Noname manuscript No.
(will be inserted by the editor)

Securing the Controller Area Network with Covert Voltage Channels

Pal-Stefan Murvay · Lucian Popa · Bogdan Groza

Received: date / Accepted: date

Abstract The Controller Area Network (CAN) is the most widely employed communication protocol for in-vehicle applications. While many of its features qualify it as a suitable candidate for future use in automotive networking, the lack of security mechanisms makes it problematic for safety-critical applications. Recently, both the research community and the industry have proposed a large number of solutions for securing CAN, but most of these solutions put additional strain on the already limited 8 byte CAN payload or require more expensive hardware. In this work we propose the use of a covert voltage channel that can be used for the transmission of additional data required by specific security mechanisms. We achieve this with the help of additional transceivers by encoding additional bits as different voltage levels in existing CAN dominant bits without affecting regular CAN traffic decoding. We demonstrate the application of our approach on both low-end and high-end automotive embedded platforms and prove its suitability for implementing authentication mechanisms and key exchange protocols over CAN while maintaining backward compatibility.

Keywords covert channel, security, controller area network, physical layer

1 Introduction

After more than two decades since its first use inside a car, Controller Area Network (CAN) remains the de facto standard for in-vehicle communication. Its reasonable cost and reliability-related features still make it the best choice for many current and future automotive applications. The downside however, comes from the security perspective since it lacks any intrinsic security mechanisms. The possible exploitation of this shortcoming was noted since as early as 2004 [36]. While ignored at first, the prospect of mounting attacks using CAN gave rise to real concerns as a result of the many reported attacks, e.g., extensive security analyses can be found in works such as [25], [7], [28].

To cope with CAN's lack of security, a number of solutions covering objectives such as key establishment [29] or authentication [12] have been proposed by both the academic and industrial sectors. In transmitting additional protocol data these solutions have to cope with the limited 8 byte payload of the CAN frame. One common approach in CAN authentication is to transmit additional frames at the cost of increased busload. Other proposals use truncated Message Authentication Codes (MACs) with the intent to fit the authentication tag in a single frame along with the payload. However, this reduces the space usable for the actual data and may also lead to increased busloads if the space is not sufficient to carry both fields in the same frame. Other lines of work modify the standard CAN behavior to allow the transmission of additional bits of information, at faster rates, within the same frame, such as the case of CAN+ [38]. However, this type of methods require modified versions of the CAN transceiver or protocol controller which bring on increased costs and a longer time-to-market which make them less appealing for the industry sector.

We explore the concept of utilizing covert channels to transmit additional information by piggybacking normally transmitted CAN frames. For this purpose,

Pal-Stefan Murvay, Lucian Popa and Bogdan Groza
The Department of Automation and Applied Informatics, Politehnica University of Timisoara
E-mail: pal-stefan.murvay@aut.upt.ro, lucian.popa.lp@gmail.com, bogdan.groza@aut.upt.ro

the covert channel encodes bits as different bus voltage levels during the transmission of dominant bits of the piggybacked frame. Transmission of covert information during recessive bits is avoided since they might be overwritten by a dominant bit sent over the covert channel due to an intrinsic property of the CAN physical layer. The resulting transmission exhibits variations in the voltage levels of dominant bits which are transparent for reguar CAN controllers but can be interpreted by analog-to-digital converters (ADC) to extract data from the covert channel. The resulting voltage levels are kept within the CAN physical layer specifications which allow correct interpretation of the original frame. This assures backward compatibility with nodes that do not implement this mechanism. While our approach relies mostly on an engineering effort, i.e., by using electrical properties of the bus and embedded devices, by it we try to solve a relevant security problem on in-vehicle networks, which is very hard to address by conventional approaches especially due to the limited payload of CAN frames. Multidisciplinary efforts are not new in the field of information security, e.g., quantum cryptography used laws of quantum physics to leverage secure key distribution, and bringing insights from other fields may prove useful as an alternative or to complement traditional approaches.

Paper contributions. In brief, the main contributions presented in this paper can be summarized along the following lines. We propose a voltage-based covert channel for CAN communication that can be implemented at the application layer with minimal hardware modifications that does not affect normal CAN traffic or bus load. We use the term “covert” in a broader sense, referring to the transfer of information via a channel that is not intended for legitimate communication and which remains transparent to the basic communication stack of the CAN protocol. Indeed, according to CAN standard specifications, the small over-voltage that we create in a natural way by letting multiple transceivers send data at the same time doesn’t affect any of the bus mechanisms (such situations also occur in normal circumstances) and it is neither viewed as a form of information transfer by regular CAN nodes. Of course, the channel remains visible to an adversary and the same holds for more conventional covert channels such as the timing covert channel proposed in [37] which is also readable by any device that has access to the bus from frame arrival times. The readability of the channel has no effect whatsoever on security since we use it only as a transportation layer for cryptographic material that is secure anyway. Nonetheless, as we discuss later, covert timing channels have a reduced data-rate and our voltage channel can achieve a much higher data-rate. We test our approach in a concrete setup and present a proof-of-concept implementation in which CAN authentication data is transported by the covert channel. We also demonstrate the use of voltage-based covert channels for implementing key agreements over CAN. Our proof of concept implementations are done on both low and high-end automotive embedded platforms proving the suitability of the voltage-based covert channel for a wide range of applications.

Our paper is organized as follows. In Section 2 we present CAN basics and existing work on voltage covert channels. Section 3 discusses on how we implement the covert channels and outlines some relevant practical aspects. In Section 4 we present some practical applications and discuss security concerns that stem from the use of voltage channels. Implementation details and our experimental results are contained in Section 5. Finally, Section 6 holds the conclusion and future work.

2 Background and related work

This section presents considerations regarding the CAN bus and existing works dedicated to alternative in-vehicle communication channels and CAN covert channels in particular.

2.1 CAN basics

Since its first introduction in automotive projects by Bosch, CAN was widely adopted in the automotive industry as a de facto standard for in-vehicle communication. The latest protocol version from Bosch, known as CAN 2.0 [31], was standardized as ISO 11898. The main body of the CAN protocol specification, covering the physical and data-link layers is contained in ISO 11898-1 [20] while parts 2 [21] and 3 [22] of the standard cover specific physical behavior for high-speed and low-speed CAN respectively.

While low-speed CAN provides increased fault-tolerance to physical line failures it can only sustain bit rates of up to 125 kbps as opposed to high-speed CAN which is built for communication at speeds up to 1 Mbps. The physical layer characteristics that we exploit to build a covert voltage channel are present in both high- and low-speed CAN. For simplicity, we use high-speed CAN as a base to illustrate the concepts presented in the remainder of this paper.

At the physical layer, CAN is implemented as a two wire differential line. The two wires used for CAN physical signaling, CAN-High and CAN-Low, are generated by the transceiver to encode one of two logical bus states: *dominant* or *recessive*. A recessive state,

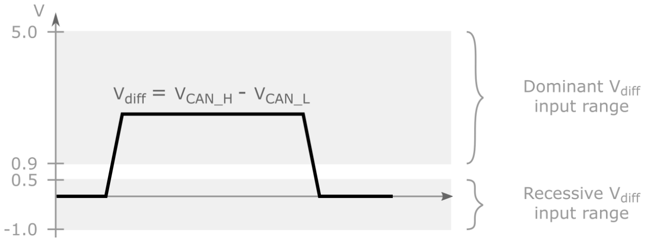


Fig. 1 Recessive and dominant differential input voltage ranges according to ISO 11898-2

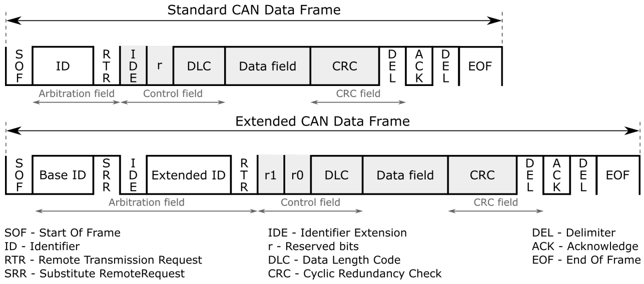


Fig. 2 Standard and extended CAN frames

interpreted as logical “1”, is asserted by the transceiver when the bus is not being actively driven by any node, while the dominant state, interpreted as logical “0”, is the result of at least one node driving the bus. This results in a wired-AND behaviour of the CAN bus. The corresponding differential voltage, i.e., $V_{diff} = V_{CAN\_H} - V_{CAN\_L}$, ranges interpreted as recessive and dominant in high-speed CAN networks are illustrated in Figure 1.

The data-link layer functionality is implemented by the CAN communication controller. The data to be transmitted over the CAN line is encapsulated in a frame along with additional fields as depicted in Figure 2. CAN specifies two types of data frames, i.e. standard and extended frames, the main difference between the two being the size of the identifier field (ID). Other fields, common to both frame types, include the markers for the frame start and end, the size of the payload field, a 15 bit CRC and receive acknowledgement. The data field can hold up to 8 bytes while the size of the ID is 11 bit in standard frames and 29 bit in extended frames.

