Cybersecurity using electronic circuits and Software

Ph D Bárbara Emma Sánchez Rinza[1], PH D .Carlos Ignacio Robledo Sánchez[1],
[1,]Benemerite Autonomous University of Puebla, Mexico, barbara.sanchez@correo.bup.mx, carlos.robledo@correo.buap.mx

Abstract–Cybersecurity is the practice of defending computers, servers, mobile devices, electronic systems, networks, and data from malicious attacks. It is also known as information technology security or electronic information security. In this work, a cybersecurity system is made by mixing electronic circuits, which then pass through software and from the correct communication between them, you can achieve an encryption, which can be quite robust, this because they do not converge between them and only generate a communication which will validate whether valid or not, the parity or verification of the bits.

Keywords—Arduino, Proteus, Protoboard, encrypt..

I. INTRODUCTION

This type of implementations are recurrent when a wireless communication is needed, an example of these can be the remote control of a car, the control of an electric gate, or the control of a smart home etc. Another example can be a token that is usually provided by banks, with the intention of having a unique key or a mechanism that only the owner carries for access to the platform [1, 2]. In this implementation by means of hardware the key mechanism to access the main program will be achieved.

This work used C language programming, which will be oriented to the Arduino, an indispensable device for the execution.

The Java language will be oriented to the main programming and together with the corresponding libraries, the communication between the circuit and the software will be achieved [3, 4].

Once the communication is achieved, a data exchange will take place in order to verify that the key or credential is the correct one in order to verify and access the main program [5].

II. METHODOLOGY

It is important that the structure of the implementation is done correctly and strictly because the communication will have to be reliable and perfectly implemented.

The scenario where the implementation will be done will be virtual, this in order to generalize and not focus only on one scenario, clarifying that taken to the physical state it will work perfectly.

Next, we will detail the different tools and what will be done for the elaboration of the structure of how it was implemented

1. The Proteus emulator is the main one, which allows interacting with several circuit components, using an Arduino, as well as the Protoboard, COMPIM, 74LS04, a circuit with 8 inputs and 8 outputs.

2. The Arduino was implemented an algorithm that when it detects an input this transmits it to the P box (permutation or implemented circuit), in turn will change state and the resulting state will be sent to the same Arduino to send it to the output zone.

3. The software implements the corresponding libraries and connects with the simulator, as it will be a data exchange, an input and an output of data will have to be implemented.

4. Validating the data to the Arduino and compares it with the predefined key that will later be used in the main window when access is achieved.

5. If everything works correctly, the first window in the form of a numeric keypad will close and then open the main window.

III P BOXES

Transpositions can be implemented using simple electrical circuits, Figure 1 below shows a device, known as a P-box (P-permutation) that is used to perform an 8-bit input transposition. If the 8 bits are designated from top to bottom as 01234567, the output of this particular P-box is 36071245.

With the proper internal wiring, a P-box can be made to perform any transposition at almost the speed of light, since no computation is required, only signal propagation. This design follows Kerckhoff's principle: the attacker knows that it is the general method is permuting bits. What he does not know is which bit goes where, and this is the key [6, 7].

Digital Object Identifier: (only for full papers, inserted by LACCEI).
ISSN, ISBN: (to be inserted by LACCEI).
DO NOT REMOVE

