the time diversity between the two operations has been defined as zero (the instances operate exactly in parallel), this control of the inputs allows adding time diversity to the dual lock-step in an easy way (if desired in the future), as illustrated in Figure 5.6.

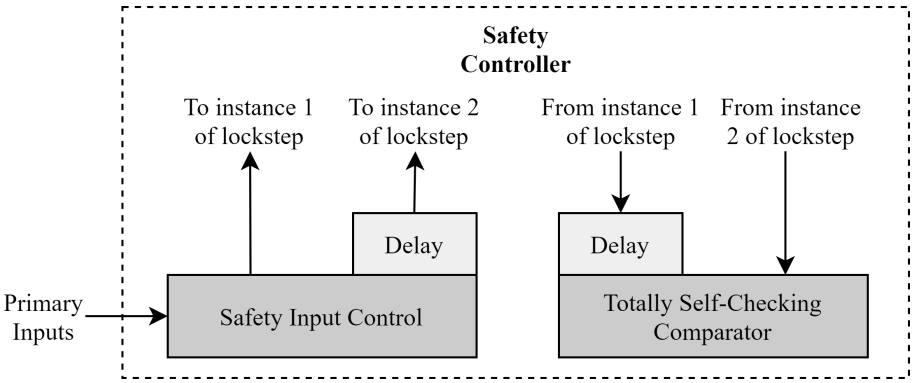

Figure 5.6: Safety Controller with data input control for the dual lock-step operation.

5.4.2.5 Safety Monitor

The safety errors of the entire design are reported to the safety monitor. In this block, the signals are aggregated and transformed into output error signals that can be accessed by an external system. These output error signals are sticky, which means that they are maintained until the core is brought to a safe state. In this case, the safe state was defined as a core reset.

5.4.3 Assumptions of Use

The implementation and operation of the safety mechanisms required some assumptions of use that specify in which manner the system may be used to ensure the proper safety operation implemented.

The first is that the clock signal for the circuit was considered coming from an external source and was assumed as always in compliance with the specified frequency and margins conditions. This means that the protection of the clock signal, if necessary, needs to be done by the external system in which this IP Core will be integrated. Assuming the clock as safety protected, it is possible to use the same signal for both instances of the lock-step, for example.

The reset signal was also considered from an external source and compliant in all cases with the specification. As this is the signal used to bring the system to a safe state, it is necessary its proper functionality, without glitches, for example. The input data and input commands in the system were also considered as correct. Then, if incorrect processing of the data happens before the input in the IP core and if that leads to improper functioning of the system, it is not ensured the safe operation of the system.

In terms of the safety operation, in case of a failure that is detected by a safety alarm, the corresponding signal shall stay active until the system is brought to a safe state. The time between the activation of the safety alarm and the application of the safe state was assumed to be less than the time it would take the system to result in a hazardous event.

5.5 Failure Modes, Effects, and Diagnostic Analysis (FMEDA)

The FMEDA was the systematic analysis technique used to evaluate the system in terms of failure rates and calculate the hardware metrics for the design that define whether the system meets the target values imposed for the specified ASIL rating or not. The FMEDA takes into account all components of the design, their functionality and eventual failure modes, the effect of these failure modes on the functionality of the design and their diagnostic coverage.

The FMEDA requires a detailed report of the area of the design, from which it is possible to estimate the number of transistors and the number of memory cells or registers per block, and the respective failure rates. In our case, it was used a 28nm ASIC technology as the reference for the area values of the design. The failure rate for the entire IP core (λ) was done using the expression

shown in 5.1, extracted from [31]. The failure rate for each block of the IP core was calculated based on the percentage of that block in the total area of the design.

$$\lambda = \{\lambda_1 \cdot N \cdot e^{-0.35 \cdot a} + \lambda_2\} \cdot \left\{ \frac{\sum_{i=1}^{y} (\pi_t)_i \cdot \tau_i}{\tau_{on} + \tau_{off}} \right\} \tag{5.1}$$

where N is the number of transistors in the circuit, a is a year of manufacturing dependent variable, τ_{on} and τ_{off} are respectively the working time and storage time ratios of the circuit, τ_i is the working time ratio for the i^{th} junction temperature of the integrated circuit, π_t is a temperature factor, and λ_1, λ_2 are technology dependent factors that can be extracted from the standard. The values of τ_{on}, τ_{off}, τ_i and π_t are based on a mission profile that is internally defined by each company.

The next step was to identify the failure modes of each block (including the failure modes of the safety mechanisms) and their impact on the total failure of the block in terms of percentage. The percentage of each failure might be estimated with different approaches: (i) based on knowledge from previous cases; (ii) based on the percentage of block area affected by the failure mode; (iii) considering the same impact and, consequently, the same percentage, for all failure modes of the block. The sum of the percentages of the failure modes for each block must be 100%. For this project, the failure modes that correspond directly to the block were considered as having the same percentage. The failure modes associated with the safety mechanisms were calculated based on the percentage of their area in relation to the block area.

For each failure mode, it was also necessary to identify their effects and determine if they were capable of violating each one of the safety goals. If the violation was possible, it was necessary to identify whether it was due to a single-point fault, a residual fault or a latent fault. Finally, for each failure mode, it was associated the corresponding diagnosis coverage of the safety mechanism (if any). From here, we had all the information needed to calculate the metrics for the design.

5.5.1 Single-Point Fault (SPFM) and Latent Fault Metrics (LFM)

The calculation of the SPFM considers all the single-point faults that occur in blocks that do not have safety mechanisms implemented and the residual faults which are not detected by any safety mechanism. The expression used for the calculation of this metric is shown in 5.2.

$$SPFM = 1 - \frac{\sum_{SRHW} (\lambda_{SPF} + \lambda_{RF})}{\sum_{SRHW} \lambda} \tag{5.2}$$

where the $\sum_{SRHW} \lambda$ is the sum of the failure rates for all the Safety Related Hardware (SRHW) components of the design.

The calculation of the failure rates for each component is the sum of all failure modes probability of occurrence multiplied by the failure rate of the component.

For the Latent Fault Metric (LFM) it was considered the failure rate of the latent multi-point faults. The expression used was:

$$LFM = 1 - \frac{\sum_{SRHW} (\lambda_{\text{MPF,latent}})}{\sum_{SRHW} (\lambda - \lambda_{\text{SPF}} - \lambda_{\text{RF}})} \tag{5.3}$$

5.5.2 Probabilistic Metric for Random Hardware Failures (PMHF)

The objective of the PMHF is to represent the overall probability of failure per hour so, instead of providing a ratio between the different failure rates of the component, as in the SPFM and the LFM, it tries to quantify the safety level of the system through an absolute value. This value is the sum of all the expected faults for the system expressed in terms of failure rates. In the case of latent faults, this metric also takes the exposure duration of the fault into account. The equation used to calculate the PMHF was:

$$PMHF = \sum_{SRHW} \lambda_{\text{SPF}} + \sum_{SRHW} \lambda_{\text{RF}} + \beta \cdot \sum_{SRHW} \lambda_{\text{MPF latent}} \tag{5.4}$$

where β is a value for integrated circuits with on-chip redundancy extracted from the standard IEC 61508 [32]. For the calculation of the metric, it was assumed the worst-case scenario which corresponds to a β equal to 0.47.

5.5.3 FMEDA Results

To ensure that the design developed met the target level for which it was planned, the metrics were computed and compared with the reference values specified in the ISO-26262. The results for the safety design developed are presented in Table 5.3 and a simplified version of the FMEDA is provided in Appendix A.

Metric	Target for ASIL-D	Result
SPFM	$\geq 99\%$	99.326%
LFM	$\geq 90\%$	96.416%
PMHF	< 10 FIT	0.0734 FIT

Table 5.3: Results for the SPFM, LFM and PMHF metrics.

As shown in the table, the system was capable of meeting all the target values for an ASIL-D rating. The value presented for the SPFM and the LFM is the value considering all the safety goals in the design. However, it was also necessary to ensure that the system satisfied the same metrics for each one of the safety goals individually. One relevant aspect is that, although the target for the safety goals was the same, their impact on the final value of the metrics was not equal, since the violation of each safety goal was dependent of the corresponding components in the design. The PMHF metric was only calculated for the entire design since it is an absolute value.