CAN employs a non-destructive collision avoidance mechanism to cope with situations when two or more nodes begin frame transmissions at the same time. For this purpose, every sender compares the value of each bit he transmits during the arbitration field (marked in Figure 2) with the bit resulting from the actual bus voltage value. Senders immediately stop transmitting when reading back a dominant bit after transmitting a recessive bit. The ID field serves as a means to prioritize frames and thus we avoid sending covert bits during this part of the frame, we only send covert information during frame fields following the arbitration.

Two more error-control mechanisms are worth mentioning. As part of the CAN error detection mechanism, it is required that all nodes acknowledge the reception of transmitted frames. This is achieved through the ACK bit. This bit is always set to recessive by the sender while all receivers generate a dominant bus level for its duration if the frame is correctly received. Another mechanism intended to alleviate errors caused by potential synchronization involves bit stuffing, by which additional bits are inserted in the frame before transmitting it. Concretely, a bit of opposite value is introduced after any 5 consecutive bits of the same value.

2.2 Related work

Concerns for the ever-increasing complexity and size of automotive wiring harnesses have led to the research of various alternatives to adding more wires when new communication channels are required. One such alternative is to use power-line communication. Other proposals involve implementing covert channels over conventional communication channels that could be exploited with either good or malicious intent. We discuss such proposals next.

Power-line communication. Several research works have considered using the in-vehicle power-lines as a communication channel. To this end the authors of [16] present a mathematical model and construct a hardware emulator for automotive power-line communication (PLC) channels. A medium access control protocol designed for automotive PLCs was proposed in [1]. This proposal provides efficient contention detection and resolution for multiple access to the channel. However, since power lines are noisier than conventional communication lines, using them for communication may prove less appealing in the presence of multiple well established in-vehicle communication protocols. The adoption of technologies new to automotive communication is a lengthy process (e.g., the case of automotive Ethernet) whereas solutions based on already proven technologies are more easily adopted (e.g., the CAN-FD extension of the CAN protocol). Consequently, steps were made towards porting the physical layer of existing automotive in-vehicle protocols from dedicated lines to PLC. Existing proposals cover protocols such as LIN (Local Interconnect Network) [27], CAN [24], FlexRay[34] and Ethernet [33].

Covert channels over conventional channels. While there is extensive literature on creating covert channels in computer networks, e.g., [4], [3], [26], there are only a few recent approaches that address covert channels on the CAN bus. For example, a timing-based covert channel is proposed in [14] for carrying cryptographic authentication data. A similar approach is considered by the authors of [37] which implement two timing-based covert channels, one based on inter-arrival times of CAN frames and another using clock offsets of transmitting nodes. The third option considered in [37] for implementing a covert channel is to embed authentication data in the least significant bit of several CAN frames with the aim to authenticate transmitters rather than messages given the low capacity of this covert channel variant. The work done in [15] brings improvements of the previous work from [14] in what regards the throughput achievable over a timing covert channel with the capability to transmit 3-5 covert bits per frame. However, the data-rate of these approaches (based on timing information) is very low, up to several hundred bits per second (bps) while a voltage covert channel may achieve a much higher data-rate as we later discuss in the experimental section. The particular case of covert key sharing over CAN is treated in [13]. Several approaches of achieving this using CAN's non-destructive arbitration mechanism and time-delays are presented.

Other works consider the use of covert channels over existing wireless mediums. One such article presents the use of FM Radio Data System for transmitting covert data to an in-vehicle Android-based radio receiver for malware deployment [10]. The presented approach requires that a malicious app, capable of receiving and acting on covert commands, is installed on the target Android system. Another use case targets the establishment of covert channels in vehicular ad-hoc networks. The authors of [19] present the use of corrupted beacon messages for covert communication to achieve identity exchange while preserving user privacy.

Out-of-band channels. Other lines of work consider the use out-of-band channels such as sound or light for various security applications. One such proposal introduces a mechanism for mobile-to-vehicle pairing based on the use of light and sound as out-of-band channels [17]. An optical laser based channel is proposed in [9] for achieving non-forwardable authentication between vehicles. Another line of work [32], also focused on vehicle to vehicle communication, uses both optical and acoustic side channels to build a session key agreement mechanism.

3 Voltage-based covert channel

This section describes the procedures for creating the voltage-based covert channel over CAN and outlines important aspects for its implementation. The conceptual setup is summarized in Figure 3. We defer the discussion on the adversary model for Section 4 in order to place it in the context of our specific covert channel implementation.

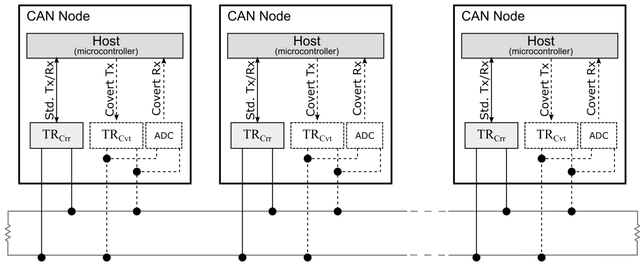


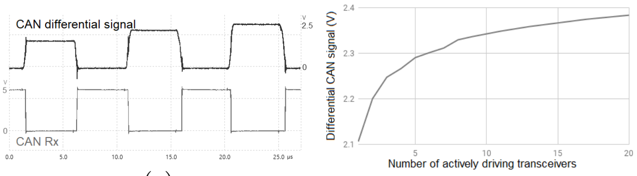
Fig. 3 CAN covert channel implementation concept

3.1 Manipulating CAN bus levels

Building a covert channel based on the bus levels of the CAN lines requires the ability to encode additional bits of information within normal CAN transmissions without affecting the correct interpretation of standard CAN traffic. The differential voltage range that defines the two CAN logical symbols (recessive and dominant) as specified by the CAN specification [21] are depicted in Figure 1. The thresholds are designed for coping with voltage variations caused by elements such as noise, ground shift or the case when multiple nodes are driving the bus at the same time, e.g., during arbitration. We exploit this feature to encode additional information as different voltage levels within the range specified by the CAN standard.

While the use of dedicated HW (hardware) would be preferable for achieving this functionality providing such a HW design is out of focus for our work. Instead, since the time to market for a dedicated integrated circuit would delay its availability for new applications we focus on providing an immediately available solution. We do this by using the signals generated by off-the-shelf CAN transceivers as we detail next. Since recessive levels are achieved when none of the network nodes is actively driving the bus, due to the weakly biased bus, we rely on dominant bits alone as carriers for covert information.

Concurent transmissions. The accepted input differential voltage for the dominant symbol ranges between 0.9 and 5V while the typical voltage level generated by a transceiver outputing a dominant level is situated arround the 2V value [21]. When multiple transceivers are driving the bus simultaneously (i.e. simultaneous generation of dominant levels) the resulting differential bus level increases slightly with each additional driving



(a) (b)

Fig. 4 Bus voltage levels resulting from simultaneous transmissions: (a) Example of bus voltage and received bit values for 1 to 3 simultaneous CAN transmissions of dominant bits, (b) Effect of the number of concurrent actively driving transceivers on dominant voltage level

transceiver. This is caused by the output transistors of driving transceivers being in parallel which results in a smaller voltage drop over each transistor and, consequently, leads to a greater differential voltage. While the CAN specification [20] sets no specific limits for the number of nodes that can simultaneously generate dominant levels, the maximum number of nodes on a CAN bus is limited by the electrical characteristics of the employed transceivers. The resulting differential bus signal remains within the specified limits regardless of the number of nodes simultaneously driving the bus as long as all the bus nodes operate within specified conditions.

In Figure 4 (a) we depict how the resulting differential voltage and the value decoded by receivers are affected by up to three CAN nodes, each using the same transceiver chip, i.e., the MC33742 system basis chip with integrated high speed CAN transceiver, when they are simultaneously driving the bus. All nodes start by generating recessive levels followed by one, then two and finally all three generating dominant bits at the same time along with additional recessive bits as separators between dominant levels. While the recessive levels do not exhibit variations, the dominant levels display an increasing differential voltage in accordance to the number of actively driving transceivers. In this case we measured an increase of 680mV in the differential bus voltage after activating the second driving transceiver and an additional 380mV when the third one was also set to generate a dominant level. Since the resulting dominant differential voltage is kept within the specification ranges it is always correctly interpreted by the receiver as being dominant (i.e. logical 0). The voltage increase generated by additional driving transceivers depends on supply voltage and transceiver characteristics. The relative voltage increase also decreases with each additional actively driving transceiver as illustrated by the plot in Figure 4 (b) generated by simulating a network with up to 20 nodes simultaneously outputting dominant levels.

