Trade-Off Analysis of Safety and Security in CAN bus communication

Luca Dariz, Michele Selvatici, Massimiliano Ruggeri
IMAMOTER-CNR
Via Canal Bianco, 28, Ferrara, Italy
{l.dariz,m.selvatici,m.ruggeri}@imamoter.cnr.it

Gianpiero Costantino, Fabio Martinelli
IIT-CNR
Via G. Moruzzi, 1, Pisa, Italy
{gianpiero.costantino,fabio.martinelli}@iit.cnr.it

***Abstract*—While safety is thoroughly applied to the development of transportation systems, one of the prominent challenges of the last years is the integration with security concepts, required by the ubiquitous connectivity and increased value in data. A joint safety/security design can expose sometimes to trade-offs, since safety and security requirements may not match perfectly or even be incompatible to a certain degree. On the other hand, well-known solutions or best-practices in one field may create new issues in the other. This paper analyses an example of this joint design, that is the combination of integrity with encryption considering the constraints of a typical CAN network and real-time traffic. The analysis is presented considering different attacker models, packet fragmentation issues and the residual probability of error of the combined scheme.**

I. Introduction

Cars are not anymore just vehicles that allow passengers to go from one place to another. Cars, or broadly speaking vehicles, offer several services and connections that turn them into Cyber-Physical Systems (CPS). APPs, sensors, park and driving assistants are not only available on luxury vehicles but, nowadays, they are considered standard features present in entry-level model of cars. These features, such as the Internet connectivity, enlarge the attack surface of vehicles and, in particular, traditional communication protocols developed to work on isolated environments could not maintain the same level of robustness when new variables are taken into account. This is the case of the Controller Area Network (CAN bus) standard used by Electronic Control Units (ECUs) into vehicles to exchange messages. The CAN bus use messages of 64bits length, which depending on the payload set, can enable specific functionality of the vehicle, for instance enabling the accelerator of a car. Moreover, CAN messages do not embed security properties such as: *Authentication*, *Integrity* and *Confidentiality*, and this lack of security makes new generation of vehicles exposed to cyber-security threats. To this purpose, we cite the attack on the Jeep Cherokee performed by Valasek and Miller [1] in 2015 where the authors showed how to hack and remotely control a Jeep Cherokee. This attack exploit a security flaw in the infotainment system of the car, and to fix this flaw, the Fiat Chrysler was forced to push a software update [2].

Security issues are a major challenge for connected vehicles, as reported in [3], and this is now recognised to have important consequence also on vehicle's safety [4]; it is clear then that both aspects are fundamental in the design of an intelligent transportation system, especially regarding the communication protocols and the message protection schemes.

In this work, we propose a solution to turn CAN messages into a Security by Design format by having "on board" authentication, integrity and confidentiality properties. In particular, our solution is built through a Message Authentication Code (MAC) that is then encrypted with an additional key. In this way, the created message guarantees authentication and integrity through the MAC and confidentiality with the additional encryption. This defence strategy is studied and applied against a model of attacker that applies both a Honest-But-Curious (HBC) or Fully Malicious attack strategy. Furthermore, our solution is evaluated from a safety point of view, in particular regarding the residual probability of error. This is necessary since the outcome of the security MAC i.e. accept or reject a particular message is a form of error detection which could reveal also transmission errors (e.g. caused by noise), and the message containing the MAC could bring safety-critical information. For example, this scheme could be applied to SAE J1939 Torque/Speed Control 1 message, which embeds a 4-bit checksum within the 8-byte CAN payload. The residual probability of error is first evaluated using an ideal block cipher model, then simulation results are presented for a specific implementation choice. We show that averaging over the secret keys and over the possible messages, the value of P_{re} depends only on the length of the integrity tag used to decide whether the message is valid or not, and this is *independent* from how the integrity tag is generated. On the other hand, we show how the worst case combination of key and message reduces massively the ability of this scheme to detect transmission errors, assuming a simplified channel model. The worst-case P_{re} is then simulated, with considerations on the difficulty of finding the worst-case combination of key and message. The main contributions of this paper is to analyse a message protection protocol from both a security and safety point of view, and highlight the trade-offs that result from the analysis.

