Design and Implementation of a Secure CAN Controller with Message Authentication

[1]Dr.Chethana R, [2]Madhava Theertha R Vedavyas, [3]Harshini P, [4]Mohammed Sameer Khazi, [5]Mythili Pai

[1]Assistant Professor, Department of Electronics and Communication Engineering, S J B Institute of Technology, Bengaluru, India
rchetana@sjbit.edu.in
[2]UG Scholar, Department of Electronics and Communication Engineering, S J B Institute of Technology, Bengaluru, India
madhavatheertha200525@gmail.com
[3]UG Scholar, Department of Electronics and Communication Engineering, S J B Institute of Technology, Bengaluru, India
p.harshini2005@gmail.com
[4]UG Scholar, Department of Electronics and Communication Engineering, S J B Institute of Technology, Bengaluru, India
mohammadsameerkhaji@gmail.com
[5]UG Scholar, Department of Electronics and Communication Engineering, S J B Institute of Technology, Bengaluru, India
paimythili@gmail.com

Abstract - The Controller Area Network (CAN) protocol is used extensively within the automotive industry between electronic control units. Classical CAN does not include message authentication or freshness mechanisms. This paper describes a low resource usage Field Programmable Gate Array (FPGA) implementation of a secure communications controller inspired by Controller Area Network. The design uses an iterative AES-128 core, 64 bit demonstration authentication tag, 15 bit cyclic redundancy check (CRC) and a 32 bit message counter. At the transmitter a custom 264 bit frame is created and at the receiver side the frame is rebuilt. Data is only accepted if it passes the CRC and authentication tag checks and the message counter is larger than the previous accepted counter value. The design was written using Verilog HDL and analysed via simulation and implementation in Xilinx ISE. The design uses 3086 lookup tables and 1031 registers. The minimum period is 8.214ns and the estimated power is 40mW. There were 151 test cases that passed during simulation. The design uses a CAN inspired frame and not a classical CAN frame. The authentication tag uses a demonstration method.

Key Words: Controller Area Network, Message Authentication, AES-128, FPGA, Replay Protection, Verilog HDL

1. INTRODUCTION

Modern vehicles use ECU's that send short messages to control various parts of the vehicle, such as the powertrain, braking, steering and body electronics. Controller Area Network (CAN) buses were developed to allow reliable communication within an electrically noisy environment. This protocol does not offer authentication and thus a malicious party with access to the bus would be able to spoof messages or replay messages. Cyclic Redundancy Check is used to identify any accidental errors but there is no cryptographic security within the protocol.Recent research has investigated using authenticated encryption, light-weight encryption, detection, physical layer detection and message authentication schemes to secure CAN and related networks [1]-[15]. Detection will identify any suspicious messages. By using cryptography the receiver can authenticate a message by verifying the tag of the message using a secret key. Using hardware accelerated techniques can be attractive, if processing occurs at a constant rate and low resource usage is required.This work demonstrates a sender and receiver using Verilog HDL that has been implemented on a Xilinx Spartan-6 XC6SLX9 FPGA. This example contains an example tag, implementing AES-128 encryption, CRC-15 and Counter checks. The main contribution is a resource efficient iterative design that has been simulated. This contains spoofing, corrupted, replayed and truncated frames. It uses a CAN inspired frame format but does not contain all the classical CAN link layer features.

2. RELATED WORK

An authenticated encryption, traceability and replay detection CAN security engine was proposed by Rasheed et al. highlighting the use of lightweight cryptographic techniques to provide multiple security features within an embedded network [1]. CAN intrusion detection using anomaly detection techniques was investigated by Tanksale and others [2]. Roy et al. demonstrated a 2-level authentication scheme using FPGA, AES, and elliptic curve cryptography within an Unmanned Aerial Vehicle [3]. Hediyal et al. proposed SCAN-C as a lightweight cryptographic primitive for CAN communication [4]. Secure time synchronisation of CAN was investigated by Musuroi and Groza highlighting the use of timing within CAN [5].