3.2 Bit encoding

Covert transmissions are encoded in the voltage levels of dominant bits of regular CAN frames which we call *carrier frames*. We distinguish between two possible cases depending on whether the content of the carrier frame is known or not in advance to the sender node. In what follows, we discuss general aspects of bit encoding based on the assumption that the carrier frame content is known a priori. We dedicate a separate section for examining the particular case of carrier frames with unknown content.

System requirements. We consider that each node that transmits data over the covert voltage channel is equipped with a CAN transceiver, denoted as $\mathrm{TR}_{\mathrm{Cvt}}$, dedicated to covert communication. This is in addition to the regular transceiver required for standard CAN communication used as carrier which we denote as $\mathrm{TR}_{\mathrm{Crr}}$. As illustrated in Fig. 3, $\mathrm{TR}_{\mathrm{Cvt}}$ is connected to the same CAN bus as the corresponding $\mathrm{TR}_{\mathrm{Crr}}$. $\mathrm{TR}_{\mathrm{Crr}}$ is controlled by a conventional CAN communication controller while $\mathrm{TR}_{\mathrm{Cvt}}$ is controlled directly by the application layer or through specialized pulse generation circuitry on the microcontroller (e.g., timer modules). This setup is required both for nodes that generate the carrier frame for their own covert transmission and nodes that use a carrier sent by another node. In the first case $\mathrm{TR}_{\mathrm{Crr}}$ is used to transmit the carrier frame which establishes the baseline dominant voltage level for the covert channel while $\mathrm{TR}_{\mathrm{Cvt}}$ is used to generate the covert bits. In the second case, $\mathrm{TR}_{\mathrm{Crr}}$ is required for standard CAN receive operations (i.e. error detection and acknowledgement) while $\mathrm{TR}_{\mathrm{Cvt}}$ is dedicated solely for covert transmissions. Achieving both standard CAN frame receive operations and covert channel transmissions in an efficient manner while using a single transceiver would require either specialized circuitry or the use of a software-defined CAN controller [6] for carrier frame decoding. While preferable, the use of specialized circuitry increases costs and time-to-market. The software-defined CAN controller approach limits the achievable communication bit-rate as reported in [6]. Therefore, we consider that the use of two transceivers would be a more suitable approach and provide experimental data as proof for its feasibility.

Encoding approach. Covert channel bits are encoded using the same convention as employed by the basic CAN protocol. A logical 1 (recessive) covert bit is generated by outputting a recessive signal using the transmitter's $\mathrm{TR}_{\mathrm{Cvt}}$ and is seen on the bus as the normal dominant voltage of the carrier frame. The logical 0 (dominant) covert level is implemented by setting the $\mathrm{TR}_{\mathrm{Cvt}}$ to output a dominant level and is seen on the

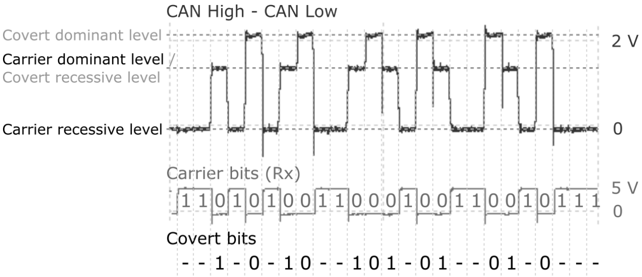
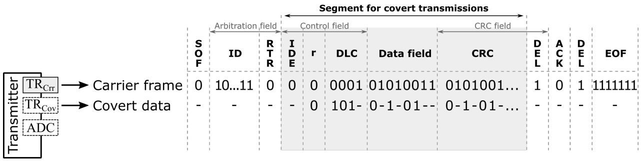


Fig. 5 Physical layer signaling resulted when encoding the '101010101010' bit sequence on a random carrier signal



	SOF	ID	RTR	IDE	r	DLC	Data field	CRC	DEL	ACK	DEL	EOF
Carrier frame	0	10...11	0	0	0	0001	01010011	0101001...	1	0	1	1111111
Covert data	-	-	-	-	0	101-	0-1-01--	0-1-01-...	-	-	-	-

Fig. 6 Encoding six sequences of '01' bits on a random carrier frame

bus as a higher dominant voltage level resulted from a TR_{Cvt} and a TR_{Crr} simultaneously driving the bus. The example in Fig. 5 illustrates the signals generated at the physical layer by a node encoding a two bit sequence, i.e. '10', repeated six times, as a covert transmission over a random carrier. Twelve dominant bits of the carrier frame are required to encode this sequence since covert bits can only be encoded over dominant carrier bits. Therefore, all recessive bits in the carrier frame are ignored for covert channel encoding as indicated by dashes in Fig. 5. All bits in the covert transmission correspond to dominant carrier bits with covert '1' bits exhibiting a normal dominant voltage level while all '0' bits correspond to an increased dominant voltage level.

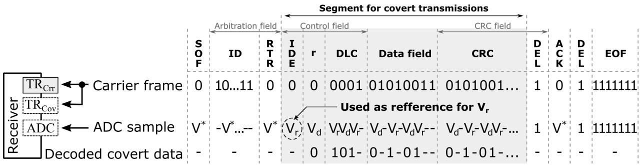
Location within carrier frame. Reliable utilization of the covert channel requires the prevention of unwanted concurrent transmissions during covert bits. By design, the CAN protocol allows concurrent bit transmissions from two or more nodes during the arbitration and acknowledgement field as explained in section 2.1. As a consequence, covert transmissions should only be encoded during frame segments in which only the carrier frame sender is allowed to affect bus signaling in accordance to the standard CAN specification [21]. Therefore, covert bits are not encoded during the start of frame, arbitration field and acknowledge bit since multiple nodes are allowed to drive the bus during these fields. The remaining usable fields are the control, data and CRC fields (grey fields depicted in Fig. 2 and 6) while the remaining trailing frame bits are unusable since they are always set to recessive. The first dominant bit of the control field (IDE in standard frames or r1 in extended frames) is not used for covert bit encoding to provide decoding nodes with a reference value for the carrier frame dominant voltage level (corresponding to the covert channel logical 1). Fig. 6 illustrates the encoding process for the covert transmission of a sequence of 2 bits (i.e., '01'), repeated six times over a random carrier frame. As expected, the carrier transmission, sent trough TR_{Crr}, occupies all the frame fields while the covert transmission, sent via TR_{Cvt}, only uses the dominant bits in the designated frame fields (marked with grey background).

3.3 Bit decoding

System requirements. Since regular CAN transceivers are only able to decode the carrier frame, an ADC (analog-to-digital converter) is required for sampling the voltage levels representing the covert bits directly from the CAN high and CAN low lines. Two analog to digital conversions are required for each covert bit to capture both the CAN high and CAN low lines if the differential voltage is used for decoding. However, using only one of the two bus lines and hence, a single conversion per encoded bit is sufficient to detect the different voltage levels. Oscilloscope protocol decoders commonly provide this one-wire CAN decoding feature [35, 30]. We prove the feasibility of this approach by implementing it on our experimental platforms.

Decoding approach. Sampling during recessive levels of the carrier is not required since they do not carry any covert information. During dominant bits, the minimum ADC sampling rate should be twice the employed bit rate according to the Nyquist rate. However, this can be reduced to one sample per bit since the covert channel bit rate as well as the start and end of dominant carrier levels are known to receivers. While the covert channel bit rate is established at system setup, senders can identify dominant carrier bits by using the Rx line of TR_{Cvt}. Therefore, one sample per covert bit time is sufficient. To account for consecutive dominant bits, ADC sampling is triggered by the detection of a dominant level on the bus and is repeated until the bus signal transitions back to recessive. Loss of synchronization with the carrier transmission, causing misaligned conversions, is avoided since no more than 5 consecutive dominant bits can be found in the carrier frame. This is due to the CAN bit stuffing mechanism which automatically introduces a bit of opposite value after any 5 consecutive bits of the same value.

