INCANTA - INtrusion detection in Controller Area Networks with Time-covert Authentication

Bogdan Groza, Lucian Popa, and Pal-Stefan Murvay

Faculty of Automatics and Computers,
Politehnica University of Timisoara, Romania
bogdan.groza@aut.upt.ro, lucian.popa.lp@gmail.com, stefan.murvay@gmail.com

Abstract. We explore the use of delays to create a time-covert cryptographic authentication channel on the CAN bus. The use of clock skews has been recently proposed for detecting intrusions on CAN, using similar mechanisms that were previously exploited in computer or mobile networks in the past decade. However, the fine-grained control of timers easily allows controllers to adjust their clock potentially making such mechanisms ineffective as we argue here and was also proved by a recent research work. We exploit this potential shortcoming in a constructive sense, i.e., the accuracy of arrival times on in-vehicle buses and the fine-grained control of timer/counter circuits on automotive controllers allows us to use time as a covert channel to carry cryptographic authentication. Based on this procedure we propose an effective authentication and intrusion detection mechanism that is fully back-ward compatible with legacy implementations on CAN. Our proposal directly applies to any modern in-vehicle bus, e.g., CAN-FD, FlexRay, etc.

1 Introduction and motivation

We are at a decade of research on attacks and countermeasures for in-vehicle networks. From proof-of-concept attacks on laboratory setups [11] to attacks on real-world vehicles [16], [3], [21], [22] we are witnessing each year more and more threats to the future of automobiles. Without proper countermeasures, such attacks may jeopardize the development of future technologies such as self-driving cars, autonomous intersection management systems, etc. Many security proposals were brought to attention by research works. Various cryptographic authentication techniques are explored from regular message authentication codes [10], [30] to well established protocols in sensors networks such as the TESLA protocol [9] or group key-sharing between nodes [8]. Attention is also payed to efficient allocation of signals in each frame [19]. Other works account for the physical layer in order to discard forged frames by error flags [17], hide authentication bits within regular CAN bits [37] or distinguish between nodes based on signal characteristics [26]. Particularities of the physical signalling on the bus have also been exploited to securely share a cryptographic key [12], [25].

Recently, the design of intrusion detection for the CAN bus has been explored by several research works. Solutions include the use of entropy [27], [20], inclusion of anomaly detection sensors [28], the analysis of voltage levels on the bus [5] or the use of cryptographic authentication [2]. Hardware implementations based on the error-confinement mechanism of CAN are discussed in [7]. Artificial intelligence techniques have been also recently employed by the use neural networks in [13], [14], [34], machine learning [36] and regression learning [18]. Other techniques include hidden Markov models [29], multivariate time series [35] and finite-state automatons [33].

Still, the most basic feature of the communication on the CAN bus which can be used to build intrusion detection mechanisms is the periodicity of messages on the bus. As industry implementations usually demand simplicity, such mechanisms cannot be neglected. Using frame periodicity to detect intrusions was discussed in several research works, e.g., [24] and [32]. Further, the periodicity of messages can be exploited to extract clock skews (which is a unique fingerprint due to physical imperfections in oscillators) and identify the sender of the message as discussed in [4]. The use of clock skews has been previously explored in computer networks [15] and was also applied to smart-phones [6]. However, as we point out in the analysis from the forthcoming section, the fine grained control of time-triggered interrupts and the low-level access to the system clock, may easily allow an embedded device to mimic the clock-skew of another. This was already proved by recent research in [31] which proposes cloaking attacks and may render mechanisms such as the work in [4] ineffective in detecting intrusions. In contrast, in this work we exploit this fine-grain control of timer-counter circuits in a constructive manner and envision the design of a time-covert cryptographic authentication and intrusion detection system for the CAN bus, i.e., INCANTA (INtrusion detection in Controller Area Networks with Time-covert cryptographic Authentication).

Our work is organized as follows. For clarity we begin by presenting the experimental setup in Section 2, this comprises high-end automotive-grade controllers as well as industry standard tools, e.g., CANoe, that are used both for simulating real-world in-vehicle traffic and measuring delays. In section 3 we discuss some theoretical notions on measuring clock offsets and present our first experimental results on measuring delays in our setup. Then, in section 4 we embed authentication information in delays, i.e., we create a time-covert authentication channel, and present experimental results. Section 5 holds the conclusions of our work.

2 Experimental setup

For gathering frame arrival timestamps we employed off-the-shelf devices and applications to build an experimental setup. Our setup, as suggested in Figure 1, consists on three nodes linked over a 500kbit/s CAN bus. Two of the nodes were implemented on AURIX development boards, while the third was a Vector VN1610 PC to CAN adapter. The Vector VN device was connected to a PC running CANoe 8.0.35 used to record all frames sent over the bus along with their arrival timestamps. A second set of frame arrival timestamps was recorded on one of the AURIX development boards which acted as a receiver node. To assure consistency of the results, the receiver node was generally either the VN1610 or a TC277 board which used the System Timer module to generate a 10ns base tick for recording the local time.

The target bus traffic was generated by the second of the AURIX-based nodes. Table 1 presents specifications for the 6 different AURIX boards featuring three different Infineon AURIX microcontrollers used as sender nodes in our data recording setup.

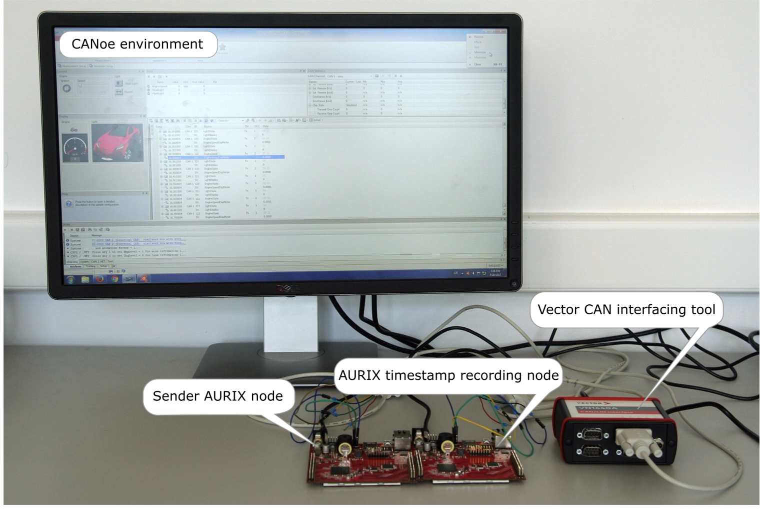


Fig. 1. Experimental setup used for gathering frame arrival timestamp data

The sender node was set to send a cyclic message once every 100ms. The timing functionality is implemented on the AURIX nodes using the on-chip System Timer module configured in the Compare Match Interrupt Control mode to generate interrupts at 100ms.

