TAURUM P2T: Advanced secure CAN-FD architecture for road vehicle

TAURUM P2T: Advanced Secure CAN-FD Architecture for Road Vehicle

Franco Oberti[1,2], Ernesto Sanchez[1], Alessandro Savino[1], Filippo Parisi[2], and Stefano Di Carlo[1]
[1]*Control and Computer Eng. Dep., Politecnico di Torino* Torino, Italy
[2]*PUNCH Torino S.p.A.*, Torino, Italy

***Abstract*—Interconnected devices are growing very fast in today's automotive market, providing new and complex features that cover very different domains. This vast and continuous requirement for new features brings to impact areas categorized as real-time safety-critical devices, opening the possibility to add potential vulnerabilities. By analyzing the security vulnerabilities within vehicle networks, this paper aims at proposing a new generation of a secure architecture based on Controller Area Network (CAN) called TAURUM P2T. This new architecture looks at mitigating the vulnerabilities found in the current network systems of road vehicles by introducing a low-cost and efficient solution based on the introduction of a Secure CAN network able to implement a novel key provisioning strategy. The proposed architecture has been implemented, resorting to a commercial Multi-Protocol Vehicle Interface module, and the obtained results experimentally demonstrate the approach's feasibility.**

***Index Terms*—CAN-bus, Rolling secret key, Automotive, Secure Embedded System, Secure CAN Network, ECC, No secret key Infrastructure.**

I. Introduction

Nowadays, automotive control systems are increasingly operating in a hostile environment. Consequently, there is a quest to make these systems resilient to cyber-attacks, thus avoiding strategic assets' exposure. Unluckily, this trend introduces severe issues for the automotive industry. The World Forum for Harmonization of Vehicle Regulations (WP.29), through the UN Economic Commission for Europe (UNECE), [1] planned new ECE Regulations for Vehicle Cybersecurity and Software Updates [2], [3]. Starting from 2023, carmakers must apply these regulations to all permanently and seamlessly connected road vehicles. They introduce non-negotiable conditions for getting approval and market access to the entire UNECE WP.29 member countries. In this context, the next years will be challenging for the automotive sector, requiring new cyber-security monitoring, detection, reporting, and response capabilities across the entire vehicle life-cycle and the entire supply chain. Failing in fulfilling WP.29 requirements means a production roadblock with a huge loss of money.

In the automotive domain, the Controller Area Network (CAN) is the main communication protocol. While designed to obtain high reliability and be employed in a high-noise environment, it is also a potential security threat source. Security mechanisms currently implemented to guarantee authenticity and integrity of CAN communications may be severely impacted by throughput limitations and limited key availability that represent a strong constraint on feature development of permanently and seamlessly connected road vehicles.

Authors contacts: {franco.oberti, alessandro.savino, ernesto.sanchez, stefano.dicarlo}@polito.it and filippo.parisi@punchtorino.com

This paper analyzes and discusses specific security vulnerabilities connected to the current CAN network architecture employed in road vehicles. It then introduces TAURUM P2T, a new Secure CAN Flexible Data rate (CAN-FD) architecture that increases the current security level in road vehicles by addressing the identified vulnerabilities and remaining strictly compliant to the WP.29 regulation. TAURUM P2T introduces the following main contributions:

- increased security with limited cost and hardware resources;
- implementation of a rolling secret key system;
- privilege separation;
- secret key auto-generation without external key infrastructures.

The paper is organized as follows: section II introduces the basic organization of a state-of-the-art vehicle CAN network while section III discusses the main vulnerabilities of this type of network. Section IV overviews the proposed secure architecture, section V provides experimental results, and finally section VI summarizes the main contributions and concludes the paper.

II. Automotive CAN Network overview

Today's vehicles include several (at least 70) Electronic Control Units (ECUs) handling various subsystems [4]. A non-exhaustive list of common ECUs includes Engine Control Module (ECM), Transmission Control Module (TCM), Adaptive Cruise Control (ACC), Electronic Stability Control (ESC), Anti-Lock Brake System Module (ABS), Body Control Module (BCM), Telematics Control Unit (TCU), Onboard Diagnostic System (OBD), Diesel Exhaust Fluid Controller (DEFC), Variable Geometry Turbine (VGT), Chassis Control Module (CCM), Light Control Module (LCM). As the reader may notice, communication is essential since often a subsystem must control actuators or receive feedback from other subsystems.

Vehicle's ECUs communicate employing the CAN protocol. The messages transmitted over the vehicle's CAN network have heterogeneous requirements in terms of accessibility (i.e.,

visibility outside the vehicle) and security (i.e., confidentiality, integrity, and authenticity). The CAN-bus is a very flexible multi-cast serial bus that supports the software implementation of a wide range of safety, security and convenience features, thus saving the cost and complexity of "hard-wired" implementations. This paper refers to the CAN Flexible Data rate (CAN-FD) version of the protocol introduced by BOSCH [5].