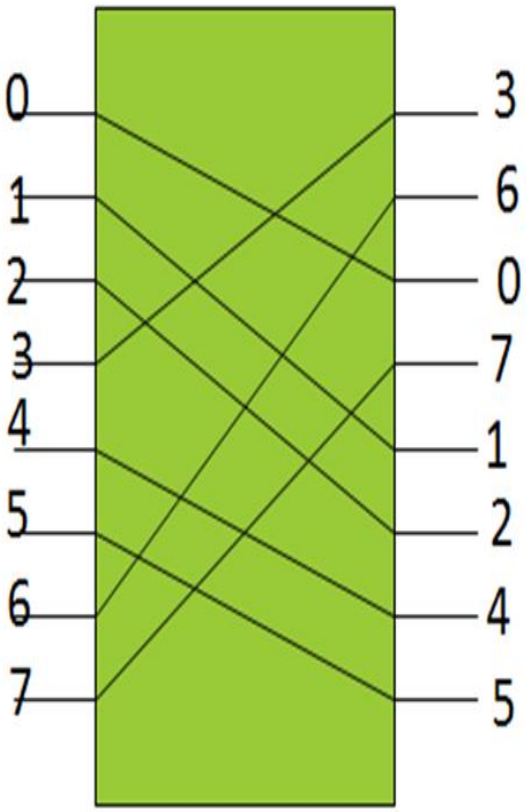

Figure 1. Basic elements of the product cipher Box P

IV IMPLEMENTATION

WE PROCEED TO IMPLEMENT THIS AS FOLLOWS

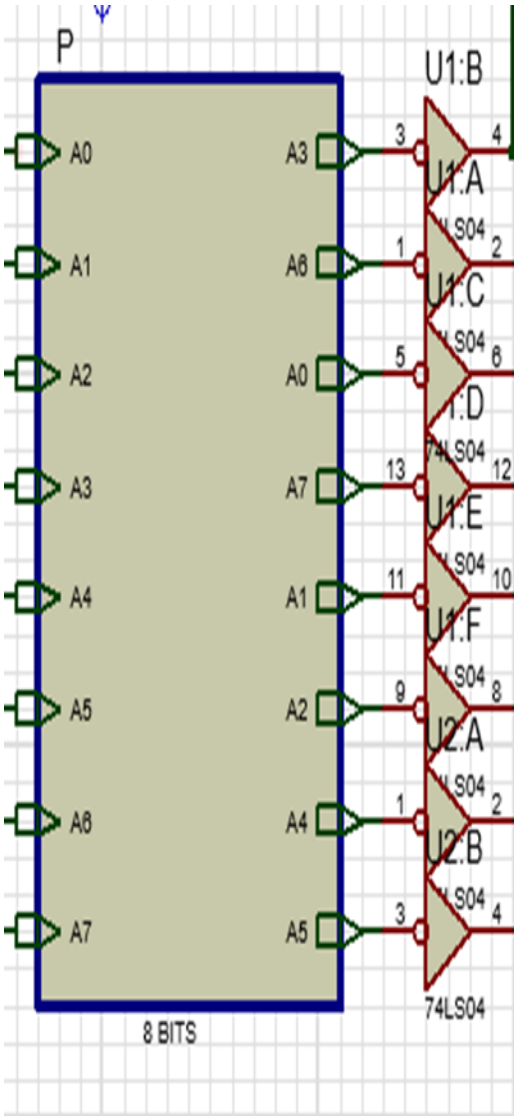

Figure 2. Permutation box where only the bits are permuted, for this arrangement

A) ENCRYPTION METHOD.

- The encryption method will be by means of change of position and permutations which will be a favorable combination, we will use the P box (permutation) that performs the rearrangement and by means of negation components, will change the state see figure 2.
- The arduino will be standard using a total of 16 connections 8 inputs and 8 outputs, this device will be programmed to provide incoming data to the input of the P box after the corresponding permutations will return through another input and then send it to the main program see figure 3.

- We will continue with the implementation of the software, which in this case will only perform the validation of the incoming data by the Arduino, for simplicity only the input key was implemented as valid, it will close the current window and allow passage to the main window, if not, it will display a warning message [8,9].

