Journal of Circuits, Systems, and Computers
Vol. 27, No. 5 (2018) 1850082 (17 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0218126618500822

MACsec-Based Security for Automotive Ethernet Backbones*

Berardino Carnevale[†,§], Luca Fanucci[†,¶],
Samson Bisase[‡,‖] and Harman Hunjan[‡,**]

[†] *Department of Information Engineering,*
University of Pisa, Via Caruso 16, Pisa,
I-56125, Italy

[‡] *Renesas Electronics Europe Ltd,*
Dukes Meadow, Milliboard Rd,
Bourne End, Buckinghamshire,
SL8 5FH, United Kingdom
[§] *berardino.carnevale@for.unipi.it*
[¶] *luca.fanucci@for.unipi.it*
[‖] *samson.bisase@renesa.com*
[**] *harman.hunjan@renesa.com*

Received 26 January 2017
Accepted 28 August 2017
Published 12 October 2017

The increasing complexity of automotive electronics and the communication of cars with the external environment have led to extensive security issues. The car industry is moving towards the use of Ethernet backbones to improve the performance and reduce the complexity of in-car networks. In this paper, we propose a security solution for automotive Ethernet-based communications. We designed a hardware Media Access Control (MAC) layer based on the MAC Security Standard (MACsec) that considers the specific constraints of the automotive world in terms of latency, throughput and area. From a security point of view, our solution guarantees the confidentiality, integrity and authenticity of data. Furthermore, the system can be configured before synthesis to meet the security needs of the context in which the Ethernet communication is used. We synthesized our architecture on a low-power 28 nm standard-cell CMOS technology, which is appropriate for automotive microcontrollers. The results show that our implementation is suitable for 100 Mbps, 1 Gbps and 10 Gbps Ethernet speeds introducing less than 350 ns of latency. The size of the circuit varies from 285 to 622 kgates depending on the required level of security and the required features.

Keywords: Ethernet; security; automotive; VLSI; network.

*This paper was recommended by Regional Editor Piero Malcovati.
[§] Corresponding author.

1. Introduction

The continuous addition of features in modern cars has increased the complexity of their internal networks and their connectivity capabilities with the external environment, which has led to security issues similar to those faced by the Information Technology (IT) world. At the top of the vehicle market, there can be up to 100 Electronic Control Units (ECUs) in a car that are all interconnected. ECUs are also often arranged into protocol sub-domains such as Controller Area Network (CAN), Local Interconnect Network (LIN) and FlexRay, and each inter-domain communication often must travel through gateways and bridges.[1] In addition, the car-to-car and car-to-infrastructure exchange of data will dramatically increase over the next few years.[2] This wide range of networks, protocols, and data could provide potential attackers with several entry points to maliciously interact with the car, thereby jeopardizing the health or even the life of passengers. Indeed, the data flowing over a car's internal network are often related to safety features, and the exploitation of security issues has even more dramatic consequences than IT systems. A typical example could be the "brake by wire" approach commonly used in modern cars in which brakes are controlled through electrical means. In this context, a "wire" can easily be a communication network carrying the brake information/request. An attacker could then interact with it causing unwanted brakes or, worse, deactivating the required effect causing potential repercussions on human lives.

To improve the performance and reduce the complexity of in-car networks, the automotive industry is moving towards Ethernet solutions for in-car backbones.[3] Such an approach would guarantee the optimization of the network because different sub-domains would communicate by using only Ethernet gateways without involving specific domain-to-domain connections, which increase the complexity of the network. The high performance of the Ethernet protocol would also protect against any reduction in safety caused by an insufficient level of performance. It is therefore clear that automotive security should also include Ethernet backbones because no section of the network should be unprotected.

To work well for the automotive industry, the Ethernet protocol requires security against any undesired disclosure of information or unauthorized interaction with the data flow. The MACsec[4] defines the countermeasures required for the MAC layer of the Ethernet Protocol to obtain an appropriate level of protection in terms of confidentiality, integrity and authenticity. The standard explains how the Ethernet frame should be encapsulated in a MACsec frame to fill the gap in terms of security. The security countermeasures are based on the usage of symmetric encryption algorithms.

In this paper, we present a MACsec-compliant Hardware (HW) implementation that is specifically targeted at full-duplex Ethernet for automotive applications. A Software (SW) solution alone would not meet the specific performance requirements of the automotive world, thus leading to unacceptable safety consequences.[5]

We therefore tailored our solution to the typical requirements of network controllers for cars such as low latency and high throughput supporting 100 Mbps, 1 Gbps and 10 Gbps speeds. We also designed a solution aimed at minimizing the area and leading to an easy integration of the system into existing or future car network controllers.

The major contributions of this work are the following:

- An approach to design a complete system that guarantees confidentiality, integrity and authenticity of data in Ethernet-based car networks
- The selection of the appropriate subsystems of the implementation to meet the strict requirements of the automotive world
- A comprehensive evaluation of the trade-off between security and performance when designing an automotive version of the MACsec
- A configurable solution whose level of security can be modified in order to meet the security needs of the target system

The remainder of the paper is organized as follows: Section 2 illustrates the related work, Sec. 3 provides an overview of the reference MACsec standard, Sec. 4 describes the proposed MACsec-based automotive architecture, Sec. 5 analyzes the results of the implementation, and finally the conclusions are presented in Sec. 6.

2. Related Work

In the last years, the problem of automotive security has been the subject of an increasing number of studies from both industry and academia.

The most important automotive suppliers have shown a deep interest in the development of tailored solutions to improve the security features and find innovative countermeasures.[7,6] Unlike automotive functional safety, there is no commonly recognized standard addressing implementations, techniques and approaches that should be used while designing secure cars. Therefore, the lack of common guidance has led to custom solution that are often difficult to evaluate in terms of security level. Recently, several consortia between industry and academia have attempted to fill this gap and proposed a common approach to these needs. The most important projects are the E-safety Vehicle Intrusion proTected Applications (EVITA),[8] Car-2-Car,[9] the PREparing SEcuRe V2X communication systEms (PRESERVE).[10] They focus on the general approach of security for vehicles and communication among cars and the exchange of information between automobiles and the external environment.

