S.A.B.E.R : Secure AI-Based Encrypted ECG Rhythm-monitoring

Caio NOBRE
Lívia BELO LIMA
Mario ARAUJO
Murilo MÜLLER
Vinicius ROCCA

Supervised by
Olivier POTIN - EMSE

August/2025

Abstract

The integration of medical devices with Internet of Things (IoT) technologies has raised significant concerns regarding the protection of sensitive patient data. This work presents **S.A.B.E.R.** (Secure AI-Based Encrypted ECG Rhythm-monitoring), an embedded system designed to acquire, process, encrypt, and transmit electrocardiogram (ECG) signals in real time. The architecture is based on an FPGA platform and comprises three main stages: ECG preprocessing, a hardware-implemented lightweight cryptographic pipeline using ASCON-128, and a communication interface for secure transmission. Experimental validation through testbenches, hardware simulations, and FPGA implementation demonstrates reliable operation with low latency, efficient resource utilization, and compliance with lightweight cryptography standards. The results confirm the feasibility of combining real-time biomedical signal processing with embedded hardware encryption, paving the way for secure and scalable IoT-based medical devices.

1. Introduction

The rapid integration of medical devices with Internet of Things (IoT) technologies has introduced new opportunities for continuous patient monitoring and remote healthcare. Among the physiological signals commonly analyzed, electrocardiograms (ECGs) play a central role, enabling early detection of cardiovascular conditions. However, transmitting such data over untrusted networks exposes patients to privacy and security risks, making data protection a critical requirement in the design of modern medical systems.

To address this challenge, we propose **S.A.B.E.R.** (Secure AI-Based Encrypted ECG Rhythm-monitoring), a hardware-based framework for secure acquisition and transmission of ECG signals. The system leverages an FPGA implementation that integrates three tightly coupled stages: (i) preprocessing, including segmentation of raw ECG signals into fixed-length frames; (ii) a lightweight cryptographic pipeline implementing ASCON-128 directly in hardware to ensure data confidentiality and integrity in real time; and (iii) a communication interface responsible for packaging and transmitting the encrypted output to external devices or servers. All modules are coordinated by finite state machines (FSMs), which manage data flow between acquisition, encryption, and communication.

On the server side, the transmitted frames can be decrypted and visualized. Additionally, artificial intelligence algorithms may be applied to assess heartbeat normality, providing decision support for automated diagnosis. Experimental validation through FPGA prototyping confirms that the system processes and encrypts ECG frames efficiently, with minimal latency and optimized logic resource usage.

In summary, this work demonstrates the feasibility of integrating real-time biomedical signal processing with embedded lightweight cryptography on FPGA platforms. The proposed prototype represents an important step toward secure, energy-efficient, and scalable IoT medical devices, contributing both to patient data protection and to the advancement of trustworthy digital healthcare technologies.

2. Context and Scope of the project

2.1 Objectives and challenges

The primary objective of this project is to design and implement an embedded system capable of acquiring, processing, encrypting, and transmitting electrocardiogram (ECG) signals in real time, while ensuring data confidentiality and integrity. The system must operate efficiently on an FPGA platform, integrating lightweight cryptography with biomedical signal processing, and remain compatible with the constraints of IoT-based medical devices.

Several challenges arise in achieving these goals. First, the continuous nature of ECG acquisition requires a seamless integration between signal preprocessing and encryption, without introducing latency that could compromise real-time monitoring. Second, the cryptographic core must be lightweight enough

to meet the limited resource availability of FPGA implementations, yet strong enough to guarantee robust security. Third, efficient coordination between modules through finite state machines is necessary to manage data flow reliably. Finally, the system must ensure interoperability with external servers, where decrypted signals may be further analyzed by artificial intelligence algorithms for anomaly detection.

By addressing these objectives and challenges, the project demonstrates the feasibility of combining secure cryptography with biomedical signal processing in resource-constrained environments.

2.2 System Architecture Overview

Figure 1 presents an overview of the S.A.B.E.R. system. The ECG signal is first acquired through an analog-to-digital converter (ADC) and stored in a data register, which organizes the samples into 64-bit words. A multiplexer selects between signal data and associated data, which, together with the 128-bit key and the internally generated *nonce* value, are provided to the ASCON-128 cryptographic core implemented in hardware.

