Journal of Systems Architecture 136 (2023) 102835

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

A qualitative cybersecurity analysis of time-triggered communication networks in automotive systems

Raimund Kirner [a],[*], Peter Puschner [b]

[a] University of Hertfordshire, Hatfield, United Kingdom
[b] TU Wien, Vienna, Austria

ARTICLE INFO

Keywords:
Security
Denial of service (DoS)
Time-triggered communication
Time-Sensitive Networking (TSN)
CAN bus

ABSTRACT

Security is gaining increasing importance in automotive systems, driven by technical innovations. For example, automotive vehicles become more open systems, allowing the communication with other traffic participants and road infrastructure. Also, automotive vehicles are provided with increased autonomy which raises severe safety concerns, and consequently also security concerns—both concerns that interweave in such systems.

In this paper we present a qualitative cybersecurity analysis by comparing different time-triggered (TT) communication networks. While TT communication networks have been analysed extensively for dependability, the contribution of this work is to identify security-related benefits that TT communication networks can provide. In particular, their mechanisms for spacial and temporal encapsulation of network traffic are instrumental to improve network security. The security arguments can be used as a design guide for implementing critical communication in flexible network standards like TSN.

1. Introduction

Distributed real-time systems need deterministic real-time communication [1]. The *Time-Triggered Architecture* (TTA) has been developed as a real-time communication system based on time-controlled (aka time-triggered) message forwarding [2,3]. The TTA is designed for the use in safety–critical applications [4]. There have been different instantiations of the TTA, but also other *time-triggered* (TT) communication systems inspired from it. While it is established that TT systems have characteristic properties that are not only suitable for safety–critical applications, we want to highlight in this article that they are also beneficial for security purposes. In particular we show that these concepts are useful to provide security in the automotive domain.

An early automotive communication network is *Controller Area Network* (CAN), developed by Bosch in 1986. CAN is typically used in automotive vehicles for the realisation of distributed real-time control systems. The computing nodes in these control systems are called *Electronic Control Units* (ECU). To connect to the individual sensors, cost-efficient sensor networks are used, for example *Local Interconnect Network* (LIN) since 2002. The TT protocol *FlexRay* has been developed to provide faster and more reliable communication than CAN [5]. Despite the arrival of FlexRay, $CAN\ Bus$ is still more widely used, as it is cheaper. For multi-media applications the *Media Oriented Systems Transport* (MOST) bus [6] has been developed, with 23 megabaud initially. However, the automotive in-vehicle communication is currently moving towards Ethernet-based solutions, like *Time-Sensitive Networking* (TSN) [7].

The development of $CAN\ Bus$ was never done with security in mind, thus it provides many attack surfaces [8]. There have been many extensions of $CAN\ Bus$, aiming to improve its security [9].

This article presents a qualitative cybersecurity analysis of TT communication systems. A fundamental outcome of the cybersecurity analysis is that the structured communication in TT communication systems can provide a defence against certain security attacks. Based on this outcome we give recommendations of how Ethernet-based solutions like TSN can adapt the mechanisms found in structured TT communication to provide similar security properties.

The article is structured as follows: Section 2 introduces the network properties and network attacks we consider in our security analysis. Section 3 describes the different communication networks we use for our comparison. A comparison of the different network types with respect to the considered properties and attacks is given in Section 4. Related work is studied in Section 5. Finally, Section 6 concludes this article.

2. Threat model

In this section we describe the threat model of our security analysis of time-triggered communication networks. Section 2.1 describes the

[*] Corresponding author.
E-mail addresses: r.kirner@herts.ac.uk (R. Kirner), peter@vmars.tuwien.ac.at (P. Puschner).

https://doi.org/10.1016/j.sysarc.2023.102835
Received 10 October 2022; Received in revised form 13 January 2023; Accepted 3 February 2023
Available online 8 February 2023