Fig. 7 illustrates the decoding process for the covert transmission depicted in Fig. 6. ADC sampling starts with the first dominant bit in the frame (i.e., the SOF bit) and continues during all dominant carrier bits until the complete frame is received. All samples are buffered



	SOF	ID	RTR	IDE	r	DLC	Data field	CRC	DEL	ACK	DEL	EOF
Carrier frame	0	10...11	0	0	0	0001	01010011	0101001...	1	0	1	1111111
ADC sample	V^*	$-V^*$...--	V^*	V_r	V_d	$V_rV_dV_r$-	V_d-V_r-V_dV_r--	V_d-V_r-V_dV_r-...	1	V^*	1	1111111
Decoded covert data	-	-	-	-	0	101-	0-1-01--	0-1-01-...	-	-	-	-

Fig. 7 Decoding six sequences of '01' bits on a random carrier frame

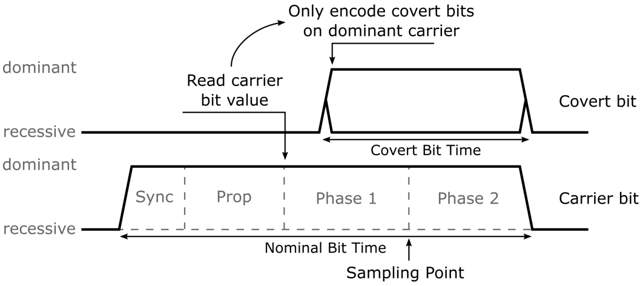


Fig. 8 Encoding covert bits in carrier signal of a priori unknown value

and processed together with the carrier frame received through $\mathrm{TR}_{\mathrm{Crr}}$ and $\mathrm{TR}_{\mathrm{Cvt}}$. The sampled voltages, denoted as V^*, covering the frame preamble are discarded since they are not used to encode covert data as explained in the previous section. The frame preamble can be used for synchronization purposes between the covert data sender and receiver or between multiple covert data transmitters. The acknowledge bit is also dropped since it is the result of other nodes acknowledging the received frame. The first dominant bit following the arbitration field is used to establish a reference voltage level corresponding do recessive (logical 1) covert bits, denoted as V_r. All subsequent ADC samples that match this reference within specified tolerance margins are also marked as V_r and considered as recessive covert bits. All other samples exceeding the reference level are considered as dominant covert bits (logical 0) and denoted as V_d. The covert bit sequence is then reconstructed based on this classification.

3.4 Frames of unknown content as carriers

As already stated, while transmitting covert bits, a node must ensure that it does so only during the dominant bits of the carrier frame to avoid interfering with normal CAN traffic. Achieving this is made more difficult if the content of the carrier frame is not known before its transmission. To make covert transmissions possible, the value of each carrier bit has to be determined before generating the covert voltage levels.

The nominal CAN bit, as depicted in the lower half of Figure 8, is composed of 4 segments: synchronization segment, propagation segment and two phase segments. This structure, devised to support node synchronization during CAN transmissions, accounts for propagation delays. The actual bit sampling occurs between the two phase segments. Limits on bus length are imposed depending on the employed bit rate so that the time required for the propagation of a CAN signal over the longest path does not exceed 25% of the nominal bit time. Therefore, a node transmitting covert data must check the value of the current carrier frame bit right after the propagation has ended. If the carrier bit is dominant, the covert bit signaling can be activated for the remainder of the carrier bit duration. This process is suggested in the upper half of Fig. 8.

While the technique may seem straightforward, implementing it poses a series of challenges. The transmitting node has to be able to check the carrier bit value and generate the covert signal without exceeding the duration of the carrier bit (e.g., 2µs for a 500kbps bit rate), while maintaining synchronization with the carrier transmission. Moreover, the transmitter node has to perform a bit-by-bit decoding of the frame to determine the segments that can support covert transmissions (as explained previously). Finally, the receiver of covert data should be able to decode the covert bits which may be shorter in duration than the carrier bit. Depending on target communication bit rates and platform capabilities some platforms may not be able to implement the covert channel functionality. Consequently, this use case will require more expensive electronics and thus we leave it only as a possibility while focusing on the first use case (using known carrier frames) which is more relevant from a security perspective.

3.5 Channel capacity

The capacity of a voltage based covert channel depends on a number of factors such as: number of covert bits encoded per carrier bit, the carrier bit rate, the number of dominant bits in the carrier frame and the sampling rate/time of the ADC. We now discuss on channel capacity.

The first limitation of channel capacity comes from the maximum amplitude of the voltage that can be encoded, divided by the resolution of the ADC. This would lead to the well-known Shannon-Hartley rate of $f \log_2(1 + A/\varDelta V)$ where f is the pulse rate of the channel, A is the maximum amplitude and $\varDelta V$ is the resolution of the ADC. However, it would be out of scope for our work to push the voltage on the CAN bus to the maximum acceptable level as this may cause errors and we want to keep CAN communication undisturbed. For this reason, we prefer to keep the signaing to a single bit per sample, i.e., $A = \varDelta V$ which means that the

ADC can distinguish between the case when there is a 0 or 1 on the covert channel.

This would further limit the capacity to f, the pulse rate, which in our case can be approximated to the average number of dominant bits (zeros) that are sent per second (only dominant bits in the data-field are to be considered). While one may see recessive bits (ones) as problematic and in fact some frames may indeed carry a large number of ones, there is an easy procedure to circumvent this problem. The content of the frame can be XOR-ed with a random (non-secret) value which will assure a uniform distribution of zeros and ones. In fact, such mechanisms which modify the content of the frame to avoid stuffing bits that come with consecutive ones and zeros have been previously proposed in [5]. To compute the random, non-secret value, one can simply map the identifier of the frame via a pseudo-random number generator (PRNG) to a 64 bit value that is XOR-ed with the datafield. This will assure an average of 32 bits set to zero inside each data frame. Given that each CAN frame carries about 50% overhead besides the data-field (because of IDs, CRC, stuffing bits, etc.) and that around half of the bits of the datafield may carry one bit, the data rate of the covert channel will be around 25% of the regular bus rate. This happens by encoding a single covert bit in one dominant bit and higher bit-rates are achievable if more than a single voltage level is used. This would be however out of scope for the current work.

3.6 Network topology considerations

While CAN is most commonly implemented based on a bus/line topology, specific system requirements, such as traffic segregation, have led to the implementation of CAN networks using other basic or hybrid topologies. The ability to implement the proposed covert voltage channel between a set of network nodes depends on the ability to relay both the carrier frame and the covert information to all nodes involved in the covert data exchange. Achieving this when the nodes share the same bus is straightforward as the signals produced by a transmitter are available to all nodes physically connected to the same CAN line.

To implement this covert channel between nodes located in different sub-networks, i.e., nodes that do not use the same physical bus, we need gateway ECUs (that connect such sub-networks) to be able to replicate not only the data frames but also the covert information for all the receiver nodes. If the covert channel is used to exchange additional authentication data, i.e., the first use case illustrated in the next section, there are no additional technical challenges for implementing this mechanism on gateway ECUs since they can also be equipped with the required additional CAN transceivers. Usually, gateway ECUs have increased computational and communication capabilities in comparison to regular ECUs, making the deployment even easier. For covert key-exchange, i.e., the second use case outlined in the next section, the gateway can no longer relay the information since this key exchange requires direct participation of the two end-points in order to mask the exchange bits. To facilitate a key sharing between distinct parts of the network, the gateway can separately engage in the key exchange with each node from a distinct sub-network and further use this secret shared key to establish a common secret key between nodes in distinct sub-networks. This can be achieved by regular cryptographic authentication protocols and the technicalities behind this are out of scope for the current work.

4 Covert channel applications

In this section we first discuss some applications that are exemplary for the proposed methodology and present experimental results.

4.1 Exemplary applications: covert authentication and covert key-exchange

We discuss the two main applications that we target with the proposed methodology.

The first application is the most of obvious and it consists in *authenticating data frames*. As already stated, a significant number of works present various solutions for providing authenticated communication over CAN. But the common disadvantage of these works stems from the additional authentication payload that has to be transmitted along with the frame. The usually employed approach is to send this information in the payload of normal CAN frames either as part of the same frame as the authenticated data or as separate frames increasing the bus load. This disadvantage can be alleviated by transmitting authentication data over covert channels. As discussed in the previous section, assuming a uniform distribution of zeros and ones in frames, and since only dominant bits may carry covert data, each frame will carry around 32 authentication bits. Consequently the security level that is achievable is around 32-bits which is even better than current specifications as the AUTOSAR standard for secure on-board comunication prescribes only 24 bits of security [2].

The second application that we propose consists in *covert key-exchange*. It is obvious that classical key

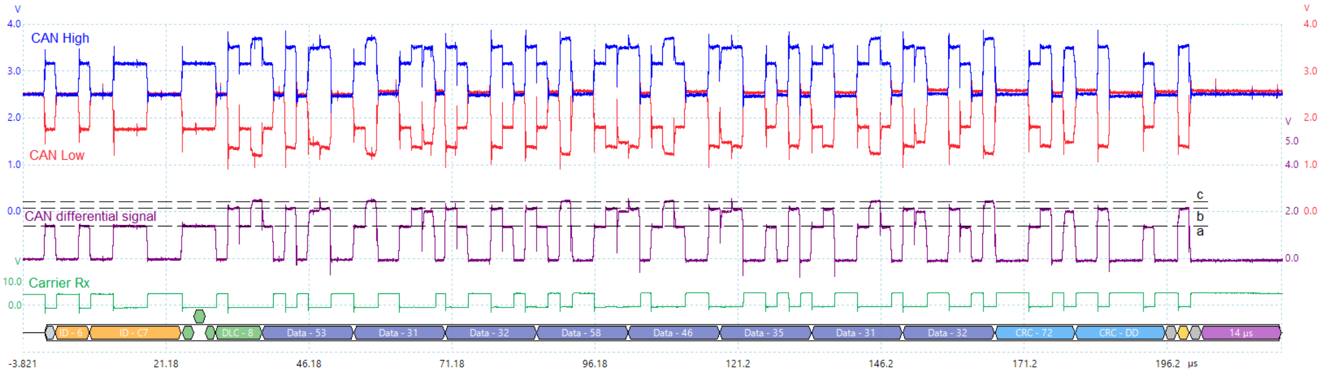


Fig. 9 Covert channel key exchange at 500kbps on the S12xF platform: a - signal threshold when both covert bits are recessive, b - signal threshold when one covert bit is recessive while the second is dominant c - signal threshold when both covert bits are dominant