Table 1. Features of AURIX development boards employed in our experiments

Microcontroller characteristics	**Development board model**		
	AURIX TC224_TFT	AURIX TC277_TFT	AURIX TC299_TFT
RAM	96KB	472KB	728KB
FLASH	1MB	4MB	8MB
EEPROM	128KB	384KB	384KB
Top frequency	133MHz	200MHz	300MHz
CAN nodes	3	4	6
Employed board count	2	2	2

To provide more realistic results under normal bus operating conditions, a separate set of message arrival timestamps was recorded while generating additional bus traffic in the previously described setup. The additional traffic consisted of traffic recorded on a real-world high-end vehicle and replayed on our CAN bus setup by CANoe through the VN CAN adapter. The additional traffic consists of ≈ 100 different CAN message types sent on event or periodically with various cycle times. By introducing the recorded traffic the busload increased from 0.49% to around 50%.

3 Analysis of clock accuracy in automotive-grade controllers

We begin with some theoretical foundations then proceed to a practical analysis of delays on the automotive-grade platforms of our setup.

3.1 Theoretical background

Existing definitions from [23] provide sufficient theoretical background on clock offsets, skews and drifts that characterize differences between clock measurements. These were used in the works from [4] for in-vehicle networks, [15] for computer networks and [6] for smart-phones over wireless-networks. In all these scenarios, delays are used to identify a particular sender. We stay to the same notions but make small modifications according to our needs.

Distinct to the case of a general clock-adjustment scenario in computer networks, e.g., [15] or [6], we are missing the time-stamps of each participant and rely only on local clocks. Subsequently, we want to infer on the clock offset based on local timestamps and also from the a-priori knowledge of the precise time intervals at which frames are broadcast. We note that delays are generally fixed in automotive applications and thus we can infer on the intended delay since this is usually a hardcoded constant $10, 50, 100, 500, 1000ms$, etc.

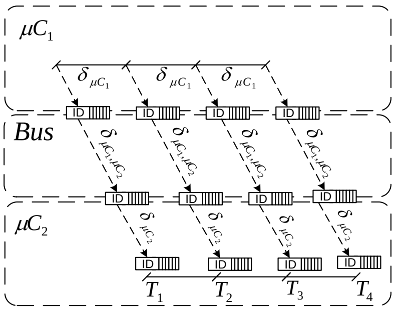
Figure 3.1 shows how the local timestamps are formed and how they account for delays that are expressed as random variables. Whenever principal μC_1 sends a cyclic frame, the frame is sent at delay $\delta_{\mu C_1}$ which is a random variable that accounts for imperfections in the local clock of μC_1. Subsequently, the frame travels on the bus and propagation delays are accounted which are again represented by a random variable $\delta_{\mu C_1, \mu C_2}$ (the propagation delay includes delays due to arbitration loss or mere propagation of the packet on the bus, etc.). Finally, the time-stamp of the frame is quantized on μC_2 this time accounting for new clock imperfections due to the local clock of μC_2 that are represented in the random variable $\delta_{\mu C_2}$. In principle, random variable $\delta_{\mu C_2}$ has a mean that is smaller than the mean of $\delta_{\mu C_1, \mu C_2}$ which can get significantly larger when the busload is high, while the mean of $\delta_{\mu C_1}$ is much larger than any of the two as it accounts for the delays at which the frame is sent.

We formally refine the timing metrics that we use in the following definition in attempt to set a theoretical background for our experimental measurements.

Definition 1. Let ΔT be the random variable that accounts for the delay between consecutive occurrences on the bus of frame identified by its ID sent by principal μC_1 and let $\mathbb{T}_{\mu C} = \{\Delta T_1, \Delta T_2, ..., \Delta T_n\}$ be a recording by principal μC_2 of n consecutive values of variable ΔT, i.e., $\Delta T_i = T_i - T_{i-1}, i = 1..n$, $T_0 = 0$. We define the following non-random variables that correspond to reference clocks:

i. $\mathcal{C}^{\star}_{\text{ideal}}(t) = t\delta$ where δ is the intended constant delay between frames,
ii. $\mathcal{C}^{\star}_{\text{min}} = tv_{\mathsf{min}}$ where v_{min} is the minimum value in $\mathbb{T}_{\mu C}$,
iii. $\mathcal{C}^{\star}_{\text{med}} = tv_{\mathsf{med}}$ where v_{med} is the median of the values in $\mathbb{T}_{\mu C}$,
iv. $\mathcal{C}^{\star}_{\text{mean}} = tv_{\mu}$ where v_{μ} is the mean of the values in $\mathbb{T}_{\mu C}$.

Subsequently, we define the cumulative clock offset of principals μC_1 and μC_2 with respect to reference clock $\mathcal{C}^{\star}_{\blacklozenge}$ where placeholder $\blacklozenge \in \{\text{min}, \text{mean}, \text{med}\}$ as $\mathcal{C}_{\mu C_1, \mu C_2}(t) -$



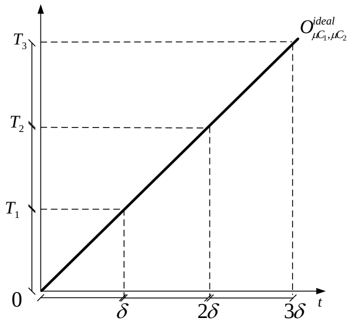


Fig. 2. The travelling time for a frame from μC_1 over the CAN bus to μC_2

Fig. 3. Offset of the recorded time-stamps in case of ideal clocks

$\mathcal{C}^\star_\blacklozenge(t)$ where $\mathcal{C}_{\mu C_1,\mu C_2} = \sum_{i=1,t} \Delta T_i$. In the following subsection we analyze the variation of delays with respect to these four reference clocks. While in the protocol description we analyze variations only with respect to $\mathcal{C}^\star_{\text{ideal}}$, evaluating the other three clocks should not appear meaningless since without proper evaluation it would have been improper to rule them out as possible indicators.