The encrypted output is stored in a dedicated register and then transmitted via a communication interface to an external server. On the server side, the received frames can be decrypted and visualized, and optionally processed by artificial intelligence algorithms for automated assessment of signal normality.

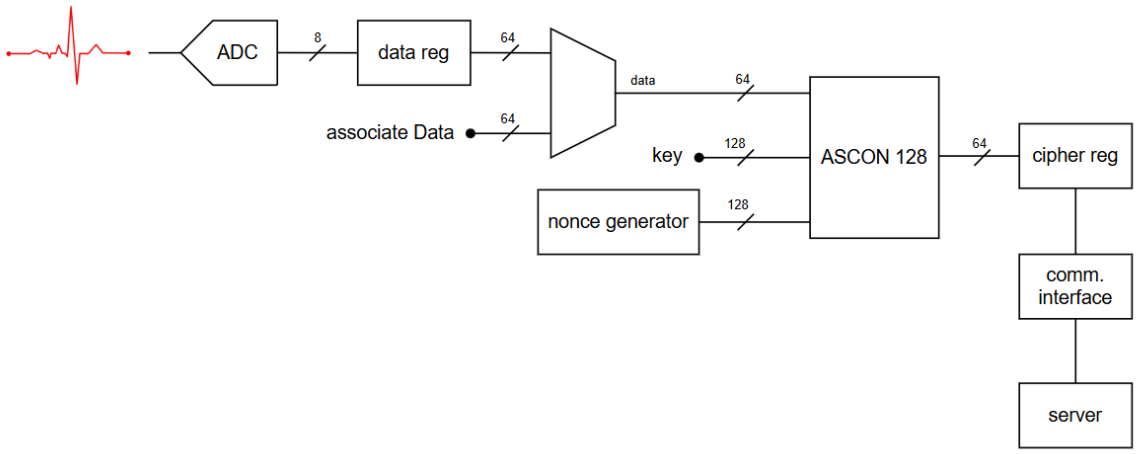

Figure 1: System Architecture Overview

2.3 Acquisition and Preprocessing

The acquisition process begins with the conversion of the analog ECG signal into digital values through an analog-to-digital converter (ADC). Each sample is represented using 8 bits, ensuring sufficient resolution for subsequent analysis. The ADC is configured to perform continuous conversions.

The digitized samples are then stored in a data register that performs sequential grouping. Every eight samples are combined to form a 64-bit block, which corresponds to the word size required by the ASCON-128 cryptographic core. Since the ADC continuously provides new values, an *add* signal manages the insertion of data into the register. This signal samples the input at 360 Hz and is generated by a finite state machine (FSM).

As a result, the register is constantly updated with sequential values, and every eight occurrences of the *add* signal a new 64-bit block is produced. Furthermore, to standardize the block generation, the 23rd block contains only five values from the ADC, with standard padding added to complete the 64-bit size.

This procedure ensures compatibility between the continuous flow of biomedical data and the input interface of the encryption pipeline.

2.4 ASCON128 cryptography bloc

The ECG data encryption project uses ASCON128 implemented in SystemVerilog. The ASCON-128 algorithm was chosen because it features a simple permutation design, a small number of rounds, efficient operations, and compact state and key sizes. It performs encryption and authentication in a single step, classifying it as a lightweight algorithm suitable for implementation on resource-constrained devices.

To implement encryption, the ASCON cipher suite is applied in its simplified mode and encryption is authenticated using associated data (AEAD). Accordingly, the message to be processed and the system initialization consist of:

- 32 bits of associated data, which forms a single 64-bit block after the *padding* step;
- 248 bits of plaintext, resulting in four 64-bit blocks after *padding*;
- 128 bits of the encoding key K;
- 128 bits of an arbitrary number *nonce* N;

2.4.1 ASCON 128 algorithme description

The encryption mode requires the nonce to be unique, but it uses a secret key during both the initialization and finalization phases to ensure stronger security, as illustrated in the diagram in Figure 2.