establishment protocols (e.g., the Diffie-Hellman key-exchange) can benefit from using covert channel transmissions to transmit protocol-related messages in data-frames that otherwise carry regular data. Another approach however, is the key agreement protocol proposed by Mueller and Lothspeich [29] which exploits the wired-AND behavior of CAN's physical layer. When two CAN nodes transmit bits at the same time, the resulting bus signals correspond to the result of performing the AND operation on the two transmitted sequences. Mueller and Lothspeich's method [29] by which two nodes agree upon a common key consists of the following steps. First, each of the nodes generates an array of random bits of a given length. This array is then modified so that for each bit in the initial array a new bit is added representing the negated original bit. Both nodes then simultaneously send their bit sequences over CAN and read back the resulting sequence. The received sequence is used by each node to identify the original bit sequence of its counterpart. This is the reason for which sending the inverse of the original sequence is also required, otherwise a node sending a 0 (dominant) cannot identify the value sent by its counterpart (the result on the bus will be dominant regardless of the value of the bit transmitted by the second node). Those bits from the initial sequence that are leaked to eavesdroppers have to be dropped since they are compromised. Bits are leaked when the nodes simultaneously transmit a bit that is 1 (either in the original or the complement sequence). The resulting bus level in this case is 1 (recessive), which makes it obvious to any eavesdropper that both nodes sent a 1 (due to the wired-AND nature of CAN). As a result of these steps, each node can extract the bit sequence sent by its counterpart which represents the inverse of its own sequence. The non-leaked bits (roughly 50% of the transmission) for the secret key, see [29] for more details.

The disadvantage of the method as proposed by Mueller and Lothspeich in [29] is that it requires modified CAN controllers and cannot be deployed in standard CAN implementations. This is because the resulting traffic cannot be decoded by other nodes as valid frames which will lead to the transmission of error frames. By using covert channel transmissions this shortcoming can be alleviated since the resulting traffic will be interpreted as a correct frame (i.e. the carrier frame). Moreover, due to the difference in the dominant voltage levels when one or two nodes are simultaneously transmitting dominant bits, there is no need to transmit the complement of the original bit sequence. This reduces the required bit transmissions to half, when compared to the original protocol proposed by Mueller and Lothspeich.

For implementing Mueller and Lothspeich's key agreement over the voltage-based covert channel the nodes involved need to have a priori knowledge of the content of the carrier frame. The node designated to be the transmitter of the carrier frame assumes the role of the key agreement initiator. The initiator begins the process by starting the carrier frame transmission. The second node synchronizes with the carrier by using bits in the frame preamble up to the end of the arbitration field. Alternatively a security designated node can initiate the process as the carrier transmitter, with the other two nodes synchronizing to this frame. Starting with the second dominant bit following the arbitration field both nodes simultaneously transmit their random bit sequences over the covert channel and read the bus levels using the ADC. Four voltage levels will be generated on the bus corresponding to four cases: (i) carrier and covert bits are recessive, (ii) carrier bit is dominant while covert bits are recessive, (iii) carrier bit is dominant and only one node generates a dominant covert bit and (iv) carrier bit is dominant and both nodes generate a dominant covert bit.

Once the transmission sequence has ended, each node decodes the received sequence and identifies the bits sent by its counterpart. Bits generated by both nodes transmitting the same covert bit value are leaked since they generate distinctive voltage levels. These leaked bits are dropped leaving each node with a bit sequence representing its counterpart non-leaked sequence. Figure 9 illustrates the signals generated during the key exchange procedure over the covert channel with markers for the distinct voltage levels.

While in the case of frame authentication the covert channel capacity will vary according to the frame content, when implementing the key agreement mechanism the channel capacity can be fixed and maximized by design if no actual meaningful information has to be transmitted in the carrier frame. The maximum number of dominant bits can be obtained in a frame with and 8 byte payload consisting of all 0 bits. If needed, a specific frame ID can be chosen to maximize the number of dominant bits in the frame CRC, but the 64 bit payload is sufficient for negotiating a 64-bit key and more frames can be used if a larger key is needed.

4.2 Security discussion

Adversary model. Our work is focused on the design of the covert channel while for the security of the data that is transmitted on it we rely on basic cryptographic building blocks. Still, in our setup, we do assume the existence of a generic adversary which has access to the CAN bus that carries both normal and covert transmissions. We consider that the adversary gains access to the CAN bus to eavesdrop on the existing communication and potentially mount spoofing, replay or DoS attacks. This can be achieved either by compromising an existing network node (e.g., by direct access to the vehicle interfaces [25], [18] or even remotely [28]) or by infiltrating a new node on the bus (e.g., through a physically accessible network point such as the diagnosis port or cables accessible under the hood).

The adversary has knowledge of the covert channels employed and has the ability to receive and generate covert transmissions. We assume that the attacker has no access to any keys stored on any of the legit nodes or the compromised node.

Resilience to specific attacks should be achieved by proper cryptographic designs, we discuss next only concerns that arise from the specific use of voltage channels.

Denial-of-Service. It is a well known fact that CAN is vulnerable to Denial-of-Service (DoS) attacks. This CAN vulnerability was among the first to be reported [36] and is caused by intrinsic protocol characteristic such as the wired-AND behavior of the physical layer and the arbitration mechanism. Standard CAN DoS attacks reported in literature will not prevent covert communication. These attacks consist in either generating a continuous dominant level on the bus or continuously transmitting messages with the highest priority. Voltage-based covert communication can be still employed by using the attacker transmissions as carrier signals while timing-based covert communication is prevented by such attacks. In our experiments, we were successful in achieving covert transmissions while an attacker node maintains a constant dominant level on the bus. A DoS attack targeting the covert communication should thwart the ability to decode transmissions. For this an attacker should reduce the voltage increase caused by covert transmissions to a level below the detection threshold. As shown in Figure 4 (b), this can be achieved by generating simultaneous transmissions from multiple transceivers which requires the attacker to either place multiple transceivers on the bus or gain control of a sufficient number of existing nodes. Alternatively, the attacker could superimpose its own covert transmission over the legit one with the intent of impede correct decoding of the covert transmission. However, as shown in section 3.4 achieving this without dedicated HW would be difficult. If successful, such an attack can be detected by both the sender and receiver by analyzing the generated dominant voltage level.

Physical probing. The basic key-exchange mechanism described by Mueller and Lothspeich [29] is vulnerable to a series of physical layer probing attacks as demonstrated in [23]. These attacks consist in identifying original bit values and their transmitter based on minute differences in signal form, voltage level and bit timing. Achieving these attacks requires dedicated sampling circuitry as those found in oscilloscopes. The generated key can be computed once the attacker obtains the bit sequence transmitted by each node. By using the covert voltage channel to implement the key exchange the features of the resulting signal are influenced by both the carrier signal and the covert signaling. This makes the process of identifying the source of individual covert transmissions more complex but not impossible by means of a device with a high enough sampling rate and vertical resolution (e.g., an oscilloscope). Fortunately, the same article that proves the vulnerabilities [23] also proposes solutions which are applicable to the case of a voltage-based covert channel. Time-related features can be masked by introducing small random variations in the bit start and duration timings which could be generated from the application layer. Similarly, features related to voltage and signal characteris-

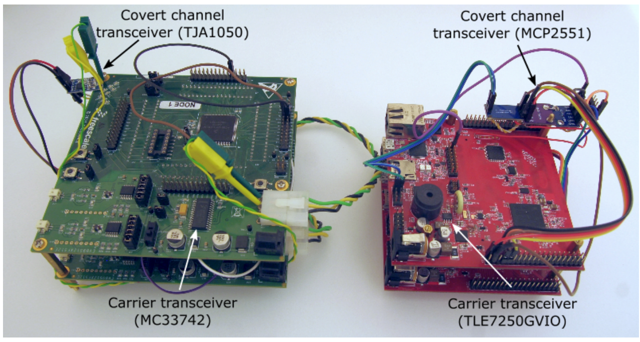

Covert channel transceiver (TJA1050)
Covert channel transceiver (MCP2551)
Carrier transceiver (MC33742)
Carrier transceiver (TLE7250GVIO)

Fig. 10 Experimental setup employed to implement covert channel applications (Aurix-based development boards on the right, S12X-based development boards on the left)

tics can be masked by random variations of the voltage and characteristics of generated dominant signal levels. This would indeed require specialized transceivers or off-the-shelf transceivers combined with additional HW. For example, in our experiments we were able to influence the output voltage of transceiver dominant levels through small variations in the transceiver supply voltage.