In a standard automotive CAN network, country-based Standard Vehicle Regulations (e.g., emission legislations) define several messages, and external inspectors must read them. The consequence is that these messages are in clear text, and vehicles are equipped with an On-Board Diagnostic (OBD) port. This port enables monitoring and read data on the CAN-bus. Therefore, to comply with emission standards, a vehicle CAN network can implement just two of the CIA triad's security pillars: integrity and authenticity [6]. Integrity and authenticity of CAN data frames are implemented using Cipher-based Message Authentication Code (CMAC) signatures [7], [8]. To avoid the implementation of replay attacks [9], a rolling counter is usually included in each transmitted frame [10].

III. Automotive CAN Network Vulnerabilities

Attackers interested in violating vehicle's ECUs by exploiting CAN vulnerabilities are mainly of two types:

- *The vehicle owner*: not interested in damaging its good. He usually aims at obtaining better vehicle performance or tampering with annoying features, like diagnostic.
- *A professional attacker*: driven by, for example, unscrupulous competitors aiming at damaging company reputations.

A. Man in the Middle attacks

Man in the Middle (MitM) attacks to the CAN network are the preferred class of attacks among vehicle owners. They exploit external devices to create a malicious CAN partial gateway. Two settings are possible, as shown in Figure 1: (A) exploiting OBD port access, and (B) placing an external CAN module downstream to the victim module.

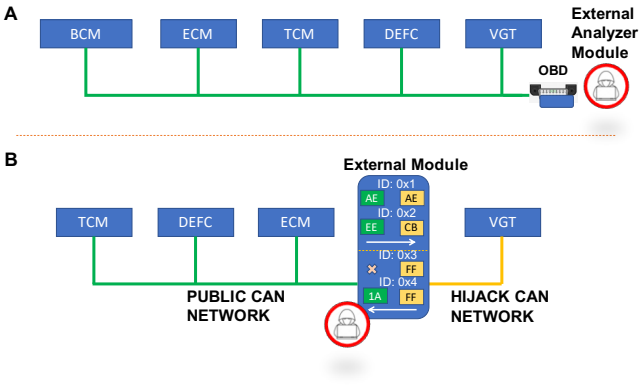


Fig. 1: Man in the Middle attack schemes

The simpler implementation exploits the OBD port through an external analyzer module (Figure 1A). This gives the attacker control over several diagnostic services. Moreover, it enables sniffing vehicle public CAN network traffic, injecting CAN frames, and potentially modifying data frames under certain physical limits.

A more efficient MitM attack requires an external CAN device placed as a CAN gateway downstream to the victim ECU. Figure 1B shows an example of this attack. The malicious CAN gateway physically isolates the right side of the network for conditioning the data frame delivered to the victim and vice versa. In this example, the frame with `ID 0x2`, directed to VGT is modified while the frame with `ID 0x1` remains unaltered. On the opposite side, the frame with `ID 0x3` produced by VGT is deleted by the malicious gateway, while the frame with `ID 0x4` passes with modified data. Roughly speaking, with this MitM configuration, an attacker can:

- Intercept and then suppress specific messages;
- Inject messages to emulate something when CMAC does not protect those messages;
- Intercept and then modify CAN data frame payloads with desirable data. This is possible with a *direct attack* when CMAC is not implemented or disabled, while an *indirect attack* is needed in all other cases. An *indirect attack* is a method to bypass the CMAC signature by sniffing and reusing CAN frame messages performing a reply attack [9]. It works pretty well when a rolling counter is not implemented but can also work when the rolling counter is present for all steady data frames.

In general, MitM attacks have a significant impact on warranty costs. Tampering with the vehicle parameters increases vehicle damage risks. In case of damage, the external devices used to mount the attack can be easily removed, making it impossible to prove a tampering action that would lead to a loss of warranty.

B. Denial of Service attacks

Denial of Service (DoS) attacks are the preferred class of attacks mounted by the second category of attackers, with the main purpose to destroy a company's reputation. The attackers exploit the Electrical/Electronic (E/E) architecture of the vehicle. Typically, they look for infotainment system exploits by leveraging the available apps. The attacker, in this case, tries to grant public CAN-bus access to force a bus off or create a task overrun event. The attacker can exploit the presence of OBD Bluetooth devices associated with unofficial apps. In automotive, losing CAN communication or miss real-time deadlines due to task overrun is considered a safety-critical event. In this condition, the vehicle must apply a proper safety recovery action with potential impact on customers. For this reason, in a secure and safe E/E architecture, a CAN gateway/firewall is usually inserted between the OBD port and infotainment system to the rest of the public CAN network.