Other research has focused on additional countermeasures. Wang et al. investigated using physical channel fingerprinting to detect rogue devices on fieldbus networks [6]. Open set intrusion detection was investigated by Zhang et al. on vehicular ad hoc networks [7]. LIDS was proposed by Yu et al. as a light-weight intrusion detection scheme for CAN [8]. Al Isawi et al. used an adaptive neuro fuzzy model to detect attacks against Electric Vehicle CAN [9]. Conditional privacy preserving authentication was proposed by Seifelnasr et al. for cross domain vehicular communication [10].

Recently learning based solutions have been proposed such as a federated two stage intrusion detection system for non-independent identical distributed CAN data [11]. Graph transformer anomaly detection was investigated by Veerati et al. [12]. Wang et al. proposed a dynamic security protocol for intra vehicle CAN [13]. Message authentication using payload compression and tag padding was investigated by Zhang et al. [14]. These papers highlight the balance between cryptographic strength, bus overhead, cost and traffic detection capabilities. This paper focuses on deterministic tag verification using FPGA. The tag generation is intended to be for demonstration purposes only, and does not provide the same security properties of a standardised MAC schema.

3. PROPOSED SYSTEM AND FRAME FORMAT

The system consists of a transmitter, serial CAN link and receiver. Upon request to transmit, the transmitter will increase a 32-bit message counter by one. The AES input is created based on the 128-bit data and address/counter pair. From here the authentication tag can be calculated. A CRC will also be calculated over the data. The frame serializer will then take the fields and serialize them. The receiver will then synchronize using the start bit and sample each field. From here the CRC and authentication tag will be independently calculated. Only if the CRC and authentication tags match, and the received message counter is higher than the last accepted message counter, will the frame be accepted. The message counter will only be updated if the message is accepted.

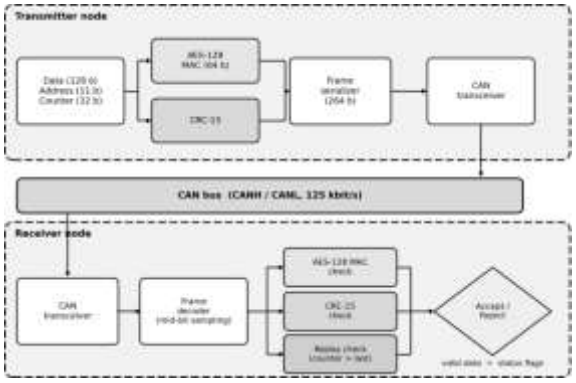


Fig. 1. Block-level secure CAN-inspired transmitter and receiver architecture.

This frame has a total length of 264 bits. This contains all the fields shown in Table 1. The data section is 128-bits which is larger than the 8 bytes of classical CAN. This means this is not a bit-true implementation of ISO 11898. The arbitration, bit-stuffing and acknowledge slot have been removed from this example as it focuses around the authentication path.

[Note: Start (1), Address (11), Control (6), Data (128), Message Counter (32), Authentication Tag (64), CRC (15), End Of Frame (7) = 264 Bits]

4. HARDWARE IMPLEMENTATION

4.1 Iterative AES-128 and tag generation

The initial full unrolled version of AES required approximately between 9500-9800 lookup tables. This was significantly more than the 5720 LUTs available on the XC6SLX9. The architecture uses a single AES Round datapath which is repeatedly reused to calculate round keys during operation. The 11 clock cycle AES requires 220ns (50MHz board clock) to complete operations (excluding any serialisation delays and controller overhead).

The data, address and counter values from the demo tag are all concatenated to form the 128bit AES input. The AES-128 output is then XORed together with the top and bottom 64bits to create a 64bit output. This is performed the same way on the receiver. The result is not a full guarantee due to the custom mapping of the AES. As the top and bottom portions are XORed together, different combinations of the inputs may produce the same AES input. The result should be treated as a demonstration of the architecture only, and not a MAC ready for production.