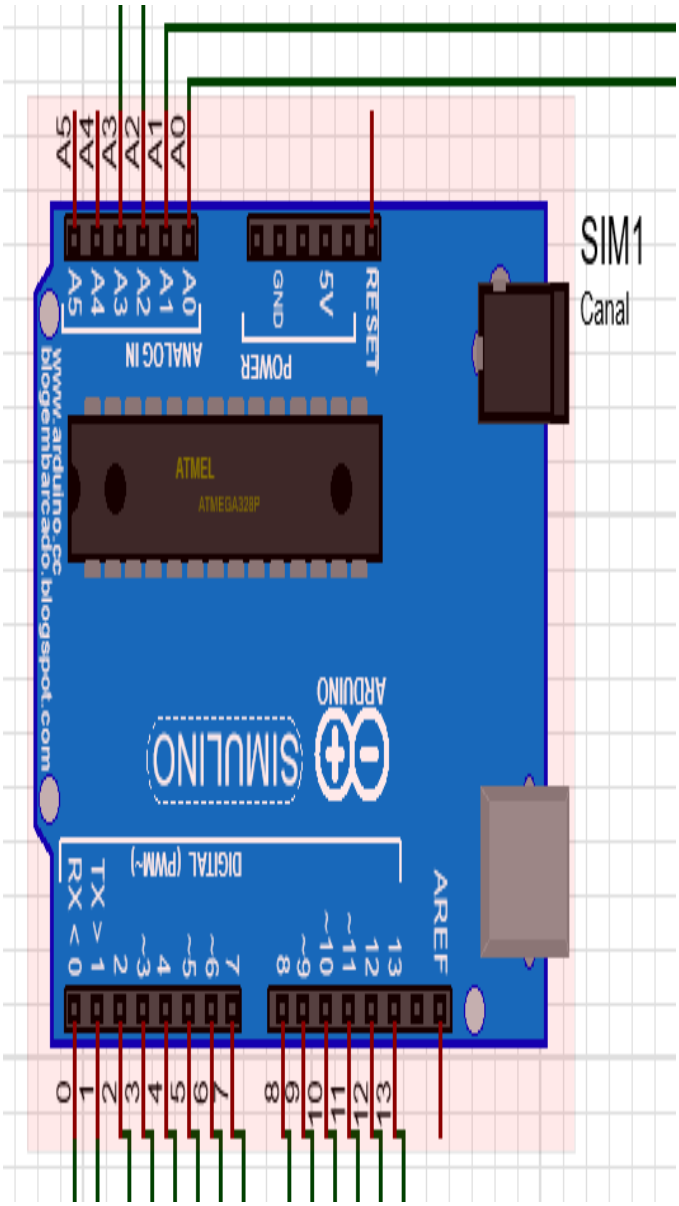

Figure 3: Arduino standard

```
String temporal;
temporal=Arduino.printMessage();
System.out.println("recibido "+temporal);
if(temporal.equals("10101010")){
    Client iniciar = new Client();
    iniciar.setVisible(true);
    iniciar.Llave11.setText(temporal);
    dispose();
}else{
    Error();
}
```

B) Interface.

To centralize in the implementation of circuits and connectivity, around the interface will be simple and comfortable in the aspect that a keyboard is handled, where digits will be partially divided into "1" and "0" but at sight will have or will be numbered from 1 to 9, see figure 4.

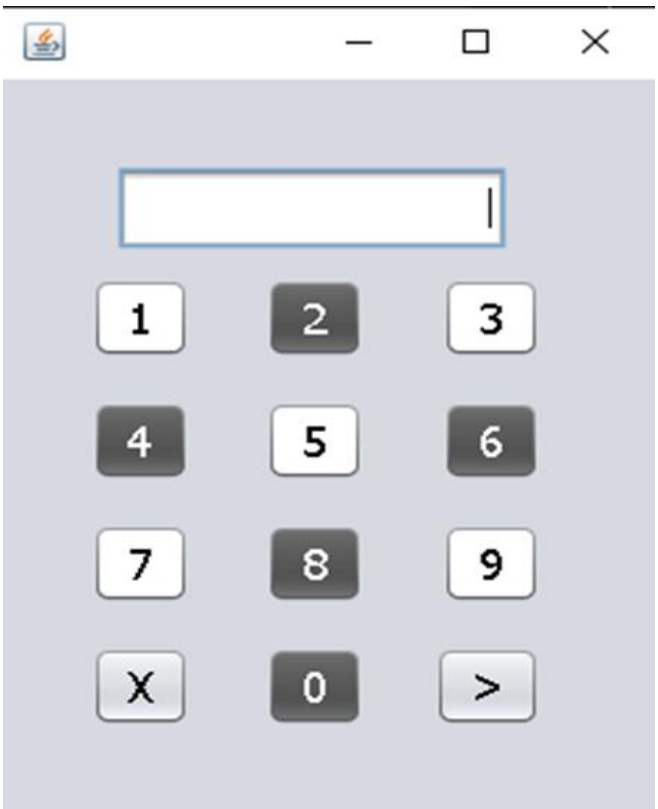

Figure 4. Access keyboard

Each light or white digit will internally function as 1 and the black buttons will function as 0, this implementation is mostly to add some complexity when entering any digit (key) see figure 4.

V RESULTS.

When the password implementation is correct, access is allowed, and in the case of an incorrect password, access is not allowed.

The password "10101010" was used, remembering that according to the rearrangement of bits and permutation the data entry will be different, in order to validate this the following sequence is entered, "00100111" which after validating it with the circuit will have an output equal to that of the password which will allow access to the main screen, see figure 5.