C. Automotive Cyber-Security Key Provisioning Infrastructure

Computing CMAC signatures to guarantee CAN data frame integrity and authenticity requires a shared key provisioning

infrastructure. It is important to mention that not all the ECUs in a road vehicle require to exchange CMAC protected data frames. The security architecture defines the total amount of keys needed for CMAC calculation for each secure vehicle [11]. Since CMAC computation requires hardware acceleration, the maximum number of secret keys a vehicle can handle is strictly related to the Crypto Engine storage availability and key's length. Considering a current standard Crypto Engine for automotive applications, the available key storage is around 256B. Assuming 16B key length, it can potentially store 16 keys. The newest generation of automotive Crypto Engines should increase this availability to 1 Kbyte, thus accommodating 64 16B keys. Still, car-makers must properly handle these secrets.

Let us consider a big car-maker selling 10 Million secure vehicles per year. Assuming the use of different secret keys for each vehicle in the entire fleet and that every vehicle uses next-generation Crypto Engines handling 64 16B MAC keys, the total amount of storage required to handle the keys would be approximately 9GB. If complementary information, such as Vehicle Identification Number (VIN), module part-numbers, etc., is considered, the memory space becomes three times the aforementioned. Interestingly, the IT infrastructure for managing those numbers is not so prohibitive. Still, it requires a lot of investment in security since data need to be shared among different worldwide actors: manufacturing plants, suppliers, services, and dealers (Figure 2). In this context, it is not always easy to maintain trust environments to avoid leakages. Any violation compromises the entire vehicle fleet. A desirable target is to dismiss the IT infrastructure having local key provisioning directly at the vehicle level, with a self-build method to mitigate the above risks.

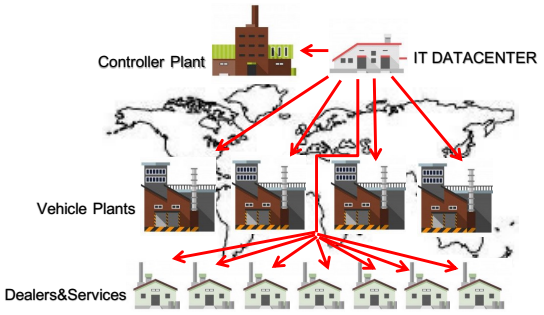


Fig. 2: Generic shared secret key proliferation

IV. TAURUM P2T

A. Architecture

TAURUM P2T is an Advanced Secure CAN-FD Architecture for road vehicles. TAURUM P2T separates ECUs data communication from security and key provisioning management using two separated CAN networks (Figure 3). The *Public CAN* network (depicted in black) and accessible through the standard CAN Gateway (CGTW) transmits the standard vehicle CAN traffic. The *Secure CAN* network (the red one) exchanges sensible information to handle shared keys and security violations. All frames in this second network are encrypted, and the network is only accessible through the TAURUM P2T Secure Gateway (SGTW). The SGTW establishes privilege levels and manages secret keys required to compute MAC signatures. The keystones of the TAURUM P2T are:

- a sharing key mechanism governed by the SGTW and able to define separated *trust zones*;
- an SGTW controlled sub-domain management of the bus for ensuring segregation;
- a rolling MAC secret key infrastructure to implement a countermeasure to MitM and reply attacks.

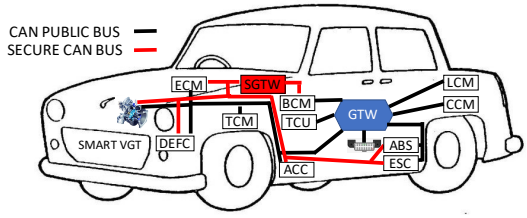
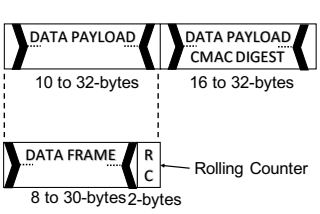
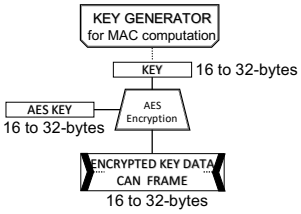


Fig. 3: TAURUM P2T Advanced Secure CAN Network for Automotive.

The TAURUM P2T architecture can only be applied to CAN-FD networks that provide data rates up to 8 Mbps, as per CAN FD's specifications [5] and, most importantly, extend the size of each data frame up to 64 bytes, which is mandatory to allocate space for security-related information. TAURUM P2T supports two types of CAN messages: the Public CANF-FD frame transmitted over the Public CAN (Figure 4a) and the Secure CAN-FD frame transmitted over the Secure CAN (Figure 4b).



(a) PUBLIC CANF-FD Frame: a payload 26 bytes to 64 byte long divided into data payload and CMAC digest required for authenticity and integrity purposes



(b) SECURE CAN-FD Frame: encrypted message sent by the SGTW to all the secure modules for key update and security management purposes.