The major concern during the development was the SPFM, since the tolerance for faults was only 1%. The LFM was met with a good margin, which means that the decision of non-protection of the ECC used in the FIFOs did not really have a significant impact on the final metric. It is also possible to observe that, if the area was a restriction, it was possible to decrease it by reducing the complexity of some safety mechanisms and consequently reducing their detection of latent faults. Finally, the PMHF was considerably far from the 10 FIT limit, which was expected, since it was not possible that such a small component as an IP core could have a high probabilistic value of failure.

Chapter 6

Power Analysis

In this chapter, it is described the methodology used for the power analysis performed on each one of the IP core configurations (the default and the ASIL-D), which allowed their comparison in terms of vulnerability to power consumption-based side-channel attacks. The power analysis procedure explained here comprises two phases: first, the extraction of the power consumption in the form of power traces and then, the processing and analysis of those power traces.

6.1 Power Traces Extraction

The extraction of power traces was made through the simulation of the physical hardware implementation of each one of the configurations, using the same 28nm ASIC technology that was used for the area estimation in the safety analysis. The objective was to obtain power traces that were highly accurate to the ones that would be obtained through a physical extraction from, for example, a test-chip. This phase included then five steps: synthesis, place and route, extraction of timing information, simulation and finally, the estimation of the power consumption of the IP core to generate the power traces. The steps included in the power extraction, the main inputs required for each one of these steps and the respective resulting products, are shown in Figure 6.1.

The synthesis converts the RTL design to the specific technology gate-level netlist, requiring for that the RTL design, the technology library and the design constraints. The place and route stage takes this gate-level netlist and creates the physical implementation of the design, according to the specification of the manufacturing process. Inherent to this stage of place and route is the clock tree synthesis, which ensures a proper distribution of the clock for the entire design. The clock tree synthesis is a fundamental step for the purpose of this project since it is directly related to the power consumption of the core, in specific to the dynamic power consumption. Although there are works that extract power traces from an estimation based solely on the gate-level netlist (as for example in [33]), this approach is not sufficient to produce accurate power traces, due to the lack of timing information at that level.

As stated in 3.2.1, power consumption can be divided into two main groups: static power consumption and dynamic power consumption. With the netlist created from the place and route

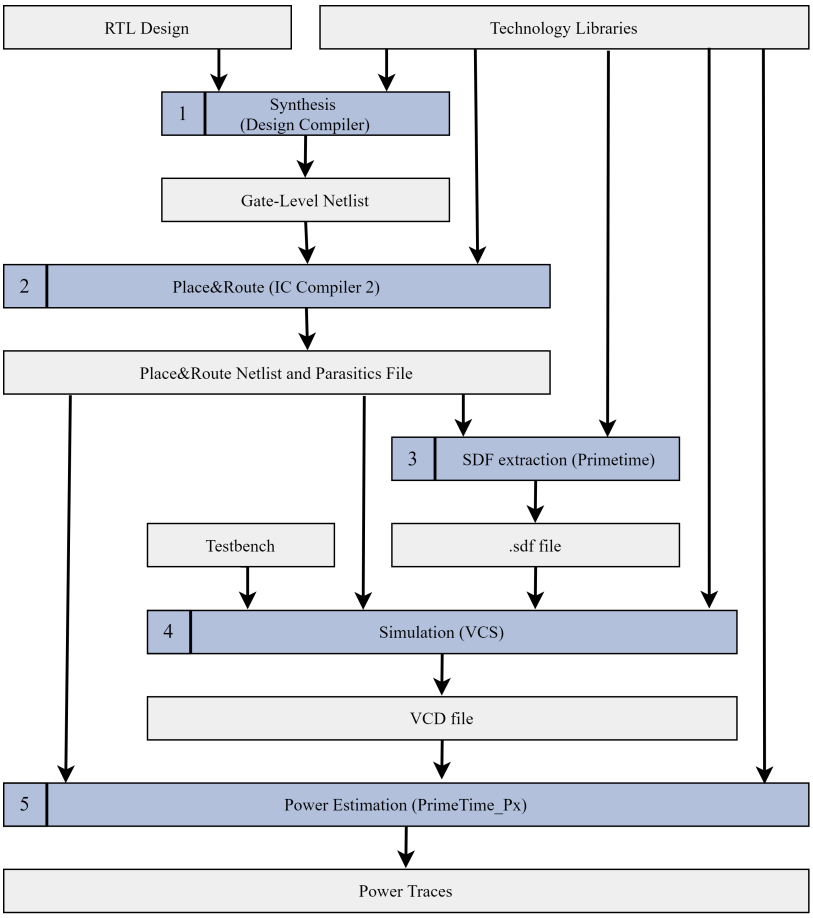

Figure 6.1: Power Traces Extraction Flow.

phase, and using the technology libraries, it was already possible to estimate the leakage and internal power for the operation of the design. However, we still did not have any model for the switching activity, that could provide information about the operations executed. This model can be created through the activity logs resulting from the device simulation for a given testbench. However, to extract the desired model from the simulation, it is necessary to include detailed timing information about the design. This is extracted after the place and route phase to a SDF (Standard Delay Format) file, which contains all timing information such as the interconnect delays, the path delays and hold and setup timing checks of the design. The simulation resulted in a Value Change Dump (VCD) file. This is an event-based format file, with logs for any value changes and the respective time of occurrence for all signals in the design.

The analysis of the work products from the place and route step, the technology libraries and the VCD file allowed then to generate the power traces for each one of the IP core configurations. However, one should note that while the synthesis, place and route and SDF extraction are only dependent on the design and the constraints defined, the simulation is also dependent on the testbench, including the actual data processed by the system under simulation. This means that to create power traces for different encryption keys and input plaintexts, the first three steps were fixed and only needed to be executed once. But, on the other hand, the simulation and power

waves extraction needed to be repeated whenever the testbench was changed, either to modify the input data or the simulation conditions. Since for the context of this project it was required to obtain power traces for various keys and plaintexts, the steps of simulation and power estimation were automated using Shell scripts.

All the steps described above were done using Synopsys Electronic Design Automation (EDA) tools. The synthesis was done using the Design Compiler, the place and route and clock tree synthesis using IC Compiler 2 and the simulation using VCS. The SDF extraction was done using PrimeTime, and the power estimation using PrimeTime_PX, which is a variant of the PrimeTime focused on the power analysis and power estimation of designs. The testbench was developed using the Universal Verification Methodology (UVM) environment, in System Verilog. The control scripts for the Design Compiler, IC Compiler 2, PrimeTime, PrimeTime Px and VCS were made using the Tool Command Language (TCL). These tools were run on the Community ENTerprise Operating System (CentOS).

6.1.1 Timing Information

As stated before, the timing information is crucial for the power traces because the timing at which each signal transition occurs has a direct impact on the patterns of the corresponding power traces. Then, since the objective here was the comparison of two configurations (the default and the ASIL-D), the timing behaviour of both should be maintained as similar as possible. For that reason, it was necessary to ensure that not only the same testbench conditions were applied for both configurations but also that the different phases to achieve the physical implementation were made under the same timing constraints.

However, these timing constraints in most of the cases create timing violations (hold and setup) in the design, especially at the end of the place and route step, which need to be solved to run the simulation. This can be done using PrimeTime, which will automatically upsize some cells to increase their drive strength, in case of setup violations, or insert buffers along the paths and downsize the cells, in case of hold violations. These changes are then mapped to the final netlist using the IC Compiler. But since these timing violations and their solutions are design-dependent, it is possible that they are not equal for the two configurations and that may lead to different timing characteristics between them. The same applies to the phase of clock tree synthesis, which was specific for the internal arrangement of each one of the configurations. Thus, although one should try to maintain the same timing characteristics for both configurations, there are transformations inherent to the process described above that are not possible to control.

6.1.2 Simulation Environment

The specification of the simulation characteristics made in the testbench should be such that it allows us to create the best possible scenarios to compare the two configurations. The first option in the testbench was to perform only encryption operations. This simplification is valid since in the AES-GCM the encryption and decryption operations are similar and then, the vulnerability in each

one of the operations is the same. To facilitate the identification of a complete execution of the algorithm in the power traces, each message to encrypt was only composed by one block of data (128 bits). As the AES algorithm is a block cipher, if the message contained more than 128 bits, it would originate multiple blocks of data, which would be encrypted sequentially, making it difficult to identify the start and the end of the algorithm execution. The clock period was defined to 10 *ns* (although the design constraint in previous steps was done with a pessimistic consideration of a clock period of 2*ns*), requiring 330 *ns* (33 clock cycles) to encrypt each block. In each simulation executed, multiple encryptions were made, spaced from 500 *ns*. The input data for the blocks, with exception to the cases specifically mentioned, was randomly generated, as well as the encryption key for each process. Although the data has been randomly generated, it was certified that the sets of data and keys used to simulate both designs were the same.