On the academic side, several researchers have focused on this innovative topic by means of a "destructive" approach[11] (i.e., finding and exploiting security vulnerabilities) or "constructive" research on tailored solutions.[12] One of the first works on this topic can be found in Ref. 13, where the authors show the feasibility of an attack on a modern car at both theoretical and practical levels. The authors exploited the vulnerability of the car communication protocols in their original version as an entry point of car hacking. The lack of a uniform direction and a standard in the field of car security has led to other work to find a systematic way of defining the attack vectors and the possible threats of a car[14] as a driver for a secure solution. Furthermore, other researchers have attempted to summarize all automotive security issues in a comprehensive way to obtain a complete picture of the state of the art.[15]

This work pursues the design of secure of in-car hardware by addressing the threats derived from an inappropriate security level in communication channels. Several works have shown the vulnerabilities of the bus protocols used in the automotive world and attempted to find suitable countermeasures. In particular, the main target of these studies has been the CAN protocol[16] because it is the most widespread network technology in automobiles. Nevertheless, some researchers have proposed secure implementations of the same standard based on higher layers of the communication stack.[17] Furthermore, other work described the security issues and solutions for less common but equally important technologies such as FlexRay[18] and LIN[19]

The Ethernet protocol is a relatively new solution for in-car buses, so as yet there are no comprehensive studies of such technology in the automotive environment. Nevertheless, the low level of security given by the Ethernet standard in its original version is commonly known and has been highlighted in several research activities.[20] For this reason, some works proposed custom solutions to fill this gap in terms of reliability.[17] Incidentally, an approach tailored to a specific solution might not satisfy the needs of car manufacturers. Indeed, this industrial area usually prefers to establish its plans on a standard, such as ISO 26262[21] in the safety field. However, some works have attempted to implement the MACsec standard for Ethernet Passive Optical Networks (EPONs), but only a few of them have contextualized it in the automotive world.[22] Nonetheless, none of the cited studies carefully evaluate the trade-off of the car industry and the HW–SW partitioning that represents a pivotal step when designing secure solutions.

3. MACsec-Based Ethernet Security

MACsec guarantees confidentiality, integrity and authenticity of data by encapsulating an Ethernet frame into a MACsec-compliant one. The Advanced Encryption Standard Galois Counter Mode (AES-GCM),[23] an authentication-improved version of the Advanced Encryption Standard (AES), is the core algorithm used by MACsec to perform encryption and decryption of data in Transmission (TX) and Reception (RX), respectively. It also guarantees integrity and can use both 128-bit and 256-bit symmetric keys. The authenticity of network peers is achieved based on the integration of MACsec with the IEEE 802.1X-2010 standard,[24] which describes how network nodes are discovered, authenticated and how the cryptographic parameters (e.g., encryption keys) are assigned to each component of the communication channel. IEEE 802.1X-2010 defines a Secure Association (SAS) to connect network peers that want to use the MACsec features and a Secure Channel (SC) to allow point-to-point communication. MACsec also performs replay protection checks, thus preventing the retransmission of frames. Each MACsec entity collects and stores several statistical aspects of the traffic over the network to monitor the data flow and detect potential misbehavior.

A MACsec frame is composed of the Destination Address (DA), the Source Address (SA), the Secure TAG (SECTag), the Secure Data and the Integrity Check Value (ICV). Figure 1 depicts the MACsec encapsulation and decapsulation performed in TX and RX. As shown, in TX, an original Ethernet frame is divided into two main sections: the MAC addresses (DA and SA) and the User Data (i.e., the remaining part of the frame). The MAC addresses are not modified in the MACsec frame, the User Data section is optionally encrypted into Secure Data, and finally the ICV is computed and appended to the end of the frame in TX. The ICV covers the integrity of MAC addresses and the Secure Data. The SECTag, which provides all security information (e.g., SAS, SC, association numbers, key length and replay protection data), is inserted between the MAC addresses and the Secure Data. The length of the ICV is always 16 bytes. In RX, the system decrypts the Secure Data in the User Data and checks that the "received ICV" matches the "computed ICV". The ICV and the SECTag are then stripped, and only the result of its check is forwarded. The original frame is then de-capsulated from the MACsec frame. Encryption/decryption and ICV computation are executed as specified by the AES-GCM. The Packet Number (PN) within the SECTag is for replay protection and uniquely identifies the frame. It is always a 32-bit section, but the TX and RX sides can also work with 64-bit PN internally extending it. We will call the 32-bit PN Normal Packet Numbering (NPN) and the 64-bit PN Extended Packet Numbering (XPN). The length of the SECTag can be 8 or 16 bytes depending on whether an 8-byte Secure Channel Identifier (SCI) is included.

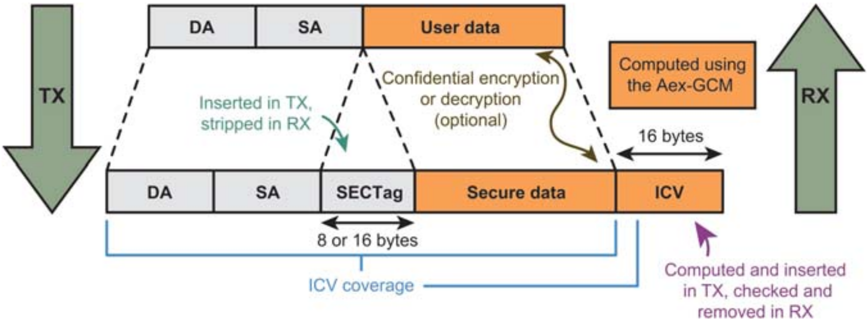

Fig. 1. MACsec frame encapsulation and decapsulation.