3.2 Experimental measurements of delays

We first perform experiments in order to choose which of the four metrics $\mathcal{C}^\star_{\text{ideal}}$, $\mathcal{C}^\star_{\text{min}}$, $\mathcal{C}^\star_{\text{med}}$, $\mathcal{C}^\star_{\text{mean}}$ is best suited for our approach. We also analyze the case when forced constant delays are added to each packet. For illustration purposes, these delays are fixed to ± 100, ± 250 and ± 500 clock ticks. In case of the TC277 boards (the main actor of our experiments) 1 tick of the System Timer is the equivalent of 10ns. Then we focus on the impact on changing delays by very small variations (in the order of hundreds of clock ticks) on the measurements from the receiver side. Nonetheless we discuss the behavior of lower priority IDs which may lose arbitration and thus add more to the propagation delay.

Our experimental measurements clearly point toward $\mathcal{C}^\star_{\text{ideal}}$ as the best reference value for the clock drift. Indeed, $\mathcal{C}^\star_{\text{ideal}}$ is a constant fixed at design time, but it is also easy to determine by empirical evidence on an existing network since manufacturers always choose fixed delays, e.g., 50ms, 100ms.

We now comment why the other indicators, i.e., $\mathcal{C}^\star_{\text{min}}, \mathcal{C}^\star_{\text{med}}, \mathcal{C}^\star_{\text{mean}}$ do not seem to offer a better approximation. The main problem consists in the gap between measurements on a free bus and measurements on a bus that features regular network traffic. We do rely on the minimum, median and mean of the first 100 received packets (by increasing this value the results do improve but not enough to justify the use of these three reference clocks) the subsequent plots are done over the next 1400 frames. When the bus is free of additional traffic the variations between the mean and median values are very small and results computed over a limited number of packets prove to be a bad indicator. The plots appear to be mixed between distinct delays with no obvious separation. This is plotted on the upper side of Figures 4 and 5. When the bus becomes loaded, variations increase by two orders of magnitude and become stable, this is plotted on the lower side of Figures 4 and 5. For $\mathcal{C}^{\star}_{\text{min}}$ apparently there is good separation both in the case of a free bus and a bus that is loaded with regular network traffic, this is depicted in Figure 6. However, the variations are not correctly aligned with the value of the delay, which suggests that the difference toward the minimum value is again a poor separator. For the same measurements $\mathcal{C}^{\star}_{\text{ideal}}$ proves to be a very good classifier with or without network traffic, measurements are presented in Figure 7.

In Figures 4, 5, 6 and 7 we have contrasted between independent measurements on the Infineon board and CANoe since the similarities between the independent measurements prove the correctness of the results. The plots suggest that using the clock drift from a single packet may lead to wrong classifications but the cumulative clock drift computed over a few dozen packets is a very good separator to identify a specific delay.

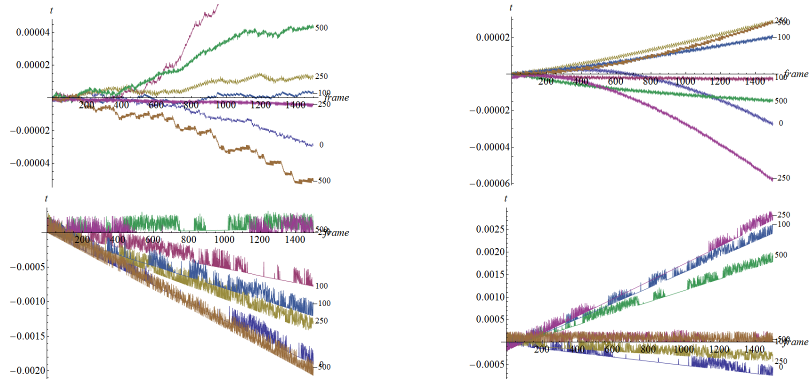


Fig. 4. $\mathcal{C}^{\star}_{\text{mean}}$ for a frame sent from an Infinenon TC277 as measured on another Infineon board (left) and on CANoe (right) - results over a free bus (up) or with network traffic (down)

3.3 Forcing delays on the bus

We now discuss the impact on manipulating delays at fine grain clock modifications. This discussion is essential for embedding the authentication values in delays which is the objective of the INCANTA protocol.

Figure 8 shows these delays (±100, ±250 and ±500 clock ticks) when they are measured from another TC277 board without (left) and with additional traffic (right). In Figure 9 we show the same delays when measured from a CANoe/VN device at the same time. Here we choose to present the plot at finer grain by taking only 500 packets, otherwise these plots are consistent with the plots already shown in Figure 7. The slope of the lines are distinct since the clock of the reference clock from the VN CAN adapter device is distinct. Overlapping the plots in the two cases (with and without traffic) for each reference clock in Figure 10 shows that traffic does not cause significant changes

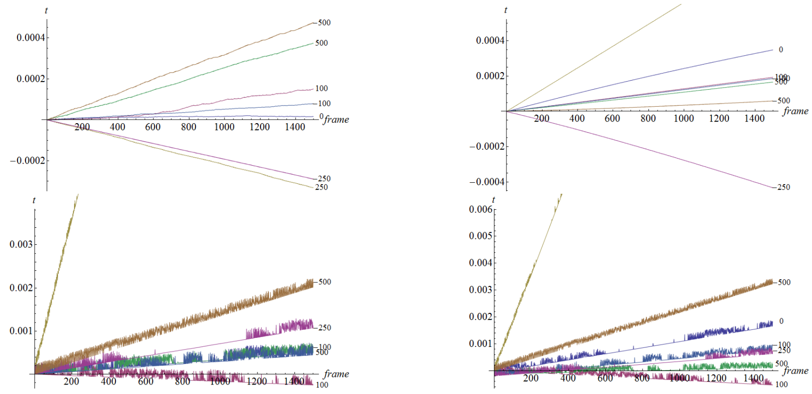


Fig. 5. $\mathcal{C}^{\star}_{\mathrm{med}}$ for a frame sent from an Infinenon TC277 as measured on another Infineon board (left) and on CANoe (right) - results over a free bus (up) or with network traffic (down)