The timing annotation for the simulation was done from the SDF file considering the information relative to the typical process corner. The dump of the signals was made for the VCD file considering a time resolution of 1 *ps*. This value was chosen in order to be much smaller than the clock period, so that it was possible to capture with high the instants of transition of the signals, which is necessary for accurately estimating the power consumption traces.

6.1.3 Power Estimation

The methodology used by PrimeTime PX to generate the power traces for a given simulation is based on events. That is, it associates the power consumption with changes on the signal values (which it calls events). If there are no signals changing their value, than PrimeTime will make an interpolation between the points to create the power waveform. This is the main reason why detailed timing information to construct the power traces is needed. If the simulation does not take into account the real clock tree and its associated delays, the change in the signals' value will happen at exactly the same time causing a spike in the power consumption instead of its distribution along the time as in a real case scenario.

6.2 Power Traces Processing

Once the power traces were extracted, it was necessary to perform the second phase of the power analysis, which consisted of processing and analysing the relevant information of these power traces. In the scope of this work, it was useful to have access to the power consumption for all the hierarchy levels of the design, to allow a direct inspection of each block. Then, for each simulation, it was extracted information relative to the power traces for all the hierarchy levels. This resulted in output files that contained a huge quantity of information and which required a significant amount of memory. It was possible to read these files in waveform viewers such as the Synopsys Custom WaveView, but it was not possible to read them in the usual data processing tools, such as Matlab. Because of this, it was necessary to filter the files which contained the power traces, extracting for each desired analysis the level and the specific time window of interest. This filtering process was automated using a parser script, developed in Python.

Chapter 7

Results

In this chapter, the resulting power traces for the default and the ASIL-D configurations are presented and a comparison between both is conducted to understand the impact of the functional safety measures on the security vulnerabilities of this IP core. First, it is made a comparison considering the total power consumption of each configuration. Then, the power traces of the most vulnerable blocks in the design are inspected. Finally, it is analysed whether the difference in the leakage of information between the two configurations, really represents a greater vulnerability of one of them in relation to the other.

7.1 Top-Level Power Traces Analysis

The changes implemented in the design to achieve an ASIL-D rating implied a considerable amount of logic which affects not only the static power consumption of the core, due to the additional number of elements in the design but also the dynamic power consumption due to the higher number of signals now switching on the design. In this section, the power traces for the total power consumption of each configuration are analysed to understand the impact of the additional functional safety logic.

The power traces extracted when performing a power analysis contain both the phases in which the IP core is operating and the others in which it is without any activity. The second ones are not relevant for an attack since they are only related to static power consumption. So, a power analysis starts typically with the identification of the timing windows in which the IP core is operating. This can be done by a direct inspection of the power traces because the additional dynamic power consumption of the operations leads to a modification in the power traces that permits to detect when the activity starts and ends. In our case, this association between the variations in the power traces and the operations can be confirmed, since the power traces resulted from a testbench created for this specific purpose and therefore, the moments of start and end of operation were well known. Figure 7.1 illustrates the power traces for a complete encryption, for each one of the configurations, considering the same block of data, key and testbench conditions for both.

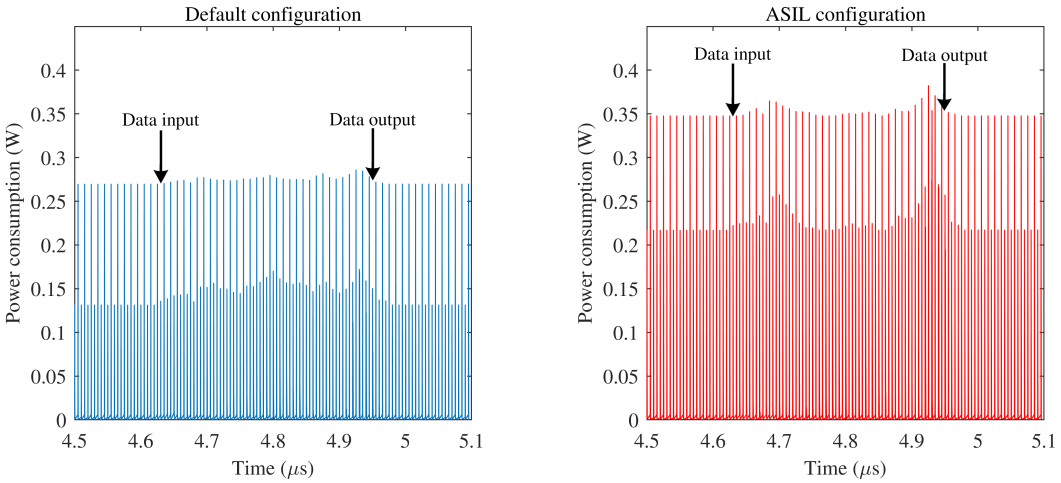

Figure 7.1: AES-GCM power consumption for default and ASIL-D configurations. In each graph it is represented when the operation starts (data input) and when it ends (data output).

When looking at the power traces, the first characteristic that stands out is the existence of two base levels of power consumption for each graph. Using the power trace for the default configuration as example, the lowest base level is around 0.14 W and the highest at 0.27 W. As this is a clocked synchronous circuit, the power consumption along time appears as peaks (at this time scale, approximated by vertical lines). In each configuration, the peaks at the lowest base level are related to the rising edge of the clock transitions and the peaks at the highest base level are related to the falling edge transitions. Between the peaks of the clock transitions, since there are no signals switching (which means an absence of events), PrimeTime is not capable of calculating the power consumption and does a linear interpolation between the consecutive events. This results in no variation of the power traces between the peaks, where there are no events. In both configurations, the logic associated with the computations of the algorithm is synchronous with the rising edge of the clock and therefore, the lowest base level is the most relevant for our analysis.

The comparison of the graphs shows that the power consumption in the ASIL-D configuration is higher, which was expected due to the additional logic that this configuration requires to ensure functional safety. In both, there is a well identifiable difference between the static power consumption of the IP core and the power consumption when it is operating, which makes it easy the identification of the start and end of the operation. However, when we look in detail for the shape of both power traces, the default configuration seems to have a higher relief than the ASIL-D configuration, especially for the zone around 4.8 μs, where the ASIL-D configuration is almost flat. This behaviour is not what was expected, since the additional logic in the ASIL-D configuration, such as the redundancy in the operations, should cause higher protuberances in the shape of the power trace. One important thing to note is that the time scale used to allow the representation of a complete encryption, led to a compression of the power peaks in the trace, causing them to be represented only as a vertical line with the amplitude equal to the maximum consumption at that

peak. This compression hid useful information from the traces as illustrated in Figure 7.2, where it is shown an example of the real shape of each one of these peaks for both configurations.

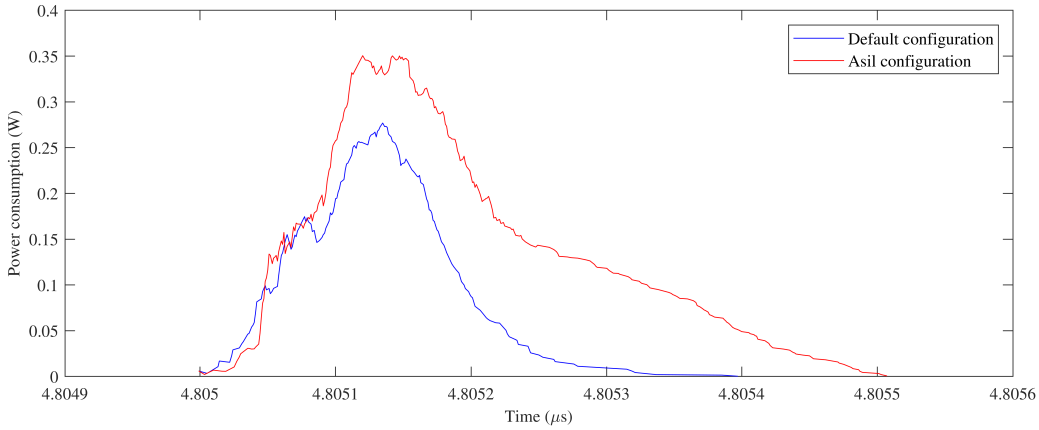

Figure 7.2: Power peak comparison between default and ASIL-D configurations.