Malicious intents. Covert channels may also be used with malicious intents. Indeed, we should note that adversary nodes can take advantage of such covert channel to exchange information aiding them in launching a complex coordinated attack. Our work however, explores such channels only in a constructive sense. Preventing covert channel communications for other purposes than the ones prescribed by system design could be done by implementing an intrusion detection system (IDS). Since the communication model employed over the covert channel is defined at design time, the IDS can monitor covert communication and report any transmission that does not fit the prescribed model as misuse. Alternatively, the IDS can prevent the successful reception of the malicious message by superimposing random covert bits over the original transmission making it impossible to correctly interpret the covert message. The IDS functionality can be implemented either on a central gateway node connected to all network sub-buses or on separate nodes each dedicated to a subbus.

5 Implementation details and evaluation results

For testing the ability of different microcontroller classes to implement the voltage-based covert channel mechanism we selected two classes of embedded devices. The first is a member of the NXP S12XF family covering the low to mid end of the market, while the second is from the Infineon Aurix family and is destined for high performance applications. The two resulting experimental platforms are depicted in Figure 10.

5.1 NXP S12XF

We employed EVB9S12XF512E development boards each equipped with an NXP S12XF512 microcontroller and an MC33742 system basys chip providing the CAN transceivers used to implement the carrier line communication. The S12XF512 microcontroller is equipped with 32 kbytes of RAM, 512 kbytes of FLASH and can support operating frequencies of up to 50MHz for peripheral modules and 100MHz for the CPU. For implementing the covert channel we attached an additional external transceiver circuit (TJA1050) to each development board.

Encoding covert information. The transmitter is either tasked with sending both the carrier frame and the covert information has a priori knowledge of the content of the carrier frame. For encoding covert bits, the transmitter first prepares the bit sequence to be transmitted over the covert channel based on the content of the covert frame. First, all the bit positions corresponding to dominant bits of frame fields unusable for encoding as well as to all recessive bits are set to 1 corresponding to recessive levels during covert transmission. The actual covert information is then mapped on the remaining dominant bits.

The transmission of the generated bit sequence is implemented by using the microcontroller timer output compare functionality. The timer output compare channel is configured to change output values on the associated digital pin at the same rate as the covert channel bit rate. At the start of each covert bit time the pin state is set in accordance to the corresponding position in the pre-generated bit sequence. A 0 in the sequence requires the pin to be set to low while a 1 requires it to be set to high. The output compare pin is connected to the Tx pin of the transceiver used for covert transmissions. Figure 11 illustrates the bus signals generated when encoding a repeating sequence of '01' bits in a CAN frame transmitted at a bit rate of 500kbit/s. A one-to-one covert to carrier bit ratio was used. The encoded dominant bits are clearly visible as greater variations in the dominant levels. The carrier frame is correctly decoded as illustrated by the transceiver Rx line and the oscilloscope protocol decoder.

The trigger for initiating the covert information transmission is given by the SoF bit of the carrier frame. The covert channel transmission ends once the entire bit sequence was processed.

Decoding covert information. The ADC module of the S12XF microcontroller is employed for decoding

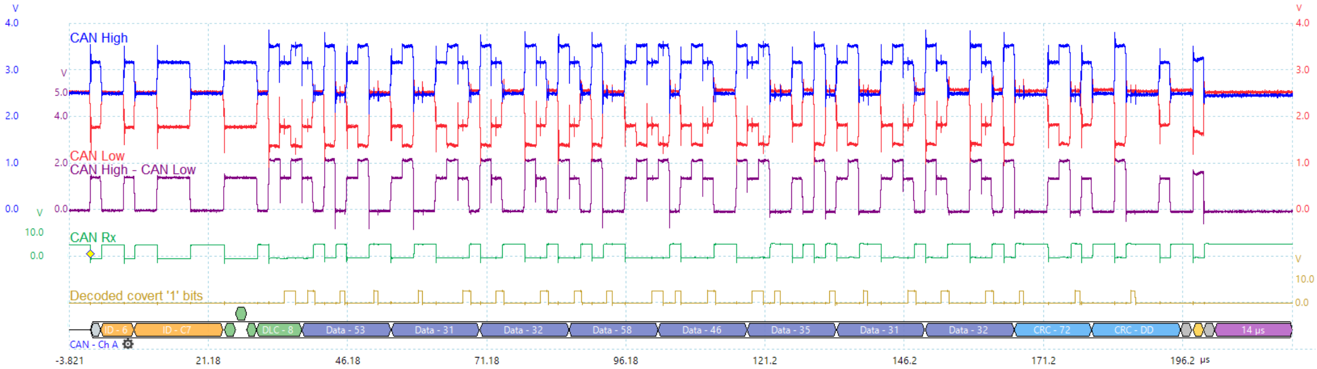


Fig. 11 Encoding and decoding a sequence of covert bits in a frame sent at 500kbps using the S12xF platform

covert information. It was configured to enable one conversion using 8 bit resolution per covert bit. The maximum conversion rate allowed on this platform is roughly equal to 2µs (the bit duration for a 500kbit/s rate). Conversions are made only during dominant bits and are triggered by a transition to the low level of the Rx pin (corresponding to the receive of dominant levels). Repeated conversions are made until the Rx pin transitions to high (recessive level). A set of continuous conversions last for at most 5 carrier bit times due to the bit stuffing rule. All conversion results are buffered and processed once the complete carrier frame is received. In Figure 11 we illustrated the decoding of dominant covert levels by signaling their detection through toggling a digital pin. Uneven duration of the illustrated decoded bits is caused by performance limitations of the S12 platforms as the conversion delay and post conversion processing did not always allow for timely signaling the bit end while decoding the bits in real-time at 500kbps communication bit rate. The covert bits are always correctly interpreted when processing is done after the completion of the carrier transmission.

Performance evaluation. In our tests, we were able to encode covert data covert bits over a known carrier at up to 500kbps on the S12X platform as illustrated for test transmissions in Figure 11. Note that here we refer only to the maximum sustainable bit rates during dominant levels of the carrier frame while the resulting data rates will vary depending on the number of dominant carrier bits.

While achieving the top bit rate of 500kbps the processing capabilities of the S12X are stretched to the limit leaving little to no space for other processing on the main S12 core. However, this load can be reduced by assigning tasks to the XGATE coprocessor. Achieving key exchange over the covert channel is still possible at 500kbps (Figure 9 exemplifies key exchange at 500kbps) on the S12X but the overall load of simultaneous encoding/decoding covert bits makes it difficult to perform other tasks during the key exchange mechanism.

5.2 Infineon TriCore

We used two application kits from Infineon equipped with the following automotive state-of-the-art, high-end microcontrollers: an Aurix TC277, which can operate at up to 200MHz and provides 472kB of RAM and 4MB of FLASH and an Aurix TC297, which can operate at up to 300MHz and comes with 728kB of RAM and 8MB of FLASH. Both boards come with an Infineon TLE7250GVIO CAN transceiver which enables high speed CAN communication. An external Microchip MCP2551 CAN transceiver supplied, controlled and monitored by the main AURIX kit was used for implementing the covert channel. Two pins from the transmitter node were connected to the TX and RX pins of the transceiver. In our experiments, the transmitter of both the carrier frame and covert data was implemented on the TC297 while the TC277 is used as a receiver or as the second node in a key exchange operation.

Encoding covert information. The implementation for the transmitter is based on two application modules which are provided by Infineon along with the AURIX documentation: the CAN application module which was used to transmit carrier frames and the GTM (Generic Timer Module) application which controls the transmission of covert information. The microcontroller pin connected to the MCP2551 TX pin is controlled from a timer interrupt service routine to transmit covert bits. Similar to the S12X handling, the covert information is prepared beforehand as a bit array based on the prior knowledge of the carrier frame content (only recessive bits are sent over the covert channel during recessive or unusable dominant bits in the carrier to prevent CAN protocol errors).

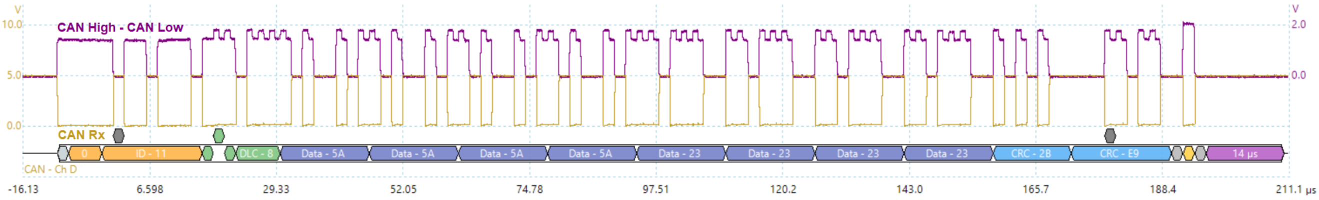


Fig. 12 Covert channel transmission at 1Mbps using the Aurix platform

Decoding covert information. The receiver node implementation consists of the configuration of the ADC application module. An interrupt triggered by the GTM TOM Timer was used to trigger the ADC sampling of the CAN_H voltage level once every microsecond. The ADC was configured to use 8 bit resolution and the sampling time was configured to 500ns. In this configuration, the conversion time necessary for the ADC to provide the digital value of the voltage was 1µs. The configuration of the timer interrupt and the first conversion trigger is done when the RX pin of the transceiver toggles to the low state due to the start bit of the carrier frame.