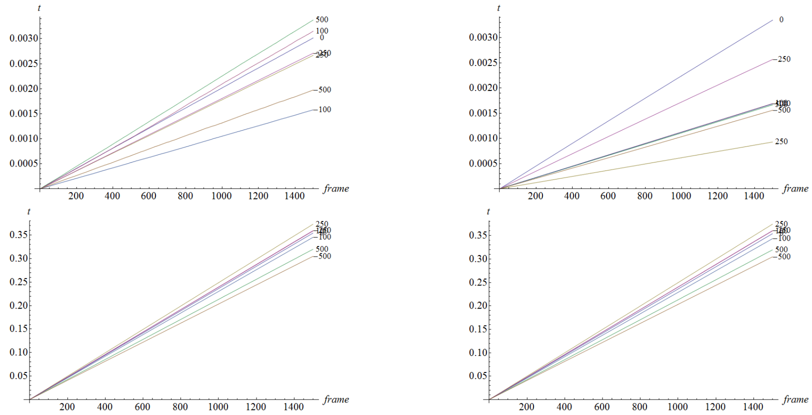


Fig. 6. $\mathcal{C}^{\star}_{\min}$ for a frame sent from an Infinenon TC277 as measured on another Infineon board (left) and on CANoe (right) - results over a free bus (up) or with network traffic (down)

in the clock skew over multiple packets. While small variations can be encountered, the slopes are close for the two cases.

Low priority IDs. Switching to low priority IDs leads to significant changes in the delay at which packets arrive. Computed over multiple packets, the clock skew remains the same, but individual packets may come at significant delays. In Figure 11 we illustrate delays for one of the TC277 boards in the case with no additional traffic on the bus (i), a bus loaded at 50% with delays computed on a packet with the highest priority ID (ii) and delays in case of the lowest priority extended ID (iii). In the later case, it is easy to note that inter-packet delays may have large variations. In part (iv) we overlap these 3 plots and the result confirms that the clock skew is the same for the node. Small variations appear in the case when the delay is set to 0, ± 100 and ± 250 between the

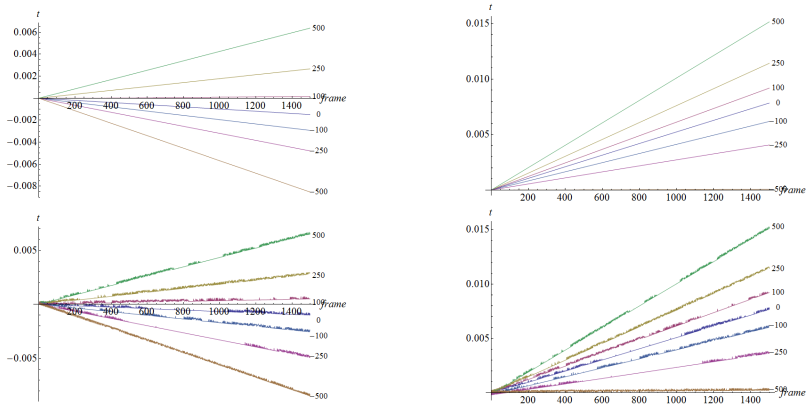


Fig. 7. $\mathcal{C}^{\star}_{\text{ideal}}$ for a frame sent from an Infinenon TC277 as measured on another Infineon board (left) and on CANoe (right) - results over a free bus (up) or with network traffic (down)

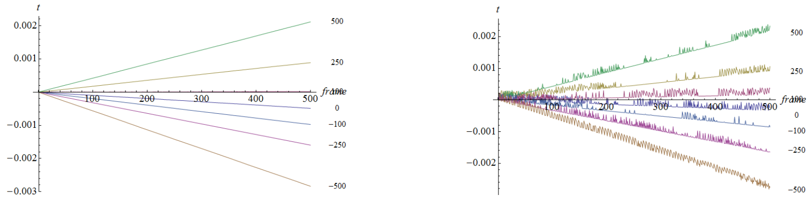


Fig. 8. Delays for a frame sent from an Infinenon TC277 as measured on another Infineon board without (left) and with network traffic (right)

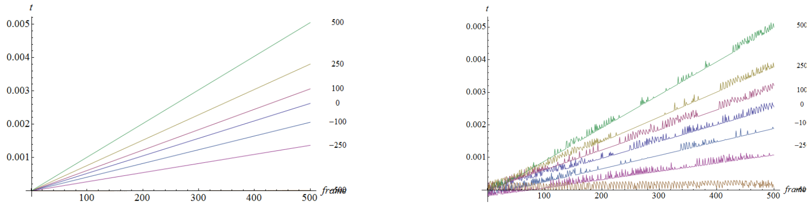


Fig. 9. Delays for a frame sent from an Infinenon TC277 as measured from CANoe/VN CAN adapter without (left) and with network traffic (right)

three cases but the slope of the line that mediates these values is still close, making identification possible.