4.2 CRC and replay protection

The CRC block is run independently to the tag generation. This is so that we can detect accidental data corruption. The tag verification ensures modification of data without a matching secret key based tag is rejected. The receiver will also ensure that the 32-bit counter provided is larger than the last accepted one. If it is duplicate or old then it will be rejected. Only updating the stored counter if all checks pass means that a corrupted frame cannot cause the replay state to advance.

4.3 Verification and tools

The modules were created using the Verilog HDL language and can be tested using the Xilinx ISE 14.7 suite. This included simulation, synthesis, place and route, static timing analysis and XPower estimation. The test plan covered many test cases including AES test vectors, valid frames, corrupted CRC, altered tag, altered address, repeated frames and truncated frames. Top level test benches have been included in this report however the design has not been validated on actual FPGA boards. Power estimation tools have been used instead of measuring actual power on a physical board.

5. RESULTS AND DISCUSSION

All 151 checks passed successfully. Rejection has been tested with frames failing CRC/tag checks, non-increasing counters and truncation. However, the design cannot guarantee that it will be resistant to all methods of attack.

Resource Utilization

The FPGA used as a target for implementation was the Xilinx Spartan-6 XC6SLX9 which had 5720 LUTs available. The transmitter and receiver used a total of 3086 LUTs (~53.9%). 1031 registers were used.

Timing and Power

The minimum period was found to be 8.214ns (~121.7MHz), meeting the clock constraint of 50MHz. It is estimated that 40mW of power will be used.

Verification and Communication

All 151 checks passed during simulation. The custom frame of 264 bits was sent at 125kbit/s taking ~2.112ms.

Overall the resource utilization is within the limits of the device. However, the total utilization is higher than the 60% required for the project. The minimum period is less than the

20ns required for a clock of 50MHz. Due to the iterative nature of the design there is a trade-off between speed and logic. This means it is better suited for the smaller FPGA selected. As the frame is 264 bits and set to be sent at 125kbit/s it will take ~2.112ms to send. This means that the main source of latency will be sending the frame.The 64-bit custom tag does not use AES-CMAC and would require a standard MAC .scheme. The frame has only been authenticated and not encrypted so can be read by any observers on the bus. It has been assumed that there will be one static key. Key distribution, rotation and storage have not been considered. More design would be required to reset and keep the counter running. The custom frame does not include arbitration, bit stuffing or acknowledgements. Results have been found using simulation and tool analysis.

5.1 SIMULATION WAVEFORMS

Figures 2-9 show some typical traces from simulation outputs for the AES core, S-box, CRC block, transmitter, receiver and combined transmit/receiver tests. The signal names and flags shown indicate what activity is occurring within the module and allow monitoring of the signals. These figures are examples of simulation results and are not actual measurements from CAN Bus.



Fig. 2: Simulation waveform of AES-128 encryption core.

The signal waveforms show the clock, reset, start, plaintext, key, ct (ciphertext output) and done signals. The error and check counters can be seen being watched by the test bench.



Fig. 3. AES S-box input / output and testbench signals.

The input and output of the S-Box can be seen by the activity on the a and y buses. The seen, i, errs and checks signals give insight into the test sequence and the respective checking counters.



Fig. 4 CRC-15 Calculation Waveform

This demonstrates how the CRC output is calculated using the data input and enable signal. This also shows the comparison/reference signal. Overall showing how the individual Error Detection block works.



Fig. 5 CAN Style Transmitter Frame Generation Waveform

Shows the transmitter with TX, Transmit, Reset/Button inputs and internal frame elements. The error and check counters can be seen.



Figure 6. Output signals from receiver showing data restored.

Shows the waveforms of the output signals from the receiver showing the restored address, data, counter, authentication and CRC signals. Error and check counts are shown.



Figure 7. Top level signals showing acceptance/rejection of data.