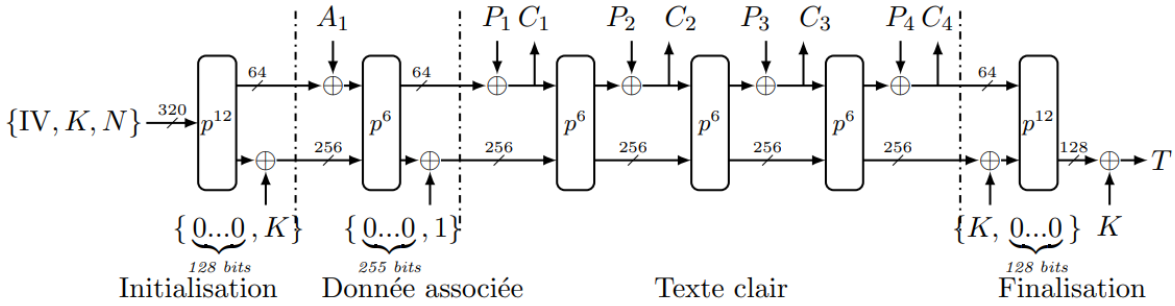

Figure 2: ASCON128

The operation mode can thus be divided into four parts:

- **Initialization:** The system takes as input an initialization vector, the secret key, and a unique arbitrary number. In addition, twelve rounds of the permutation p are applied to the initial state, followed by an XOR with the secret key K;
- **Associated data:** ASCON processes the associated data by applying an XOR between the initial state and the associated data, followed by six permutations, and finally an XOR with a 1-bit separation constant;
- **Plaintext:** ASCON processes the plaintext P in four 64-bit blocks and produces the corresponding ciphertext blocks;
- **Finalization:** The authentication tag is generated and retrieved.

It should be noted that a 12-round permutation is used for both the initialization and finalization phases.

2.4.2 Encryption process

The overall architecture of ASCON features a permutation block with XOR operations, a control block implemented via a finite state machine, and two blocks for counting rounds and data blocks.

First, the permutation block is constructed, where a transformation p is applied to each type of permutation (p^6 and p^{12}), defined as:

$$p = p_c \circ p_s \circ p_c \tag{1}$$

and integrated with the *xor_begin* and *xor_end* operators.

Next, the state machine is designed, here implemented as a Moore machine, and the counters required by the model are coupled accordingly.