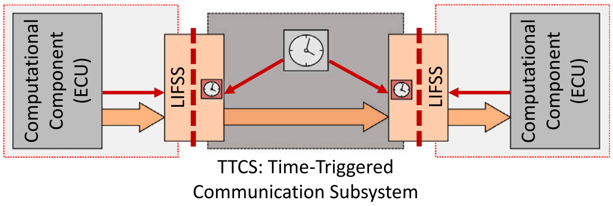

Computational Component (ECU)
LIFSS
LIFSS
Computational Component (ECU)
TTCS: Time-Triggered Communication Subsystem

Fig. 1. Time-triggered system model [10].

foundations of time-triggered communication systems, which sets the architectural focus for the security analysis. Section 2.2 lists the identified security-related communication structures and processes that we use in our security analysis. Section 2.3 describes the concrete security attacks that we consider in our threat model.

2.1. Time-Triggered Communication Subsystem (TTCS)

In a distributed time-triggered (TT) system, the real-time communication network consists of the *time-triggered communication subsystem* (TTCS) and the *computational components* (CCs), in the automotive jargon also called *Electronic Control Units* (ECUs). The CCs connect to the TTCS via the *linking interface subsystem* (LIFSS), as shown in Fig. 1.

The TTCS is an autonomous subsystem that transports messages in a time-predictable way. The red arrows coming out from the central clock in Fig. 1 indicate that the transmission of messages is time-controlled. This central clock providing a global time is only a virtual concept, as each LIFSS has its local clock, and via clock synchronisation they create their joint view of a global time. The sender CC and receiver CC are temporally decoupled from the TTCS. The sender CC writing a message into the buffer of its LIFSS is called an information-push interface, as the sender CC has temporal autonomy of when to do so. The receiver CC reading a message from the buffer of its LIFSS is called an information-pull interface, as the receiver CC has also temporal autonomy of when to do so.

In its simplest form, the TTCS communication consists of periodical communication rounds, where each CC has a communication slot assigned. The CC is only allowed to send within its own communication slot. To increase time predictability, one can align the task scheduling with the TTCS communication [11]. Assigning the slots to individual tasks of the system is also an optimisation problem to minimise communication latency [10].

As discussed in more detail in Section 3, there are different realisations of the TT communication model with so-called bus guardians to ensure the LIFSS of a CC can only transmit within its own communication slot.

2.2. Identified security-relevant communication properties

In the following we describe security-related communication structures and processes we use in the security analysis. These terms themselves are relevant for communication networks in general and are not specific to TT communication.

Network segmentation: is the separation of a network into individual network segments, which are connected via gateways. In case that two network segments are of the same networking model, then their connecting gateways is called a *network switch*.

From a security perspective, network segmentation is beneficial, as it allows to control via the gateway what messages are forwarded between the network segments.

Message authentication: is the assurance that a message comes from a certain sender. Message authentication is not to be confused with device authentication, as the latter is used to allow a node to send and receive messages on the network. Message authentication is a by-product of message non-repudiation, but message authentication could be also achieved with methods less resource-intensive than message non-repudiation.

From a security point of view, message authentication is important for a system, as certain actions are supposed to be triggered only by a certain node or set of nodes.

Message non-repudiation: is the proof that the received message came from a certain sender. Message non-repudiation is a stronger concept than message authentication, as message non-repudiation provides verifiable proof that the message content came from the sender. Message non-repudiation is typically achieved by adding some signature to the message that links a sender identifier and the message content together.

Crash-detectability of node: is the ability to notice whenever a node has failed with a fail-silent behaviour. One way to implement crash-detectability would be to require nodes to send a so-called *heartbeat signal*, which is a message sent at periodical intervals. Sending such heartbeat signals in addition to the application-specific messages causes an overhead on the network bandwidth. Thus, it is of interest if the regular behaviour of the communication bus includes a message pattern that works as heartbeat signal.

Crash-detectability is primarily a safety concern, but it is also a security concern, as it allows to detect situations where an attack has crashed or deactivated a network node.

Data confinement: is about mechanisms to limit the physical accessibility of data among individual nodes within the network. Data confinement is not about hiding data via encryption, but rather preventing data to physically reach non-intended network nodes.