Shows the output signals from the receiver indicating whether the frame was accepted or rejected. Valid_data and frame_toggle signals are also shown.



Figure 8. Output signals showing transmitter activity and frame information.

Shows the activity of the transmitter when the send button was pressed. The frame buffer along with address, data, MAC and CRC signals are displayed. The check and error counts are also being monitored.



Fig. 9. Combined transmit and receive simulation waveform.

This waveform shows the combined signals from the transmitter and receiver, plus the signals; frame_ok, replay_attack, crc_error, mac_error, valid_data and frames_seen. This also shows the test bench exposing tamper and cut stimulus signals to view any faults.

5.2 RTL HIERARCHY AND MODULE INTEGRATION

To aid the separation of transmit and verification logic within the design, a module hierarchy was created. At the highest level of the hardware components, modules were created for transmitting and receiving, the modules share clock and reset signals. The transmitter was wrapped up, so that the transmit request and transmit data signals could be passed in and serial data and transmit signal sent out. Similarly the receiver was wrapped up to output status signals and received data. This allows the logic to be tested in isolation and module level/integrated simulations to be performed.

Looking at the RTL hierarchy (below) we can see a total of 2 modules within the top_can module. The transmitter has a can_tx module to create the frame and the receiver has the validation logic. By separating the design into two halves less coupling will occur between the transmitter and receiver logic. The transmitter has a serializer component to ensure the correct order of fields are present. The receiver will decide if all the fields have passed based on the configuration.

The receiver top level has individual signals to tell us if the frame was valid, repeated, had a failed CRC or failed authentication. During simulation and mapping to LED status, this is helpful to see. It also shows clearly what the logic is doing and how it has decided if it should accept or reject the frame. The frame may have been decoded, however it will only be accepted if it has passed all the checks for its integrity, authentication and freshness.

5.3 CRYPTOGRAPHIC AND ERROR-CHECKING DATA PATH

The data from the data field, address and counter fields are all combined to create a 128bit Input for the AES-128 block. The AES block is implemented iteratively meaning the AES block only uses a single datapath, this is due to the fact that all ten rounds aren't implemented. This calculation is also performed at the receiver using the fixed key to create the 64bit value to compare to the tag. This reduces the amount of logic required as there is no need to perform an AES decryption.

The tag was created by XORing the top half and bottom half of the AES output. This is only done to demonstrate how this would work. It is not an example of an AES-CMAC. The address, counter and payload have been XOR'd. This causes issues when certain combinations of address and counter values are used they will create the same value going into the AES block. It shows how a key based verification scheme would work but this shouldn't be considered a secure protocol. Instead a standard MAC/Authenticated Encryption should be used. It should be clear what fields are being authenticated.

The CRC-15 has been calculated separately. This gives an extra check that there hasn't been any accidental corruption. This is very different to the secret key tag. The CRC is designed to detect multiple random errors. Finally the freshness check keeps track of the last accepted counter value. If the counter value is higher then new frames will be allowed through. Only if all conditions are met will the counter be updated. If the frame has been rejected then the receiver won't progress.

5.4 FPGA SYNTHESIS AND IMPLEMENTATION EVIDENCE

An evaluation of the design was carried out using Xilinx ISE 14.7 for the Spartan-6 XC6SLX9 FPGA. This includes screen shots of the different aspects of the design including RTL schematic, synthesis data, device usage and timing. The top level RTL schematics show the separation of modules and the top level implementation report shows the most useful data in terms of the resources being used in the full top level design.

Figure 10 shows the top level RTL hierarchy, Figure 11 shows the internal top level schematic showing how the transmitter and receiver modules are connected. Figures 12 and 13 show the transmitter and receiver schematics. These schematics are provided by the synthesis tool to give a structural representation of the design and do not replace waveform and hardware testing.