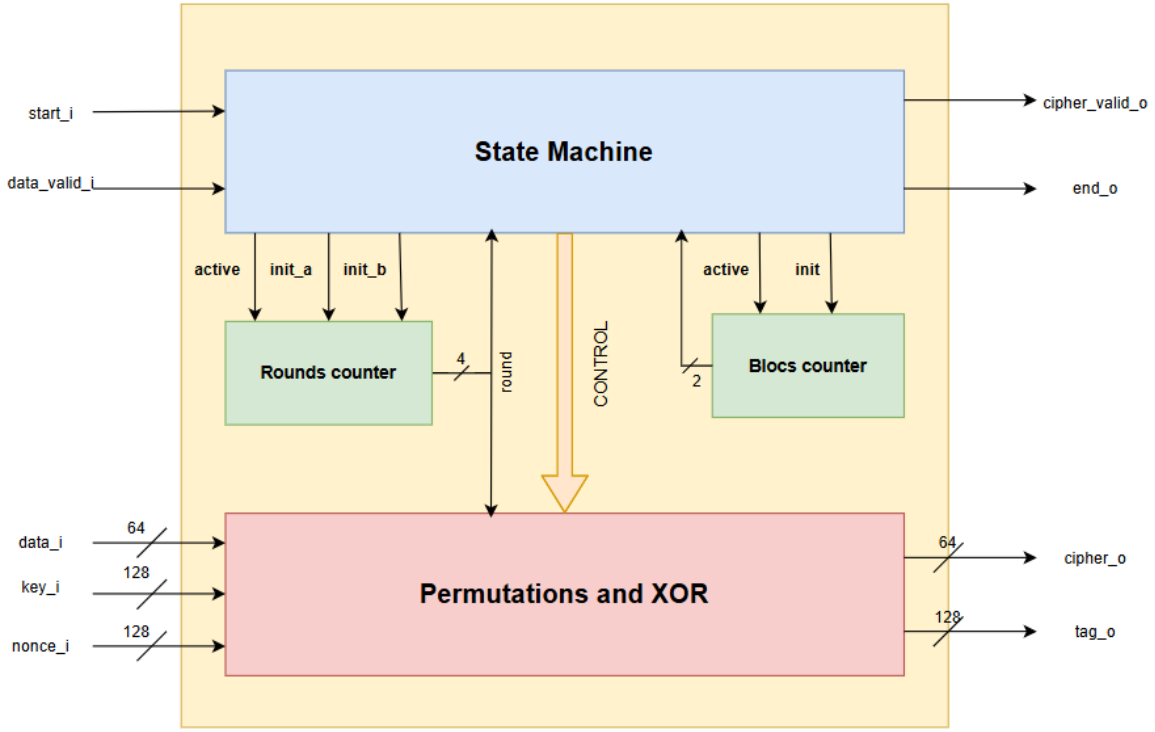

Figure 3: ASCON128 Global architecture

Finally, all the blocks are integrated, as shown in Figure 3, to achieve a functional and efficient cryptographic system..

2.4.3 Constant Addition p_c

Following the ASCON128 algorithm presented in the previous section, the round transformation p_c is first defined as adding a round constant c_r to the state register x_2 for each round. In other words:

$$x_2 \Leftarrow x_2 \oplus c_r \tag{2}$$

The round constants are listed in the following table:

Round r of p^{12}	Round r of p^6	Constant c_r
0		00000000000000000f0
1		00000000000000000e1
2		00000000000000000d2
3		00000000000000000c3
4		00000000000000000b4
5		00000000000000000a5
6	0	0000000000000000096
7	1	0000000000000000087
8	2	0000000000000000078
9	3	0000000000000000069
10	4	000000000000000005a
11	5	000000000000000004b

Table 1: Round constant addition for both permutations.

2.4.4 The Substitution Layer p_s

The substitution layer p_s modifies the state S by applying a parallel 5-bit column substitution using the S-box defined in Table 2.

x	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
$S(x)$	04	0B	1F	14	1A	15	09	02	1B	05	08	12	1D	03	06	1C
x	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
$S(x)$	1E	13	07	0E	00	0D	11	18	10	0C	01	19	16	0A	0F	17

Table 2: Substitution table.

2.4.5 The Linear Diffusion Layer p_l

The linear diffusion layer p_l applies a diffusion operation to each 64-bit row, or register x_i. The operation is defined as:

$$x_i \leftarrow \sum_i x_i \tag{3}$$

For this project, the following operations are performed:

$$x_0 \leftarrow \Sigma(x_0) = x_0 \oplus (x_0 \ggg 19) \oplus (x_0 \ggg 28) \tag{4}$$

$$x_1 \leftarrow \Sigma(x_1) = x_1 \oplus (x_1 \ggg 61) \oplus (x_1 \ggg 39) \tag{5}$$

$$x_2 \leftarrow \Sigma(x_2) = x_2 \oplus (x_2 \ggg 1) \oplus (x_2 \ggg 6) \tag{6}$$

$$x_3 \leftarrow \Sigma(x_3) = x_3 \oplus (x_3 \ggg 10) \oplus (x_3 \ggg 17) \tag{7}$$

$$x_4 \leftarrow \Sigma(x_4) = x_4 \oplus (x_4 \ggg 7) \oplus (x_4 \ggg 41) \tag{8}$$

2.4.6 Development of the Permutations p^6 and p^{12}

Following the transformations, two XOR gates and the required multiplexer (MUX) were incorporated to complete the encryption process. The diagram below presents the final schematic.

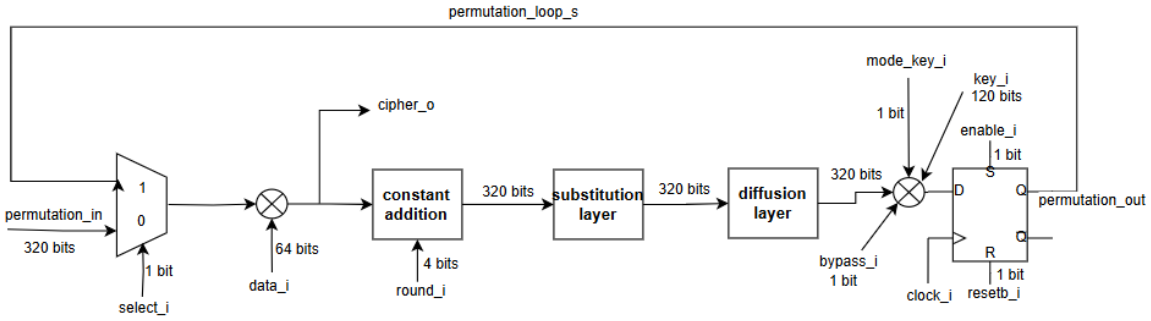

Figure 4: Development of the permutations

2.5 ECG characteristics

For this project, specific segments of the ECG signal were analyzed to support the future implementation of artificial intelligence methods for anomaly detection. Accordingly, the parameters illustrated in the figure were taken as the reference.