The superimposition of both graphs reveals that the peaks of the ASIL-D configuration are not only higher, as it was already visible in Figure 7.1, but also wider. The fact that they are wider is justified by a double reason. The first is the additional combinational logic inherent to the safety mechanisms applied in the ASIL-D configuration, which increased the propagation delays of the design. The second is the decisions and optimisations made by the compiler during the different phases of synthesis, place and route and clock tree synthesis, that due to the additional logic in the ASIL-D configuration, led to differences in the implementation.

Thus, these wider peaks seem to be the reason for the absence of relief around the 4.8 *ns* in Figure 7.1 for the power trace of the ASIL-D configuration. The additional power consumption due to the safety mechanisms not only produced higher peaks in amplitude but also resulted in a higher distribution of the power consumption along the time. To prove this, the graphs in Figure 7.1, were transformed using a moving average window of 10 *ns* (one clock cycle). This converted the instantaneous power consumption into a representation of the average power consumption in a window of one clock cycle around each instant of time. The result is shown in Figure 7.3.

With the average window transformation, the distribution of the power consumption along the clock cycle has now an impact on the amplitude envelope of the power consumption waveform and confirms the assumption that was made before. The wider peaks in the ASIL-D configuration flattened the power peaks of some zones of the power trace, masking their variation in Figure 7.1. Moreover, this transformation reveals that the power consumption patterns for both configurations are in fact very similar in shape, however with a higher amplitude in the case of the ASIL-D configuration as expected.

For the power analysis, the higher the protuberances in the power traces due to the operation are, the easiest it is for an attacker to identify the operation of the device. Then, the ideal case for a secure system against power analysis attacks would be to have the power consumption trace as flat as possible. From these graphs above, it is possible to conclude that having lower power peaks is possible if, instead of having the operations of the design happening at the same time, we could

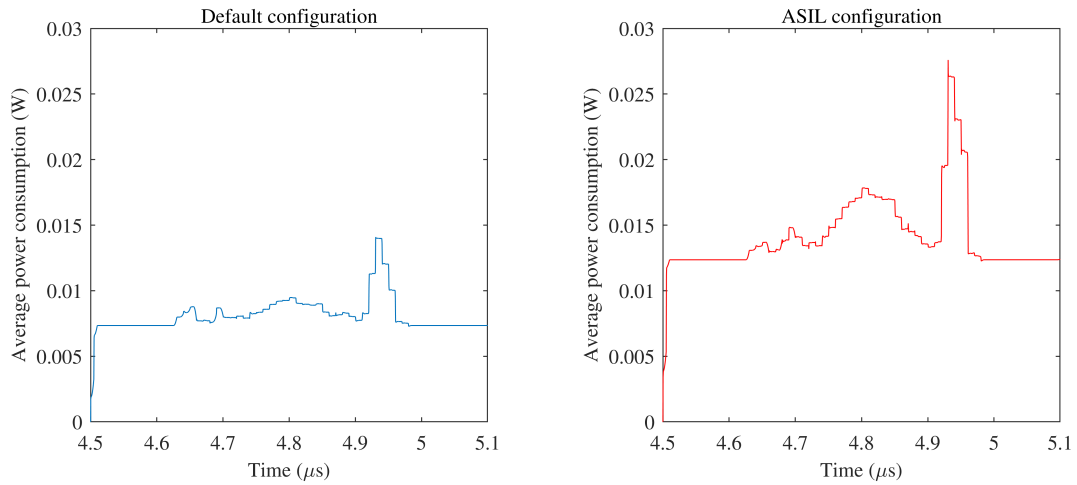

Figure 7.3: AES-GCM power consumption trace with a transformation of a moving-average window of 10 *ns*.

distribute them along the clock cycle and, consequently distribute the corresponding dynamic power consumption along the time too. One possibility for that in the ASIL-D configuration is to establish a delay (inferior to one clock cycle) in the blocks that are implemented in lock-step, between the operation of the two instances. In that way, the operation of both would not coincide and the instantaneous dynamic power consumption would be smaller and more distributed along the time.

7.2 Typical Key Leaking Points Analysis

In the previous section, we saw the impact of implementing functional safety on the total power consumption of the core. However, from an inspection at this level, it is not possible to conclude the effective vulnerability of the system since the additional increment in the power consumption could happen in blocks that do not provide any useful information for an attack. Then, since from the simulations executed, we had the power consumption traces for all the blocks in each configuration, the power consumption of the most relevant blocks could be inspected. Two blocks that are the main sources of information about the encryption key were analysed: the Pipeline Cipher and the Key Expander. These blocks are responsible for the existence of points of interest in the power traces, that is, characteristic protuberances in the power traces that are associated with specific operations. These points of interest allow attackers to define the specific moments to focus their analysis.

For the Pipeline Cipher the points of interest are the rounds of encryption, which since we are using a 128-bit key, are ten. For the Key Expander, the points of interest are the eleven key expansions required for the algorithm. As mentioned in 5.4.2, the main safety mechanism implemented in these blocks was their operation in lock-step.

The graphs with the instantaneous power consumption from the Pipeline Cipher block for both configurations is presented in Figure 7.4.

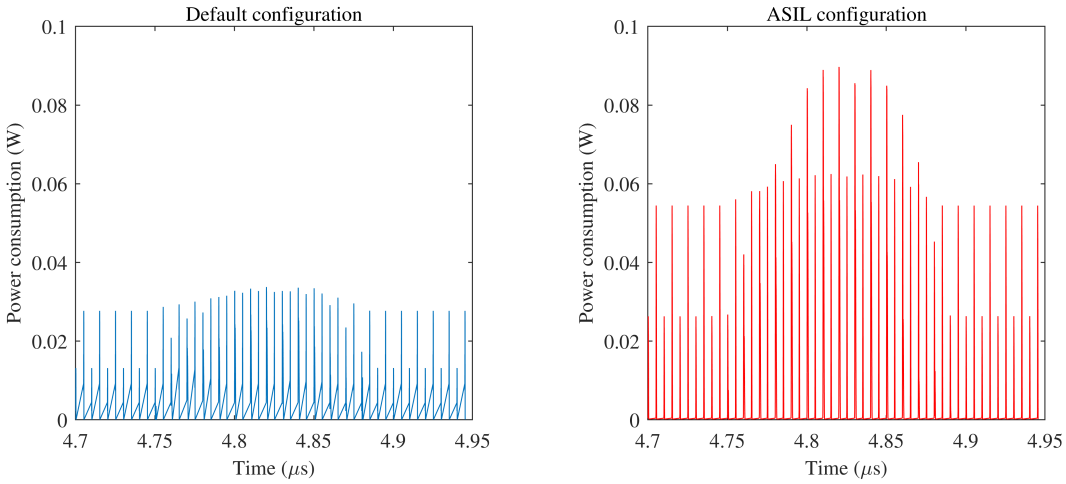

Figure 7.4: Pipeline Cipher power consumption for default and ASIL-D configurations

At first glance, it is evident that the ASIL-D configuration has a power consumption much higher than the default configuration, even for the base-level where the core is not encrypting data. This means that the additional static power consumption of the ASIL-D configuration for this block has a high impact on the power trace. As this static power consumption does not provide any useful information for an attack and is only higher in the ASIL-D configuration due to the additional number of cells, it could lead to erroneous ideas from the comparison of both power traces. With the purpose to solve that, both power traces were normalised, minimising the differences in the power traces due to the static power consumption. This normalisation is shown in Figure 7.5.

With the normalisation, the only major difference that continues to exist between both is in the ten identified peaks which correspond to the ten rounds of encryption. This difference that still remains between the power traces is now only due to the additional dynamic power that exists on computing the ten rounds in lock-step.

For the case of the Key Expander, the corresponding power consumption traces are shown in Figure 7.6. Also in this block, the identification of the points of interest (the eleven key expansions), became much more evident in the ASIL-D configuration.

