Full-protocol safety analysis of CINNAMON

Luca Dariz
IEEE Member,
Email: luca.dariz@ieee.org

Gianpiero Costantino
Istituto di Informatica e Telematica
Consiglio Nazionale delle Ricerche
Email: gianpiero.costantino@iit.cnr.it

Ilaria Matteucci
Istituto di Informatica e Telematica
Consiglio Nazionale delle Ricerche
Email: ilaria.matteucci@iit.cnr.it

***Abstract*—The gap between safety and security solutions in the automotive domain is still far from being filled. Till now, the automotive industries have been mainly devoted to provide safety relevant solution. The increasing adoption of electronic solutions to regulate vehicles' functionalities moves the attention also to cyber-security issues. In this paper, we present a full-protocol analysis of the CINNAMON intra-vehicle communication protocol, showing how it is able to satisfy both security and safety AUTOSAR requirements. In particular, we include in the analysis the synchronization messages, which were previously excluded.**

I. Introduction

Embedded systems are now almost always interconnected with some external network, and often with Internet, possibly through some kind of gateway.

This is especially true for industrial control systems and vehicular networks. In these areas however the impact of connectivity is even higher than on commodity systems (e.g., home automation, weather stations, etc) because of functional safety requirements. Historically, in these applications the safety-critical network has been isolated from external systems, and in some cases even from other internal networks, to avoid interference with other applications (e.g., powertrain network on vehicles) and in other cases to overcome bandwidth limitations (CAN Bus is often configured at 250 or 500 kbit/s). However, enhanced connectivity implies a higher exposure of the system, and this enables interference from remote with high security implications, and by extension high safety implications.

In the automotive context, in particular in-vehicle networks, the safety and security requirements are traditionally analysed separately. Protective measures are implemented in two separate protocol layers: taking as a reference the AUTOSAR standard, the Secure Onboard Communication (SecOC) profiles [1] for security and the End-to-End (E2E) profiles [2] for functional safety. In recent years some attention was put on the safety-security joint interaction and design, for example in the context of automotive ethernet [3], engine control [4], service design [5], but also in general tools for embedded system design [6]. In all these examples, the safety and security protection measures are meant to be separate protocol layers, with some duplicated mechanism and functionality.

This paper proposes a method to analyze and use a secure communication protocols, based on CINNAMON [7], which is shown to satisfy both the SecOC requirements and the E2E requirements, with the key advantage of maintaining a reduced packet size, and still be usable on classic CAN 2.0 networks with 8-byte data packets. As opposed to previous work, the analysis will be performed considering not only a single data packet, but the full protocol, so including the message rate information and other management packets. However the purpose of this analysis is not to cover the full AUTOSAR protocol stack and layers, but we will focus on the analysis and interpretation of the CINNAMON protocol only, with reference to the SecOC and E2E AUTOSAR modules.

The paper is structure as follows: next section recall the AUTOSAR security and safety profiles. §III proposes a way to integrate security and safety layers according to the AUTOSAR guidelines. §IV provides a safety interpretation and analysis of the CINNAMON protocols. In §V we describe the simulation setup used to obtain the results presented in $VI, where we show the results of the analysis on a simple application scenario. Section §VII concludes the work.

II. AUTOSAR Security and Safety layers

AUTOSAR handles security and safety aspects separately. It defines a security protocol layer [1] and an E2E protocol layer [2]. Each layer has dedicated *profiles* for security and functional safety [8], respectively. A *profile* is an assignment of values to configuration parameters.

A. AUTOSAR Security Layer

The AUTOSAR Security Layer aims to guarantee authenticity and integrity of the transmitted data. In the AUTOSAR Specification of Secure On-Board Communication document [1] a security profile is defined as the composition of parameters referred to possible Message Authentication Codes (MAC) and freshness value calculation algorithms. These four parameters can be instantiated to obtain different security profiles. In particular:

- **Profile 1**: 24 bit of MAC and 8 bit of Truncated Freshness Value, no matter how it is calculated.
- **Profile 2**: 24 bit of MAC and no Freshness Value.
- **Profile 3 (or Jaspar)**: 28 bit of MAC and 4 bit of Truncated Freshness Value from a 64-bit long Freshness Value.