4. MACsec Design for Automotive Applications

Our MACsec implementation for in-car networks consists of the TX core, the RX core, a Secure Register Set (SRS), a Secure Hardware Extension (SHE) and a synchronization interface. Figure 2 depicts the high-level view of the proposed architecture. The system is designed to be integrated between the original MAC and the Logic Link Control (LLC) of the Open System Interconnect model. As highlighted in Fig. 2, the RX and TX cores work in the HW clock domain, whereas the SRS and the SHE work in the SW clock domain. The synchronization interface allows for reliable Clock Domain Crossing (CDC) when the two domains want to communicate using typical CDC techniques (i.e., double flip-flops, recirculation multiplexers and toggle synchronizers). In the following sections, we identify each Ethernet speed with its MAC-Physical (PHY) commercial interfaces: Media-Independent Interface (MII), Gigabit Media-Independent Interface (GMII) and 10-Gigabit Media-Independent Interface (XGMII). We use this equivalence because in our implementation, we expect the MACsec layer to work at the same frequency as the PHY and the MAC. We therefore use the following equivalences and the following clock frequencies for each configuration:

- 100 Mbps ↔ MII, $f = 25\,\text{MHz}$
- 1 Gbps ↔ GMII, $f = 125\,\text{MHz}$
- 10 Gbps ↔ XGMII, $f = 156.25\,\text{MHz}$ or $f = 312.5\,\text{MHz}$.

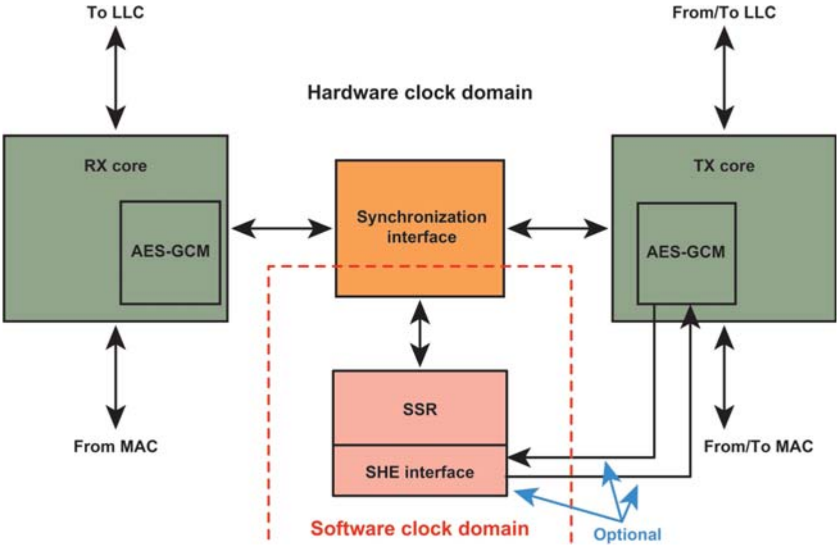

Fig. 2. High-level architecture of our MACsec implementation.

A detailed description of the PHY-Media MAC commercial interfaces is beyond the scope of this work. Indeed, in our design, we require to be compliant with the working frequency and throughput only.

In the following subsections, the five mentioned sub-components of the designed system are described. The focus of this work was the complete design of the sub-components with the only exception of the AES-GCM. The design of the AES-GCM has uniquely involved the choice of the state-of-the-art solutions meeting the automotive requirements and trade-offs.

4.1. *The secure register set*

The SRS has an Advanced Peripheral Bus (APB) interface to give it as much flexibility as possible and provide easy integration with different types of SW modules. The SRS is sub-divided into two sections: one for statistics and one for cryptographic keys/replay protection capabilities.

The MACsec updates the statistics section with real-time information on the traffic and can be reset/preset to specific values by the SW. The number of statistical registers can easily reach several Kbytes in size and thus occupies a large area. We thus implemented a flexible approach that instantiates only a reduced sub-set of registers with a synthesis parameter. Indeed, depending on the context in which the MACsec is integrated, the user can choose to instantiate a light version with only high-importance security information (i.e., number of corrupted frames) or a full version also including low-importance security information (i.e., delayed packets). For the sake of simplicity, in the following sections, the light version will be called Half statistic (HS) mode, and the full version is called Full statistic (FS) mode. The MACsec standard requires the FS implementation; but in some low-risk environments, the choice of HS mode solutions may be appropriate.

The keys and PN must be managed carefully due to the security importance of such information. Indeed, to update the keys, a SHE-compliant approach was implemented. The SHE is the automotive implementation of secure HW–SW interaction proposed by the Hersteller Initiative Software (HIS)[25] consortium, a group of the car manufacturers Audi, BMW, Diamler AG, Porsche and Volkswagen. The SHE requires a Secure Boot that authenticates the SW and specific integrity/authenticity checks on the HW–SW interaction. The Advanced Encryption Standard-Cipher-based Message Authentication Code (AES-CMAC)[26] algorithm is used for integrity checks. The AES block within the TX core is multiplexed with the SHE extension to prevent logic duplication. The decision to multiplex the AES block of the TX instead of the RX is because the TX flow can be stopped if the AES core is busy while the RX receives data directly from the physical bus whose flow cannot be interrupted. Multiplexing the AES logic with the RX path would have required large buffers with high area occupation. Nonetheless, the user can also choose to avoid such a muxed approach when increasing the security level of the system. Indeed, despite the increased area of the system, a non-muxed approach guarantees a clear division between HW and SW and reduces the HW entry points.

4.2. *The AES-GCM core*

The AES-GCM core is aimed to be the best trade-off for automotive applications by means of a low area/latency and high throughput solution. There are some choices to be made during the design of the AES-GCM related to: the byte substitution step of the AES, the Galois Field Multiplier for the ICV calculation, and the number of AES rounds implemented in the HW.