Data confinement is important for security, as it can help to implement *confidentiality* of information. The benefit of data confinement is that it tends to require less computing overhead and causes less response-time delay compared to data hiding via encryption. These reduced resource demands are crucial in automotive communication systems where computing power is limited and many services have strict real-time requirements.

Data encryption: is used to convert data into a secret code via a key to prevent unauthorised access. The inverse process of converting a secret code via a key is called decryption. There are different types encryption available, e.g., symmetric or asymmetric encryption, etc. The compared network types themselves do not imply the usage of any encryption method. Thus, if encryption is needed, it has to be implemented on top of the compared network types. While encryption is standard in many application domains, in vehicular networks it is often omitted, as encryption might interfere with the real-time requirements, causing extra communication delay. For concrete network technologies there might be an additional prohibiter to the use of encryption as that might increase the messages to be sent, causing problems with the available network bandwidth for the used application.

Bus domination: is the ability by one or more network nodes to saturate the available communication bus bandwidth with their messages, suppressing any message of other nodes. Bus domination can be caused by priority-based network arbitration without measures to prevent it.

The possibility of bus domination is a critical security threat, as it would allow a misbehaving network node or nodes to monopolise the communication bus.

2.3. Network attacks

In the following we discuss the security attacks relevant to the communication network that we consider in our threat model. In particular we look at DoS attacks, which can address either the communication medium via bus/network jamming or the network nodes. DoS attacks targeting the network (Network-DoS) are aiming to prevent the exchange of normal messages by blocking the normal application-specific messages. DoS attacks targeting the network nodes (Node-DoS) do not focus on preventing normal application-specific messages, but rather on preventing a network node from providing its intended service. Node-DoS can be done by overloading the network node with spurious requests or by trying to put the network node in a state where it its intended service is disabled.

Bus jamming via bus domination: is a form of Network-DoS where the ability of bus domination is exploited for a *denial of service* (DoS) attack against all other nodes with lower bus arbitration priority. Bus jamming via bus domination is based on sending valid messages from a higher-priority node, thus forcing the lower-priority nodes into an indefinite waiting till network access becomes available.

While bus jamming via bus domination is a serious threat in networks that exhibit bus domination, it also means that network nodes of lower priority cannot perform such an attack on nodes with higher priority.

Bus jamming via protocol violation: is a form of Network-DoS where an attack is done by sending messages that violate the behaviour that is demanded by the bus/network protocol. For example, in a controller–responder network a responder node is normally only allowed to send when requested by the controller node. If a slave node has been compromised it can start sending messages on its own, causing a bandwidth shortage on the communication bus/network.

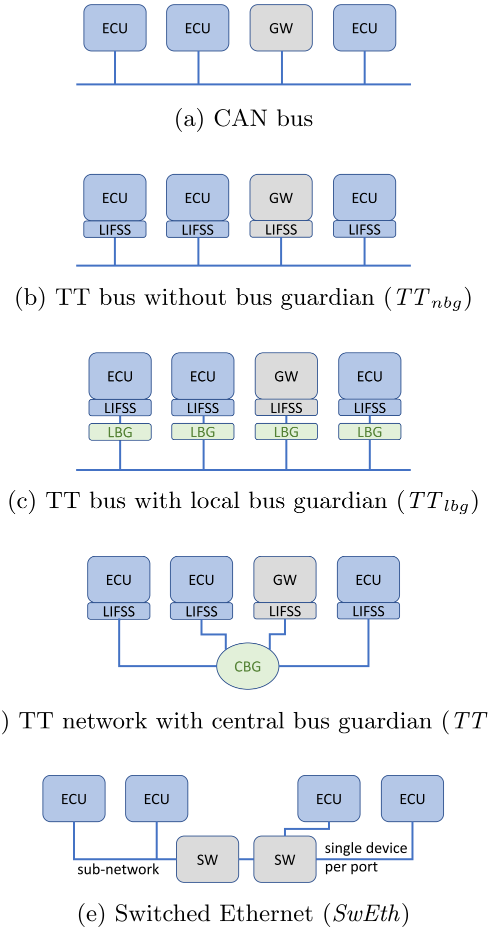
A key characteristic of bus jamming via protocol violation is that it allows to identify the compromised network node rather swiftly by an observer on the network.