AUTOSAR provides how the MAC and the Freshness Value have to be calculated and the order in which the secured frame has to be composed and decomposed. AUTOSAR assumes that all ECUs have the cryptographic keys to handle MACs [9]. Moreover, an external Freshness Manager may provide counters to both sender and receiver to support the freshness of exchanged frames.

AUTOSAR describes three different approaches to building the freshness counter, two based on counters and one on timestamps:

1) Freshness Value Based on **Single Freshness Counter**: the FVM supplies the FV to nodes connected to the network and increases the counter each time a message is sent in the communication channel. From the communication point of view, this method allows to detect repetition, loss, insertion, incorrect sequence, blocking communication errors.
2) Freshness Value Based on **Single Freshness Timestamp**: it is necessary to keep clocks synchronized within the system. The FVM must guarantee the resolution and accuracy of the time values, so that the freshness value will be the same for both the sender and all recipients.
3) Freshness Value Based on **Multiple Freshness Counter**: four counters are used to achieve FVs. Here, the FVM is implemented in a separate master ECU that manages two of the four counters, namely the Trip and Reset Counter. These counters are incremented according to well specified criteria within the AUTOSAR document.

B. AUTOSAR Functional Safety Profiles

The AUTOSAR Functional Safety Profiles are currently three [8]:

- **Profile 1** includes a 4-bit running counter, check of receiving timestamp, an explicit Data ID to state the source of the message and a 8-bit CRC.
- **Profile 2** includes a 4-bit running counter, check of receiving timestamp, an implicit Data ID (derived from the running counter but not included in the packet) and an 8-bit CRC.
- **Profile 4** includes a 16-bit running counter, check of receiving timestamp, explicit Data ID and a 32-bit CRC.

While Profile 4 is not suited for standard CAN messages because of payload length restrictions (but could be used on CAN-FD), Profiles 1 and 2 are feasible also on standard CAN. All of them are similar to the first freshness approach, i.e., Single Freshness Counter, for how the counter is computed.

An implicit Profile 2 requirement is that the counters for different sources must not be synchronized, otherwise it would be impossible to identify with certainty the correct Data ID.

III. Integrating Security and Safety Layers

With the goal of integrating both security and safety protective measures in one single protocol, we need to slightly change the design perspective. Both the SecOC and E2E profiles are designed as independent protocol layers with a single well-definite purpose, and using a number of protective measures based on the errors and threats they want to prevent. Fig. 1 shows an example of combination of SecOC and E2E as separate protocol layers.

Here, we concentrate on having a single protocol layer that covers the functionality of both SecOC and E2E profiles, relying on generic protective measures. The main reasons why this is possible and advantageous are:

- lower overhead of a single protocol layer instead of two separate protocol layers.
- at the receiver side, the decision whether a packet is valid is absolute, i.e. a packet cannot be “half valid” due to a single fault either in SecOC or E2E profiles.
- at the receiver side, the processing order of SecOC and E2E determines how an error is detected. The first layer to be processed can “shadow”an error that would be detected also by the second, or that would cause a different diagnostic.
- there are many similarities in the mechanisms used in both the SecOC and E2E profiles, in particular the usage of a counter.
- both safety and security checks need to work end-to-end from the source to the destination of the communication.

Our working assumption is that the protective measures required by both SecOC and E2E can be implemented by sharing the same mechanisms in a single protocol layer. The basic mechanisms can be defined in a quite general way as an authenticated integrity check and a sequence number, and they will be used to verify both security properties and E2E properties. Moreover, for security purposes we also add an encryption layer, which does not require a dedicated field in the protocol layer, but influences how the general mechanisms works. Similarly, for safety purposes, we add a reception timestamp as a requirement, which can be taken in the receiver protocol stack, without additional fields in the packet. In general, error detection will result from one of the two general mechanisms directly, or by combining the mechanisms with some other contextual information. By having all error detection in a single stage, there is no more shadowing due to the encapsulation order of SecOC and E2E layers. Instead, when an error is detected, additional work will be required to decide, if possible, the root cause of it, that is if it is due to an attempted attack or to a communication issue. The root cause analysis can also be deferred to an offline stage, while online