In terms of device resources the final device usage report showed that there were 3086 slice LUTs which is roughly 53.9% of the 5720 available. There were 1031 slice registers used meaning roughly 9.0% of the 11440 available. There were 18 bonded I/O pins used meaning roughly 17.6% of the 102 available. However, the synthesis report in Figure ? also shows that it was estimated that 3188 LUTs would be used in the design during an earlier stage of the synthesis process. This is from a different report so shouldn't be considered as the same reading. The device usage is therefore reported here.

The timing report (figure ?) shows that the minimum period was 7.421ns meaning that during synthesis it was estimated that the design would run at approximately 134.76MHz. The project report says that after placing and routing the minimum period was 8.214ns meaning that the design would run at approximately 121.7MHz. This is consistent as the reports from synthesis and place and route will differ. The result after routing is below the target of 20ns for the clock period to run at 50MHz. However, the design summary from ISE shows that the timing constraints were met. Finally, the power consumption of 40mW was an estimate from the tool.

5.5 RESOURCE AND TIMING SUMMARY

The design and timing information can be seen in Table 2. The iterative design was able to satisfy the LUT constraint as the project only required less than 60% of the available LUTs. This has also drastically reduced the amount of logic needed, by reusing rounds in the design. It is estimated that the original fully unrolled design would have used between 9500-9800 LUTs. Due to this the design was able to fit onto the selected FPGA. However, there has been an increase in the amount of clock cycles required to complete the AES encryption. It takes 11 clock cycles to encrypt the data. This means the encryption will take approx 220ns to complete (Clock Rate = 50MHz). This low latency is not an issue when transferring the custom frame serially.

There are 264 bits of information in the custom frame. With a bit rate of 125kbit/s this should result in a nominal transfer time of 264 / 125,000 = 2.112 ms. This does not take into account any additional delay caused by the design itself. Therefore it can be seen that the encryption time is not an issue when transmitting the frame. This only relates to the simplified frame used for the prototype and cannot be assumed to work with a standard CAN controller.

5.6 VERIFICATION SCOPE AND LIMITATIONS

There were 151 simulation tests completed throughout the project. There are screenshots of waveforms of the AES core, S-box logic, CRC logic, transmitter and receiver. There are also additional signals

on the receiver such as frame_ok, replay_attack, crc_error and mac_error to help see why certain frames have been accepted or rejected. They are example traces taken from the simulator and should be viewed in conjunction with the testbench check and error counts.

During verification there were AES standard test vectors used along with frames with corruption. Address changes, re-play attacks and frame shortening. This was carried out within the simulation environment and through analysis of the FPGA tools. The design has not been tested on real hardware therefore power consumption has been estimated using the tool. The 264-bit frame is not an accurate bit-true classical CAN data frame. It contains a 128-bit data field and there is no bus arbitration, bit stuffing and acknowledge slot. Due to these limitations there is limited conclusions that can be drawn regarding compatibility and performance in a real vehicle.

A more product focussed evaluation should consider testing on real hardware using CAN transceivers. Consideration should be given to testing clock skew and electrical effects. An AES-CMAC or authenticated encryption algorithm should be implemented instead of the custom AES. Freshness state should persist between resets. Secure key management should be defined. To be compatible with classical CAN the frame and transport should be compatible with the relevant CAN standards. These requirements are things to consider for future engineering.

Table 2. Summary of reported implementation metrics.

Metric	Reported value	Interpretation
Slice LUTs (final utilisation)	3,086 / 5,720 (53.9%)	Below the 60% design target
Slice registers	1,031 / 11,440 (9.0%)	Sequential storage usage
Bonded I/O	18 / 102 (17.6%)	Top-level pin usage reported by ISE
Synthesis timing estimate	7.421 ns; 134.76 MHz	Synthesis-stage report
Post-place-and-route period	8.214 ns; 121.7 MHz	As recorded in the project report
Estimated power	40 mW	Tool estimate; not measured on board

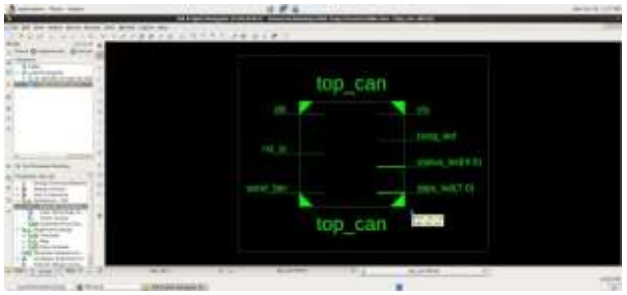


Fig. 10. RTL schematic of the combined top-level CAN controller.