This paper is structured as follows: (§II) reviews the state of the art with regard to MAC algorithms and best practices, as well as error detection mechanisms; (§III) presents the message protection scheme discussed in this paper, discussing some design choices. In (§IV), we introduce the attacker model

and we make a security consideration on the MAC size. (§V) exposes two aspects usually relevant for the safety of the system, that are packet fragmentation and the probability of residual error P_{re}. Finally, (§VI) discusses other message protection schemes and their security and safety properties, compared to the scheme proposed here, while (§VII) concludes the paper with some motivation for future research directions.

II. Related Works

The following works refers to lightweight Message Authentication Code solutions for devices that have limited computational resources, like processors and memory. Chowdhury and Dasbit in [5] introduce *LMAC*, a Lightweight Message Authentication Code (MAC) of 64bits dimension for Wireless Sensor Network that uses hash based symmetric key MAC. The authors show that LMAC is secure against passive and active attacks and it has a low overhead compared with other similar solutions. Another 64bits MAC, called *Chaskey* is presented by the authors of [6]. Chaskey uses 128bits key to generate a MAC which length is of 64bits or more. The authors say that Chaskey generates MAC that are suitable for 32bit Microcontroller and that it does not suffer of MAC truncation [7]. Gong et al. in [8] show two versions of lightweight MAC of 64 and 128bits called *TuLP-64* and *TuLP-128* that are resources efficient and are though for body sensor networks. Then, Fouda et al. present in [9] a lightweight MAC suitable for Smart Grid communications in which two devices reach mutual authentication by sharing a session key exchanged using Diffie-Hellman and a hash-based authentication code technique.

Regarding safety properties, such as the residual probability of error, Schiller and Mattes have analysed different ways of using nested CRC codes, for example in [10]. However, when it comes to cryptographic algorithms the kind of errors treated are related to security properties and possible vulnerabilities, see for example the survey of Barenghi et al. in [11]. To the best of our knowledge, an explicit model for the residual probability of error of a system using cryptographic algorithms has not been developed, although the statistical properties of symmetric ciphers are sometimes studied in-depth, but always in the context of security, see for example [12].

III. Message protection scheme

The main scheme analysed in this paper is based on encryption, and is represented in Figure 1. In general, a message μ with length μ_{size} bits is combined with an integrity tag $\tau = H(k_2, \mu)$ of length τ_{size} , where k_2 is the *authentication key*. The combined $\mu||\tau$ is then encrypted to obtain the ciphertext $C = ENC(k_1, \mu||\tau)$, where k_1 is the *encryption key*; the ciphertext is then transmitted on the CAN bus. The receiver receives C', decrypts it and checks if $\tau' = H(k_2, \mu')$, with $\mu'||\tau' = DEC(k_1, C')$, to decide if the message is valid.

There can be some variations in this scheme; for example the integrity code can be appended to μ to form the plaintext (also known as MAC-then-encrypt approach) with $C = ENC(k_1, \mu||\tau)$, or it can be excluded from encryption

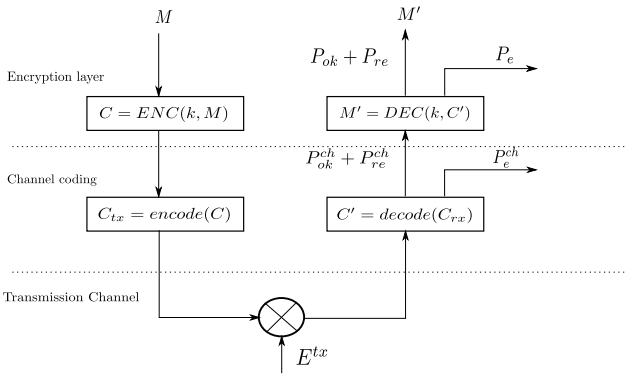

Fig. 1. Message protection scheme.