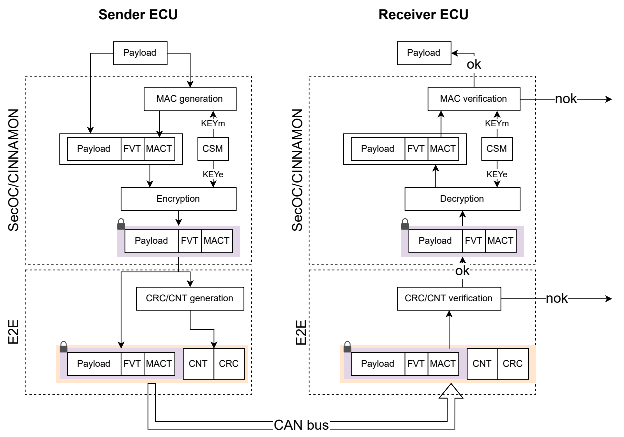

Fig. 1. Frame generation with combined SecOC/CINNAMON and E2E.

the raw data are collected.

IV. Functional Safety analysis of CINNAMON

A. CINNAMON in a Nutshell

Starting from the consideration that Confidentiality is not consider in AUTOSAR, in [7] it has been proposed the CINNAMON (Confidential, INtegral aNd Authentic on board coMunicatiON) module that builds and sends the secured data using a single CAN frame. CINNAMON inherits the first six parameters are inherited from SecOC and add two new parameters to specify both the encryption algorithm (`algorithmEncryption:String [0..1]`) and the freshness value algorithm (`algorithmFreshnessValue:String [0..1]`) (Table I). In particular, it relies on Chaskey MAC, which is robust under tag truncation [10] and on SPECK64/128, a lightweight block cipher publicly released by the NSA [11].

TABLE I
Example of a CINNAMON Profile

Parameter	Configuration Value
`algorithmFamily`	Chaskey
`algorithmMode`	Chaskey_MAC
`algorithmSecondaryFamily`	not set
`SecOCFreshnessValueLength`	64 bit
`SecOCFreshnessValueTruncLength`	8 bit
`SecOCAuthInfoTruncLength`	24 bit
`algorithmFreshnessValue`	Single Counter
`algorithmEncryption`	SPECK64/128

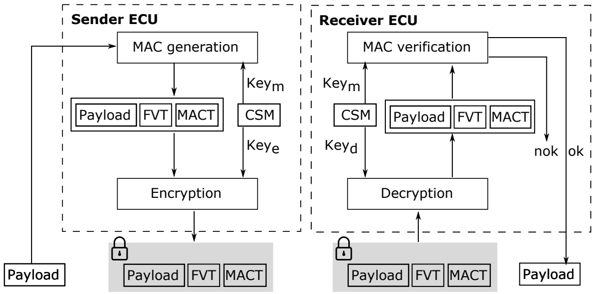

Fig. 2. CINNAMON Frame Generation and Verification [12]

CINNAMON inherits SecOC authentication and integrity mechanisms, reviewed in Fig. 2. The CINNAMON module turns an AUTOSAR secured CAN frame into a CINNAMON secured CAN frame; its data field is presented in Fig. 3. A CINNAMON secured CAN frame is formed by reducing the dimension of the payload. Then, a freshness value is used to guarantee that the frame content is fresh. To complete the data field, an additional block is used for the MAC, which ensures authentication and integrity. Finally, the entire 64 bits of the payload are encrypted to ensure confidentiality.

Let us consider a sender ECU and a receiver ECU. Before sending a payload, the sender generates the MAC starting from the payload and possibly the *Freshness Value* (Fig. 2) provided by the Freshness Manager (an ECU may decide to ignore the Freshness Value). So, the secured CAN frame is composed by the payload, the truncated MAC (MACT in Fig. 2) and, optionally, the truncated freshness value (FVT).

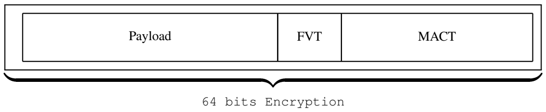

Fig. 3. The CINNAMON Secured CAN Data field

The receiver has to validate the CAN frame before accepting it and does this by verifying the MAC. In fact, the receiver generates a freshness value for verification (FVV) according to the chosen method, the counters (Fig. 2) received by the Freshness Manager and the previously received freshness value (the latest received counter in Fig. 2). Then, it calculates the MAC by using the received payload and the FVV. If the outcome equals the received MACT, then the payload is accepted, otherwise it is discarded.