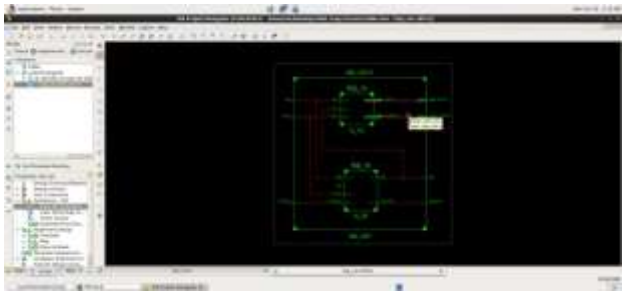


Fig. 11. Top-level RTL hierarchy showing transmitter and receiver integration.

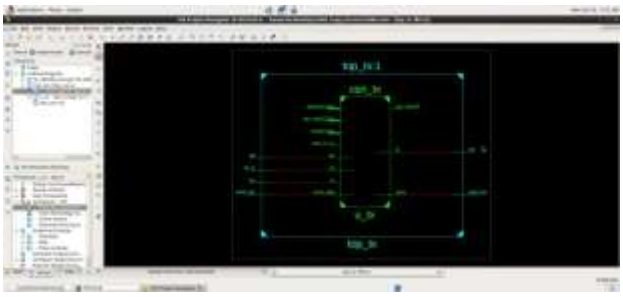


Fig. 12. Internal RTL structure of the transmitter path.

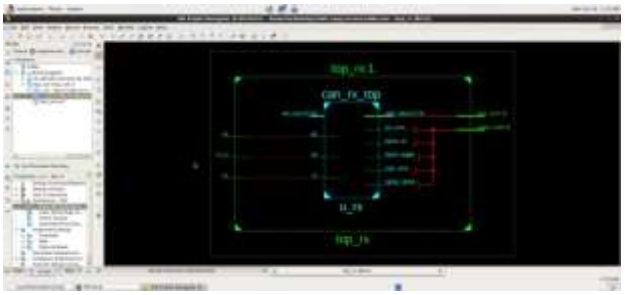


Fig. 13. Internal RTL structure of the receiver and its verification outputs.



Fig. 14. ISE device utilisation summary for the implemented top-level design.



Fig. 15. ISE timing summary reporting the synthesis-stage minimum period.

5.7 ATTACK-CASE INTERPRETATION

There are several checks here. To begin with the receiver will try to decode the frame fields and use the check CRC. A new authentication tag can be created and the freshness of the tags can be checked by comparing the counter to the last accepted value. Using the statuses it becomes easier to know where the issue is. If the CRC fails then something must have gone wrong during the transmission or with the frame fields. If the MAC fails then it means that the value created using the shared key does not match. If there has been a replay then the counter will not be new.

The table below shows what results we would expect at the receiver for the main scenarios mentioned in the project report. This shows the policy we will take to verify the frame but should not be seen as the new probability of an attack being successful. 151 checks have been within the simulation. Screenshots have been taken to show certain signals in the design during certain times.

Table 3. Expected handling of representative frame conditions.

Input condition	Expected receiver behaviour

Unmodified, fresh frame	Accept only when CRC, tag and counter checks pass.
Payload or address alteration	Reject when the recomputed authentication tag differs.
Corrupted field / CRC mismatch	Raise CRC error and do not release payload data.
Previously accepted counter	Raise replay indication and reject the frame.
Truncated or incomplete frame	Do not treat the incomplete field sequence as valid data.
Wrong shared key	Tag comparison should fail, preventing acceptance.