(encrypt-then-MAC approach) with $C = ENC(k_1, \mu)||\tau$. In both cases the MAC has to be computed using the original message μ. Sometimes the MAC-then-encrypt approach is considered less secure, for example see [13], but in this paper the scheme has no padding and a fixed length of the message, so these considerations do not apply. Considering the CAN bus, the first approach is more practical since there exist encryption algorithms with 64-bit block size, equal to the maximum payload of a CAN message; in this case there is no need for additional data to perform the encryption, and the ciphertext is computed as $C = ENC(k_1, \mu||\tau)$. On the other hand, if the plaintext is different from the block size (like in the encrypt-then-MAC approach), the cipher must be used in counter mode, or a stream cipher can be used. Either way, there needs to be additional information shared between the sender and the receiver, e.g. a nonce, to perform the encryption; in this case the ciphertext is computed as $C = ENC(k_1, I, \mu)||\tau$ with I being the shared information.

Another variant is to avoid the use of two different shared keys and define the integrity code as $\tau = H(\mu)$, where $H(\cdot)$ is a hash function like SHA1 or an error-detection code of the CRC family.

In this paper we consider only different possibilities for the integrity tag $\tau = H(\mu)$, which can be a proper Message Authentication Code, a hash function or a CRC; we do not consider then the encrypt-then-MAC approach, so we can define the plaintext $M = \mu||\tau$.

IV. Security considerations

A. Attacker Model

To model our attackers, we take into account that they can have locally or remotely access to the vehicle to compromise the CAN bus network by forging or altering messages that are verified as valid by the recipients. For instance, attackers were able to exploit a bug of the authentication system of the vehicle and remotely they access the CAN bus infrastructure

Attack	*Goal*	Defence
Impersonation	Forging or altering messages that are valid by the recipient	Confidentiality Authentication Integrity
Replay	Re-use messages that are considered valid by the recipient	Authentication
Sniffing	Read content of messages	Confidentiality

TABLE I
SUMMARY OF ATTACK AND DEFENCE STRATEGIES.

using a classic IP connection, and once inside the vehicle, they are able to forge valid message or even altering their contents.

Our defence strategy is to avoid that one or more attackers are able to forge valid messages, keeping enabled confidentiality of proper messages generated. Also, we aim at identifying messages that were altered, i.e., *losing of integrity*. So, our defence strategy uses three security properties that are:

Authentication: A recipient should be able to verify whether the message is sent by a legitimate sender;

Integrity: A recipient should be able to verify whether the message has been altered during its transmission;

Confidentiality: it guarantees that the content of the message is not revealed to an illegitimate entity, as it can happen with the Man-in-the-Middle (MITM) attack;

We apply our defence strategy against attackers who can play the following attacks:

Honest-but-Curious (HBC): Also known as *Passive Attack*; an attacker may exploit the information legitimately gleaned by capturing messages exchanged over the CAN bus infrastructure, but he/she will not perform any malicious activity to harvest it.

Fully Malicious (FM): Also known as *Active Attack*; an attacker is able to forge or alter messages that are considered valid, after a verification step, by the recipient. So, the attacker strategy is to succeed in at least one of the following attacks:

- *Impersonation* attack: the attacker is able to forge or alter messages that are considered valid by the recipient;
- *Replay* attack: the attacker is able to re-use valid messages with a malicious or fraudulent aim;
- *Sniffing* attack: the attacker is able to read the content of any messages exchanged through the CAN bus infrastructure:

In Table I we show which defence strategy we adopt to block or mitigate each of the above attack strategy.