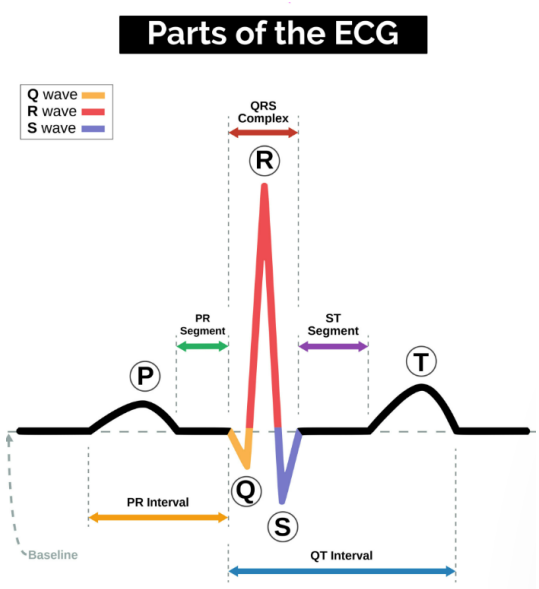

Figure 5: Parts of a ECG signal [5]

2.6 Wi-Fi + Platform

To complement the system, a web interface was developed that enables real-time access to ECG signals via Wi-Fi. The platform provides user authentication and a dedicated area for the visualization of collected and decrypted signals. This ensures not only secure data transmission, but also convenient remote access for patients or healthcare professionals.

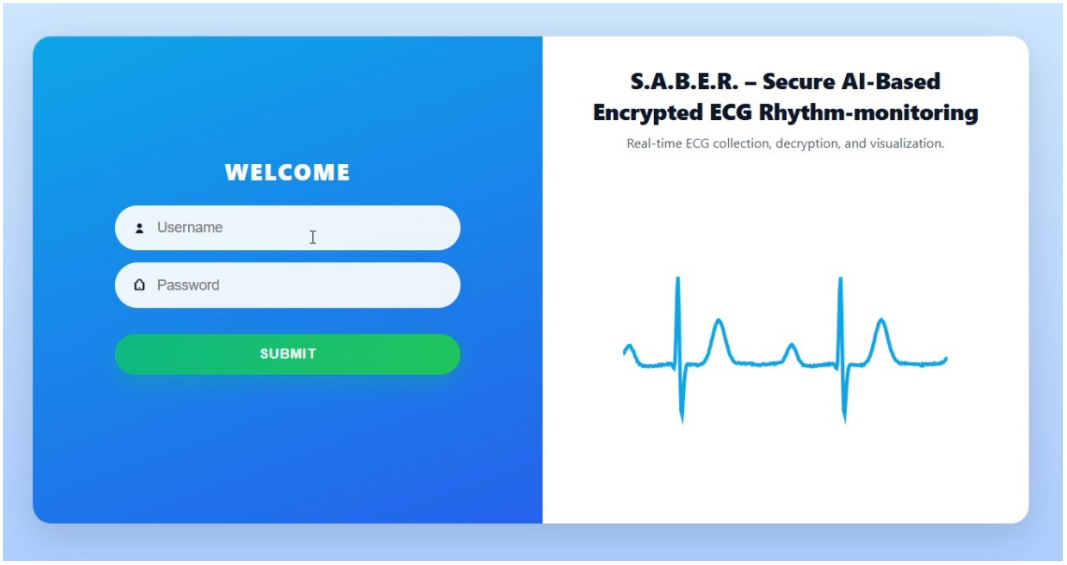

Figure 6: Authentication screen of the S.A.B.E.R. platform.