Fig. 4: TAURUM P2T Frames

The Public CANF-FD frame uses CMAC to guarantee integrity and authenticity. Therefore, it contains the plain data and the CMAC digest. By profiling the CMAC computation time on real automotive hardware (see Section V), we identified that the worst case in throughput performance is given by a configuration using 256 bit for data and 256 bit as CMAC digest. This configuration is the most secure from a cryptography standpoint, requiring secret key updates at a slow rate. Similar security levels can be obtained with fewer digest bits at the price of an increased key update rate, thus allowing

to trade-off between digest's length and key updates. In the case of a steady condition, a frame could contain the same data over several transmissions. Even with CMAC implemented, a reply attack is always possible. For this reason, the plain data block reserves two bytes for implementing a rolling counter protecting the system from these attacks.

To compute CMAC digests, ECUs must share a secret key. To better handle CAN security, TAURUM P2T introduces the possibility of setting privilege levels (PL) in the communication (Figure 5). Each PL holds a dedicated secret key (K_{PLi}) used for MAC signature computation at that level.

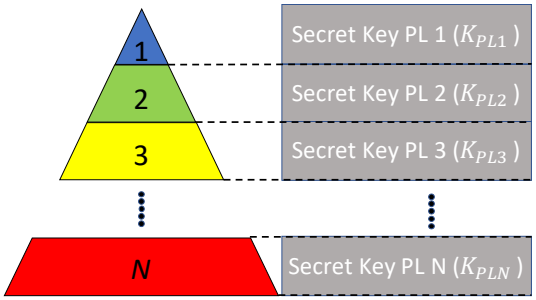


Fig. 5: TAURUM P2T Privilege Hierarchy Block Scheme. Lower numbers indicate higher privilege levels. Level 1 usually represents the SGTW.

Privilege separation is a fundamental mechanism introduced by TAURUM P2T. Every secure ECU also referred to as a secure node (SN), is associated with a PL in the hierarchy. An ECU working at $\mathrm{PL_i}$ holds all secret keys form $\mathrm{PL_i}$ to $\mathrm{PL_N}$ (i.e., $K_{PLi}, K_{PLi+1}, \ldots, K_{PLN}$). It, therefore, can communicate with its counterparts at the same PL or with those at lower PLs. In case of an attack to an ECU or secret key leaks, only the affected PL and the lower PLs are compromised for a certain period, i.e., until the activation of recovery countermeasures or update of the secret keys.

TAURUM P2T privilege separation also implements an additional feature useful to handle specific constrained security requirements. Road vehicles are often equipped with so-called secondary controllers. Usually, these modules have reduced hardware capability for meeting security requirements. Through TAURUM P2T, it is possible to define a security sub-domain, where the strength of the secret keys is reduced (e.g., 8B or less) to better fit the system throughput constraints. This requires a more frequent update of the secret keys.

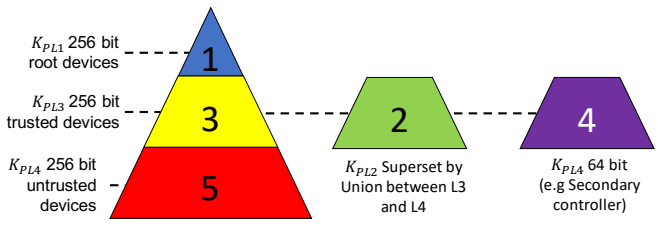


Fig. 6: TAURUM P2T Privilege Hierarchy with different key size.

Figure 6 provides an example where a privilege level acts as a security sub-gateway. In this example, the SGTW works at level 1, while all untrusted devices work at level 5. The rest of the secure modules work at level 3, except the secondary controller assigned to level 4. Finally, level 2 is assigned to the sub-domain gateway module, thus keeping the secondary controller isolated from the other nodes of the CAN network.The way privilege levels are assigned is application dependent and aims at fulfilling the security requirements of the architecture.

B. Secure CAN and Key Provisioning

The TAURUM P2T secure CAN performs key provisioning, sharing the secret keys (K_{PLi}) required by all SNs for CMAC digest calculation. Communication on this bus must be fully secure and guarantee confidentially, integrity and authenticity. Communication on the secure CAN is encrypted through symmetric cryptography based on the Advanced Encryption Standard (AES), implemented with the Cipher Block Chaining (CBC) modality [12]. The PL secret keys (K_{PLi}) used for CMAC digest calculation are also used for secure communication at different PLs on the secure CAN. To keep a high level of security, these secret keys are periodically rolled. The rolling time and the digest size are parametrized to ensure the highest flexibility.