B. Security considerations on encrypted MAC and MAC size

While *authentication* and *integrity* are reached with the Message Authentication Code, the *confidentiality* property is achieved by encrypting $\mu||\tau$. It avoids that an attacker is able to sniff the content of the messages exchanged among ECUs making harder the "job" of an attacker. Without encryption an attacker is capable to sniff messages, but not easily forging valid messages. Instead, with encryption the attacker's knowledge is still minor, and the probability to forge a valid message is tied to the **guessing attack**[1] plus the encryption of $\mu||\tau$. In [14] it is highlighted the importance of having a robust MAC to be resistant against the guessing attack. In particular, the Dworkin says that a sound MAC is achieved when its size is greater than 64bits, i.e., $\tau_{size} \geq 64$-bits. But, due to CAN-bus message size restriction, i.e., 64bit in total, it is very hard to keep those inequality true and we need a workaround to guarantees the security properties. To this purpose, we can apply two different strategies: i) concatenating a MAC which size is at least 64-bit, or ii) limiting the number repeated trials made by an attacker before considering invalid the key that generates the MAC. The first solution may cause the fragmentation issue that we detail better in §V-A. Instead, the second solution can lower the τ_{size} requirement. To this purpose, always in [14] it is explained how to calculate the right τ_{size} depending on the following two bounds:

MaxInvalids: as the limit on the number of trials of an attacker before the key is retired;

Risk: the highest acceptable probability for an inauthentic message to be accepted as valid;

Then, due the above parameters, the τ_{size} should satisfy the following inequality:

$$\tau_{size} \geq lg(\frac{MaxInvalids}{Risk}). \quad (1)$$

So, from equation 1 if the system can tolerate up to 30 (2^5) messages before considering the key invalid, and the system can accept 2^{-11}, i.e., $Risk = 2048$, chance of inauthentic messages; in this case, the inequality 1 is satisfied for any value of τ_{size} that is greater than or equal to 16. So, considering that the size of a CAN-message is generically 48bits (μ_{size}) and the maximum bandwidth of the communication channel is 64bit ($bandwidth_{max}$), we obtain that $\mu_{size} + \tau_{size} \leq bandwidth_{max}$. This inequity defines the lowest condition to have a $\mu||\tau$ message for CAN having $MaxInvalids = 2^5$ and $Risk = 2048$.

V. SAFETY CONSIDERATIONS

A. Fragmentation issues

It is well known that a single CAN message is too short for the proper implementation of many security properties. However there are use cases where using a second CAN message (e.g. for authentication purposes) is not acceptable since it introduce unnecessary latency in the complete reception of a message and it increases the residual error rate in the communication (see for example the appendix D.5.2 of ISO 15998 [15]). In this case the application requirements limit the security level achievable without changing the network communication protocol stack, and is fundamentally due, for a point-to-point communications, to the limited payload length of a single CAN 2.0 bus message.

[1]The guessing attacks is defined as the probability to forge a valid MAC after a number of trials.

B. Probability of Residual Error

The analysis of the probability of residual error P_{re} in a communication protocol is usually a difficult task. However, the use of encryption simplify a lot the calculation, provided that a measure of P_{re} is given for the channel coding, which correspond to P_{re}^{ch} with reference to Figure 1. Here P_{re} is obtained assuming Shannon's ideal cipher model, which has been used in other works like [16], and the results are validated through simulations.

1) Theoretical model: According to Shannon's model, an ideal cipher is a random family of permutations, chosen independently for each possible key. More precisely, suppose $\mathcal{K}$ is the set of all keys and $\mathcal{M}$ is the set of all messages. An ideal block cipher is a map $ENC : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{M}$ where, for each key $k \in \mathcal{K}$, the function $ENC_k(\cdot) = ENC(k, \cdot)$ is a random permutation on the message set $\mathcal{M}$ (independent of any other permutation).

In this context, it is more useful to fix a specific pair (k, m) of key and message, and consider the cipher text message set as:

$$\mathcal{C}' = \{C' : C' = C + e', e' \in \mathcal{E}'\} \tag{2}$$

while the plain text message set is:

$$\mathcal{M}' = \{M' : M' = M + e, e \in \mathcal{E}\} \tag{3}$$

where the set $\mathcal{E}'$ is the set of all possible undetected error vectors after channel decoding and $\mathcal{E}$ is the set of all error vectors after decryption. Recalling from section III that $M = \mu||\tau$, the probability of residual error can be defined as:

$$P_{re} = P(\mu \neq \mu', \tau' = H(\mu')) \tag{4}$$

with $M' = \mu'||\tau'$. The distribution of the values in $\mathcal{E}$ depends on the distribution of $\mathcal{E}'$, in a different way for each different pair (k, m), since the function $E(k, \cdot)$ will correspond to a different and independent permutation.