Bus jamming via access conflicts: is a form of Network-DoS where compromised network nodes try to send frequently messages on the bus, causing access conflicts on the medium, with the result that those network nodes involved in the access conflict have to abort their send operation and try again later. With bus jamming via access conflicts the attack pattern may not even result in additional messages sent over the network, as in the extreme case it can cause only a sequence of access conflicts, which prevents the normal sender from successfully sending a message.

For individual events of bus jamming via access conflicts it is not possible to identify who is the compromised node, as it will take the observation of multiple send attempts to identify a misbehaving node.

ECU-DoS: is a node-DoS. ECU stands for *Electronic Control Unit*, which is a common name for computing devices in a network in the automotive domain. We distinguish between network node and ECU, as, depending on the type of communication bus, a network node can besides the ECU also include other components. For example, a TT bus includes also a local bus guardian, as shown in Fig. 2(c). Theoretically, a node-DoS could attack the ECU as well as other components like the local bus guardian.

The ECU-DoS attack aims to prevent an ECU from providing its intended service. This could be either done via request flooding to overload the ECU, or to put it into state where it does not aim to provide the service at all.



(a) CAN bus

(b) TT bus without bus guardian (TT_{nbg})

(c) TT bus with local bus guardian (TT_{lbg})

(d) TT network with central bus guardian (TT_{*})

(e) Switched Ethernet ($SwEth$)

Fig. 2. Visualisation of network types.

2.4. Detection of network attacks

In this section we discuss detectability of the network security attacks listed in Section 2.3.

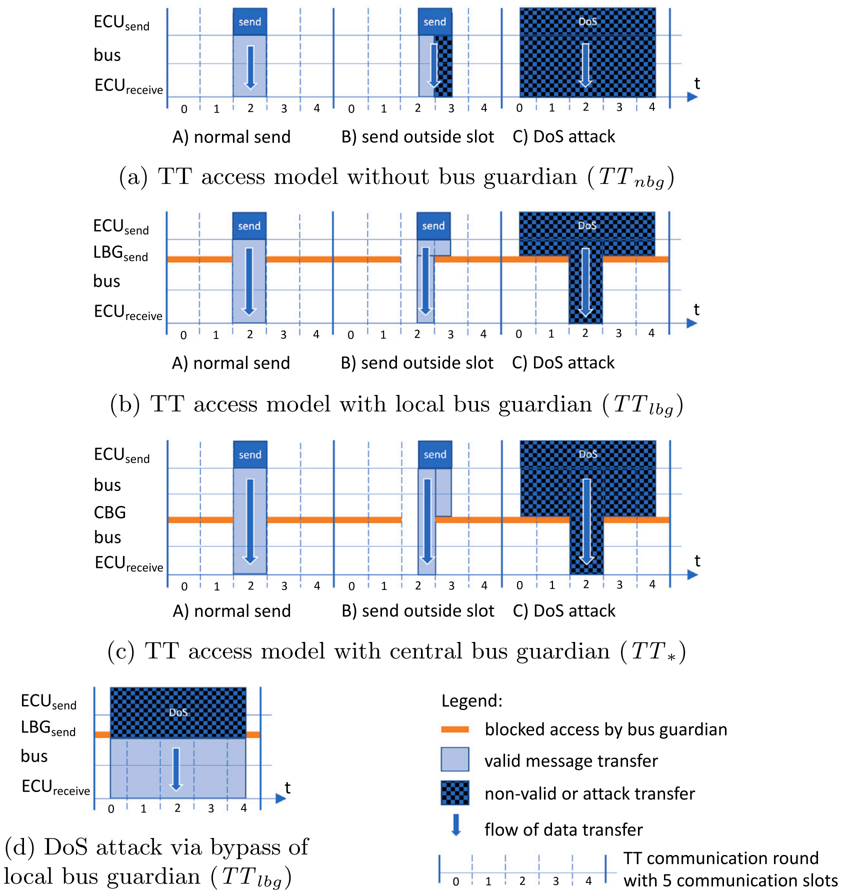
DoS detection: is about the detection that a DoS attack is taking place or had been taken place. Bus jamming via protocol violation tends to be quite swiftly detectable, because the correct behaviour according to the communication protocol can be monitored, and as soon as a violation is detected, it can be recognised. Bus jamming via bus domination cannot be detected immediately, as by definition, all the DoS traffic can be still valid messages. Similar, a DoS attack based on bus jamming via access conflicts cannot be detected immediately, as the occurrence of access conflicts can be a normal behaviour, and only after observing and unusual pattern a detection is possible. The time frame that an ECU-DoS can be detected differs depending on what established behaviour a node has to have in a certain communication system.