Repeated randomized trials. To get a convincing image on the correctness of the results we proceed to a set of randomized trials that consist in taking small portions of the trace at random positions and compute the variation of clock on the smaller data set. We consider both the case of a free bus and that of a bus loaded with regular network traffic. We performed these tests with 4 Infineon controllers: TC277, TC297, TC224 and a second TC224. Moreover, these tests are again performed for all the induced delays on packet arrival: ±100, ±250 and ±500, Figure 12 graphically depicts the results. For

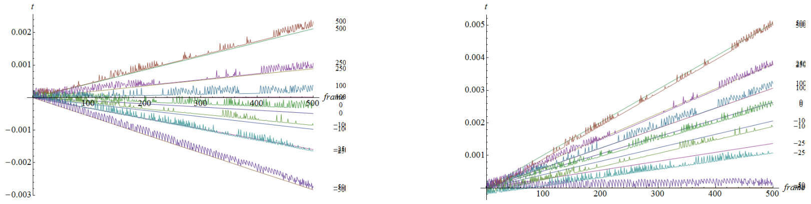
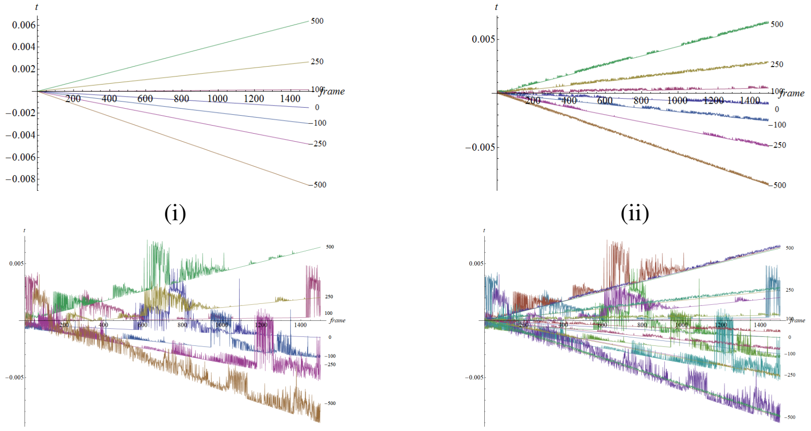


Fig. 10. Delays for a frame sent from an Infinenon TC277 as measured by an Infineon board (left) or from CANoe/VN CAN adapter (right) with and without traffic (overlap)



(i)

(ii)

(iii)

(iv)

Fig. 11. Delays for TC277 on several testbeds: a bus with no traffic (i), with regular traffic and high priority ID (ii) and regular traffic with lowest priority extended ID (iii) and overlap of the three cases (iv) (the slope is similar in all cases)

each board the first set of data which is encircled in the figure represent the packet arrival time without any forced delay, the subsequent tests are for plus 100, 250 and 500 ticks and it can be clearly seen that the arrival time increases and is correctly determined by the receiver. Then the delay decreases by 100, 250 and 500 ticks and so does the arrival time measured by the receiver. In the case when regular traffic is present, variations exists due to the obvious fact that the bus may not be free when data is sent and the time-stamp of these packets alone does not offer sufficient information to identify a sender node (or the intended delay). Fortunately, even with a loaded bus for the majority of packets the delay is correctly estimated as can be seen in Figure 12. In the next section we do discuss how to exploit this delay in order to hide authentication information and how additional traffic influences the accuracy of our estimation (some packets do deviate significantly from the expected delay and unavoidably they will contribute to the false positive rate).

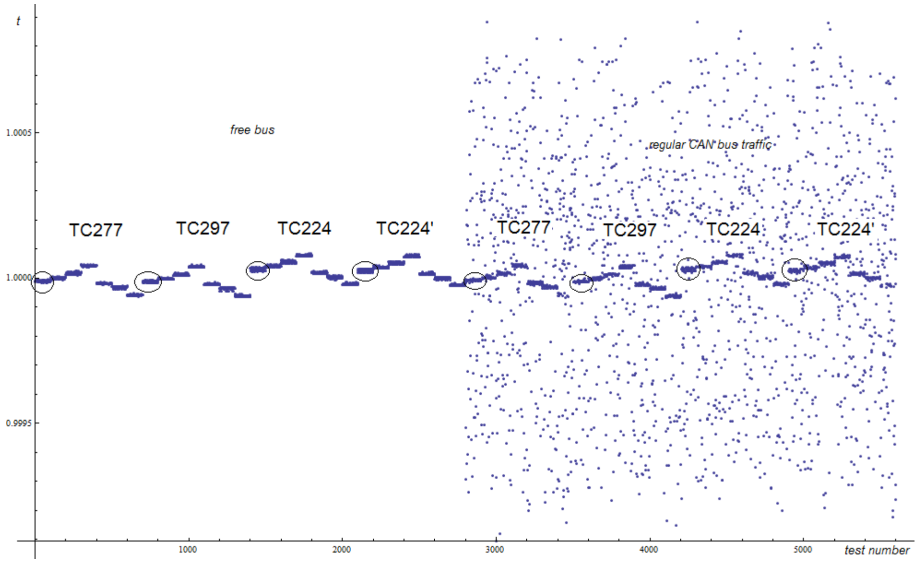


Fig. 12. Results after 100 tests at randomized locations in the trace for all 4 Infineon controllers: TC277, TC297, TC224 and a second TC224 and the six added delays ±100, ±250 and ±500 with (right) and without (left) regular network traffic

4 The proposed protocol and results

In this section we give a brief overview of the proposed protocol which uses a covert timing channel to embed authentication tags. Then we present experimental results.

4.1 Protocol overview

The protocol description is written for frames which are cyclic in nature. Fortunately the large majority of CAN bus traffic fits into this category. Frames that are on-event can be treated distinctly provided that there is a reference frame for computing the delay. For example one can use the delay toward the previous cyclic frame as a covert channel. It is out of scope for the current work to address on-event frames.

We consider that a shared secret key k exists on each ECUs from the CAN bus. We do not discuss how this key is shared since procedures for this are well known. The INCANTA (INtrusion detection in Controller Area Networks with Time-covert cryptographic Authentication) protocol consists in the following set of actions that are to be followed by each node:

1. $\mathsf{SendCyclic}(id, \mathrm{m})$ is the procedure triggered at some fixed delay δ for a frame with identifier field id at which the sender computes the tag $tag = MAC_{\mathsf{k}}(i, id, \mathrm{m})$ where i is a counter that is incremented for each ID. The sender then sets $\mathsf{T} = \lfloor tag \rfloor_\ell$ and performs a wait operation $wait(\mathsf{T})$ then broadcasts message (id, m),

2. $\mathsf{RecCyclic}(id, \mathrm{m})$ at which a message having identifier id and data-field m is received at time t, the receiver computes $tag = MAC_{\mathsf{k}}(id, \mathrm{m})$ and $\mathsf{T} = \lfloor tag \rfloor_{\ell}$ then checks if $|\mathsf{t} - i\delta + \mathsf{T}| \leq \epsilon$ and if this fails it drops the frame and reports an intrusion by returning $\mathsf{Intrusion}$ (here i is the counter for the corresponding frame).