B. Functional Safety Analysis

To analyse the secure communication from a safety perspective, we take as a reference the list of faults described in the AUTOSAR E2E profiles, which itself refers to the ISO 26262, part 6 [13] standard for each sender and receiver. Similar considerations can be found in other domain-specific standards related to functional safety like ISO 25119, and also in the higher-level standard ISO 61508.

If we consider AUTOSAR E2E profiles, which are the native AUTOSAR mechanisms to protect safety-related communication, they employ a mix of i) data integrity, with an additional CRC; ii) running counter/alive counter; iii) port- or PDU-specific identification; iv) timeout detection, which can be implemented on the receiving side or on the sender side (with an explicit acknowledge).

If we allow re-use of the CINNAMON fields already employed for security purposes, we can obtain a similar coverage if we add a reception timeout:

- **data integrity** - it would be ensured by the MAC. This will add to the usual error detection by means of a CRC at the physical layer, and plays the same role of the CRC in the E2E profiles.
- **running number** - this is the same data field used to ensure freshness. It allows to detect repetition, loss, insertion and incorrect sequence.
- **sender/receiver identification** - This can be achievable by means of authentication, and strongly depends on the key distribution and usage. In particular, it requires that the authentication key is not used by any other ECU. If authentication keys are specific only to one pair of sender and receiver, it can allow to detect incorrect addressing.
- **timestamping on rx side** - it allows to measure packet rate and inter-arrival time, which can be used to detect loss and delay independently from the specific packet format, as no additional information is required in the payload.

TABLE II

COVERAGE OF TRANSMISSION ERRORS OF VARIOUS COUNTERMEASURES, ACCORDING TO ISO 26262 AND AUTOSAR E2E PROFILES. THE COVERAGE OF CINNAMON IS INDICATED WITH $\checkmark^{C}$AND THE COVERAGE OF THE E2E PROFILE 1 WITH $\checkmark_{A}$. $\checkmark^{C2}$INDICATED CINNAMON WITH THE TIMESTAMP-BASED FRESHNESS. CELLS WITH $\checkmark$INDICATE A MECHANISM CONSIDERED BY FUNCTIONAL SAFETY STANDARDS BUT NOT PRESENT IN CINNAMON.

	Running number	Timestamp	Timeout	Ack	Peer Identification	Data Integrity
Repetition	$\checkmark^{c}{}_{a}$	$\checkmark^{c2}$				
Loss	$\checkmark^{c}{}_{a}$			$\checkmark$		
Delay		$\checkmark^{c2}$	$\checkmark^{c}{}_{a}$			
Insertion	$\checkmark^{c}{}_{a}$			$\checkmark$	$\checkmark^{c}{}_{a}$	
Masquerade					$\checkmark^{c}{}_{a}$	
Incorrect sequence	$\checkmark^{c}{}_{a}$	$\checkmark^{c2}$				
Corruption				$\checkmark$		$\checkmark^{c}{}_{a}$
Asymmetric communication						$\checkmark^{c}{}_{a}$
Partial receive	$\checkmark^{c}{}_{a}$					
Access blocked	$\checkmark^{c}{}_{a}$		$\checkmark^{c}{}_{a}$			

As visible in Table II, both the E2E profiles and CINNAMON can provide a complete coverage of communication errors. In particular, CINNAMON has the same coverage as E2E Profile 1 and 2.

Compared to the packet-level analysis [14], where only data packets are considered independently, the full protocol analysis requires to evaluate the packet stream and to consider synchronization messages, if required by the freshness mode.

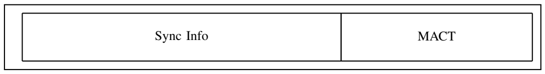

Fig. 4. The CINNAMON Secured CAN Sync field

Synchronization packets are assumed to be composed as in Figure 4, with an authentication tag but no encryption. The Sync Info will be composed of the Trip Counter and reset Counter, as described in section 11.4.2 of [1].

We then analyse the four basic mechanisms more in detail, with reference to both data and sync packets, and finally we will extend the analysis to the full protocol.