As discussed in subsection III-C, the secret key distribution infrastructure is among the main challenges for the carmakers in developing a large fleet of interconnected vehicles. TAURUM P2T removes this bottleneck. In TAURUM P2T, all secrets are locally generated by the SGTW and securely shared with all connected nodes. This solution reduces the need to find trusted users and sustain a secure infrastructure.

Figure 7 outlines the different steps of the TAURUM P2T key provisioning implementation. During the first vehicle initialization at the plant (step 1), the SGTW performs a network discovery phase to map all SNs connected to the Secure CAN (i.e., those that require exchanging CMAC signed frames on the public CAN). It then generates using its local Crypto Engine the first set (time 0) of all PL secret keys ($K^0_{PL1}, \cdots, K^0_{PLN}$) and securely stores this information (step 2). After a complete network discovery, the SGTW handles the key provisioning node by node.

To establish the first root of trust between the SGTW and a given secure node ($\mathrm{SN_i}$) with privilege $\mathrm{PL_M}$, TAURUM P2T resorts to elliptic-curve cryptography (ECC) [13]. Every ECU connected to TAURUM P2T stores the same curve as public data in its flash. Curve25519 has been selected to balance the high efficiency of computation, thus fitting the considered hard real-time environment perfectly [14]. ECC shared keys are used to provision MAC secret keys during the network's initialization or when an attack is detected. They make it possible to build a secure point-to-point network between the SGTW and each ECU.

The SGTW and the SN start the establishment of the first root of trust (step 3) by generating a public/private key pair ((Kg_{PB}, Kg_{PR}) for the SGTW and (Kn_{PB}, Kn_{PR}) for the

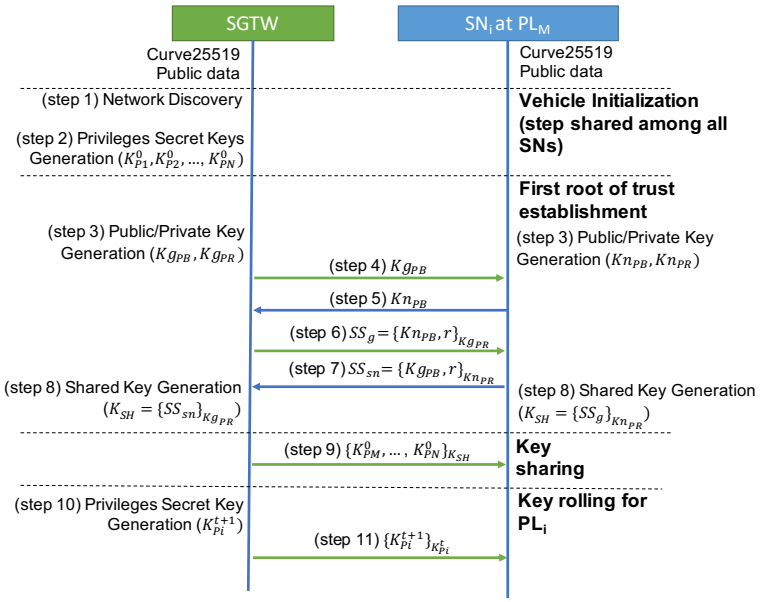


Fig. 7: TAURUM P2T Secure CAN key provisioning protocol based on symmetric cryptography.

secure node). The SGTW uses a different key pair for every node. The SGTW and SN exchange their public key (steps 4 and 5) and use it to build two shared secrets (SS_g and SS_{sn}), adding a nonce to the received public key. These secrets are exchanged after encryption using the local private keys (steps 6 and 7). The shared secrets are used to generate the first shared key K_{SH} (step 8). This shared key is used to securely transfer the secret keys starting from $\mathrm{PL_M}$ (the PL of the SN) down to $\mathrm{PL_N}$ ($K^0_{PLM}, \cdots, K^0_{PLN}$ in step 9). At this point, the node holds the secret keys and can start communicating with other nodes on the public network using CMAC signed frames.

Generated keys are valid for a limited time frame. Each PL sets a rolling parameter to decide when it must roll its related key. Whenever the rolling key time of $\mathrm{PL_i}$ expires, the SGTW generates a new key (step 10) and then transmits the new key to all nodes connected to that level using the previous key. The secret key update is not only time-based but can also be event-based. An update can be forced by a specific event, like init, shutdown controller procedure, etc.

TAURUM P2T implements a deprecated key functionality. When the violation of an ECU is detected, the SGTW can mark the related PL secret key as deprecated. Figure 8 shows an example of this mechanism. Starting from a valid condition with several ECUs connected at $\mathrm{PL_3}$ (Figure 8A), the SGTW detects a compromised DEFC module (Figure 8B). The secret key for K^t_{SH} is then marked as deprecated (Figure 8C). All ECUs connected at the same PL or higher are informed and receive a new key K^{t+1}_{SH} encrypted using their K_{SH}. This, in fact, isolates the compromised node on that level through privilege downgrading.