Performance evaluation. In our tests, we were able to encode covert bits over a known carrier at up to 1Mbit/s on the Aurix as illustrated, for a test transmission, in Figure 12. The efficient timer module of the Aurix platform allowed us to generate bits of even smaller duration. Unfortunately, we were not able to generate the corresponding covert CAN signals since the transceiver employed for this task (i.e. the MCP2551) is unable of achieving bit rates higher than 1Mbit/s. However, this is a strong indication that the mechanism could be implemented at higher bit rates and even over CAN-FD communication.

On the Aurix platform only one of the three cores and one timer interrupt are tasked with the covert channel transmissions allowing sufficient spare processing power for other tasks. Furthermore, we also managed to achieve the key exchange by only making use of one core and one timer interrupt.

5.3 Communication performance and error rates

As stated in the related work section, the throughput that is achievable by more conventional covert timing channels is much more limited when compared to the solution that we propose in this work. We now make a brief performance comparison with related work then we analyze the expected transmission error rates.

The work in [37] suggests a throughput of 22.5bps for the timing channel and this is further extended to 200bps when the least-significant bit of the frame is also added to the covert channel [37]. This results in a channel that tops somewhere below 300 bps. In contrast, our approach can send one covert bit for each regular dominant bit in the CAN data-field. For a 500kbps bit rate, assuming that half of the bits in an 8 byte data-field are dominant, this results in 32 bits extracted in roughly $200\mu s$ (this is the average duration of an 8 byte frame on the bus). For a regular bus-load of 30-50% around 1500-2500 frames are sent each second which would result in a data-rate of 48-80 kbps, a higher data-rate and incomparable to the data-rate of a timing channel from [37]. The data-rate is further increased by using the bits of the CRC field as well and even more by using multiple voltage levels. We do however assume that the covert channel will be used for securing regular CAN frames and thus only a limited amount of covert bits would be necessary.

Bit error rate. During the experiments that we carried out we encountered no transmission errors on the communication channel. To get a crisper image on the reliability of the channel and the potential error rate we now use some recent results in this direction. The bit error rate (BER) of a communication channel is determined by the ability to correctly decode the associated dominant voltage levels in the presence of noise. One specific analysis of communication channels added over existing CAN signaling is done by the authors of [8] which analyze a high-speed CAN transmission scheme that uses a passband modulated signal on a standard baseband CAN carrier providing data rates that exceed 1 Gbps. Based on measurements of the noise level in a vehicle, the BERs that the authors achieve are lower than 1×10^{-6} for modulated signal amplitudes as low a 0.2V. Our approach generates a modulated signal with amplitudes in the order of several hundred millivolts (at much lower frequencies). Therefore, we can consider 1×10^{-6} as an upper bound of the BER although it may be lower for our signaling inside a real car since we target a lower data-rate. As already stated, no bit errors were recorded during our experiments in a laboratory environment. According to [11], the BER measured for standard CAN communication is 3.1×10^{-9} in normal industrial environments and increases to 2.3×10^{-7} in the case of aggressive environments (at a distance of 2 meters from high frequency arc-welding equipment).

Table 1 Comparative performance results for CAN bus covert channels

Covert channel	Type	Bits/frame	Throughput (at 2500 frames/second)	BER	Send authentication tag (24 bit tag)	Key exchange (128 bit key)
TACAN-IAT [37]	timing	1	2.5 kbps	<1%	9.6 ms	n/a
CANTO [15]	timing	3-5	7.5-12.5 kbps	0.95%	2-3.2 ms	n/a
This work	voltage	32	80 kbps	<0.0001%	0.4 ms	1.6 ms

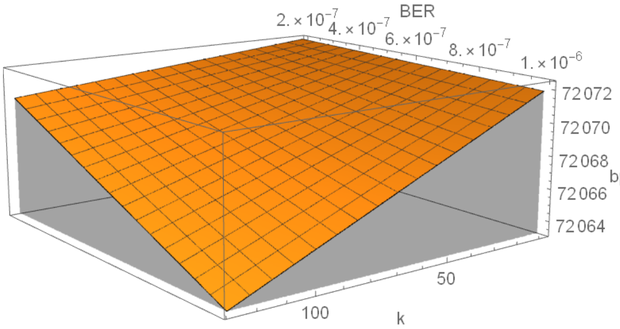


Fig. 13 Expected data rate of the covert channel according to package size k and BER

While the 1×10^{-6} BER level is several orders of magnitude higher than that of standard CAN communication, it would have little impact on our communication channel. The reason for this is that we intend to send over this channel cryptographic material that is limited in size, e.g., 32-64 bit authentication tags (current automotive standards require even less than this [2]) or 128 bit symmetric cryptographic keys. Given the estimated BER the packet error rate (PER), i.e., the probability that a packet will have at least one error easily follows as $\mathrm{PER} = 1-(1-\mathrm{BER})^k$ where k is the number of bits that we intend to send. Concretely, for a 32 bit tag the error rate is as low as 3.19×10^{-5} and even for the 128 bit key the error rate is no larger than 1.27×10^{-4} which will require that, on average, one key out of 10.000 keys is re-transmitted. In contrast to this, the work in [37] reports a much higher bit-error rate for timing covert channel: in the worst case around 40% of the bits are swapped and by increasing the size of a running average window the bit error rate drops to less than 0.1% but this is still three orders of magnitude higher than 1×10^{-6}.

To sum up this analysis, by considering a usual datarate of 500kbps and a bus-load of 50%, in Fig. 13 we show the estimated maximum capacity of the channel in the presence of transmission errors. We considered for this analysis an effective data rate of 288kbps which corresponds to the best case data bandwidth when CAN runs at 500kps (note that CAN frames have high overheads due to the IDs, stuffing bits, etc.). Subsequently we considered that half of these bits are dominant and the BER is in the range of 1×10^{-6} reported by the work in [8] down to 2.3×10^{-7} as predicted by [11]. Regardless of the frame size, from 24 bits (the minimum authentication tag size according to [2]) up to 128bits (the size of an AES key), having a package error rate of $\mathrm{PER} = 1-(1-\mathrm{BER})^k$ which would require re-transmission, the maximum data-rate of the covert channel remains steadily around 72kbps.

Error handling. Regular CAN frames employ a 15 CRC field for error detection which limits data throughput capabilities of the protocol. When used for security-related applications, as described in section 4, the proposed covert channel does not necessary require the use of a dedicated bit field for error detection. Error detection is straightforward without additional mechanisms since our channel carries cryptographic material for which any alteration will be easily detected. When used for authentication, any covert transmission error will result in an authentication failure. Similarly, in the case of the key exchange mechanism the final key verification step will fail when covert transmission errors occur.

Following detection, the message that contains the error will be re-transmitted. The implementation of an error correction mechanisms would be inefficient in this case due to the limited bit space in the covert channel and corresponding computational costs. Similar measures to the CAN error confinement mechanism can be put in place to limit the number of re-transmissions preventing any misuse by malicious nodes.

Comparative performance analysis. Table 1 presents a comparison of our work with other related works on covert channel implementations for CAN. For each of the considered works, the table illustrates the covert channel type, throughput, BER, number of covert bits per frame, the time required to transmit a 24 bit authentication tag (minimum tag size recommended by [2]) and the time for a 128 bit key exchange. The throughput is calculated based on the number of covert bits that can be carried by an 8 byte frame without errors, considering a CAN baudrate of 500kbps and a 50% busload (i.e., 2500 frames are transmitted each second). For the voltage covert channel implementation we consider that only the data field is used for carrying covert bits. Given the small amount of covert bits encoded in each frame, the timing based covert channels provide lower throughputs: TACAN-IAT [37] is limited at 2.5 kbps while CANTO [15] (improved version of INCANTA [14]) can go up to 12.5 kbps. Using our proposed approach the average throughput is at 80 kbps which can be increased even more (to 97.5 kbps) if including the CRC field as a carrier for covert bits. Transmitting a 24 bit authentication tag using voltage covert bits takes 0.4ms (i.e., the time required to transmit the 8 byte carrier frame). Both of the time covert approaches achieve this with a longer transmission time since they require several frame transmissions to transmit the complete 24 bit tag. A 128 bit key exchange can be done in 1.6 ms using the voltage channel while the cited related works did not cover this operation. The two timing covert channels feature BER of around 1% while our proposal provides a considerable reduction of the BER.