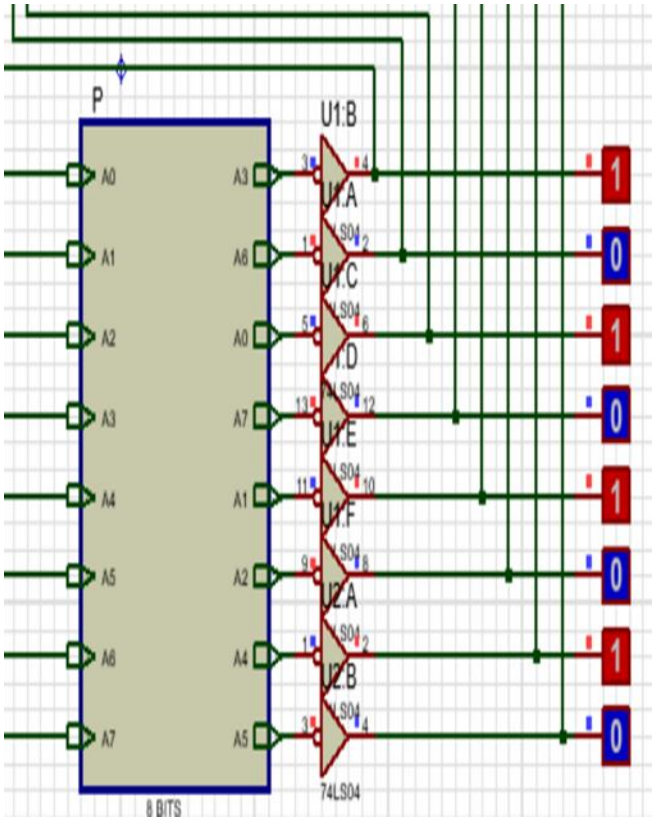

Figure 5. Password circuit

When entering the data will pass through the communication channel, duly to the Arduino, it will channel them to the P box and then pass them through its permutation, at the end it will send them to the output channel of the Arduino, some logical states were attached to validate the type of state that passes through the communication channels see figure 5.

As shown in Figure 5, the logical states show a sequence of ones and zeros that if we compare it with the password will be identical "10101010" which indicates that when communicating with the program it will validate the access and it will be possible to enter the main program, see figure 6.

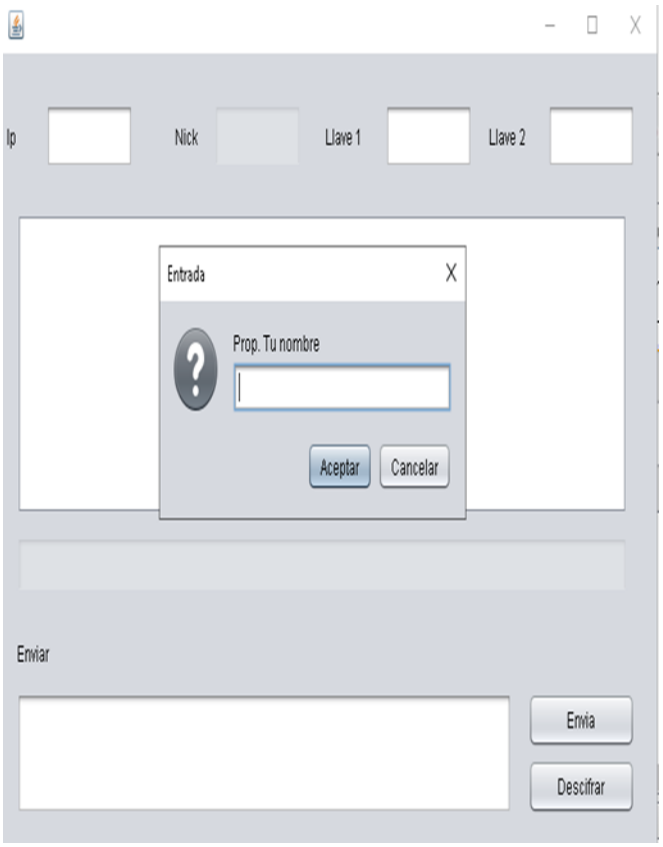

Figure 6. Main screen

As a result it gives us access to the main screen, which indicates that the validation was successful and the implementation was correct, see figure 6.

Only as an observation an invalid credential will be entered to see what it shows, in this case it will be "11001100".

In the implemented circuit it will show an output of the form "11010100" see figure 7.