In the description above, we assume that the CAN message having the identifier id is sent at delay δ while the expected arrival time differs by a small constant ϵ which compensates for both synchronization error and propagation/computation delays. The desired security level is denoted by ℓ. For practical purposes this must be set to a value that introduces a reasonable delay. For example, by using $\ell = 16$ and assuming signed values the maximum delay would be $\pm 2^{15}$ ticks, i.e., $327680ns$, and the $327\mu s$ should be negligible considering a frame that regularly arrives at $100ms$.

4.2 Results on embedding authentication in delays

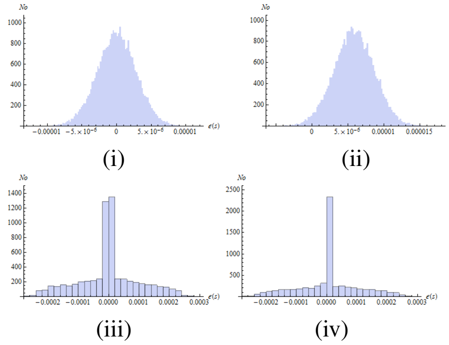
The analysis in this section deals with deviations of the arrival time from the expected value and with the analysis of the adversarial success rate in injecting forged frames. The broader image in Figure 12 makes it clear that in case when regular bus traffic is present, frame arrival time may deviate by large amounts. We do discuss how this affects the detection rate and the improvements that can be done.

Deviations from the expected value. Due to existing traffic (and also due to measurement imprecisions but by a much smaller amount) some of the genuine frames may be marked as potential intrusions, i.e., the false positives rate FPR. To begin with, in Figure 13 we show the deviations as recorded in our experiments in case when there is no additional traffic on the bus (i)-(ii) and in case when the bus is loaded with additional traffic (iii)-(iv). Clearly, the distribution of delays is Gaussian in both cases but it is greatly influenced by existing traffic. Figure 14 proves that these deviations are mostly independent on the receiver's clock by illustrating the distribution of delays as recorded both on CANoe vs. the Infineon board. Indeed, the distributions are similar which suggests as expected that the busload is the only cause for the delays.

Detection rate, true negatives and false positives. The experiments are performed both with MD5 and SHA256 as the underlying hash functions (messages are hashed along with a secret key as required by regular Message Authentication Codes) and the delay is fixed to the last 16, 20 or 22 bits of the resulting authentication tag. MD5 is known to be insecure but this is irrelevant to our experiments since our security level is even lower than that of MD5. Experimental data is used to compute the acceptance rate for legitimate frames, i.e., the true negatives rate TNR, while we estimate the success rate of an adversary, i.e., false negatives rate FNR, synthetically as:

$$\epsilon_{adv} = \frac{\epsilon}{\theta 2^{\ell}}$$

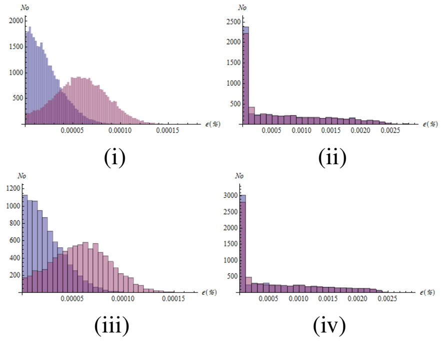
Here ϵ is the delay tolerance for accepting a frame, θ is the value of a tick (in seconds) and ℓ is the security level. This comes from the assumption that an adversary can at best insert a frame at some random point and hope that it will match the expected delay. Figure 15 shows the contrast for the advantage of legitimate frames when compared to frames injected by an adversary. Plots (i) and (ii) address the case without



(i) (ii)

(iii) (iv)

Fig. 13. Delay deviations on a free bus recorded by Infineon receiver (i) vs. CANoe trace (ii) and on a bus with regular traffic recorded by Infineon receiver (iii) or CANoe trace (iv)



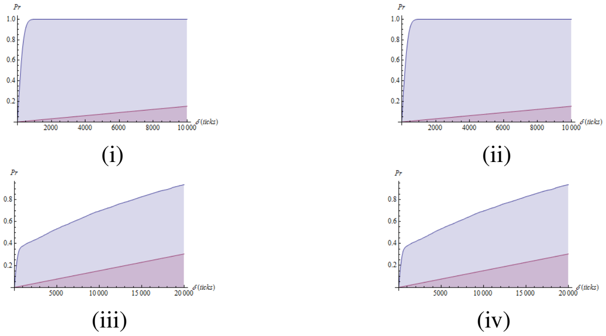
(i) (ii)

(iii) (iv)

Fig. 14. Comparative deviations between delays on Infineon and CANoe traces with MD5 (i-ii) and SHA2 (iii-iv) on a free bus (left) and a bus with regular traffic (right)

additional bus-load. Since there are no delays, the probability of accepting a legitimate frame quickly increases to 1 by using a tolerance of several hundreds ticks. The bus-load however greatly affects the acceptance rate and in case when additional traffic is present the tolerance needs to be increased to more than ten thousands ticks. While the advantage of an adversary is much higher, the acceptance rate for legitimate frames is still superior. The results are also summarized in Table 2 for the case of a free bus and Table 3 for the case of a bus loaded with regular traffic. As expected on the free bus, by considering a tolerance of around one thousand ticks, legitimate frames are accepted at a 99.9% rate while a false negative can occur between 1.40% and 0.02% according to the security level. This is a very good detection/acceptance rate. For the case of a bus loaded with regular network traffic only by a tolerance close to 20,000 ticks we get an acceptance rate of 90% for legitimate frames and from 1.80% to 0.47% false negative rate at 20–22 bit security for the embedded authentication delay. This is satisfactory but not perfect and may be improved by better allocation of bus traffic which out reach for our current work. For 16-bit authentication tags hidden in delays the false negative rate is somewhat high at 27% which suggest that 20-22 bits should be preferred.