1) Running number: In principle, it could be enough to consider only the part which is included in the packet, which is the truncated freshness value (FVT). We also need to consider the periodicity of this value. This will not work reliably if a sufficiently old packet is replayed. However, since replay attack protection is handled by the SecOC module, even the short value would be enough to perform basic control flow checks, assuming the time window for delayed and reordered packets is shorter than the periodicity of FVT. The FVM needs then to be enhanced with some control flow checks based on the same FVT used to detect replay attacks.

Two different perspectives emerge from the elaboration of the running counter. From a safety perspective, the admitted running counters on rx side must enable diagnosis of some packet loss, without losing the functionality but at the same time detecting the anomaly. This error, if confirmed, may trigger an emergency action, but it can also remain a transient error, in which case the error condition will be reset after some time. This detection is usually performed with a state machine. On the other hand, from a security perspective, allowing more than one counter decrease the strength of the replay attack protection, because some more replayed packets could then be accepted. This could be solved if the full FV is used for authentication instead of FVT, so a replay attack will result in an authentication failure.

Some additional considerations are needed for the synchronization message: we need to evaluate the impact of packet loss on synchronization, considering that this message is not acknowledged by the receiver. Thus the FV management master cannot know if both the sender and the receiver are synchronised to the same ResetCounter.

There are two possible ways to enhance the reliability of the synchronisation

- if we consider only the CAN bus, as in this paper, we could rely on the bus-level acknowledge if the FV management master, the sender and receiver are connected to the same CAN bus. However, this provides still a limited guarantee of synchronisation, because it only guarantees that the message was correctly received, not that it was correctly understood and the ResetCounter updated.
- in case of de-synchronisation (e.g. because of single packet loss) the communication between sender and receiver can still be possible, thanks to the SecOC specifications. The receiver can then know if the sender has a different synchronisation, for example because it receives continuously messages with a newer ResetCounter. This information could be propagated to the FV management master to restart the synchronisation procedure after a given number of packets with a newer ResetCounter is received. However, there is no such mechanism in AUTOSAR, and if there is a de-synchronisation between sender and receiver, it needs to wait for the next sync message; this could be considered as a transient error at diagnostic level. Adding this mechanism will have an impact on the security of the communication, so we consider the recovery procedure of SecOC.

While the first depends on a specific network topology, the second allows to detect a partial de-synchronisation, and could be treated as a warning condition, which will be cleared by another synchronisation message.

2) Timestamping: Since we do not include timestamp information in the packets, we can only capture the receiving time. While this is sufficient to verify the packet rate and inter-arrival time, and to handle packet timeouts, further considerations are needed for the presence of the SecOC layer.

Even in case of multiple freshness values the verification time is constant, thanks to the procedure described SecOc specifications. Furthermore, the processing time on a real hardware is limited and well below the message repetition time; as shown in [15], it is in the order of tenths of microseconds. This is negligible compared to the message repetition time, which is in the order of tenths of milliseconds. For this reason, the timestamp can be acquired in software, without the need of special hardware support, but as part of the usual real-time processing of incoming packets.

3) Sender identification: Authentication allows us to verify the identity of the sender, even if this information is not explicitly encoded in the packet. If we assume that different keys are used for different purposes, we can also have a minimal separation from non-safety related traffic. It is worth noting that deriving identity from the authenticating key has different failure modes compared to using an explicit identifier. An additional field would impose some constraints on the message payload, so checking this field for correctness decrease the residual probability of error. On the other hand, trying to authenticate the packet with a given key does not add this stronger protection with respect to transmission errors. This implicit identification is similar to the E2E Profile 2.

4) Data integrity: Integrity of the data is ensured, against transmission errors, by means of a CRC in the CAN protocol. Note that in the CAN protocol also other consistency checks are performed, e.g. on specific bits of the frame header, see [16] for an overview. Additionally, the CINNAMON protocol adds a MAC field for security purposes, which ensures integrity against attackers. From a cryptographic point of view, the strength of the MAC against transmission errors can vary considerably depending on the specific key and message, with the average being equal to the guessing probability usually considered for security purposes (see [14] for more details on a single packet analysis), and the worst-case being equal to no protection against transmission errors.