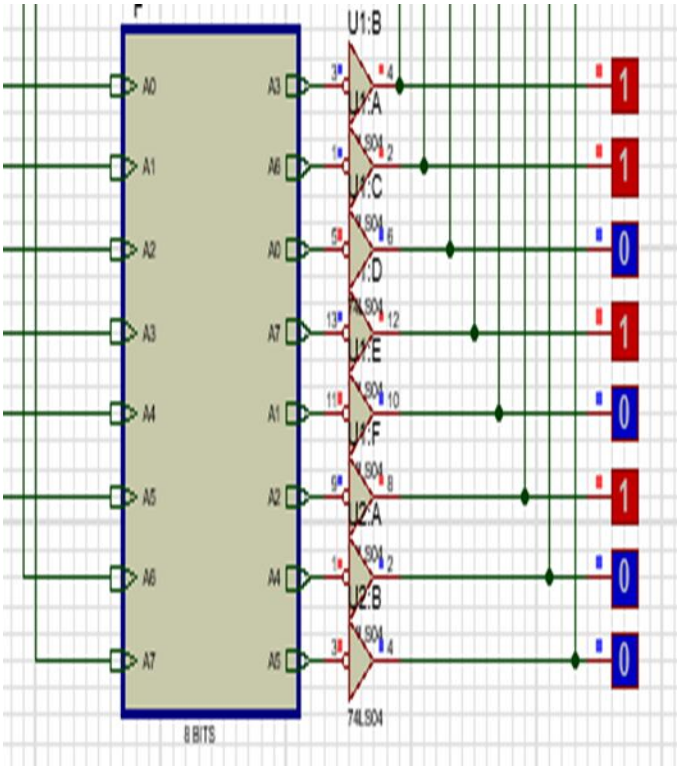

Figure 7. Circuit with invalid password

As a result, it will show that the credentials are wrong and will ask to enter them again, see figure 8 [10,11,12].

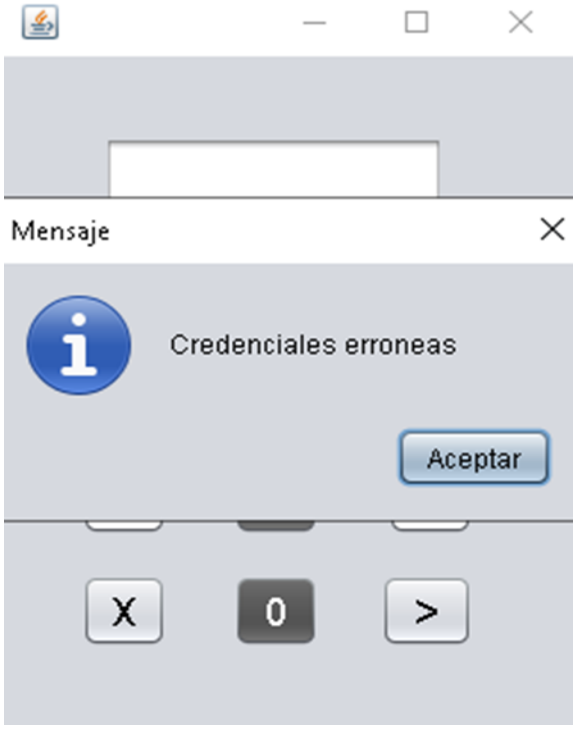

Figure 8. When the password is wrong, it will not allow access to the system.

VI. CONCLUSIONS

Considering the implemented software and hardware gives us as a result a robust implemented security, which helps us to know that by means of physical and digital tools can be mixed, in such a way to innovate for future implementations.

These types of blends are already being used in the automotive sectors, residential, banking, etc.

This is important since it is the trend in cybersecurity and what would be missing in this project would be to increase biometric security.

REFERENCES

[1] Pino Caballero Gil. (2002). Introduction to Criptography. España: Editorial Ra-Ma..

[2] Luis Hernández Encinas. (2016). The Criptography. United States: CSIC.

[3] Germán Tojeiro Calaza.. (2008). Proteus. Spain: MARCOMBO, S.A.

[4] Óscar Torrente Artero. (2013). ARDUINO Practical training course. Madrid, Spain: Alfaomega Grupo Editor.

[5] Carlos Taranilla de la Varga. (2018). Criptography (Science Outreach) United States.: GUADALMAZAN.

[6] Bárbara E. Sanchez-Rinza, et, Chat aplication with a codified information traveling option, World Journal of Research and Review, Issue 4 Vol 4,(2017) P1-3.

[7] Bárbara E. Sanchez-Rinza, et, Instant messaging application using cascading encryption based on the Vigenere and Hill algorithms. Cisci 2018,vol 17, (2018), pag154-159

[8] Barbara Emma Sánchez Rinza (2020). Development of a distributed system for monitoring electricity consumption in companies. ELECTRO magazine, Vol 42, PP 83-89.

[9] Barbara Emma Sánchez Rinza. (2020). Fire monitoring through the Internet of Things, ELECTRO magazine, vol42, pp 107-112.

[10] Bárbara Emma Sánchez Rinza, Jorge A. Cabrero, Mario Rossainz, (2021), Residential security through telegram Bot and PIR motion sensors, ELECTRO magazine, vol 43, pp 137-142.

[11] Sánchez, B. Bigurra, Diana. et all. De-Encryption of a text in Spanish using probability and statistics. 18th International Conference on Electronics, Communications and Computers: 2008.

[12] Sánchez, B. Cruz, S. Cesar decryption algorithm, but the method of frequency points in the Spanish language. International Journal of Engineering and Innovative Technology: 2013.