From the comparisons for the Pipeline Cipher block and the Key Expander it is possible to conclude that the points of interest for a hypothetical attack are now more visible in the ASIL-D configuration and then, it is now easier for an attacker to determine the best moments in time to execute the attack.

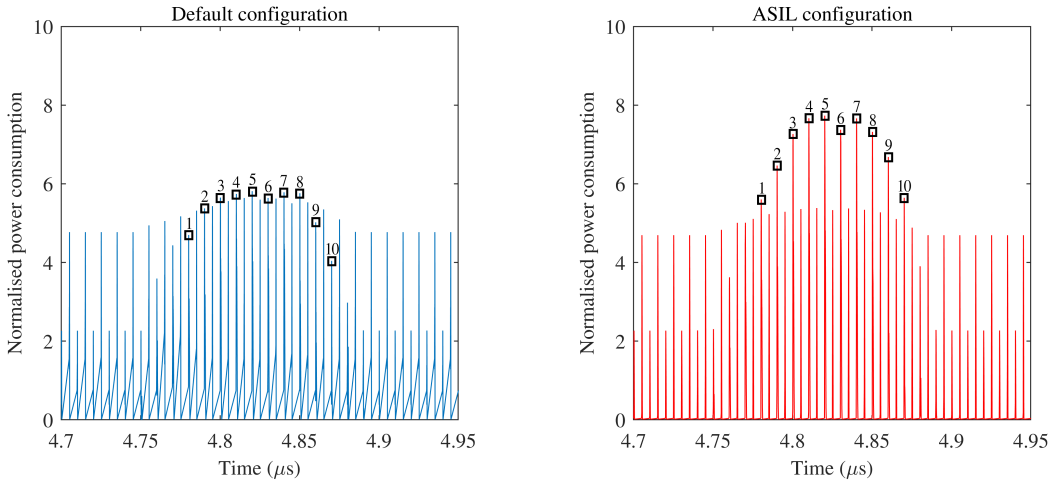

Figure 7.5: Pipeline cipher normalised power consumption for default and ASIL-D configurations. The markers and the associated numbers identify the power peak related to each round of encryption.

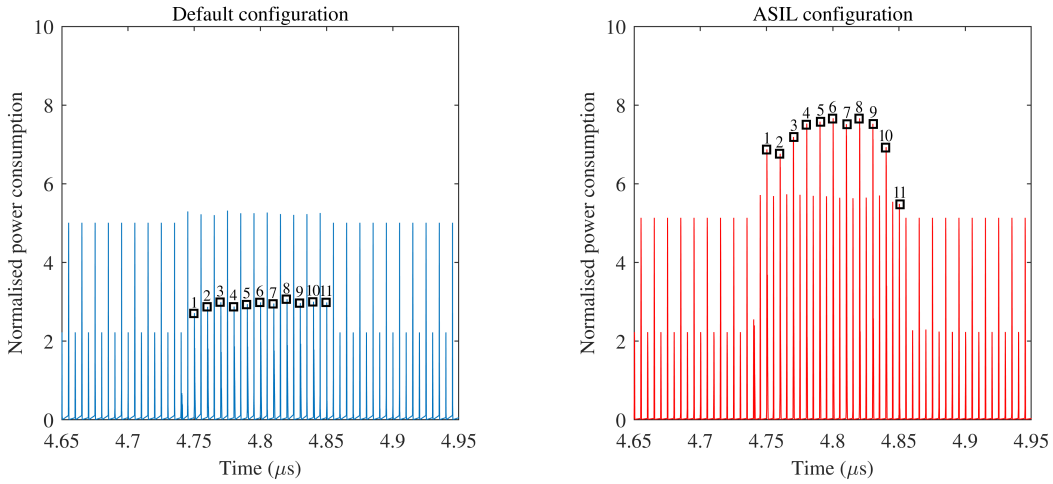

Figure 7.6: Key Expander normalised power consumption for default and ASIL-D configurations. The markers and respective numbers identify the power peak related to each key expansion.

7.3 Information Leakage Analysis

The objective of power analysis and in specific the different attack methodologies presented in 3.2.2, is to exploit the behaviour of the system, which is reflected in the power traces, to draw conclusions about the encryption key. Although for that the identification of points of interest is an important step, what can really reveal the value of the encryption key is the variability of the power traces due to different plaintext values and, especially, due to different keys used in the encryption. Then, in the following subsections, it was analysed the sensibility of variations to the encryption key and the plaintext of the two configurations.

7.3.1 Key Variation Effect

To evaluate the sensibility of the power traces to key variations a simple test was executed: for each configuration, two power traces resulting from encryptions with different keys were extracted. Then, the difference between the power traces was calculated. As explained in section 3.2.2, the attacks normally explore only variations of one byte (or, in some cases, just one bit) in the key. The reason for this is that the more bits are changed, the more difficult it is to correlate patterns in the power traces with values of the key. So, for our tests, it was also only considered a variation of one byte between the keys considered. For one of the keys, the value of the most significant byte was, in hexadecimal, 0x7f and, for the other, the value of the same byte was 0xff (or 255 in decimal). The value of the remaining bytes for both keys was the same. Figure 7.7 presents the result of the power traces' difference for both configurations.

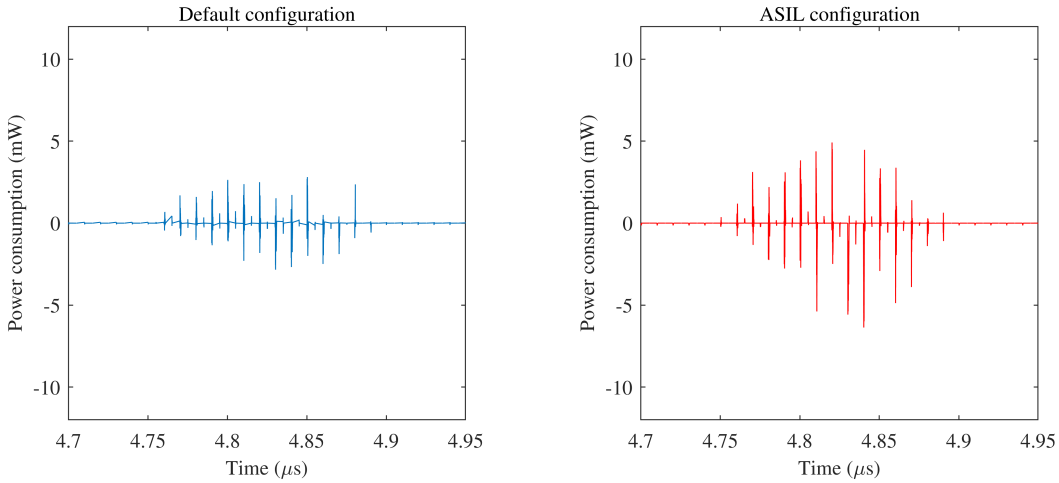

Figure 7.7: Pipeline cipher power consumption difference due to a variation of one byte in the encryption key, for each configuration.

One important thing to take into account is that these were power traces collected from simulations, that is, in an ideal case with no noise as it would exist in a physical extraction. Then, in noisy environments, some of the smaller differences would presumably be masked. Thus, the higher these differences are, the easiest would be for an attacker to extract from the power traces information about the operation of the device. Considering this, and looking at the graphs, it is possible to observe that the differences for the ASIL-D configuration are significantly higher, which means that this configuration is much more exposed to attacks and offers high chances of a successful attack.

To evaluate if these conclusions could be generalised, and were not just a specific case for the selected bytes, the same test was extended for all the 256 possibilities for the value of the most significant byte of the key. The reference power trace for the subtraction was the one that resulted from the simulation that had the most significant byte of the key equal to 0x7f. As with this approach we would have 256 graphs for each configuration, for each one it was calculated the

root mean square difference. This value of the root mean square difference for each one of the 256 possibilities was then aggregated in a graph, which is shown in Figure 7.8. The value of the root mean square difference for the power trace with the same most significant byte as the reference power trace is annotated in the graphs by the decimal value of 127.