The choice of using the average or the worst case residual rate depends on how the key is diagnostically considered. For security purposes, the key must be assumed to be completely unpredictable (i.e. uniformly distributed), and its value can be changed during the system lifetime. However, each different value of the key will result in a different average residual error rate, because it causes a different error distribution. So, we define P_{re} as the residual error rate of the CINNAMON protocol; additionally, from the safety analysis we can derive an upper bound P_{re}^{up} that must not be exceeded.

The more conservative estimate would be to consider the worst residual error rate with the worst key value: however, if the key is periodically renewed every T_r, we will have a different value of P_{re} every T_r, and it is possible to estimate $P(P_{re} < P_{re}^{up})$ over a given period of time. The long term temporal average of P_{re} will be asymptotically equivalent to the average over all possible keys, which is the guessing probability.

In the previous model [14] the value of P_{re} was estimated over all combinations of key and messages. However, considering the different messages only adds a scaling factor, and does not change the average P_{re} value. Suppose that for a given m' we have $P_{re}^{(m',k)}$, averaged only on all possible keys, i.e. considering all possible permutations done by the encryption and decryption process. Then, a single message will already cover all possible mappings of the error vectors, and the average will be independent from the specific m'.

C. Residual Error Rate

To compute the residual error rate of the MAC integrity check for a single packet, we need to define a reference error model, which we will then use to compute the packet error rate. One such model is the one proposed by Charzinski; for our purposes we only need to consider the error patterns resulting on the data payload. The error model is based on a classical Binary Symmetric Channel Model, with the exclusion of bursts up to 15 bits and 1-bit errors. These characteristics derive from the characteristics of the CAN physical layer encoding, which employ a stuffing rule and a 15-bit CRC. For a more detailed explanation, we refer to [16].

Then, to obtain the residual error rate of the whole protocol, we need to consider the probability of having an undetected error while considering the complete packet flow. In general, we can compute it as:

$$\Lambda = 3600\,\Gamma\,\nu\,\eta\,R(p) \quad err/h \tag{1}$$

where 3600 is the factor to obtain the failure rate per hour, $R(p)$ is the residual error probability for a single encrypted packet, ν is the transmission rate of the packets, η is the number of packets transmitted with the given rate, and Γ is a safety margin. This formula is derived from ISO 15998 Annex D [17], where a simple message-based control system is presented. The choice to use *err/h* is due to easier comparison with dangerous failure rates with values usually found in functional safety standards as a requirement for different safety integrity levels (e.g. SIL levels).

V. Simulation Setup

The simulation results for the E2E Profile 1 presented in the next section were implemented with a dedicated C++ program, where the encryption and decryption phases are performed for both protocols. The value of P_{re} is obtained by iterating over different user payloads and encryption/authentication keys, and checking the decryption result when an error pattern is applied to the encrypted packet. Encryption is done with the SPECK64/128 algorithm and the MAC value is computed with Chaskey. In both cases, we took the reference C implementation from the original papers. The CRC algorithm is implemented using the crcany library from M. Adler [18].

VI. Application Example

It is not easy to find in literature a reference application with sufficient details to evaluate a communication protocol like CINNAMON. Some example of automotive architectures are available, but without details on the message flows, and most related work is about the architecture definition at software level, for example the Automotive Architecture Framework [19], Archietctural Design Framework [20]. In AUTOSAR there is also a concrete example use case for functional safety, but it does not involve communication as it requires one single control unit.

To analyse the performance of the freshness mechanism of CINNAMON for transmission error detection, we define in this section a simple reference scenario, then we analyse the error detection capabilities with reference to AUTOSAR and ISO 26262 requirements for functional safety. This is done only for illustration purposes, and does not necessarily reflect a real automotive application, but rather illustrates the analysis for a simple message flow.

In real applications, the message flows are likely more complex than this example; however a similar analysis on the residual error rate is required anyway by functional safety standards. In our example we address the issue of computing $R(p)$ with a protocol embedding security properties. With more complex scenarios, this result can be composed with the usual rules of probability.

We assume to have a control application where secure data are to be exchanged. As a simple example, for illustration purposes, suppose there is a Control Unit (CU) which sends a periodic message to the Actuation Unit (AU), and this message depends on additional information received by different Sensor Unit (SU). The relevant network diagram is shown in Fig. 5. In a concrete implementation, these units could correspond for example to a Vehicle Control Unit, Engine Control Unit, Transmission Control Unit, Anti-Lock Braking System Unit, depending on the specific vehicle and application. Also, suppose that this communication flow needs to be protected with AUTOSAR SecOC, and the authentication and encryption keys are known and pairwise distributed on each communication link.