Nonetheless we find it relevant to note that the chance of legitimate frames to be accepted increases rapidly after only several hundred ticks in tolerance and then reaches a first 0-growth point (a point where the acceptance probability does not increase at the next tick that is added in tolerance). At the 0-growth point the advantage of the adversary is extremely small, e.g., 0.0002% to 0.4%, while the acceptance rate (TNR) of legitimate frames is much higher, e.g., 4.89% to 25%. While such an acceptance rate is too low to be useful for practical scenarios, the discrepancy between the TNR and FNR can be positively exploited by deciding the intrusion over several consecutive frames. This happens because in case of legitimate frames there is a high chance that at least one frame fits the expected arrival time while it is less likely for any of the adversarial frames to match the expected arrival time. For example, if each authentication tag is computed over the content of the previous k frames, then each frame may eventually benefit from a tag embedded in k distinct delays. The probability for at least a single frame out of the k frames to yield the correct delay which matches the authentication tag is $1 - \mathrm{FPR}^k = 1 - (1 - \mathrm{TNR})^k$ while the probability that an adversary injects one frame with a correct delay remains $1 - \mathrm{TPR}^k = 1 - (1 - \mathrm{FPR})^k$. Substituting with data from the second row of Table 3 we get for $k = 8$ a correct classification rate of $1 - (1 - 0.26)^8 = 91\%$ while the false negative rate is $1 - (1 - 0.0044)^8 = 3\%$. This is only a quick estimation and further experiments in this direction will be subject of future work for us.



(i) (ii) (iii) (iv)

Fig. 15. Advantage of genuine frames in front of adversarial frames at a security level of 16 bits with no bus-load as recorded by the Infineon receiver (i) or CANoe trace (ii) and with bus-load by the Infineon receiver (iii) or CANoe trace (iv)

Table 2. Detection rate in case of a bus with no traffic

	Tolerance at 90% TNR	FNR at 90% TNR	Tolerance at 99.9% TNR	FNR at 99.9% TNR
16 bit ((MD5)	469	0.70%	941	1.40%
16 bit (SHA256)	505	0.77%	1033	1.57%
20 bit (SHA256)	510	0.04%	981	0.09%
22 bit (SHA256)	504	0.01%	998	0.02%

Table 3. Detection rate in case of a bus with regular traffic

	0-growth point tolerance	TNR at 0-growth	FNR at 0-growth	Tolerance at 90% TNR	FNR at 90% TNR
16 bit ((MD5)	312	25.00%	0.4%	18289	27.00%
16 bit (SHA256)	291	26.00%	0.44%	17976	27.00%
20 bit (SHA256)	65	4.89%	0.006%	19596	1.80%
22 bit (SHA256)	121	9.16%	0.002%	19814	0.47%

5 Discussion and conclusion

Our work proves that the cyclic nature of in-vehicle communication and the accuracy of timers on automotive-grade controllers can facilitate the creation of an efficient time-covert authentication channel on the CAN bus.

INCANTA stays at the borderline between a conventional intrusion detection mechanism and regular cryptographic authentication. By relying on cryptography our detection system should be superior to conventional intrusion detection systems which can be easily fooled by messages build to satisfy intrusion classification rules (e.g., simulating the delay of another device/frame). Since false-negatives are still present, we do not eliminate the need for a conventional IDS which may easily coexist with the current solution. Since the security level that can be embedded in delays, i.e., 16-20 bits, is obviously lower than the size of a regular cryptographic MAC, e.g., 128 bits, we cannot claim to achieve perfect security in a cryptographic sense. However, recently introduced standards in automotive security require only 24 bits of security [1] for authentication values used on in-vehicle modules (this may may be enough for real-time communication). Depending on the network traffic, we may get closer to this limit.

Besides relying on cryptography which makes it more solid than regular intrusion detection systems, INCANTA has at least two merits: first it is fully back-ward compatible and second it does not increase the bus-load which is already at its limit on the CAN bus. Nonetheless, the solution is bus independent and can be ported on modern buses such as CAN-FD, FlexRay or BroadRReach without much modifications. Limitations do exist as in the case of a loaded bus a small rate of false positives and true negatives does occur. But we hope that this limitation can be overcome by better allocation of the rest of the traffic from the bus. For the moment this was out-of-reach for our work and we used a real-world trace from a vehicle bus that was not specifically designed for our experiments. As future work we do believe that better allocation of the traffic on the bus can lead to excellent results and we hope that our work opens road in this direction where clever engineering can merge with cryptographic techniques to build efficient intrusion detection by using covert timing channels on CAN or other in-vehicle buses.

Acknowledgement. We thank the reviewers for their comments which have helped us to improve our work. This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS-UEFISCDI, project number PN-II-RU-TE-2014-4-1501 (2015-2017) `http://www.aut.upt.ro/~bgroza/projects/cseaman/`.

References

1. AUTOSAR. *Specification of Secure Onboard Communication*, 4.3.1 edition, 2017.
2. A. Boudguiga, W. Klaudel, A. Boulanger, and P. Chiron. A simple intrusion detection method for controller area network. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–7. IEEE, 2016.
3. S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno, et al. Comprehensive experimental analyses of automotive attack surfaces. In *USENIX Security Symposium*. San Francisco, 2011.