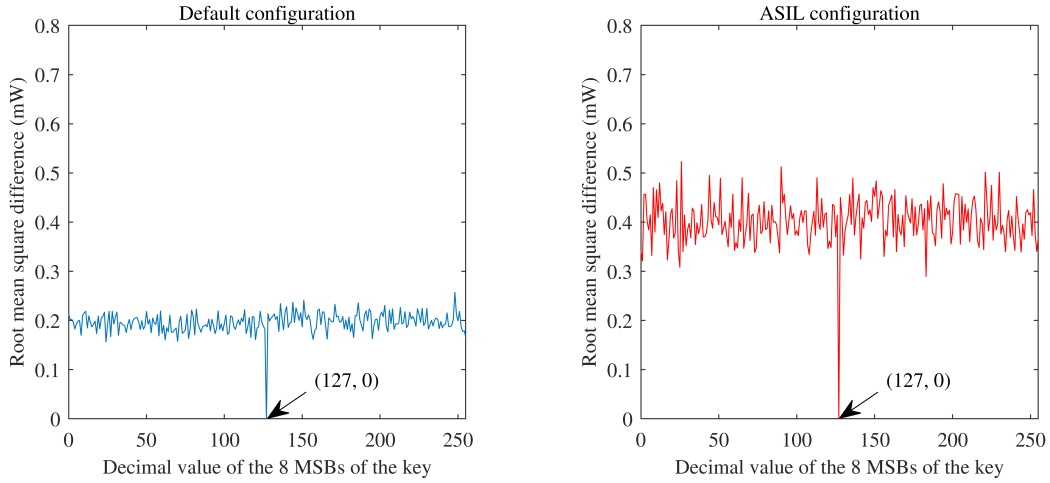

Figure 7.8: Impact of key variation on Pipeline cipher power consumption.

As one can see, although the variation between the power traces was not constant, the variation for the ASIL-D configuration is always higher independently of the value of the bytes selected. For the power trace in which the value of the most significant byte was equal to the reference value (0x7f), the difference between the power traces was zero as it was expected.

7.3.2 Plaintext Variation Effect

The same approach used for the variations of the key was repeated to compare the behaviour of both configurations with variations in the plaintext. Then, applying the same encryption key for two simulations and using two plaintexts that differ in the most significant byte (one using 0x7f and the other 0xff), it was calculated the difference between the power traces. The result is presented in Figure 7.9.

Also for the plaintext, it was computed the result of the variation of the 256 possibilities for the most significant byte of the block, using a reference power trace (the one that results from the plaintext with the value 0x7f). The root mean square difference for each combination is presented in Figure 7.10 for both configurations.

These results for the plaintext had a similar result from the ones obtained for the variation of the key. It is clear that for attacks based on key or based on plaintext variations, the ASIL-D configuration provides more useful information about the system through the analysis of the power traces.

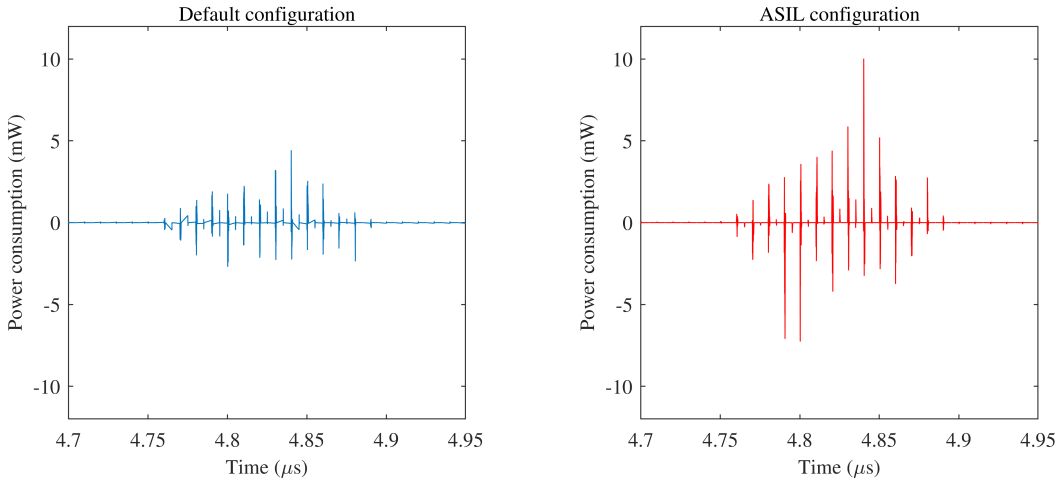

Figure 7.9: Pipeline cipher power consumption difference due to a variation of one byte in the plaintext, for each configuration.

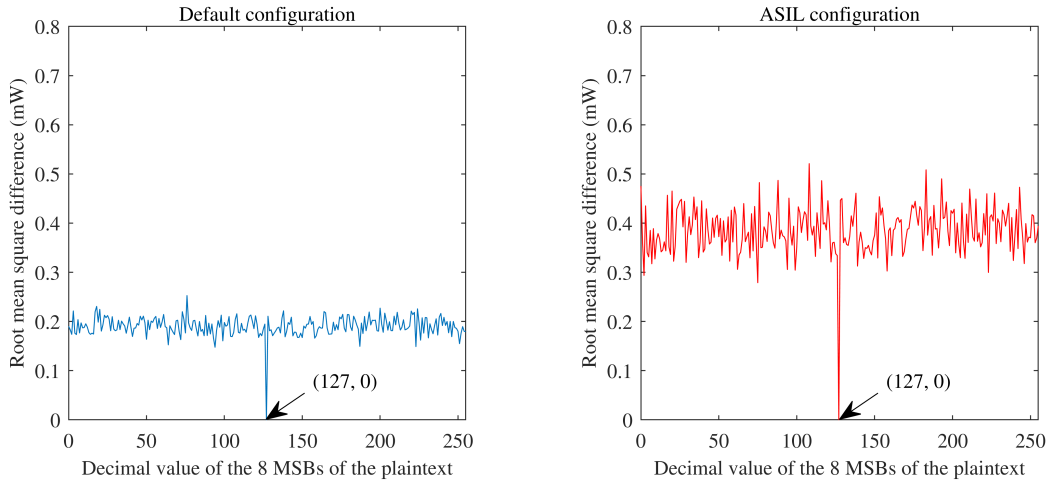

Figure 7.10: Impact of plaintext variation on Pipeline cipher power consumption.

Chapter 8

Conclusion and Future Work

The AES has been widely adopted in modern automobiles as a solution to ensure security of the data travelling between increasingly sophisticated electronic systems. However, the implementations of this algorithm for the automotive industry, require adaptations for adding safety measures to comply with the functional safety standard ISO-26262. In this dissertation, it was evaluated whether these safety measures can create or intensify implementation vulnerabilities which may allow unauthorised access to critical data that ultimately can compromise the vehicle's and people safety. With that purpose, an IP core implementing the AES was adapted to meet the ASIL-D safety requirements and was compared in terms of vulnerability to power analysis attacks with the default implementation.

This dissertation demonstrated that the safety improvements applied to the ASIL-D configuration can increase the vulnerability of the implementation, by intensifying the points of interest for attacks and increasing the leakage of sensitive information. However, this work also suggests that the adoption of certain techniques during the implementation of the safety measures can be used to mitigate the vulnerabilities created or to develop security countermeasures for the system.

The main conclusion of this work is that the dependency between safety and security goes far beyond what is currently considered in the automotive industry. If it was already known that, in a first instance, a safe system had to be secure, from this work it is possible to conclude that the safety enhancement also needs to be done taking into account the security aspects. Thus, if the automotive industry wants to evolve in the way of safer and secure automobiles, it will also need to evolve in the direction of unifying the existing independent standards that cover each one of these domains. And only then, it will be possible to establish clear guidelines so that each one of these domains can be developed without compromising the other.

The work initiated with this dissertation has then brought into question several particularities of these domains of safety and security that would be interesting to explore in the future. One of the most relevant would be to investigate how the safety measures implemented could be modified, so that instead of increasing the vulnerability of the system, they could make it more robust to attacks. The identification of the vulnerabilities done in this work demonstrated that one of the possibilities would be to distribute the operation of the safety mechanisms along the clock period,

instead of an operation at the same time. This would spread the dynamic power consumption, decreasing the power peaks and consequently flatten the power traces.