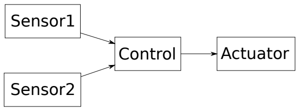

Fig. 5. Network diagram for the reference application.

We suppose that the communication flow is composed of these streams: a 50 Hz packet stream from Sensor1 to Control; a 50 Hz packet stream from Sensor2 to Control; a 50 hz packet stream from Control to Actuator.

For the case of CINNAMON with a single freshness counter, the communication protocol consist only in the stream of encrypted data message. With reference to Equation 1, we have then $\nu = 50$, $\eta = 3$. Also, we take $\Gamma = 100$ as a safety margin. and from [14] we can use:

$$R(p) \approx P_{re}^{CAN} \frac{\phi_m}{2^{\tau_s + \phi_s}}, \text{ for } 2^{\mu_s} \gg 1 \tag{2}$$

where P_{re}^{CAN} is the residual error rate of a single message on the CAN bus, without the CINNAMON protocol.

For the case of CINNAMON with multiple freshness values, we have to consider also the synchronisation message. As discussed above, this message could be received only by one of the communication units, causing de-synchronisation. However, this would not prevent the communication, thanks to SecOc, but the receiver is able to detect the error. We can also consider the communication as degraded in this case. A loss of communication would then happen only after 3 consecutive synchronisation messages are lost by the same peer. Only then the receiving procedure would not be able to successfully decode the message.

As the synchronization messages are not encrypted but they only have an authentication tag of length $\tau_s = 24$, we have:

$$R_{sync}(p) \approx P_{re}^{CAN} 2^{-\tau_s} \tag{3}$$

Assuming the synchronisation message is sent with rate ν_{sync}, we can derive the hourly failure rate as:

$$\Lambda = 3600\,\Gamma\,(\nu\,\eta\,R(p) + \nu_{sync}\,\eta_{sync}\,R_{sync}(p)) \;\; err/h \tag{4}$$

where, depending on the value of η_{sync} we can consider the degraded communication or the communication loss cases.

In Figure 6 we plot the performance of CINNAMON with various freshness models, and compare it with the plain CAN bus and with the E2E profile 1, for reference. We can observe that CINNAMON provides significantly better performance compared to the plain CAN bus, and also compared to the E2E profile 1, to a lesser extent. In particular, with E2E profile 1, the curve is different and reflect the CRC performance, which increase with low p, while in CINNAMON the performance gain from the plain CAN bus is independent from p.

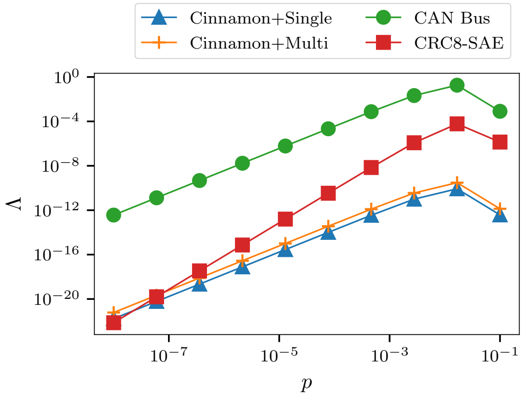

Fig. 6. Λ depending on the bit error probability p, using $\mu_s = 32$, $\tau_s = 24$, $\phi_s = 8$, $\phi_m = 3$, $\Gamma = 100$, $\nu = 10$ $\eta = 2$ $\nu_{sync} = 1$ $\eta_{sync} = 3$.

VII. Conclusion

The paper presents a full-protocol functional safety analysis of CINNAMON with reference to the AUTOSAR guidelines and ISO 26262. In particular, we model the probability of residual error caused by transmission errors. The protocol is evaluated on an application scenario to understand the impact of different freshness models, and which could be the one that better fill the gap between security and safety requirements.

As future work, we would like to implement the protocol on an embedded device, such as Raspberry or STM Discovery boards, resembling ECU capabilities, to evaluate the applicability of the protocol in a real automotive environment.

References

[1] AUTOSAR, “AUTOSAR: Specification of Secure Onboard Communication,” *AUTOSAR CP Release 4.4.0*, pp. 1–151, 10 2018.