The byte substitution mathematically represent the multiplicative inversion and affine transformation in the Galois Field (GF). There are two ways to implement this: as a Lookup Table (LUT).[27] or as a combinational network[28] The former approach is often used in Field-Programmable Gate Array (FPGA) and SW implementations due to the dedicated resources to store such an amount of data. The combinational network approach is the best solution for Application-specific Integrated Circuit (ASIC) implementations and leads to reduced area occupation. We therefore chose the combinational network approach due to the need to integrate the system in a small automotive micro-controller.

The GF polynomial multiplier works on each 128-bit block in which the frame is subdivided to compute the final ICV. To reduce the area of such an operation, we implemented the Karatsuba–Offman multiplier,[29] which reduces the area required by multiplication by subdividing it into smaller sub-multiplications and merging the intermediate results. Pipeline stages between intermediate steps reduce the critical path of the logic, although this is paid in terms of area and latency. We implemented both non-pipelined and pipelined versions. The former is synthesized when only MII and GMII are required, and the pipelined version is for when the XGMII mode is included; this choice can also be made by a synthesis parameter.

The final choice was related to the number of physically implemented AES rounds ($\mathrm{AES}_{\mathrm{pnrd}}$), and in this case our focus was on the reduction in area and latency. The number of rounds of the AES algorithm ($\mathrm{AES}_{\mathrm{anrd}}$) depends on the key size: 10 for 128-bit and 14 for 256-bit keys. We designed an outer-round pipelined AES core, which means that there is a single pipeline stage after each round and that a single round is executed during a clock cycle. This leads to short critical paths and relatively low latency (i.e., equal to the number of rounds). $\mathrm{AES}_{\mathrm{pnrd}}$ is less than or equal to $\mathrm{AES}_{\mathrm{anrd}}$ and depends on the throughput. This leads to a smaller area than implementations using higher $\mathrm{AES}_{\mathrm{pnrd}}$. Therefore, in our solution each round has a finite number of executions before the data advance to the next stage of the pipeline. $\mathrm{AES}_{\mathrm{pnrd}}$ depends on the throughput as follows:

$$\mathrm{AES}_{\mathrm{pnrd}} = \left\lceil \frac{\mathrm{AES}_{\mathrm{anrd}} \cdot B_C}{128} \right\rceil . \tag{1}$$

B_C is the throughput expressed as:

$$B_C = \frac{\text{input bit}}{\text{clock cycle}}\,, \tag{2}$$

which is the size of the AES input bus.

This means that each physical round executes at most the following AES_{anrd}:

$$\left\lceil \frac{\text{AES}_{\text{anrd}}}{\text{AES}_{\text{pnrd}}} \right\rceil . \tag{3}$$

Therefore, considering that our implementation should support the worst case of a 256-bit key and 64-bit per clock cycle (XGMII 312.5 MHz), AES_{pnrd} is 7. In any case, to provide enough flexibility, our system can be synthesized into different versions depending on whether the XGMII mode is required. If not, the system can keep the MII and GMII throughput using only one pipeline stage. Therefore, there are no area overheads, and only the minimum amount of logic is used. To increase the flexibility of the system, a synthesis parameter enables only the AES to be implemented using 128-bit keys or both the 128- and 256-bit keys. MACsec includes both possibilities, but in specific environments, the trade-off between security and resource usage could lead to an only 128-bit keys configuration.

4.3. *The TX and RX cores*

Figure 3 shows the simplified architecture of the TX and RX cores. They share similar basic blocks and for the sake of simplicity, Fig. 3 highlights the paths belonging only to the RX in red and in blue, the path relative to the TX. The input frames arrive at the output in words whose size depends on the Ethernet mode and is

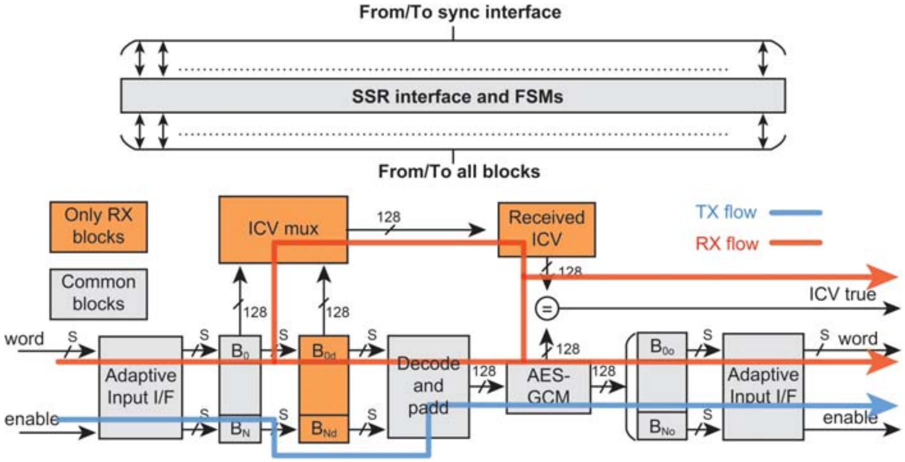

Fig. 3. High-level architecture of the TX and RX cores of our MACsec implementation.

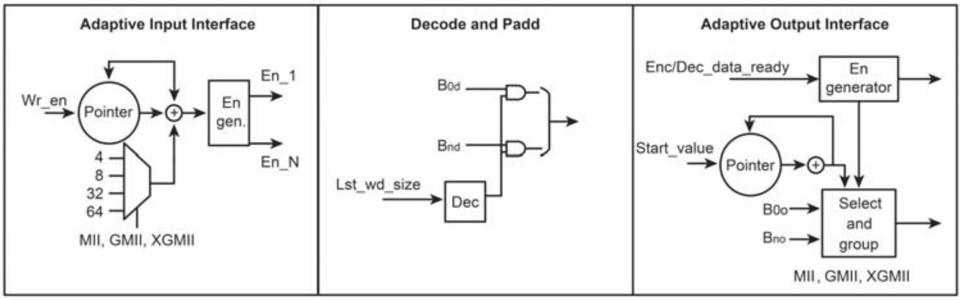

Fig. 4. Architectural details of the Adaptive Input Interface, Decode and Pad and the Adaptive Output Interface.