TAURUM P2T also includes a Short Secret Key mode providing each SN with an additional short key (e.g., 16B) to be used in specific conditions. Forcing the secure network

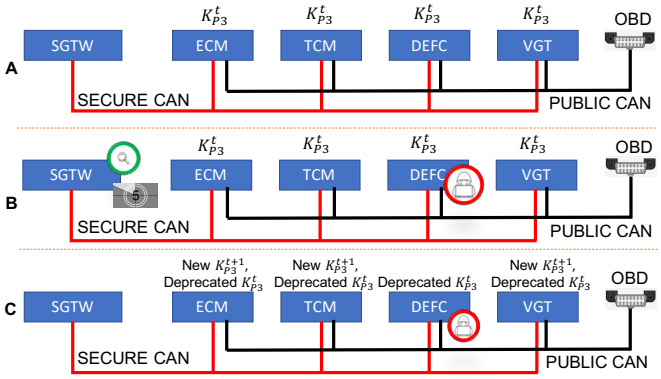


Fig. 8: TAURUM P2T Secret Key Deprecate Status.

to work with digests and keys of 16B allows saving throughput and computation resources. This mode helps to gain extra hardware resources for counterattacking or managing high throughput peaks.

To summarize, Figure 9 shows the set secret keys that every module must handle in a TAURUM P2T architecture. TAURUM P2T centralizes hardware resources into the SGTW, allowing for a more flexible and lighter security resource into the rest of the connected modules. This way, all controllers can implement an essential encryption function with limited storage capacity.

V. Experimental Results

The TAURUM P2T Secure CAN network concept was verified by simulating a real vehicle architecture, including the SGTW connected to two nodes. The implementation was based on the neoVI FIRE 2 Multi-Protocol Vehicle Interface produced by Intrepidcs [15]. The device was configured with

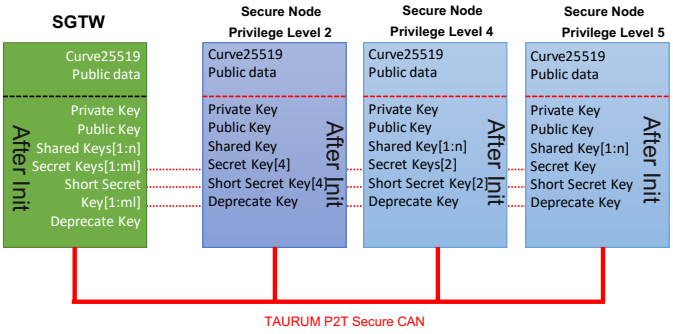


Fig. 9: Summary of TAURUM P2T shared secrets

a CANF-FD baud rate of 2Mbit/s, and TAURUM P2T was configured to manage up to five PLs (Figure 10). The full communication stack was built using Python.

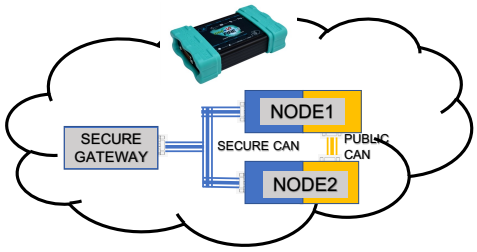


Fig. 10: TAURUM P2T simulation environment set up

The proposed simulation environment was used to experimentally validate the TAURUM P2T architecture, providing interesting information about feasibility and performance introduced by all security mechanisms.

Figure 11 shows the TAURUM P2T throughput overhead introduced by CMAC calculation with different key lengths. As discussed in subsection IV-B, TAURUM P2T introduces a Short Secret Key mode that can be set at run-time in case of need. Figure 11 shows an 11% CMAC computation saving moving from a strong 32B key down to a weaker 16B key. TAURUM P2T could use this mechanism to handle high traffic conditions like the one that could arise from DoS attacks discussed in subsection III-B. CMAC message-digest operations are not symmetric. A pure digest calculation executed by the sender is still 3 times faster than a digest verification in charge of receivers due to more instructions to execute.

Implementing the TAURUM P2T communication stack introduces extra code. Comparing the firmware of one of our sample nodes implemented without any security feature with one implementing the TAURUM P2T communication stack, we measured a 300% code overhead. Nevertheless, in our prototype, all cryptographic operations are software implemented. In real ECUs, the use of Crypto Cores would significantly mitigate this overhead.

