Instituto Tecnológico
y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

Especialidad en Sistemas Embebidos

DISEÑO DE UN SISTEMA CAN SEGURO Y SU IMPLEMENTACION USANDO AES EN DOS ECU'S

TRABAJO RECEPCIONAL que para obtener el **GRADO** de
ESPECIALISTA EN SISTEMAS EMBEBIDOS

Presenta: **JOSE ANTONIO GARIBAY CHAVEZ**

Asesor **HECTOR ANTONIO RIVAS SILVA**

Tlaquepaque, Jalisco. agosto de 2021.

DISEÑO DE UN SISTEMA CAN SEGURO Y SU IMPLEMENTACION USANDO AES EN DOS ECU´S

Garibay Chávez José Antonio
Departamento de electrónica
sistemas e informática
Instituto Tecnológico de Estudios
Superiores de Occidente
Tlaquepaque, México
jose.garibay@iteso.mx

Rivas Silva Héctor Antonio
Departamento de electrónica,
sistemas e Informática
Instituto Tecnológico de Estudios
Superiores de Occidente
Tlaquepaque, México
harivas@iteso.mx

***Abstract*—El objetivo de este trabajo es dar a conocer el diseño e implementación de un sistema seguro de comunicación CAN por medio del protocolo de encriptación AES 128 en dos tarjetas de evaluación, con comportamiento similar a las unidades de control electrónico automotrices. Se utilizaron cables de calibre reducido, dos tarjetas de evaluación de Atmel y una pantalla de cristal líquido de tamaño 16x2. Se concluyó que la información transmitida permanencia integra tanto en la transmisión y recepción con el algoritmo de encriptación, lo que sugiere que en otras área como la domótica podrían usar redes basadas en CAN con una capa de seguridad como AES.**

Palabras clave—can, aes, diseño de software, arquitectura, sistema.

INTRODUCCION

Desde septiembre del año 2020 se han presentado un considerable aumento en el número de ataques informáticos dirigidos a sitios web de instituciones financieras que ascienden a pérdidas millonarias[1], un ejemplo es el Banco de México al ser vulneradas sus aplicaciones. Últimamente parte de estos ataques contra la vulnerabilidad de datos se han dirigido a elementos automotrices como lo son las unidades de control electrónico (del inglés *Electronic Control Unit-ECU)*, las cuales operan como computadoras que procesan datos de los sensores del automóvil. Las consecuencias de estos ataques pueden ir desde el robo de datos del vehículo hasta accidentes fatales por fallos de lectura en los datos del sistema de las redes automotrices.

Un vehículo se puede proteger de ataques de prueba y error de combinaciones o conocidos también como de fuerza bruta con diversos algoritmos criptográficos y cada uno de ellos ofrece diferentes ventajas con respecto a otro; comúnmente lo ideal en estos sistemas es que ocupen poco espacio en memoria y que se adapten a cualquier arquitectura del sistema donde se requiera implementar.

Los algoritmos criptográficos surgen de la necesidad de disponer de mecanismos para evitar comprometer y a su vez aumentar la seguridad e integridad de los datos de los dispositivos. Uno de los más populares es el AES del inglés *Advanced Encryption Standard*[2], el cual integra dentro de sus características principales: operaciones con bloques de sustitución (del inglés *sustitution boxes)*, así como una sola operación que funciona tanto para encriptación y desencriptación y la posibilidad de usar o no un vector de inicialización con valores pseudoaleatorios dependiente del modo de cifrado.

El presente artículo tiene como objetivo la implementación y diseño de un sistema de comunicación alámbrico CAN entre dos nodos automotrices, estos nodos adaptan conceptos de una patente donde describe un sistema automotriz con radio digital[3].

Estos a su vez integran cada uno en su diseño un algoritmo de encriptación AES para cifrar información de un vehículo usando archivos digitales como la pista de una canción que se sintonice en la radio del automotor junto con datos de actualización de firmware, creado desde el enfoque de una variante del modelo de diseño de cascada o del inglés *waterfall* conocido como el método V *y* una perspectiva de arquitectura de sistema automotriz.

En la primera sección se describe la propuesta de implementación del sistema, así como la metodología de diseño que se adoptó para la solución del sistema. En la segunda sección se realiza la definición y evaluación de los requerimientos y arquitectura que componen el sistema el cual divide el todo en sus partes. Por su parte en la sección de análisis y requerimientos de software se desglosan las funcionalidades de cada módulo de sistema en un módulo de software y se definen en requerimientos que a su vez conforman una arquitectura de software. Finalmente se presenta el diseño detallado y la implementación en código donde para cada

módulo de software se genera un diagrama de flujo y una interfaz en lenguaje C para integrar las funcionalidades y validar el funcionamiento del sistema.

I. *Implementación del sistema*

Para lograr el objetivo de la implementación del sistema se decidió emplear una plataforma de hardware con periféricos de CAN integrados, los cuales son los kits de evaluación del fabricante Atmel el Atmel SAM V71 Xplained Ultra.