Considering all the possible pairs (k, m), each element of $\mathcal{C}'$ can correspond to all the possible values of $\mathcal{M}'$. If the elements of $\mathcal{C}'$ are uniformly distributed (such as when considering a random message attack) the probability that $\tau' = H(\mu')$, averaged over all the possible (k, m), can be computed by counting as:

$$P^{avg}(\tau' = H(\mu')) = \frac{2^{\mu_{size}}}{2^{\mu_{size}+\tau_{size}}} = \frac{1}{2^{\tau_{size}}} \tag{5}$$

which is the probability of guessing a valid message. On the other hand, when transmission errors are considered, only one of the elements of $\mathcal{C}'$ correspond to the correct message, while all other elements have cumulative probability P_{re}^{ch}. The residual probability of error in this case can be computed as:

$$P_{re}^{avg} = P_{re}^{ch} \frac{2^{\mu_{size}} - 1}{2^{\mu_{size}+\tau_{size}} - 1} \approx \frac{P_{re}^{ch}}{2^{\tau_{size}}}. \tag{6}$$

Both equations 5 and 6 are valid without making any assumption on the actual algorithm H used for computing τ. This is consistent with the usual case where H is a CRC code and there is no encryption, with the value of P_{re} approaching $2^{-\tau_{size}}$ as the probability of error per bit approaches 0.5, i.e. a uniform distribution, see for example [10].

This measure of P_{re} is, however, a measure for the average case, while from the safety point of view it is necessary to consider the worst case, that is:

$$P_{re}^{wc} = \max_{\substack{k \in \mathcal{K} \\ m \in \mathcal{M}}} P_{re} \tag{7}$$

This again can be computed by counting, but this time the actual distribution of $\mathcal{E}$, given by transmission errors, must be considered, specifically the one which maximise equation 7, which is obtained using the worst-case pair (k^{wc}, m^{wc}). Using the ideal cipher model, the distribution in $\mathcal{E}$ can be obtained from a permutation of the distribution in $\mathcal{E}'$, so a simpler way to compute P_{re}^{wc} is to take the $2^{\mu_{size}} - 1$ error vectors of $\mathcal{E}'$ with higher probability $P_{e'}$ and sum their probability. Previous work on P_{re} for the CAN bus like [17] and [18] only compute the cumulative value of P_{re}^{ch}, with assumptions also on the structure of the CAN network. This could be considered as an upper bound for the various $\mathcal{E}$, that is

$$P_{re}^{wc} \leq P_{re}^{ch} \tag{8}$$

where equality would mean that the encryption procedure, even with an integrity check, is not effective at all in detecting transmission errors when (k^{wc}, m^{wc}) are used. More specifically, this correspond to the case where the $2^{\mu_{size}} - 1$ most probable error vectors in $\mathcal{E}$ cover practically the whole amount of P_{re}^{ch}. To illustrate this problem, we consider a simplified model with a binomial distribution $\mathcal{E}' \sim B(n, k, p)$ (which would correspond in Figure 1 to the case with no channel coding and a Binary Symmetric Channel with probability of bit error p). The worst case P_{re}^{wc} can then be obtained by first listing the probability P_l of each error vector e'_l with l bit set; this list is then sorted incrementally and then the first values are taken, one error vector at a time, until exactly $2^{\mu_{size}} - 1$ error vectors are chosen.