As described before, at the very first time, the system executes the key provisioning protocol for sharing and exchanging keys to all the ECUs in the network. This process lasts no more

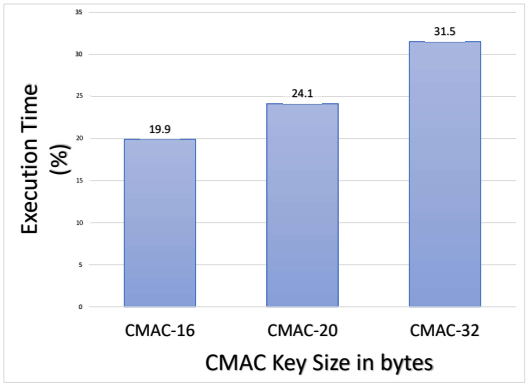


Fig. 11: TAURUM P2T Advanced Secure CAN network throughput trend profile based on HMAC used mode

than 50 ms for sharing the secret keys between a secure gateway and two secure nodes in our experimental implementation. This time strongly depends on the CAN baud-rate settings. A similar amount of time, less than 50ms, is also needed to update secret keys for each privilege level at the end of each rolling period. Being this a broadcast operation, this time is not influenced by the number of nodes. Time measurements are all performed on the prototype implementation of the system. Experimental activities also proved the concept's capability on privilege separation.

Finally, let us discuss the TAURUM P2T impact on the hardware architecture. Most of the ECUs used today in real vehicles are already multi-CAN devices, often with spare channels available. They, therefore, are already able to host two CAN-buses. Therefore, the TAURUM P2T architecture only requires adding the SGTW module and ensuring the Secure CAN cabling. This hardware overhead is mitigated by the complete lack of the IT secret key management infrastructure, with annexed security weakness described above, and impact on management costs.

VI. Conclusion

TAURUM P2T Advanced Secure CAN-FD Architecture for road vehicles fulfills the challenge to increase connected vehicles security level by keeping the cost of security under control and limiting intrusiveness in the production and supply chain. The TAURUM P2T Secure CAN-bus allows reusing with some updates today's hardware. Many devices already have Multi-CAN Controllers on board. The high flexibility combined with high conductibility permits the management of simultaneous CMAC from 128 bits to 256bits to make coexisting high performance with low-performance hardware. All highest crypto hardware resources shall not be distributed among the entire network. A secure gateway centralizes them in a single unit, getting just a central complex system for supervising all network modules made with simpler hardware.

REFERENCES

[1] UN Economic Commission for Europe, "Unece world forum for harmonization of vehicle regulations (wp.29)," 2021. [Online]. Available: https://unece.org/wp29-introduction

[2] ——, "Un regulation no. 155 - cyber security and cyber security management system," 2021. [Online]. Available: https://unece.org/transport/documents/2021/03/standards/un-regulation-no-155-cyber-security-and-cyber-security

[3] ——, "Un regulation no. 156 - software update and software update management system," 2021. [Online]. Available: https://unece.org/transport/documents/2021/03/standards/un-regulation-no-156-software-update-and-software-update

[4] A. Albert *et al.*, "Comparison of event-triggered and time-triggered concepts with regard to distributed control systems," *Embedded world*, vol. 2004, pp. 235–252, 2004.

[5] T. Nguyen, B. M. Cheon, and J. W. Jeon, "Can fd performance analysis for ecu re-programming using the canoe," in *The 18th IEEE International Symposium on Consumer Electronics (ISCE 2014)*, 2014, pp. 1–4.

[6] M. U. Farooq, M. Waseem, A. Khairi, and S. Mazhar, "A critical analysis on the security concerns of internet of things (iot)," *International Journal of Computer Applications*, vol. 111, no. 7, 2015.

[7] Network Working Group, "The aes-cmac algorithm," 2021. [Online]. Available: https://tools.ietf.org/html/rfc4493.html

[8] N. Nowdehi, A. Lautenbach, and T. Olovsson, "In-vehicle can message authentication: An evaluation based on industrial criteria," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017, pp. 1–7.

[9] M. Marchetti and D. Stabili, "Anomaly detection of can bus messages through analysis of id sequences," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1577–1583.

[10] Y. Xiao, H.-H. Chen, R. Wang, and S. Sethi, "Mac security and security overhead analysis in the ieee 802.15.4 wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2006, 04 2006.

[11] M. S. U. Alam, S. Iqbal, M. Zulkernine, and C. Liem, "Securing vehicle ecu communications and stored data," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[12] Network Working Group, "The aes-cbc cipher algorithm and its use with ipsec," 2021. [Online]. Available: https://tools.ietf.org/html/rfc3602

[13] N. Koblitz, A. Menezes, and S. Vanstone, "Guide to elliptic curve cryptography," 2004.

[14] P. Koppermann, F. De Santis, J. Heyszl, and G. Sigl, "Low-latency x25519 hardware implementation: breaking the 100 microseconds barrier," *Microprocessors and Microsystems*, vol. 52, pp. 491–497, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0141933117300273

[15] Intrepid Control Systems, Inc, "neovi fire 2 user guide," 2021. [Online]. Available: URL:https://cdn.intrepidcs.net/guides/neovifire2/neovi_fire2_ug.pdf

[16] C. Lin and A. Sangiovanni-Vincentelli, "Cyber-security for the controller area network (can) communication protocol," in *2012 International Conference on Cyber Security*, 2012, pp. 1–7.