4. K.-T. Cho and K. G. Shin. Fingerprinting electronic control units for vehicle intrusion detection. In *25th USENIX Security Symposium*, 2016.
5. W. Choi, K. Joo, H. J. Jo, M. C. Park, and D. H. Lee. Voltageids: Low-level communication characteristics for automotive intrusion detection system. *IEEE Transactions on Information Forensics and Security*, 2018.
6. M. Cristea and B. Groza. Fingerprinting smartphones remotely via icmp timestamps. *IEEE Communications Letters*, 17(6):1081–1083, 2013.
7. H. Giannopoulos, A. M. Wyglinski, and J. Chapman. Securing vehicular controller area networks: An approach to active bus-level countermeasures. *IEEE Vehicular Technology Magazine*, 12(4):60–68, 2017.
8. B. Groza, P.-S. Murvay, A. Van Herrewege, and I. Verbauwhede. LiBrA-CAN: a lightweight broadcast authentication protocol for controller area networks. In *11th International Conference on Cryptology and Network Security, CANS 2012, Springer-Verlag, LNCS*, 2012.
9. B. Groza and S. Murvay. Efficient protocols for secure broadcast in controller area networks. *IEEE Transactions on Industrial Informatics*, 9(4):2034–2042, 2013.
10. O. Hartkopp, C. Reuber, and R. Schilling. MaCAN-message authenticated CAN. In *10th Int. Conf. on Embedded Security in Cars (ESCAR 2012)*, 2012.
11. T. Hoppe and J. Dittman. Sniffing/replay attacks on can buses: A simulated attack on the electric window lift classified using an adapted cert taxonomy. In *Proceedings of the 2nd workshop on embedded systems security (WESS)*, pages 1–6, 2007.
12. S. Jain and J. Guajardo. Physical layer group key agreement for automotive controller area networks. In *Conference on Cryptographic Hardware and Embedded Systems*, 2016.
13. M.-J. Kang and J.-W. Kang. Intrusion detection system using deep neural network for in-vehicle network security. *PloS one*, 11(6):e0155781, 2016.
14. M.-J. Kang and J.-W. Kang. A novel intrusion detection method using deep neural network for in-vehicle network security. In *Vehicular Technology Conference (VTC Spring), 2016 IEEE 83rd*, pages 1–5. IEEE, 2016.
15. T. Kohno, A. Broido, and K. C. Claffy. Remote physical device fingerprinting. *IEEE Transactions on Dependable and Secure Computing*, 2(2):93–108, 2005.
16. K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, et al. Experimental security analysis of a modern automobile. In *Security and Privacy (SP), 2010 IEEE Symposium on*, pages 447–462. IEEE, 2010.
17. R. Kurachi, Y. Matsubara, H. Takada, N. Adachi, Y. Miyashita, and S. Horihata. CaCAN - centralized authentication system in CAN (controller area network). In *14th Int. Conf. on Embedded Security in Cars (ESCAR 2014)*, 2014.
18. H. Li, L. Zhao, M. Juliato, S. Ahmed, M. R. Sastry, and L. L. Yang. Poster: Intrusion detection system for in-vehicle networks using sensor correlation and integration. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 2531–2533. ACM, 2017.
19. C.-W. 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*, 7(1):11–14, 2015.
20. M. Marchetti, D. Stabili, A. Guido, and M. Colajanni. Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms. In *Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, pages 1–6. IEEE, 2016.
21. C. Miller and C. Valasek. Adventures in automotive networks and control units. *DEF CON*, 21:260–264, 2013.
22. C. Miller and C. Valasek. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA*, 2015, 2015.

23. S. B. Moon, P. Skelly, and D. Towsley. Estimation and removal of clock skew from network delay measurements. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 227–234. IEEE, 1999.
24. M. R. Moore, R. A. Bridges, F. L. Combs, M. S. Starr, and S. J. Prowell. Modeling inter-signal arrival times for accurate detection of can bus signal injection attacks: a data-driven approach to in-vehicle intrusion detection. In *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*, page 11. ACM, 2017.
25. A. Mueller and T. Lothspeich. Plug-and-secure communication for can. *CAN Newsletter*, pages 10–14, 2015.
26. P.-S. Murvay and B. Groza. Source identification using signal characteristics in controller area networks. *IEEE Signal Process. Lett.*, 21(4):395–399, 2014.
27. M. Müter and N. Asaj. Entropy-based anomaly detection for in-vehicle networks. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 1110–1115. IEEE, 2011.
28. M. Müter, A. Groll, and F. C. Freiling. A structured approach to anomaly detection for in-vehicle networks. In *Information Assurance and Security (IAS), 2010 Sixth International Conference on*, pages 92–98. IEEE, 2010.
29. S. N. Narayanan, S. Mittal, and A. Joshi. Obd_securealert: An anomaly detection system for vehicles. In *Smart Computing (SMARTCOMP), 2016 IEEE International Conference on*, pages 1–6. IEEE, 2016.
30. A.-I. Radu and F. D. Garcia. LeiA: a lightweight authentication protocol for CAN. In *European Symposium on Research in Computer Security*, pages 283–300. Springer, 2016.
31. S. U. Sagong, X. Ying, A. Clark, L. Bushnell, and R. Poovendran. Cloaking the clock: emulating clock skew in controller area networks. In *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems*, pages 32–42. IEEE Press, 2018.
32. H. M. Song, H. R. Kim, and H. K. Kim. Intrusion detection system based on the analysis of time intervals of can messages for in-vehicle network. In *Information Networking (ICOIN), 2016 International Conference on*, pages 63–68. IEEE, 2016.
33. I. Studnia, E. Alata, V. Nicomette, M. Kaâniche, and Y. Laarouchi. A language-based intrusion detection approach for automotive embedded networks. *International Journal of Embedded Systems*, 10(1):1–12, 2018.
34. A. Taylor, S. Leblanc, and N. Japkowicz. Anomaly detection in automobile control network data with long short-term memory networks. In *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*, pages 130–139. IEEE, 2016.
35. A. Theissler. Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection. *Knowledge-Based Systems*, 123:163–173, 2017.
36. D. Tian, Y. Li, Y. Wang, X. Duan, C. Wang, W. Wang, R. Hui, and P. Guo. An intrusion detection system based on machine learning for can-bus. In *International Conference on Industrial Networks and Intelligent Systems*, pages 285–294. Springer, 2017.
37. A. Van Herrewege, D. Singelee, and I. Verbauwhede. Canauth-a simple, backward compatible broadcast authentication protocol for can bus. In *ECRYPT Workshop on Lightweight Cryptography*, volume 2011, 2011.