6 Conclusion and future work

Our work proposes a novel approach for enhancing the data rate over the CAN bus by implementing a voltage based covert channel. This paves the way for the integration of security mechanisms for in-vehicle communication, which is one of the major concerns for automotive networks today. Platforms with lower computational power may encounter problems in implementing the covert channel at high bit-rates if they also need to perform regular tasks in parallel. However, this issue can be alleviated by developing dedicated HW modules for handling the covert channel or by lowering the bit-rate of the covert channel. Our experiments show that the mechanism can be implemented on low and high-end automotive platforms by using of-the-shelf CAN transceivers and standard microcontroller peripherals. All the required hardware is roughly inexpensive.

We consider as future work a HW-based implementation of the proposed voltage covert channel. Such an approach could provide throughput improvements by reducing the load on the application layer and by providing additional benefits such as the use of multiple voltage levels to allow multiple bit encoding over a single carrier bit.

Compliance with Ethical Standards

Funding: This work was supported by a grant of Ministry of Research and Innovation, CNCS-UEFISCDI, project number PN-III-P1-1.1-TE-2016-1317, within PNCDI III (2018-2020).

Conflict of Interest: The authors declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Amrani O, Rubin A (2007) Contention Detection and Resolution for Multiple-Access Power-Line Communications. IEEE Transactions on Vehicular Technology 56(6):3879–3887
2. AUTOSAR (2017) Specification of secure onboard communication
3. Berk V, Giani A, Cybenko G, Hanover N (2005) Detection of covert channel encoding in network packet delays. Rapport technique TR536, de lUniversité de Dartmouth 19
4. Cabuk S, Brodley CE, Shields C (2004) IP covert timing channels: design and detection. In: Proceedings of the 11th ACM conference on Computer and communications security, pp 178–187
5. Cena G, Bertolotti IC, Hu T, Valenzano A (2014) A mechanism to prevent stuff bits in CAN for achieving jitterless communication. IEEE Transactions on Industrial Informatics 11(1):83–93
6. Cena G, Bertolotti IC, Hu T, Valenzano A (2019) On a software-defined CAN controller for embedded systems. Computer Standards & Interfaces 63:43 – 51
7. Checkoway S, McCoy D, Kantor B, Anderson D, Shacham H, Savage S, Koscher K, Czeskis A, Roesner F, Kohno T, et al (2011) Comprehensive experimental analyses of automotive attack surfaces. In: USENIX Security Symposium, San Francisco, pp 77–92
8. Choi E, Han S, Lee J, Lee S, Kang S, Choi JW (2018) Compatibility analysis of the turbo controller area network (turbo CAN). IEEE Transactions on Vehicular Technology 67(6):5146–5157
9. Dolev S, Krzywiecki L, Panwar N, Segal M (2016) Optical puf for non-forwardable vehicle authentication. Computer Communications 93:52–67
10. Fernandes E, Crispo B, Conti M (2013) FM 99.9, Radio Virus: Exploiting FM Radio Broadcasts for Malware Deployment. IEEE Transactions on Information Forensics and Security 8(6):1027–1037
11. Ferreira J, Oliveira A, Fonseca P, Fonseca JA (2004) An experiment to assess bit error rate in CAN. In: Proceedings of 3rd International Workshop of Real-Time Networks (RTN2004), pp 15–18
12. Groza B, Murvay PS (2018) Security Solutions for the Controller Area Network: Bringing Authentication to In-Vehicle Networks. IEEE Vehicular Technology Magazine 13(1):40–47

13. Groza B, Popa L, Murvay P (2019) TRICKS - Time TRIggered Covert Key Sharing for Controller Area Networks. IEEE Access 7:104,294–104,307
14. Groza B, Popa L, Murvay PS (2019) INCANTA - INtrusion Detection in Controller Area Networks with Time-Covert Authentication. In: Security and Safety Interplay of Intelligent Software Systems, Springer International Publishing, Cham, pp 94–110
15. Groza B, Popa L, Murvay PS (2021) Canto - covert authentication with timing channels over optimized traffic flows for can. IEEE Transactions on Information Forensics and Security 16:601–616, DOI 10.1109/TIFS.2020.3017892
16. Guerrieri L, Masera G, Stievano IS, Bisaglia P, Garcia Valverde WR, Concolato M (2016) Automotive Power-Line Communication Channels: Mathematical Characterization and Hardware Emulator. IEEE Transactions on Industrial Electronics 63(5):3081–3090
17. Han J, Lin YH, Perrig A, Bai F (2014) Short paper: Mvsec: secure and easy-to-use pairing of mobile devices with vehicles. In: Proceedings of the 2014 ACM conference on Security and privacy in wireless & mobile networks, pp 51–56
18. Van den Herrewegen J, Garcia FD (2018) Beneath the bonnet: A breakdown of diagnostic security. In: Computer Security, Springer International Publishing, Cham, pp 305–324
19. Hussain R, Kim D, Tokuta AO, Melikyan HM, Oh H (2015) Covert communication based privacy preservation in mobile vehicular networks. In: MILCOM 2015 - 2015 IEEE Military Communications Conference, pp 55–60
20. ISO (2003) 11898-1–Road vehicles–Controller area network (CAN)–Part 1: Data link layer and physical signalling. Tech. rep., International Organization for Standardization
21. ISO (2003) 11898-2, Road vehicles Controller area network (CAN) Part 2: High-speed medium access unit. Tech. rep., International Organization for Standardization
22. ISO (2006) 11898-3, Road vehicles Controller area network (CAN) Part 3: Part 3: Low-speed, fault-tolerant, medium-dependent interface. Tech. rep., International Organization for Standardization
23. Jain S, Wang Q, Arafin MT, Guajardo J (2018) Probing Attacks on Physical Layer Key Agreement for Automotive Controller Area Networks. In: 2018 Asian Hardware Oriented Security and Trust Symposium (AsianHOST), pp 7–12
24. Jousse J, Ginot N, Batard C, Lemaire E (2017) Power Line Communication Management of Battery Energy Storage in a Small-Scale Autonomous Photovoltaic System. IEEE Transactions on Smart Grid 8(5):2129–2137
25. Koscher K, Czeskis A, Roesner F, Patel S, Kohno T, Checkoway S, McCoy D, Kantor B, Anderson D, Shacham H, et al (2010) Experimental security analysis of a modern automobile. In: Security and Privacy (SP), 2010 IEEE Symposium on, IEEE, pp 447–462
26. Liu Y, Ghosal D, Armknecht F, Sadeghi AR, Schulz S, Katzenbeisser S (2009) Hide and seek in time—robust covert timing channels. In: European Symposium on Research in Computer Security, Springer, pp 120–135
27. Lodi GA, Ott A, Cheema SA, Haardt M, Freitag T (2016) Power Line Communication in automotive harness on the example of Local Interconnect Network. In: International Symposium on Power Line Communications and its Applications (ISPLC), pp 212–217
28. Miller C, Valasek C (2015) Remote exploitation of an unaltered passenger vehicle. Black Hat USA 2015:91
29. Mueller A, Lothspeich T (2015) Plug-and-secure communication for CAN. CAN Newsletter pp 10–14
30. Pico Technology (2020) CAN and CAN FD bus decoding. `https://www.picotech.com/library/oscilloscopes/can-bus-serial-protocol-decoding`, accessed: 2020-04-20
31. Robert Bosch GmbH (1991) Can specification, version 2.0
32. Rowan S, Clear M, Gerla M, Huggard M, Goldrick CM (2017) Securing vehicle to vehicle communications using blockchain through visible light and acoustic side-channels. arXiv preprint arXiv:170402553
33. Strobl M, Waas T, Moehne S, Kucera M, Rath A, Balbierer N, Schingale A (2012) Using Ethernet over powerline communication in automotive networks. In: Proceedings of the 10th International Workshop on Intelligent Solutions in Embedded Systems, pp 39–44
34. Sung G, Huang C, Wang C (2012) A PLC transceiver design of in-vehicle power line in FlexRay-based automotive communication systems. In: 2012 IEEE International Conference on Consumer Electronics (ICCE), pp 309–310
35. Tektronix (2019) Debugging CAN, LIN, and FlexRay Automotive Buses with an Oscilloscope, application note. `https://download.tek.com/document/Automotive-Bus_App-Note_`

55W-61098-3.pdf, accessed: 2020-04-20

36. Wolf M, Weimerskirch A, Paar C (2004) Security in automotive bus systems. In: Workshop on Embedded Security in Cars
37. Ying X, Bernieri G, Conti M, Poovendran R (2019) TACAN: transmitter authentication through covert channels in controller area networks. In: Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS 2019, Montreal, QC, Canada, April 16-18, 2019, pp 23–34
38. Ziermann T, Wildermann S, Teich J (2009) CAN+: A new backward-compatible Controller Area Network (CAN) protocol with up to 16x higher data rates. In: Proceedings of the Conference on Design, Automation and Test in Europe, European Design and Automation Association, pp 1088–1093