4, 8, 32 and 64 bits for MII, GMII, XGMII 312.5 MHz and XGMII 156.25 MHz, respectively. For each valid word, there is an enable signal to indicate that this word is valid. Figure 4 shows some details of three blocks of the system: the Adaptive Input Interface, the Decode and Pad and the Adaptive Output Interface.

The Adaptive Input Interface takes the input words and forwards them to the first 128-bit buffer. Therefore, depending on the Ethernet mode, it increments a buffer pointer with a different value by enabling the correct buffer location. The pointer wraps when it reaches the maximum value (i.e., 128). The incremental value of the pointer is exactly BC.

The first buffer receives the arranged words and store them until the pointer in the Adaptive Input I/F wraps. When this occurs, the content moves into the second buffer or into the Decode and Pad block depending on whether we are in RXor TX, respectively. In RX, the content of the second buffer is passed to Decode and Pad only when a second wrap occurs (meaning that the following 128-bit is complete) or when the end of the frame has been reached. In the latter case, the ICV is split between the first and second buffers, and the "ICV mux" block rearranges and stores it in the "Received ICV" block.

The Decode and Pad block receives the value of the last pointer, and uses a decoder to transform this value into a bit mask. The generated mask is used to pad the last 128-bit block if its content is smaller than 128 bits (i.e., the last 128-bit block was not complete), as required by AES-GCM. The sections of the frame that do not need encryption (i.e., SA, DA, SECTag) are simply forwarded by the AES-GCM core without any modification.

Finally, the block generated by AES-GCM is sent to the output using the Adaptive Output Interface, which behaves similarly to that of the input. Indeed, the size of the word is derived from the Ethernet mode, and a pointer selects the starting address of the word in the buffer. A final STATUS word delivers some additional information at the end of the frame in RX. This is the result of the comparison between the received and computed ICV, and the PN checks for replay protection.

A set of Finite State Machines (FSMs) control the data flow, and the SRS interface takes all cryptographic keys, statistic counters, PN values and configuration signals exchanged between the SRS and the two data paths.

4.4. *Security considerations*

The usage of MACsec for automotive applications cannot guarantee protection against intrusions of potential malicious attackers owing to the nature of the standard itself. This means that such an entity can physically inject the deleterious message into the network because the standard does not define any type of prevention. Other types of countermeasures should be delegated to attack prevention, integrating them closer to the physical layer. These can be physical protections of the cables or identification of unexpected electrical levels.

On the other side, the MACsec compliant node is capable of identifying whether a message is not allowed to flow over the network by the ICV. Once MACSec has identified the misbehavior, the frame can be discarded and therefore never transmitted through the communication layers or flagged with such information. If the frame is blocked and discarded, the other layers are completely unaware that this frame was received. Conversely, if the frame is simply flagged as erroneous, the higher layers should manage this frame in a careful fashion. For the sake of clarity, we can provide a typical example involving a safety-related ECU such as the brake control unit. If such a unit receives a corrupted brake request, its MACsec immediately identifies the issue. At this point, if the MACsec discards the frame, it should signal to the system that an important feature is not working properly and that an attack causing severe health consequences might be possible. Indeed, no entity other than the MACsec can identify the issue. If the frame is passed to the higher layers anyway, the designer can choose which entity should alert the system. Once the system receives the warning, the car should be led to a safe state (e.g., a gradual brake with advice to the passengers), blocking the attacker from further interacting with the safety-system. Obviously, another type of attack unrelated to safety features (e.g., multimedia, window control) can be managed with different types of responses requiring, in some cases, human interaction. Figure 5 shows an example in which the received ECUs identifies a message failing the integrity check. In the same example, the ECUs flags the issue to the Secure Control System, which is responsible for switching the communication from the main channel to the backup channel.

The second main feature of MACsec is the optional confidentiality of messages. It is clear that in a modern car, not all information is confidential, and the disclosure of some data has no dramatic consequences. The information to activate the brake or to open the windows is usually available to the public. On the other side, there might be a commercial algorithm that the car manufacturer might choose to keep confidential. Typical examples might be any specific control system at the top of the market (e.g., car remote controls or self-adaptive lights). Furthermore, manufacturers can decide

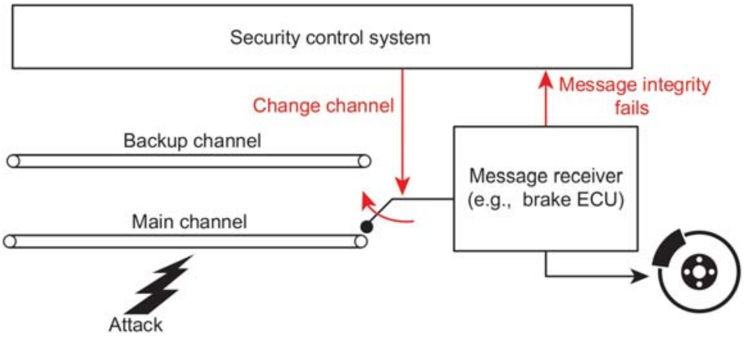

Fig. 5. Example of high-level response of the brake system in case of integrity check failure of the received message. The brake ECU fags the system that a message received from the main channel did not pass the ICV check. Therefore, the system switches from the main to the backup channel to protect the brake unit against unwanted messages.

to use custom algorithms to control simple systems to increase the security level hiding even such features.

Finally, the protection against replay attacks guarantees that any chance of reproducing a message previously flown over the network is identified. This type of misbehavior can sometimes be more important than wrong ICV identification. Indeed, a wrong ICV might be related to errors on the bus but, if the system works correctly, the same message is never re-transmitted. Having two of the same message can be considered as an attack with a higher probability, and the system response can be dimensioned as a consequence.

Therefore, the analysis of the vulnerabilities is an important step that should come before the implementation of countermeasures to avoid under/over-estimation of threats. MACsec, as every specific solution for car security, can be considered one of the candidates for in-car network security, but, as explained in this section, its integration is strongly related to the system design.