La metodología de diseño para el sistema que se adoptó es el método V. La Fig. 1 muestra la secuencia de etapas para diseño del sistema usando el método V, del cual solo se consideraron para este trabajo el análisis de requerimientos del sistema, análisis y requerimientos de software y la implementación en lenguaje C.

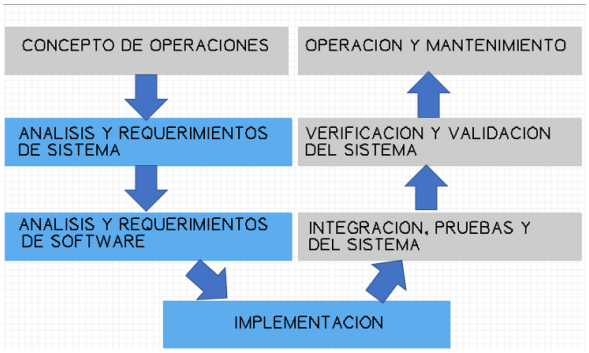

Fig. 1. Flujo de desarrollo de sistema por el método V.

II. *Arquitectura y requerimientos de sistema*

Se dividió la funcionalidad total del sistema en módulos con funcionalidades específicas, para cada módulo de sistema se definieron los requerimientos a satisfacer. La guía del INCOSE[4] fue utilizada para describir los requerimientos de una manera clara, alcanzable y específica, en la Fig. 2 se hace referencia a la arquitectura propuesta del sistema.

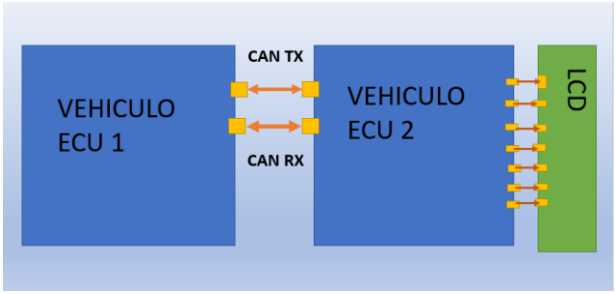

Fig. 2. Arquitectura propuesta del sistema

III. *Requerimientos y arquitectura del software*

Después como base para el diseño de la arquitectura de software que el sistema va a emplear, el siguiente proceso del método V tomando en consideración la arquitectura de sistema es describir y documentar los requerimientos de software necesarios, un ejemplo se muestra en la tabla 1.

TABLA 1. Requerimientos de software para lcd

Identificador de requerimiento	**Descripción de requerimiento de software**
	Capa de abstracción de hardware de entradas y salidas digitales.
	Los propósitos principales de esta capa son: Creación de macros y definiciones de mapeo de pines a salidas digital por medio de un archivo de cabecera. Creación y declaración de estructuras de datos para inicializar pines. Decodificar mensajes de salida a la pantalla de cristal líquido. Definir y crear interfaces para funciones como limpiar pantalla y funciones de impresión de caracteres en la pantalla de cristal líquido.
	definiciones de pines de buses de datos del LCD.
sw_req24	El archivo de cabecera del LCD deberá incluir definiciones de pines de bus de datos d variable del tipo char.
	Manejador de funciones del LCD
sw_req25	El manejador de funciones del LCD deberá incluir una función de reset.
sw_req26	El manejador de funciones del LCD deberá incluir una función de clear.
sw_req27	El manejador de funciones del LCD deberá incluir una función de activación.
sw_req28	El manejador de funciones del LCD deberá tener una función de habilitación de LCD.
sw_req29	El manejador de funciones deberá incluir una función de imprimir.
sw_req30	El manejador de funciones de LCD deberá tener una función de delay.

Luego de diseñar la arquitectura de sistema y tener un esquema de cada una de las partes con sus interfaces se hizo el diseño detallado de la arquitectura de software el cual nos describe como es el flujo de la información entre módulos es decir que entrega o recibe de una capa arriba o abajo y las interfaces para después codificar cada módulo, en la Fig. 3 mostramos un ejemplo del diseño detallado para una función.

Para validar el funcionamiento de la implementación se definió una tabla de mensajes de CAN en el que cada valor en hexadecimal equivale a una de las señales la tabla 2 muestra los valores definidos para cada señal de CAN.

TABLA 2. Mensajes de CAN

Byte7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2
0x6a	0xc1	0xbe	0xe2	0x2e	0x40
Id	marca	modelo	año	número de serie	versión de software