The main security consideration here is that although CRC-15 is great at detecting accidental errors, it is public knowledge therefore can easily be recalculated by an adversary. The tag check introduces a secret dependent action and the counter check introduces replay resistance given the assumption that the counter is shared. These are just complementary mechanisms and not encryption. As the message contents are still visible on the bus they are not encrypted. The XOR based reduction of the tag introduces some structural issues. This implementation would be a good example of an educational hardware prototype.

5.8 INTERPRETATION OF TOOL REPORTS

These screenshots show us what stage the reports are from. The design summary shows that the device usage after the design was complete was 3086LUTs and 1031 registers. Looking at the synthesis report we can see that at an early stage the number of LUTs in use was 3188. The minimum period value of 7.421ns was obtained from the timing summary. This shows that this was an estimated value given by synthesis. The value of 8.214ns was obtained from the project report meaning that this was a post place and route value. By including both values and stating where they came from we can see that they are not the same. The timing report has notes explaining that the timing values from synthesis are estimated whilst the post place and route traces will give a more accurate indication of the timing.

The design summary also shows that the design has been routed and all timing constraints have been achieved. We can therefore claim that the logic implemented meets the clock speed of 50Mhz which was set during configuration in the ISE flow. We can however not claim that signal integrity has been met on a real CAN bus, that the circuit will operate correctly between two boards or that it will operate on a vehicle level. This would require obtaining values from the hardware and testing to see whether they meet protocol standards. This wasn't within the scope of the project.



Fig. 16. ISE design summary showing implementation status and device resource totals.



Fig. 17. Synthesis timing report and its warning that the values are estimates before place-and-route.

6. CONCLUSION AND FUTURE WORK

This paper implements a resource efficient FPGA design for a secure communications controller inspired by the CAN protocol. At the transmitter stage, a counter, demonstration 64-bit tag based on AES-128 and a CRC-15 checksum are appended to a unique 264-bit frame. This is checked at the receiver stage using the tag, checksum and fresh counter. In simulation there were 151 successful checks. The overall design uses 3086 LUTs and 1031 registers. It has a minimum period time of 8.214ns and was implemented on a Spartan-6 XC6SLX9 device. An estimated power usage of 40mW.

The next step would be to implement the tag with a standardised method such as AES-CMAC, or an authenticated encryption scheme. Distinct fields would also need to be implemented using a clear cryptographic input encoding. Secure key management and counter storage would need to be implemented. This controller would need to be tested on actual FPGA boards with CAN transceivers. Additional features would need to be added such as arbitration, bit stuffing, acknowledgement and a payload format suitable for CAN or CAN FD. Only then could this prototype be considered for use within a vehicle network.

ACKNOWLEDGEMENT

We would like to thank our Project Guide Dr Chetana R Assistant Professor, Department of Electronics and Communication Engineering, SJB Institute of Technology, Bengaluru for guiding and encouraging us to suggest the ideas for our project. Without their help and guidance we would have not completed our project.

We would like to thank our Department of Electronics and Communication Engineering, SJB Institute of Technology, Bengaluru for providing us with the necessary facilities and lab equipment for designing and implementing our proposed FPGA based secure CAN communications.

We would like to thank our Head of the Department, Faculty Members and Staff Members for their help.

Lastly we would like to thank our friends and family members for constantly encouraging us.