5. Results

We implemented our solution on a CMOS technology, varying the parameters in terms of throughput, key length, replay protection type, latency, area and security level.

In terms of throughput, our system supports our three targets: MII, GMII and XGMII 156.25/312.5 MHz which correspond to 100 Mbps, 1 Gbps and 10 Gbps throughputs. It is therefore suitable for a wide range of Ethernet backbones that will be used in future cars, including high-performance applications such as multimedia and safety.

Figure 6 shows the latency of the TX and RX paths for four different speed configurations. In both RX and TX, the latency increases with the level of security

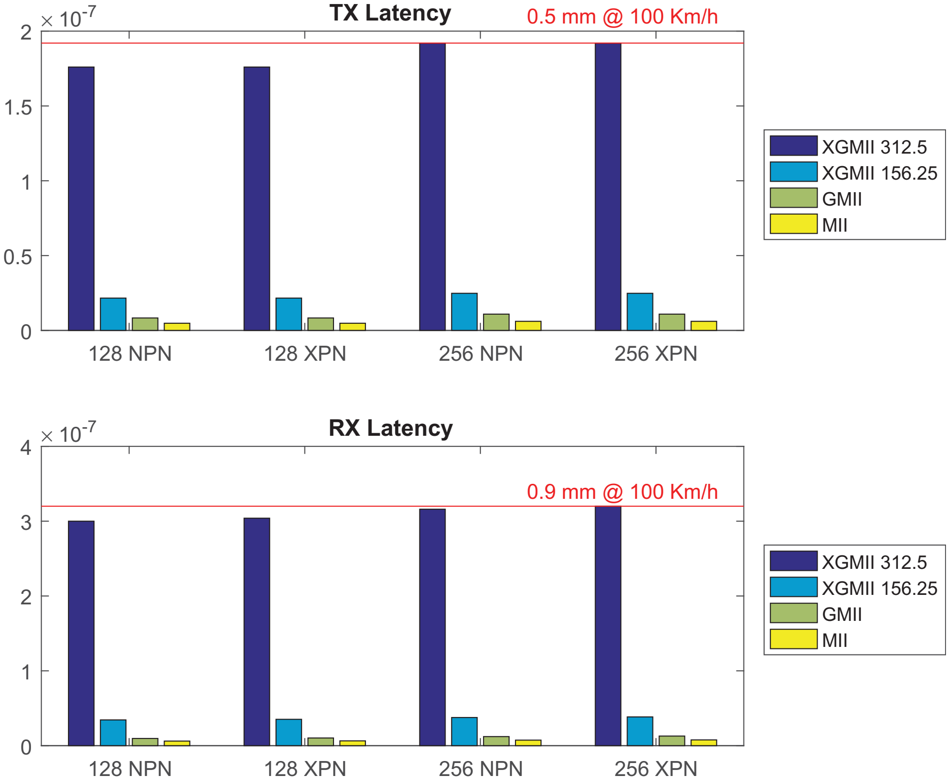

Fig. 6. Latency of each configuration for RX and TX.

for two main reasons: the AES key length and the PN approach (i.e., NPN or XPN). Indeed, the numbers of clock cycles to perform the AES encryption are 10 and 14 for 128- and 256-bit, respectively. Furthermore, the XPN requires operations on 64-bit signals, and to reduce the path and size, the architecture provides a pipelined folded implementation of 64-bit arithmetic. Therefore, the XPN and 256-bit keys require 1 and 4 clock cycles compared with the NPN and 128-bit key solutions. For each mode, the latency is smaller than 350 ns, which is acceptable in the context of automobiles. Indeed, as shown in Fig. 6, the latency introduced by our architecture is such that in the same amount of time at 100 km/h (27.8 m/s), the car travels for less than 1 mm. This means that, even in a safety-related system (e.g., the braking system), the MACsec introduces a negligible delay in the communication and does not lead to any criticality.

Figure 7 summarizes the area results of the SRS/SHE approach. As depicted, the SRS size is larger for FS than HS, owing to the large number of registers needed to store all statistical information. In addition, the register interface increases depending on two choices: the SHE implementation and the mux-ed AES-CMAC

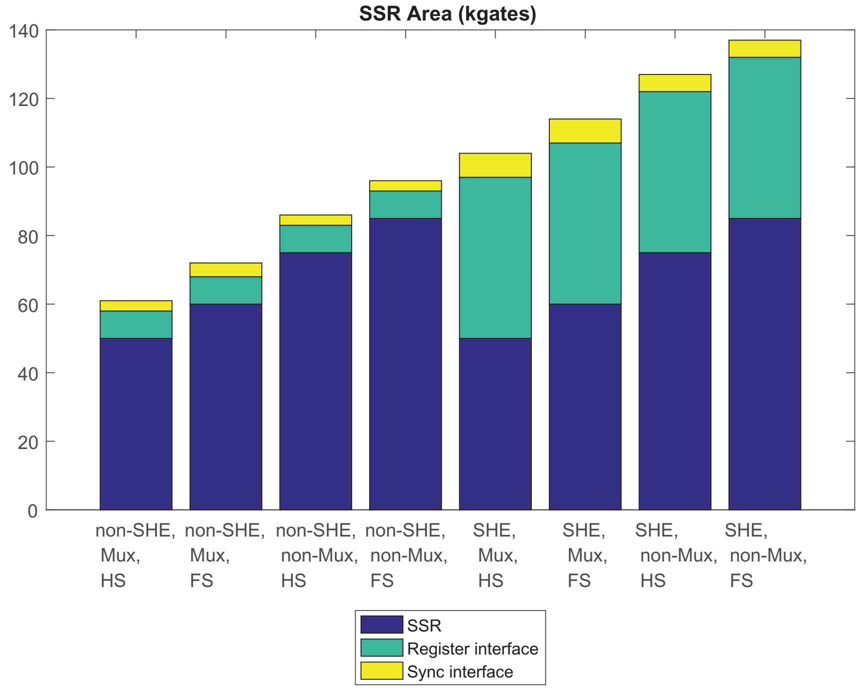

Fig. 7. Area occupation of the SRS for each configuration in kgates.