2) Simulation results: The simulations have been performed applying the encryption scheme to randomly selected (k, m) and using different error vectors to obtain an approximation of P_{re}. For the ENC cipher, DES and AES have been used, while for H we used CRC, a truncated SHA1 hash function and a truncated HMAC scheme based on SHA1. The simulation have been implemented as a C++ program using the Nettle v2.7.1 cryptographic library. The plotted results represent a normalized $\frac{P_{re}}{P_{re}^{ch}}$, where the value of 1 represent equality in equation 8. The value of τ_{size} varies from 0 to 16; greater values, which in theory correspond to a lower P_{re}, have not been simulated since they would have taken too much time to yield a result with reasonable precision; however the results are still meaningful with respect to the theoretical model. The complexity of the simulation has three factors: k, m and e'. For example, using 1000 different keys, 1000 different messages and 10000 error patterns the total

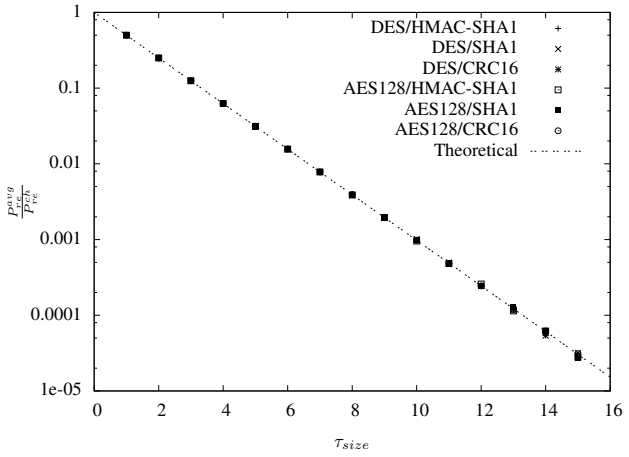

Fig. 2. Normalised average P_{re} obtained through simulation and from equation 6, for different values of τ_{size}.

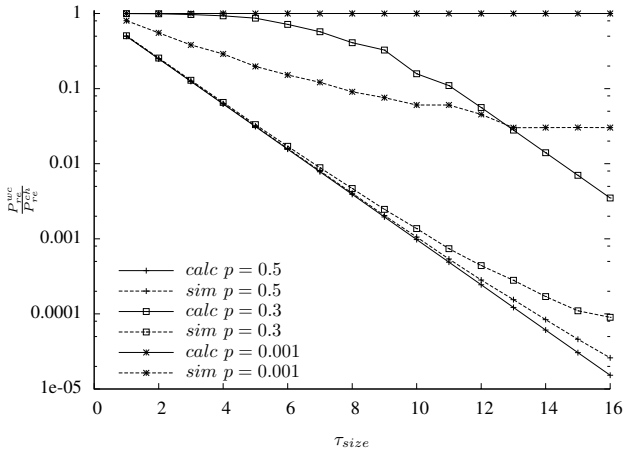

Fig. 3. Normalised worst-case P_{re} obtained with $\mathcal{E}' \sim B(n,k,p)$ through calculation and simulation using a DES/SHA1 scheme, with $\mu_{size}+\tau_{size} = 64$, for different values of τ_{size}.

number of iterations is $1000 \cdot 1000 \cdot 10000 = 10^{10}$. However each value of P_{re} is evaluated using 10000 samples, so lower values close to 10^{-4} will have a lower accuracy. This explains the convenience to simulate with low τ_{size}. In Figure 2 the normalised P_{re}^{avg} is plotted, both resulting from equation 6 and from simulations. The correspondence between the theoretical model and the simulation results is very good, and the results are independent either from the H algorithm used to compute τ and the encryption algorithm ENC. Only a small glitch is visible for $\tau_{size} = 14$, presumably due to the relatively small number of iterations. The simulations results are accurate because all the P_{re} are averaged over the pairs (k,m).

In Figure 3 the normalised P_{re}^{wc} is plotted with varying τ_{size}. The numerical values, displayed with a continuous line, are evaluated with the algorithm described in section V-B1, while the simulation results are taken as the value of P_{re}^{wc} from all the tested (k,m). In this case the simulations do not match the theoretical model; the reason is that while the theoretical