REFERENCES

[1] A. Rasheed, M. Baza, M. M. Badr, H. Alshahrani and K.-K.R. Choo, "Efficient Crypto Engine for Authenticated Encryption, Data Traceability and Replay Attack Detection over CAN Bus Network", IEEE Transactions on Network Science and Engineering, vol. 11, no. 1, pp. 1008-1025, 2024. https://doi.org/10.1109/TNSE.2023.3312545

[2] V. Tanksale, "Intrusion detection system for controller area network,", Cybersecurity, vol. 7, no. 4, 2024. https://doi.org/10.1186/s42400-023-00195-4

[3] K.S. Roy, et al., "FPGA based Dual Layer Authentication Scheme using AES and ECC Algorithms for Unmanned Aerial Vehicles", EURASIP Journal on Wireless Communications and Networking, vol. 2024, no. 91, 2024. https://doi.org/10.1186/s13638-024-02419-8

[4] N. Hediyal, B.P. Divakar and K. Narayanaswamy, "SCAN-C: Light weight cryptographic algorithm for securing CAN communications in modern vehicles", Cybersecurity, vol. 8, no. 49, 2025. https://doi.org/10.1186/s42400-024-00291-z

[5] A. Musuroi and B. Groza, "Secure Time Synchronization with Submicrosecond Accuracy in Controller Area Networks", IEEE Transactions on Industrial Informatics, vol. 21, no. 6, pp. 4522-4532, 2025. https://doi.org/10.1109/TII.2025.3541719

[6] X. Wang, et al., "Intrusion Device Detection Using Channel State Group Fingerprint of Fieldbus Networks", IEEE Transactions on Information Forensics and Security, vol. 19, pp. 4012-4027, 2024. https://doi.org/10.1109/TIFS.2024.3374596

[7] J. Zhang, Z. Pan, J. Cui, H. Zhong, J. Li and D. He, "Towards Open Set Intrusion Detection in VANETs: An Efficient Meta Recognition Approach", IEEE Transactions on Network Science and Engineering, vol. 11, no. 6, 2024. https://doi.org/10.1109/TNSE.2024.3459087

[8] Z. Yu et al., "LIDS: A Lightweight Intrusion Detection System for Controller Area Network,", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 44, no. 9, pp. 3303-3312, 2025. [DOI: 10.1109/TCAD.2025.3543431]

[9] O.A. Al Isawi, K.A. Al Jaafari, A.S. Al-Sumaiti, "Electric Vehicles CAN Bus Cyber Attacks Detection Using Adaptive Neuro Fuzzy Inference System", IEEE Access, vol. 13, pp. 90862-90874, 2025. [DOI: 10.1109/ACCESS.2025.3550970]

[10] M. Seifelnasr, R. AlTawy, and A.M. Youssef, "A Conditional Privacy-Preserving Protocol for Cross-Domain Communications in VANET", IEEE Transactions on Intelligent Transportation Systems, vol. 26, no. 4, 2025. [DOI: 10.1109/TITS.2025.3527409]

[11] Y. Zhang et al., "Federated two-stage transformer based network for intrusion detection in non-IID data of controller area networks", Cybersecurity, vol. 8, article no. 29, 2025. [DOI: 10.1186/s42400-024-00329-2]

[12] X. Zhou et al., "CGTS: graph transformer based anomaly detection in controller area networks", Cybersecurity, vol. 8, article no. 62, 2025. [DOI: 10.1186/s42400-025-00365-6]

[13] R. Veerati, K. Dorthi, K. Siripuri, S. Kuntamalla, R.K. Kotha, "Enhanced advanced encryption standard algorithm for secure communication with low latency using Verilog hardware description language", Discover Computing, 2026. [DOI: 10.1007/s10791-026-10032-0]

[14] Y. Wang, Y. Xu, Z. Liu, S. Liu, Y. Wu, "Research on Lightweight Dynamic Security Protocol for Intelligent In-Vehicle CAN Bus", Sensors, vol. 25, no. 11, article no. 3380, 2025. [DOI: 10.3390/s25113380]

[15] G. Zhang, J. Shen, J. Li, W. Qin, Y. Li, "Payload Processor: Message authentication for in-vehicle CAN bus using data compression and tag filling", Computer Networks, vol. 259, article no. 111061, 2025. [DOI: 10.1016/j.comnet.2025.111061]