routine. As expected, an SHE-compliant interface costs more in terms of area, and a non-multiplexed AES-CMAC requires additional logic to implement this function into the register interface. The AES-CMAC is part of the SHE interface and has no influence if the system does not have this security block. An SHE-compliant interface is more secure than a non-SHE one, whereas a non-multiplexed version of the AES-CMAC completely divides the SW and HW domain, increasing the level of security. The synchronization interface has negligible effects on the size. The maximum occupation is 152 kgates provided by the SHE-compliant, non-multiplexed, FS solution

Figure 8 shows the area of the TX and RX cores depending on the synthesis to support only MII and GMII modes, the largest occupation is provided by the polynomial multiplier core of the ICV calculation. In contrast, when the system also supports the XGMII mode, the value is 7 instead of 1, and the AES is the most area-intensive core. The multiplier size also increases for XGMII due to the addition of a pipeline stage to reach the higher clock frequencies (i.e., 312.5 MHz). The total occupation is 315, 352, 590 and 685 kgates for the MII/GMII-128, MII/GMII-256, XGMII-128 and XGMII-256 speed modes, respectively.

Table 1 summarizes the security level of each configuration for all explained cases and combinations of parameters. The security level is defined by a number from 1 (minimum security) to 7 (maximum security). As expected, the highest security level

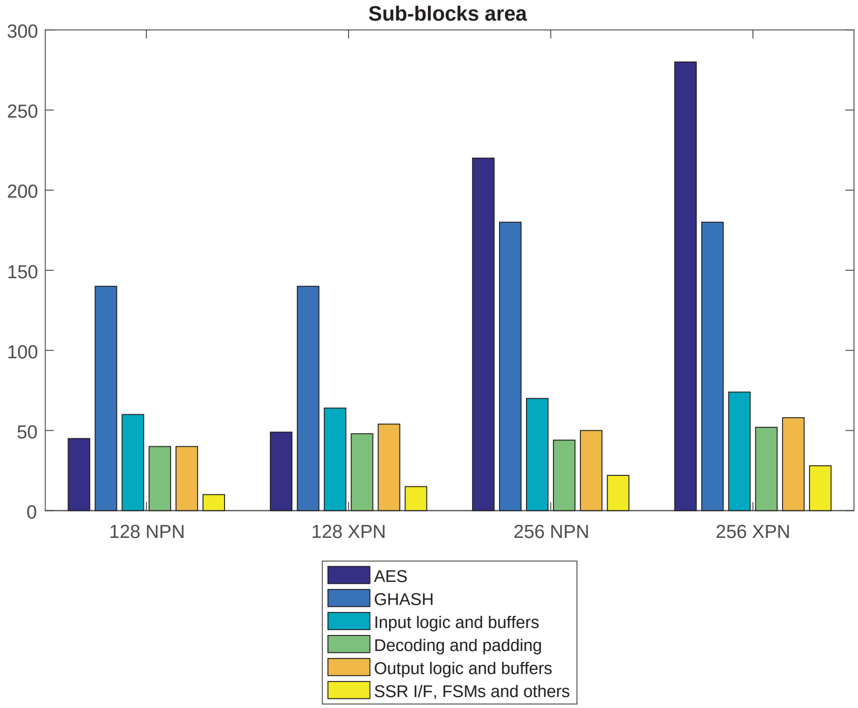

Fig. 8. Area occupation of the TX and RX cores in kgates.

Table 1. Security level versus different parameters.

	NPN, AES-128	XPN, AES-128	NPN, AES-256	XPN, AES-256
HS, Mux-ed AES	1	2	3	4
HS, non Mux-ed AES	2	3	4	5
FS, Mux-ed AES	3	4	5	6
FS, non Mux-ed AES	4	5	6	7

is reached when the AES-256, XPN, FS, non-Mux-ed version is used and this configuration corresponds exactly to the largest in terms of area.

The comparison with the state of the art shows that our work is more suitable than other studies in the context of in-car networks. Indeed, we have carefully analyzed the trade-offs of an automotive environment proposing a flexible solution that can be easily adapted to specific security needs. Furthermore, our solution adds support for 256-bit encryption functionalities that the future Internet of Things (IoT) could require. Finally, we addressed of the SW management of a MACsec compliant node, completing the pure-HW approach of other works. Table 2 shows the main differences of our work compared with previous MACsec implementations

Table 2. Comparison with other works.

	AES-128	AES-256	SW I/F	Flexibility	Automotive	Throughput (Mbps)
This work	Yes	Yes	Yes	Yes	Yes	10^2, 10^3, 10^4
Ref. 22	Yes	No	No	No	Yes	10^2, 10^3
Ref. 30	Yes	No	No	No	No	10^2

6. Conclusion

Our implementation represents an appealing solution for future Ethernet backbones and matches in-car requirements, above all including latency and throughput. We believe that our solution could be adopted for in-car network Ethernet controllers. We have described a flexible solution that is suitable for 100 Mbps, 1 Gbps and 10 Gbps Ethernet speeds and can be customized before synthesis in relation to the security needs. Future work should focus on the SW layer of the IEEE 802.1X-2010 and the interaction with the MACsec architecture and study the implementation of secure gateways that interact with MACsec-compliant networks.