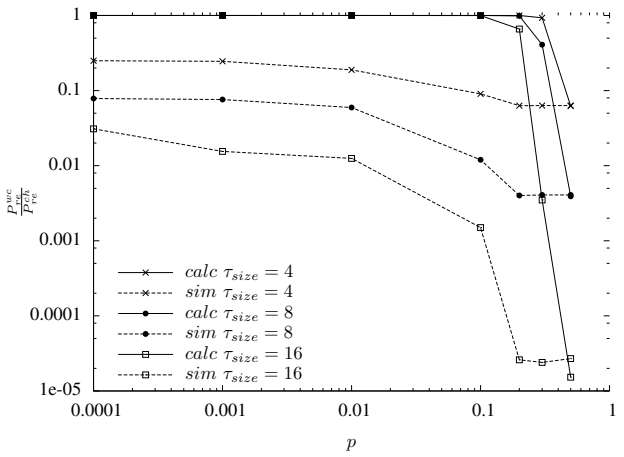

Fig. 4. Normalised worst-case P_{re} obtained with $\mathcal{E}' \sim B(n,k,p)$ through calculation and simulation using a DES/SHA1 scheme, with $\mu_{size}+\tau_{size} = 64$, for different values of p.

model assumes that (k^{wc}, m^{wc}) is known, in practice this is not true, although for some ciphers it could be feasible to calculate it. In this case a great number of (k,m) combinations are chosen and the worst case is considered. However, being unable to scan all the (k,m) space, it is unlikely to find the worst case but only a "bad" pair (k,m) is found, for which P_{re} differ significantly from the average case.

In Figure 4 the normalised P_{re}^{wc} is plotted with varying p. For low values of p, the value of P_{re}^{wc} does not change a lot, since the most probable error patterns are always the ones with 1 bit error. On the other hand, with higher p the value of P_{re}^{wc} approaches P_{re}^{avg}, which is reached with $p = 0.5$, corresponding to a uniform distribution. The issue of finding the worst case (k,m) is then different depending on the bit error probability of $B(n,k,p)$. For low p, approximately under 0.1, it is easier to find a pair (k,m) with high P_{re} since the most probable error patterns are the ones with only 1 bit error and are $\mu_{size} + \tau_{size}$. On the other hand, for higher p, the most probable error patterns are a much great number, because it is easier to find more than one bit error. This explains the difficulty of finding the pair (k^{wc}, m^{wc}) to simulate P_{re}^{wc} accurately.

VI. Discussion

In Table II different message protection schemes are compared and ordered with decreasing security properties.

The message protection schemes addressed in this paper correspond to the ENC+MAC and ENC+CRC schemes, depending on the choice of $H()$, to resist a fully malicious attacker with chosen plaintext. The main alternative scheme, which does not consider the Confidentiality property, is evaluated for reference, based on literature work. Here the trade-off appears clear comparing the ENC+CRC and plain+CRC scheme; while the first has better security properties, the latter has better safety properties under common channel models,

Scheme	Resists to	Security properties	Leaking out	Safety properties
ENC + MAC	Fully Malicious with chosen plaintext	Confidentiality Authentication Integrity	-	Strongly depends on (k, m)
ENC + CRC	Fully Malicious with chosen plaintext	Confidentiality Integrity	-	Strongly depends on (k, m)
plain + MAC	Fully Malicious	Authentication Integrity	Plaintext to FM and HBC attackers	Depends on $H()$
plain + CRC	Honest-But-Curious	-	Plaintext to FM and HBC attackers	Good under common channel assumption [10]
plain	-	-	Plaintext to FM and HBC attackers	-

TABLE II
SUMMARY OF MESSAGE PROTECTION SCHEMES.

because CRC codes are designed specifically for correcting transmission errors.

VII. CONCLUSION

The analysis of the scheme presented in this paper showed that a security property like Encryption directly influences a safety property like the probability of residual error. On the other hand, the length of a MAC code must respect the fragmentation constraints which can be imposed by real-time requirements. With respect to similar schemes without encryption, where a second CRC is used in addition to the one at the physical layer, the scheme ENC+CRC performs worse; this is due to the intrinsic properties of block ciphers, which transform the distribution of errors to uniform on average. Other message protection schemes could have a less drastic impact on the worst-case error detection capability, or even this error detection capability could be embedded into the encryption algorithm itself, but then the risk is to offer the possibility for a side-channel attack. Future works include the design and study of different cryptographic schemes, to offer a better trade-off between safety and security.