This dissertation was only focused on power analysis attacks, which were selected as the starting point due to their popularity and efficiency. Other possibilities would be the attacks that explore the timing characteristics or the electromagnetic emanations of the implementations, for example. Since this work already provides the ASIL-D compliant implementation, future works may analyse the susceptibility of this implementation to other types of side-channel attacks.

Appendix A

Failure Modes, Effects, and Diagnostic Analysis (FMEDA)

The FMEDA presented here is a simplified version of the one developed in the context of this work, which summarises the most relevant aspects of the analysis performed. The complete version used a proprietary template from Synopsys.

The FMEDA includes all the blocks of the ASIL-D configuration, along with their failure rates (expressed in FIT) and the respective failure modes. For each failure mode, it was indicated the corresponding distribution in the total failure rate of the block (in percentage) and whether this failure mode was considered or not as capable of violating each one of the four Safety Goals (SG) defined. The violation of safety goals through a single-point fault was indicated with *S* and the violation through a latent fault with *M*. If there was no hypothesis of violation, it was indicated with *N*. For each failure mode, it was also indicated the safety mechanisms used (if any) to cover eventual single-point faults and latent faults. Finally, it was shown the effective failure rates for each failure mode, which were used to calculate the Single-Point Fault Metric (SPFM) and the Latent Fault Metric (LFM) for the configuration.

At the end, the results of the FMEDA for the developed configuration are presented, which testify that the metrics for an ASIL-D rating have been achieved.

Component Name	Failure Rate	Failure Mode (FM)	FM Distribution	Safety Goal Violated (S=SPF; M=MPF; N=Safe)				Safety mechanism allowing to prevent the failure mode from violating the safety goal?	Failure Mode coverage wrt. Violation of safety goal	Detection means? Safety mechanism(s) allowing to prevent the failure mode from being latent?	Failure mode coverage wrt. Latent failures	Residual or Single-Point Fault failure rate/FIT	Latent Multiple-Point Fault failure
				SG1	SG2	SG3	SG4						
Pipeline Cipher	1.2149	Incorrect round computation	37.23%	S	N	N	N	SM3	99%	-	-	0.004628	0.000000
		Incorrect round data sampling	4.54%	S	N	N	N	SM4	99%	-	-	0.000565	0.000000
		Incorrect processing of input commands	4.54%	S	N	S	N	SM4	99%	-	-	0.000565	0.000000
		Incorrect valid command transmission	4.54%	S	N	N	N	SM2	99%	-	-	0.000565	0.000000
		Incorrect duplicated rounds logic	37.23%	M	N	N	N	-	-	SM3	99%	0.000000	0.004628
		Incorrect comparison logic	11.92%	M	N	N	N	-	-	SM3	99%	0.000000	0.001482
Key Expander	0.2393	Incorrect encryption key sampling	16.80%	N	N	S	N	SM4	99%	-	-	0.000450	0.000000
		Incorrect key expansion algorithm	9.77%	N	N	S	N	SM3	99%	-	-	0.000262	0.000000
		Incorrect ready command transmission	16.80%	N	N	S	N	SM2	99%	-	-	0.000450	0.000000
		Incorrect processing of input commands	16.80%	N	N	S	S	SM4	99%	-	-	0.000450	0.000000
		Incorrect round keys sampling	16.80%	N	N	S	N	SM4	99%	-	-	0.000450	0.000000
		Incorrect key expander duplicated logic	9.78%	N	N	M	N	-	-	SM3	99%	0.000000	0.000262
		Incorrect comparison logic	13.25%	N	N	M	N	-	-	SM3	99%	0.000000	0.000355
Context Manager	0.1667	Incorrect context index processing	32.58%	S	S	S	S	SM2	99%	-	-	0.000585	0.000000
		Incorrect context data read	32.58%	S	S	S	S	SM4	99%	-	-	0.000585	0.000000
		Incorrect context data latch	32.58%	S	S	S	S	SM4	99%	-	-	0.000585	0.000000
		Incorrect error detection logic	2.27%	M	M	M	M	SM2	-	SM2	99%	0.000000	0.000041
Context Store	0.0365	Incorrect encryption key load	22.53%	S	N	S	N	SM4	99%	-	-	0.000095	0.000000
		Incorrect key busy command processing	22.53%	S	N	S	N	SM2	99%	-	-	0.000095	0.000000
		Incorrect key sampling	22.53%	S	N	S	N	SM4	99%	-	-	0.000095	0.000000
		Incorrect key write command transmission	22.53%	S	N	S	N	SM2	99%	-	-	0.000095	0.000000
		Incorrect error detection logic	9.89%	M	N	M	N	-	-	SM2	99%	0.000000	0.000042
AES FIFO	0.0765	Incorrect empty/full command processing	23.16%	S	S	S	S	SM2	99%	-	-	0.000205	0.000000
		Address error when accessing FIFO buffer	23.16%	S	S	S	S	SM1	99%	-	-	0.000205	0.000000
		Read/write data error when accessing FIFO buffer	23.16%	S	S	S	S	SM1	99%	-	-	0.000205	0.000000
		Incorrect ECC protection	30.51%	M	M	M	M	-	-		-	0.000000	0.027040
Egress FIFO	0.0456	Incorrect empty/full command processing	27.27%	S	S	N	N	SM2	99%	-	-	0.000128	0.000000
		Incorrect address when accessing FIFO buffer	27.27%	S	S	N	N	SM1	99%	-	-	0.000128	0.000000
		Incorrect read/write data when accessing FIFO buffer	27.27%	S	S	N	N	SM1	99%	-	-	0.000128	0.000000
		Incorrect ECC protection	18.19%	M	M	N	N	-	-		-	0.000000	0.008553
GMUL FIFO	0.0527	Incorrect empty/full command processing	23.34%	S	S	N	S	SM2	99%	-	-	0.000142	0.000000
		Incorrect address when accessing FIFO buffer	23.34%	S	S	N	S	SM1	99%	-	-	0.000142	0.000000
		Incorrect read/write data when accessing FIFO buffer	23.34%	S	S	N	S	SM1	99%	-	-	0.000142	0.000000
		Incorrect ECC protection	29.99%	M	M	N	M	-	-	-	-	0.000000	0.018303

Ingress FIFO	0.2448	Incorrect empty/full command processing	27.26%	S	S	S	S	SM2	99%	-	-	0.000671	0.000000
		Incorrect address when accessing FIFO buffer	27.26%	S	S	S	S	SM1	99%	-	-	0.000671	0.000000
		Incorrect read/write data when accessing FIFO buffer	27.26%	S	S	S	S	SM1	99%	-	-	0.000671	0.000000
		Incorrect ECC protection	18.21%	M	M	M	M	-	-	-	-	0.000000	0.044862
AES Match FIFO	0.1013	Incorrect empty/full command processing	30.56%	S	S	S	S	SM2	99%	-	-	0.000314	0.000000
		Incorrect address when accessing FIFO buffer	30.56%	S	S	S	S	SM1	99%	-	-	0.000314	0.000000
		Incorrect read/write data when accessing FIFO buffer	30.56%	S	S	S	S	SM1	99%	-	-	0.000314	0.000000
		Incorrect ECC protection	8.32%	M	M	M	M	-	-	-	-	0.000000	0.008554
GMUL System	0.5487	Incorrect MAC computation	45.47%	N	S	N	N	SM3	99%	-	-	0.002519	0.000000
		Incorrect gmul operands load	1.96%	N	S	N	N	SM4	99%	-	-	0.000109	0.000000
		Incorrect processing of input commands	1.96%	N	S	N	S	SM4	99%	-	-	0.000109	0.000000
		Incorrect gmul result sampling	1.96%	N	S	N	N	SM4	99%	-	-	0.000109	0.000000
		Incorrect galois multiplier duplicated logic	45.52%	N	M	N	N	-	-	SM3	99%	0.000000	0.002522
		Incorrect comparison logic	3.13%	N	M	N	M	-	-	SM2	99%	0.000000	0.000173
GMUL Context Manager	0.2524	Incorrect context index processing	32.58%	S	S	N	S	SM2	99%	-	-	0.000909	0.000000
		Incorrect context data read	32.58%	S	S	N	S	SM4	99%	-	-	0.000909	0.000000
		Incorrect context data latch	32.58%	S	S	N	S	SM4	99%	-	-	0.000909	0.000000
		Incorrect error detection logic	2.27%	M	M	N	M	-	-	SM2	99%	0.000000	0.000063
Ifr Read	0.0359	Interface error with the next stage	45.74%	S	S	S	N	SM4	99%	-	-	0.000189	0.000000
		Incorrect operation commands processing	45.74%	S	S	S	S	SM4	99%	-	-	0.000189	0.000000
		Incorrect error detection logic	8.53%	M	M	M	M	-	-	SM2	99%	0.000000	0.000035
Ifw Write	0.0174	Interface error with the previous stage	45.51%	S	S	S	N	SM4	99%	-	-	0.000092	0.000000
		Incorrect operation commands processing	45.51%	S	S	S	S	SM4	99%	-	-	0.000092	0.000000
		Incorrect error detection logic	8.97%	M	M	M	M	-	-	SM2	99%	0.000000	0.000018
AES Fr	0.1022	Interface error with the next stage	45.73%	S	S	S	N	SM4	99%	-	-	0.000539	0.000000
		Incorrect operation commands processing	45.73%	S	S	S	S	SM4	99%	-	-	0.000539	0.000000
		Incorrect error detection logic	8.54%	M	M	M	M	-	-	SM2	99%	0.000000	0.000101
GMUL Fw	0.0333	Interface error with the previous stage	45.54%	S	S	S	N	SM4	99%	-	-	0.000176	0.000000
		Incorrect operation commands processing	45.54%	S	S	S	S	SM4	99%	-	-	0.000176	0.000000
		Incorrect error detection logic	8.93%	M	M	M	M	-	-	SM2	99%	0.000000	0.000034
Safety Monitor	0.0013	Incorrect error signaling	100.00%	M	M	M	M	-	99%	-	-	0.000000	0.001487