[2] ——, “AUTOSAR: Specification of SW-C End-to-End Communication Protection Library,” *AUTOSAR CP Release 4.3.1*, pp. 1–315, 12 2017.

[3] H. Ju, B. Jeon, D. Kim, B. Jung, and K. Jung, “Security considerations for in-vehicle secure communication,” in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, 2019, pp. 1404–1406.

[4] M. Sojka, M. Kre, and Z. Hanzlek, “Case study on combined validation of safety security requirements,” in *Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems (SIES 2014)*, 2014, pp. 244–251.

[5] H. Mun, K. Han, and D. H. Lee, “Ensuring safety and security in can-based automotive embedded systems: A combination of design optimization and secure communication,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7078–7091, 2020.

[6] L. Apvrille and L. W. Li, “Harmonizing safety, security and performance requirements in embedded systems,” in *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2019, pp. 1631–1636.

[7] G. Bella, P. Biondi, G. Costantino, and I. Matteucci, “CINNAMON: A module for AUTOSAR secure onboard communication,” in *16th European Dependable Computing Conference, EDCC 2020, Munich, Germany, September 7-10, 2020*. IEEE, 2020, pp. 103–110. [Online]. Available: https://doi.org/10.1109/EDCC51268.2020.00026

[8] AUTOSAR, “Overview of Functional Safety Measures in AUTOSAR,” *AUTOSAR CP Release 4.3.0*, pp. 1–96, 11 2016.

[9] ——. (2019) Specification of Key Manager. [Online]. Available: https://www.autosar.org/fileadmin/user_upload/standards/classic/19-11/AUTOSAR_SWS_KeyManager.pdf

[10] N. Mouha, B. Mennink, A. Van Herrewege, D. Watanabe, B. Preneel, and I. Verbauwhede, “Chaskey: An efficient mac algorithm for 32-bit microcontrollers,” in *Selected Areas in Cryptography – SAC 2014*. Springer International Publishing, 2014, pp. 306–323.

[11] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, “The simon and speck families of lightweight block ciphers,” Cryptology ePrint Archive, Report 2013/404, 2013, https://eprint.iacr.org/2013/404.

[12] AUTOSAR. (2019) Specification of Secure Onboard Communication AUTOSAR CP R19-11. [Online]. Available: https://www.autosar.org/fileadmin/user_upload/standards/classic/19-11/AUTOSAR_SWS_SecureOnboardCommunication.pdf

[13] ISO, “ISO 26262 - Road Vehicles - Functional Safety,” *International Organization for Standardization*, 2011.

[14] L. Dariz, G. Costantino, M. Ruggeri, and F. Martinelli, “A Joint Safety and Security Analysis of message protection for CAN bus protocol,” *Advances in Science, Technology and Engineering Systems Journal*, vol. 3, no. 1, pp. 384–393, 2018.

[15] G. Bella, P. Biondi, G. Costantino, and I. Matteucci, “Toucan: A protocol to secure controller area network,” in *Proceedings of the ACM Workshop on Automotive Cybersecurity*, ser. AutoSec ’19. New York, NY, USA: ACM, 2019, pp. 3–8. [Online]. Available: http://doi.acm.org/10.1145/3309171.3309175

[16] J. Charzinski, “Performance of the error detection mechanisms in can,” in *Proceedings of the 1st International CAN Conference*, Sept 1994, pp. 20–29.

[17] ISO, “ISO 15998 - Earth-moving machinery Machine-control systems (MCS) using electronic components Performance criteria and tests for functional safety,” *International Organization for Standardization*, 2008.

[18] M. Adler. (2021) Crcany suite. [Online]. Available: https://github.com/madler/crcany

[19] P. Kluge, W. Krenzer, S. Merenda, M. Gleirscher, D. Wild, and M. Broy, “Toward a holistic and standardized automotive architecture description,” *Computer*, vol. 42, no. 12, pp. 98–101, dec 2009.

[20] H. G. C. Góngora, T. Gaudré, and S. Tucci-Piergiovanni, “Towards an architectural design framework for automotive systems development,” in *Complex Systems Design & Management*, M. Aiguier, Y. Caseau, D. Krob, and A. Rauzy, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 241–258.