REFERENCES

[1] C. Valasek and C. Miller, "Adventures in automotive networks and control units," in *DEFCON 23*, 2015.

[2] ——, "After jeep hack, chrysler recalls 1.4m vehicles for bug fix," Online, 7 2015. [Online]. Available: https://www.wired.com/2015/07/jeep-hack-chrysler-recalls-1-4m-vehicles-bug-fix/

[3] B. McCluskey, "Connected cars: the security challenge for autonomous vehicles," *IET Engineering and Technology Magazine*, Feb 2017. [Online]. Available: https://eandt.theiet.org/content/articles/2017/02/connected-cars-the-security-challenge-for-autonomous-vehicles/

[4] C. Kim, "Safety challenges for connected cars," *IEEE Transportation Electrification Newsletter*, Jun 2016. [Online]. Available: http://tec.ieee.org/newsletter/june-2016/safety-challenges-for-connected-cars

[5] A. R. Chowdhury and S. DasBit, "Lmac: A lightweight message authentication code for wireless sensor network," in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–6.

[6] N. Mouha, B. Mennink, A. Van Herrewege, D. Watanabe, B. Preneel, and I. Verbauwhede, *Chaskey: An Efficient MAC Algorithm for 32-bit Microcontrollers*. Cham: Springer International Publishing, 2014, pp. 306–323. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-13051-4_19

[7] N. Ferguson, "Authentication weaknesses in gcm," Comments submitted to NIST Modes of Operation Process, May 2005.

[8] Z. Gong, P. Hartel, S. Nikova, S.-H. Tang, and B. Zhu, "Tulp: A family of lightweight message authentication codes for body sensor networks," *Journal of Computer Science and Technology*, vol. 29, no. 1, pp. 53–68, 2014. [Online]. Available: http://dx.doi.org/10.1007/s11390-013-1411-8

[9] M. M. Fouda, Z. M. Fadlullah, N. Kato, R. Lu, and X. S. Shen, "A lightweight message authentication scheme for smart grid communications," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 675–685, Dec 2011.

[10] F. Schiller and T. Mattes, "Analysis of nested crc with additional net data by means of stochastic automata for safety-critical communication," in *2008 IEEE International Workshop on Factory Communication Systems*, May 2008, pp. 295–304.

[11] A. Barenghi, L. Breveglieri, I. Koren, and D. Naccache, "Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures," *Proceedings of the IEEE*, vol. 100, no. 11, pp. 3056–3076, Nov 2012.

[12] A. Rimoldi, "On algebraic and statistical properties of aes-like ciphers," Ph.D. dissertation, University of Trento, 2010. [Online]. Available: http://eprints-phd.biblio.unitn.it/151/

[13] S. Vaudenay, "Security flaws induced by cbc padding - applications to ssl, ipsec, wtls ..." in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, ser. EUROCRYPT '02. London, UK, UK: Springer-Verlag, 2002, pp. 534–546. [Online]. Available: http://dl.acm.org/citation.cfm?id=647087.715705

[14] M. Dworkin, "Recommendation for block cipher modes of operation: The cmac mode for authentication," NIST Special Publication 800-38B, May 2005.

[15] ISO, "Earth-moving machinery - machine-control systems (mcs) using electronic components - performance criteria and tests for functional safety," The International Organization for Standardization, Genève, Switzerland, Tech. Rep. ISO 15998, 2008.

[16] C. Petit, F.-X. Standaert, O. Pereira, T. G. Malkin, and M. Yung, "A block cipher based pseudo random number generator secure against side-channel key recovery," in *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '08. New York, NY, USA: ACM, 2008, pp. 56–65. [Online]. Available: http://doi.acm.org/10.1145/1368310.1368322

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

[18] J. Unruh, H.-J. Mathony, and K.-H. Kaiser, "Error detection analysis of automotive communication protocols," in *SAE Technical Paper*. SAE International, 02 1990. [Online]. Available: http://dx.doi.org/10.4271/900699