SG	SPFM	LPFM
SG1	99.478%	96.575%
SG2	99.716%	96.685%
SG3	99.689%	97.520%
SG4	99.789%	97.030%

Metrics	Target	Overall Value
SPFM	> 99%	0.99326
LFM	> 90%	0.96416
PMHF	< 10 FIT	0.0734 FIT

References

[1] National Institute of Standards and Technology. Data Encryption Standard (DES). Technical Report FIPS 46-3, FIPS PUBS., 1999.

[2] W. Stallings. *Cryptography And Network Security: Principles And Practices 4Th Ed.* Prentice-Hall Of India Pvt. Limited, 2006.

[3] James Nechvatal, Elaine Barker, Lawrence Bassham, William Burr, and Morris Dworkin. Report on the development of the Advanced Encryption Standard (AES). Technical report, DTIC Document, 2000.

[4] N. Wiersma and R. Pareja. Safety != security: On the resilience of asil-d certified microcontrollers against fault injection attacks. In *Proceedings - 2017 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pages 9–16, 2017.

[5] National Institute of Standards and Technology. Advanced Encryption Standard (AES). Technical Report FIPS 197, FIPS PUBS., 2001.

[6] Christof Paar and Jan Pelzl. *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer Publishing Company, Incorporated, 2009.

[7] Morris Dworkin. Recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac. Technical Report SP 800-38D, National Institute of Standards and Technology, 2007.

[8] Morris Dworkin. Recommendation for block cipher modes of operation: Methods and techniques. Technical Report SP 800-38A, National Institute of Standards and Technology, 2001.

[9] Riccardo Cassettari, Luca Fanucci, and Giorgio Boccini. A new hardware implementation of the advanced encryption standard algorithm for automotive applications. In *2014 10th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, pages 1–4, 2014.

[10] Luca Baldanzi, Luca Crocetti, Matteo Bertolucci, Luca Fanucci, and Sergio Saponara. *Analysis of Cybersecurity Weakness in Automotive In-Vehicle Networking and Hardware Accelerators for Real-Time Cryptography*, pages 11–18. 2019.

[11] Satya S. Karanki and Mohammad S. Khan. Smmv: Secure multimedia delivery in vehicles using roadside infrastructure. *Vehicular Communications*, 7:40 – 50, 2017.

[12] X. Ni, W. Shi, and V. F. S. Fook. AES security protocol implementation for automobile remote keyless system. In *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, pages 2526–2529, 2007.

[13] Richard Soja. Automotive security : From standards to implementation. Technical report, Automotive Microcontrollers and Processors, NXP, 2014.

[14] Alastair Ruddle, Benjamin Weyl, Sajid Idrees, Y. Roudier, Michael Friedewald, Timo Leimbach, A. Fuchs, S. Gürgens, O. Henninger, Roland Rieke, M. Ritscher, H. Broberg, L. Apvrille, R. Pacalet, and Gabriel Pedroza. Security requirements for automotive on-board networks based on dark-side scenarios. deliverable d2.3: Evita. e-safety vehicle intrusion protected applications. *Fraunhofer ISI*, 2009.

[15] K. Sha, A. Striege, and M. Song. *Security, Privacy and Reliability in Computer Communications and Networks*. River Publishers Series in Communications. River Publishers, 2016.

[16] National Institute of Standards and Technology. Security Requirements for Cryptographic Modules. Technical Report FIPS 140-3, FIPS PUBS., 2019.

[17] François-Xavier Standaert. Introduction to side-channel attacks. In *Secure Integrated Circuits and Systems*, pages 27–42. Springer, Boston, MA, 2010.

[18] Bruce Jacob, Spencer W. Ng, and David T. Wang. Power and leakage. In *Memory Systems: Cache, DRAM, Disk*, pages 847 – 864. Morgan Kaufmann, 2008.

[19] Paul Kocher, Joshua Jaffe, Benjamin Jun, and Pankaj Rohatgi. Introduction to differential power analysis. *J. Cryptographic Engineering*, 1:5–27, 2011.

[20] Thomas Messerges, Ezzy Dabbish, and Robert Sloan. Examining smart-card security under the threat of power analysis attacks. *IEEE Transactions on Computers*, 51(5):541–552, 2002.

[21] Stefan Mangard. A simple power-analysis (spa) attack on implementations of the aes key expansion. In *Information Security and Cryptology - ICISC 2002: 5th International Conference, Seoul, Korea*, pages 343–358. Springer Verlag, 2003.

[22] Stefan Mangard, Norbert Pramstaller, and Elisabeth Oswald. Successfully attacking masked aes hardware implementations. In *Proceedings of the 7th International Conference on Cryptographic Hardware and Embedded Systems*, volume 3659, pages 157–171. Springer-Verlag, 2005.

[23] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In *Cryptographic Hardware and Embedded Systems*, pages 16–29. Springer, Berlin, Heidelberg, 2004.

[24] Owen Lo, William Buchanan, and Douglas Carson. Power analysis attacks on the aes-128 s-box using differential power analysis (dpa) and correlation power analysis (cpa). *Journal of Cyber Security Technology*, pages 1–20, 2016.

[25] M. Elaabid, Sylvain Guilley, and Philippe Hoogvorst. Template attacks with a power model. *IACR Cryptology ePrint Archive*, page 443, 2007.

[26] International Organization for Standardization. Road vehicles - functional safety. Standard ISO 26262:2018, 2018.

[27] Hans-Leo Ross. New challenges and solutions for e-mobility and automated driving. In *Functional Safety for Road Vehicles*. Springer International Publishing, 2016.

[28] G. Macher, M. Bachinger, A. Kager, M. Stolz, and C. Kreiner. Integrating SEooC components in highly automated vehicles. *Communications in Computer and Information Science*, 896:56–67, 2018.

[29] Synopsys Inc. What is ASIL (Automotive Safety Integrity Level)? - overview, 2020. URL: https://www.synopsys.com/automotive/what-is-asil.html [last accessed 2020-06-15].

[30] Simon Burton, Jürgen Likkei, Priyamvadha Vembar, and Marko Wolf. Automotive functional safety = safety + security. In *Proceedings of the First International Conference on Security of Internet of Things*, pages 150–159. Association for Computing Machinery, 2012.

[31] International Electrotechnical Commission. Reliability data handbook - Universal model for reliability prediction of electronics components, PCBs and equipment. Technical Report IEC TR 62380, 2004.

[32] International Electrotechnical Commission. Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems. Technical Report IEC 61508, 2010.

[33] Kenneth James Smith. Methodologies for Power Analysis Attacks on Hardware Implementations of AES. Master's thesis, Rochester Institute of Technology, Rochester, New York, 2009.