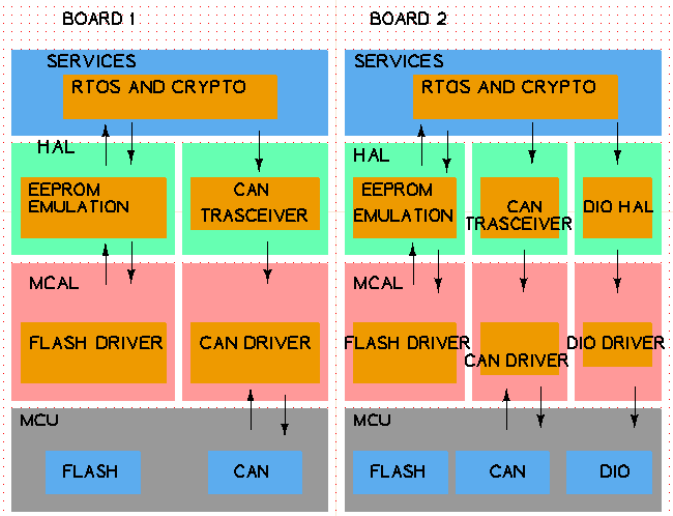

Fig. 3. Arquitectura propuesta del software

IV. *Diseño detallado e implementación*

En el diseño detallado se realizan los diagramas de flujo correspondientes que describen los pasos del algoritmo de cada componente de software que se encuentran en la arquitectura de software. En la Fig. 4 se ilustra un ejemplo para un componente del DIO (del inglés *Digital Input Output*) y función es inicializar el lcd.

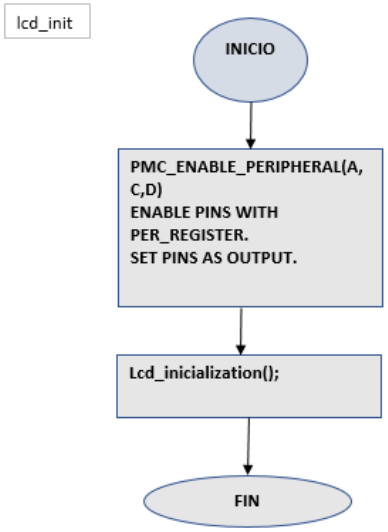

Fig. 4. Diseño detallado de función de inicialización de lcd

Se realizó la correspondiente codificación en lenguaje C de cada módulo de software con el editor de código denominado Microchip Studio. Después, cada componente fue integrado a la arquitectura propuesta del sistema en un solo proyecto en lenguaje C para verificar el correcto funcionamiento del sistema, se bajó el código a las tarjetas de evaluación. Se programaron ambas tarjetas de evaluación con sus respectivos programas, una tarjeta para la encriptación y transmisión de los paquetes de CAN y otra tarjeta para la recepción, desencriptación, decodificación y visualización de los paquetes de mensajes de CAN finalmente se configuró una terminal serial a 115200 baudios para cada tarjeta y se mandaron comandos para iniciar la transferencia de paquetes.

RESULTADOS

Al final se logró visualizar en la pantalla de la segunda unidad de control electrónico el 99.9% de la totalidad de los datos del vehículo desplegados en la pantalla como lo ilustra la Fig. 5, los cuales corresponden a los valores de la tabla de CAN después de realizar la desencriptación de los paquetes; esto se puede ver en la Fig. 6.

Fig. 5. Visualización de datos del vehículo en la pantalla

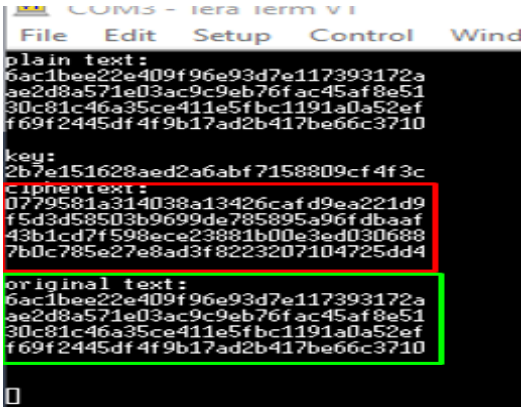

Fig. 5. Cuadro rojo datos cifrados y en verde los datos descifrados

CONCLUSIONES

La implementación propuesta del sistema de comunicación permite ver la ventaja de usar un algoritmo de encriptación como AES en un sistema automotriz por que conserva la integridad de la totalidad de los datos y además es complejo de descifrar. Esto sugiere que podrían crearse en un futuro cercano en la domótica sistemas con características semejantes a las redes CAN con capas de seguridad que incluyan el algoritmo AES.

AGRADECIMIENTOS

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT), a mi asesor el docente H. Rivas por su valiosa opinión técnica, supervisión del análisis y diseño del proyecto y finalmente al Instituto Tecnológico de Estudios Superiores de Occidente por las facilidades brindadas.

REFERENCIAS

[1] [1] «ENISA Threat Landscape 2020: Cyber Attacks Becoming More Sophisticated, Targeted, Widespread and Undetected». https://www.enisa.europa.eu/news/enisa-news/enisa-threat-landscape-2020 (accedido jul. 12, 2021).

[2] [2] Computerphile, AES Explained (Advanced Encryption Standard) - Computerphile. Accedido: jul. 12, 2021. [En línea Video]. Disponible en: https://www.youtube.com/watch?v=O4xNJsjtN6E

[3] [3] H. A. Rivas Silva, González Jímenez,Luis Enrique, Ruiz Cruz,Riemman, y Campos Rodriguez,Rául, «Sistema y método para la reprogramación de dispositivos ecu (unidades electrónicas de control) en vehículos, vía radio digital», WO2017010859A1, jul. 17, 2015

[4] [4] A. Alexandrovich y K. Igorevich, «INCOSE Guide for Writing Requirements. Translation experience, adaptation perspectives», p. 15.