References

1. S. Tuohy, M. Glavin, C. Hughes, E. Jones, M. Trivedi and L. Kilmartin, Intra-vehicle networks: A review, *IEEE Trans. Intell. Transp. Syst.* **16** (2015) 534–545.
2. J. Harding, G. Powell, R. Yoon, J. Fikentscher, C. Doyle, D. Sade, M. Lukuc, J. Simons and J. Wang, Vehicle-to-vehicle communications: Readiness of v2v technology for application, Technical report, 2014.
3. T. Steinbach, K. Müller, F. Korf and R. Röllig, Real-time ethernet in-car backbones: First insights into an automotive prototype, *2014 IEEE Vehicular Networking Conference (VNC)*, (IEEE, 2014), pp. 133–134.
4. IEEE standard for local and metropolitan area networks: Media access control (MAC) security, *IEEE Std 802.1AE-2006*, pp. 1–150, August 2006.
5. C. V. Briciu and I. Filip, The challenge of safety and security in automotive systems, in *Proc. IEEE 9th Int Applied Computational Intelligence and Informatics (SACI) Symp*, (IEEE 2014), pp. 177–181.
6. D. Brown, G. Cooper, I. Gilvarry, A. Rajan, A. Tatourian, R. Venugopalan, D. Wheeler and M. Zhao, Automotive security best practices, *White Paper*, pp. 1–17, 2015. https://www.mcafee.com/it/resources/white-papers/wp-automotive-security.pdf
7. R. Soja, Automotive security: From standards to implementation. *Freescale White Paper, Document Number: AUTOSECURITYWP REV*, 1, 2014.
8. EVITA Consortium, The evita project: E-safety vehicle intrusion protected applications, 2011. https://cache.freescale.com/files/automotive/doc/white_paper/AUTOSECURITYWP.pdf
9. Car 2 Car Communication Consortium Car 2 car communication consortium manifesto, overview of the c2c-cc system, 2007. https://www.evita-project.org/
10. PRESERVE Consortium, Preparing secure vehicle-to-x communication systems, 2011. https://www.preserve-project.eu/
11. J. Staggs, How to hack your mini cooper: reverse engineering controller area network (can) messages on passenger automobiles, in *DEFCON*, 2013. https://www.defcon.org/images/defcon-21/dc-21-presentations/Staggs/DEFCON-21-Staggs-How-to-Hack-Your-Mini-Cooper-WP.pdf

12. Q. Wang and S. Sawhney, Vecure: A practical security framework to protect the can bus of vehicles, *2014 Int. Conf. Internet of Things (IOT)*, (IEEE, 2014), pp. 13–18.
13. K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham and S. Savage, Experimental security analysis of a modern automobile, *Proc. IEEE Symp. Security and Privacy*, (May 2010), pp. 447–462.
14. S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno, Comprehensive experimental analyses of automotive attack surfaces, in *Proceedings of the 20th USENIX Conference on Security, SEC'11*, (Berkeley, CA, USA, 2011), p. 6.
15. M. Wolf, A. Weimerskirch and T. Wollinger, State of the art: Embedding security in vehicles, *EURASIP J. Embed. Syst.* **2007** (2007) 1–16.
16. S. Otsuka, T. Ishigooka, Y. Oishi and K. Sasazawa, Can security: Cost-effective intrusion detection for real-time control systems. Technical report, SAE Technical Paper, 2014.
17. D. Wampler, H. Fu and Y. Zhu, Security threats and countermeasures for intra-vehicle networks, *Proc. Fifth Int. Conf. Information Assurance and Security (IAS '09)*, Vol. 2, August 2009, pp. 153–157.
18. C. C. Wang, G. N. Sung, C. L. Wang, P. C. Chen, M. F. Luo and H. C. Hu, Physical layer design for ecu nodes in flexray-based automotive communication systems, *Proc. Digest of Technical Papers Int. Conf. Consumer Electronics*, January 2009, pp. 1–2.
19. P. Kleberger, T. Olovsson and E. Jonsson, Security aspects of the in-vehicle network in the connected car, *Proc. IEEE Intelligent Vehicles Symp. (IV)*, (June 2011), pp. 528–533.
20. T. Kiravuo, M. Sarela and J. Manner, A survey of ethernet LAN security, *IEEE Commun. Surv. Tutorials* **15** (2013) 1477–1491.
21. R. Palin, D. Ward, I. Habli and R. Rivett, ISO 26262 safety cases: Compliance and assurance, in *Proc. 6th IET Int System Safety Conf.*, (September 2011), pp. 1–6.
22. B. Carnevale, F. Falaschi, L. Crocetti, H. Hunjan, S. Bisase and L. Fanucci, An implementation of the 802.1 ae mac security standard for in-car networks, *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, (IEEE, 2015), pp. 24–28.
23. M. J. Dworkin, Sp 800-38d, recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac, 2007. http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf
24. IEEE standard for local and metropolitan area networks–port-based network access control. *IEEE Std 802.1X-2010 (Revision of IEEE Std 802.1X-2004)*, pp. 1–205, February 2010.
25. R. Escherich, I. Ledendecker, C. Schmal, B. Kuhls, C. Grothe and F. Scharberth, She: Secure hardware extension functional specification, version 1.1, *Hersteller Initiative Software (HIS) AK Security*, April, 2009.
26. H. E. Michail, D. Schinianakis, C. E. Goutis, A. P. Kakarountas and G. Selimis, Cipher block based authentication module: A hardware design perspective, *J. Circuits, Syst. Comput.* **20** (2011) 163–184.
27. A. Aziz and N. Ikram, Memory efficient implementation of aes s-boxes on fpga, *J. Circuits Systems Comput.* **16** (2007) 603–611.
28. V. K. Sharma, S. Kumar and K. K. Mahapatra, Iterative and fully pipelined high throughput efficient architectures of aes in fpga and asic, *J. Circuits, Syst. Comput.* **25** (2016) 1650049.
29. K. M. Abdellatif, R. Chotin-Avot and H. Mehrez, Fpga-based high performance aes-gcm using efficient karatsuba ofman algorithm, *ARC* (Springer, 2014), pp. 13–24.
30. K.-S. Han, K.-O. Kim, T. W. Yoo and Y. Kwon, The design and implementation of mac security in epon, *The 8th Int. Conf. Advanced Communication Technology, 2006 (ICACT 2006)*, Vol. 3, p. 4 (IEEE, 2006).