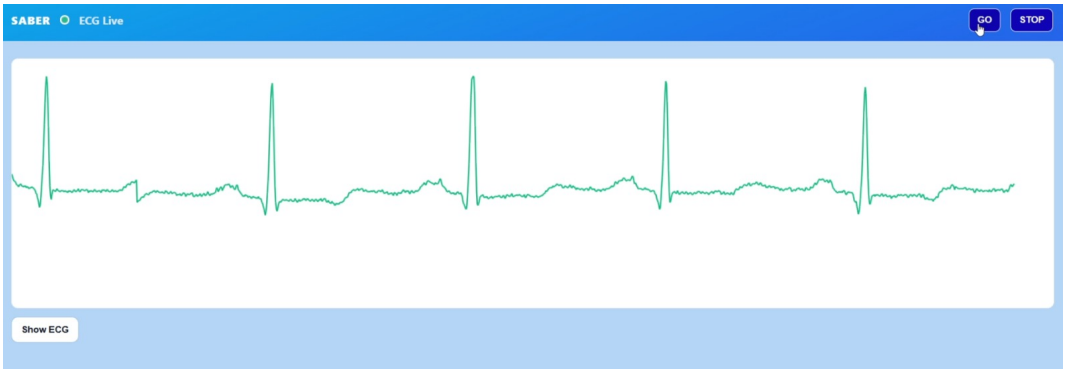

Figure 7: Real-time ECG visualization on the platform.

3. Results and Discussions

This section presents the results obtained from the synthesis and implementation of the project in Vivado, including FPGA resource utilization, power consumption, and timing verification. In addition, comparative plots between the original and the decrypted signal are shown.

3.1 FPGA Resource Utilization

After implementation, the resource utilization was found to be very low, leaving margin for future expansions. Table 3 summarizes the obtained values.

Table 3: FPGA resource utilization

Resource	**Used**	**Total**	**Utilization**
LUTs	2,028	53,200	4%
FFs (Registers)	5,173	106,400	5%
Slices	1,466	13,300	11%
BRAM	2	140	1%
DSPs	0	220	0%
IOBs	2	125	1%

3.2 Power Consumption

The power analysis indicated that the total on-chip consumption was approximately 1.55 W, with most of it related to the ARM subsystem (PS7). The FPGA logic responsible for the cryptographic pipeline presented a reduced consumption, below 0.3 W. Table 4 presents the distribution.

Table 4: Power consumption in the FPGA

Type	**Power**	**Percentage**
Dynamic	1.414 W	91%
Static	0.138 W	9%
Total	**1.552 W**	100%

3.3 Timing Verification

The timing report confirmed that all constraints were satisfied, with no setup or hold violations. The main values are presented below:

- Worst Negative Slack (WNS): 1.374 ns

- Worst Hold Slack (WHS): 0.012 ns
- All endpoints met

These results confirm that the pipeline can operate in real time with stability.

3.4 Signal Integrity

The ECG signals collected were successfully encrypted, transmitted, and decrypted. The comparison between the original and the decrypted signal showed complete overlap, with a mean squared error close to zero, ensuring full data integrity.

4. Conclusion

The **S.A.B.E.R.** project demonstrated that it is possible to perform real-time encryption of biomedical signals using an FPGA, with low resource utilization and power consumption. The results confirmed that the solution preserves the integrity of the ECG signal, ensuring confidentiality without compromising performance.

The main highlighted points are:

- Reduced FPGA resource utilization (less than 5% of LUTs and FFs).
- Total power consumption of 1.55 W, with the cryptographic core responsible for only a minimal fraction.
- Stable pipeline, with no timing violations, suitable for real-time operation.
- Full preservation of the ECG signal after encryption and decryption.

Thus, the proposed system represents an efficient, scalable, and secure solution for medical IoT and telemedicine applications, with potential for expansion to other biomedical signals and integration with different wireless communication technologies.

References

[1] Rigaud, J.-B., & Dutertre, J.-M. *Modélisation SystemVerilog de l'algorithme de chiffrement ASCON. sujet_ascon, France.*.

[2] Dobraunig, C., Eichlseder, M., Mendel, F., & Schläffer, M. *Ascon v1.2. submission to the nist lightweight cryptography competition*. Tech. Rep., TU Graz, 2019.

[3] Daemen, J. *Permutation-based Encryption, Authentication and Authenticated Encryption.*. 2012.

[4] Daemen, J. *Permutation-based Encryption, Authentication and Authenticated Encryption.*. 2012.

[5] P., Lewis, J., Matthew & M, Ben. *Understanding an ECG*. Geek Medics,2025.