[17] G. Macher, C. Schmittner, O. Veledar, and E. Brenner, *ISO/SAE DIS 21434 Automotive Cybersecurity Standard - In a Nutshell*, 09 2020, pp. 123–135.

[18] G. Macher, H. Sporer, R. Berlach, E. Armengaud, and C. Kreiner, "Sahara: A security-aware hazard and risk analysis method," in *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2015, pp. 621–624.

[19] R. G. Dutta, F. Yu, T. Zhang, Y. Hu, and Y. Jin, "Security for safety: A path toward building trusted autonomous vehicles," in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2018, pp. 1–6.

[20] R. B. Gmbh, "Can with flexible data-rate," 2012.

[21] "Iso/iec/ieee international standard for information technology – telecommunications and information exchange between systems – local and metropolitan area networks – part 1ae: Media access control (mac) security - amendment 1: Galois counter model – advanced encryption standard-256 (gcmaes-256) cipher suite," *ISO/IEC/ IEEE 8802-1AE First edition 2013-12-01 AMENDMENT 1 2015-05-01*, pp. 1–57, 2015.

[22] K. Kang, Y. Baek, S. Lee, and S. H. Son, "Lightweight authentication method for controller area network," in *2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2016, pp. 101–101.

[23] S. Woo, H. J. Jo, I. S. Kim, and D. H. Lee, "A practical security architecture for in-vehicle can-fd," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2248–2261, 2016.

[24] B. Groza and S. Murvay, "Efficient protocols for secure broadcast in controller area networks," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2034–2042, 2013.

[25] R. I. Davis, S. Kollmann, V. Pollex, and F. Slomka, "Controller area network (can) schedulability analysis with fifo queues," in *2011 23rd Euromicro Conference on Real-Time Systems*, 2011, pp. 45–56.

[26] P. Murvay and B. Groza, "Source identification using signal characteristics in controller area networks," *IEEE Signal Processing Letters*, vol. 21, no. 4, pp. 395–399, 2014.

[27] H. Nicanfar and V. C. M. Leung, "Multilayer consensus ecc-based password authenticated key-exchange (mcepak) protocol for smart grid system," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 253–264, 2013.

[28] D. Jang, S. Han, S. Kang, and J. Choi, "Communication channel modeling of controller area network (can)," in *2015 Seventh International Conference on Ubiquitous and Future Networks*, 2015, pp. 86–88.

[29] H. Chen and J. Tian, "Research on the controller area network," in *2009 International Conference on Networking and Digital Society*, vol. 2, 2009, pp. 251–254.

[30] J. Liu, S. Zhang, W. Sun, and Y. Shi, "In-vehicle network attacks and countermeasures: Challenges and future directions," *IEEE Network*, vol. 31, no. 5, pp. 50–58, 2017.

[31] M. Marchetti and D. Stabili, "Anomaly detection of can bus messages through analysis of id sequences," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1577–1583.

[32] Y. Zhang, M. Chen, N. Guizani, D. Wu, and V. C. M. Leung, "Sovcan: Safety-oriented vehicular controller area network," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 94–99, 2017.

[33] M. Barranco, J. Proenza, and L. Almeida, "Quantitative comparison of the error-containment capabilities of a bus and a star topology in can networks," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 3, pp. 802–813, 2011.

[34] P. Martí, A. Camacho, M. Velasco, and M. E. M. Ben Gaid, "Runtime allocation of optional control jobs to a set of can-based networked control systems," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 4, pp. 503–520, 2010.

[35] W. Choi, H. J. Jo, S. Woo, J. Y. Chun, J. Park, and D. H. Lee, "Identifying ecus using inimitable characteristics of signals in controller area networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 4757–4770, 2018.

[36] P. M. Yomsi, D. Bertrand, N. Navet, and R. I. Davis, "Controller area network (can): Response time analysis with offsets," in *2012 9th IEEE International Workshop on Factory Communication Systems*, 2012, pp. 43–52.

[37] C. Lin, Q. Zhu, and A. Sangiovanni-Vincentelli, "Security-aware modeling and efficient mapping for can-based real-time distributed automotive systems," *IEEE Embedded Systems Letters*, vol. 7, no. 1, pp. 11–14, 2015.

[38] B. Groza and P. Murvay, "Efficient intrusion detection with bloom filtering in controller area networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 1037–1051, 2019.

[39] P. Nuzzo, N. Bajaj, M. Masin, D. Kirov, R. Passerone, and A. L. Sangiovanni-Vincentelli, "Optimized selection of reliable and cost-effective safety-critical system architectures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2109–2123, 2020.