3. In-vehicle network types

In this section we analyse the impact of network structures on the security of different TT networks and other network types.

3.1. CAN bus (symmetric communication bus)

In this section, we discuss *symmetric communication buses* (SCB), i.e., communication buses where each node has the same communica-



(a) TT access model without bus guardian (TT_{nbg})

(b) TT access model with local bus guardian (TT_{lbg})

(c) TT access model with central bus guardian (TT_*)

(d) DoS attack via bypass of local bus guardian (TT_{lbg})

Fig. 3. Access model of different TT network types.

tion autonomy. In addition, we mention specific properties of *Controller Area Network* ($CAN\ Bus$), which is an instance of SCB.

SCBs are in contrast to other principles like controller–responder communication, where nodes have different communication roles. The CAN bus is a representative of an SCB with event-triggered communication. Fig. 2(a) shows the concept of an SCB. As shown in Fig. 2(a), an SCB can have special nodes like a gateway, but these nodes would still have the same communication autonomy as the other nodes.

For any SCB with event-triggered communication there is a need for an arbitration method to provide contention resolution. CAN bus provides a lossless bitwise arbitration method, which means that contention situations do not cause any waste of communication bandwidth. CAN's lossless bitwise arbitration is based on the principle that a logical 0 on the physical layer is dominant over a logical 1, if one node wants to write a logical 0 on the bus and another node want to write a logical 1 on the bus, then the resulting signal level on the bus is logical 0, i.e., logical 0 is the dominant signal level. The arbitration method in CAN works that way that each message starts with a bit-sequence that is the unique identifier (ID) of the sender node. When a node writes its own unique ID on the bus, the node at the same time samples the signal level on the bus. If there is a mismatch between the sent signal level and the sampled signal level, this means that another node at the same time also tried to start sending, and the node that observed the signal mismatch interrupts its message transmission attempt. While this resolves the contention without any loss of bandwidth, it at the same time implies that the ID of each node introduces a priority ranking in the contention resolution. For example, an ID starting with a 0 is higher prior than an ID starting with a 1, and so on. Since the ID of each node is unique, this way the CAN bus provides a total order of the communication priorities of all nodes. In particular, the node with the highest-prior ID, i.e., the least ID number, is able to dominate the bus by sending continuously messages, which would block any write attempt on the bus from all other nodes.

From a security perspective, the possibility of bus domination of CAN is a serious concern, as it allows denial-of-service (DoS) attacks on the bus by a node. However, the extent of the bus jamming that a particular CAN bus node is capable of doing, depends on the ID of that node. This makes CAN quite outstanding among the SCBs, as the maximum severity of a bus jamming attack is different for each node. In contrast, with another SCB like non-switched Ethernet, each node would be able to perform a bus jamming attack on the bus with maximum extent. While the classic purpose of the CAN bus arbitration method was to implement access priorities without loss, this mechanism would also open doors to optimise the limitation of bus jamming attacks for the higher protection of the more critical services.

3.2. *TT without bus guardian (TT_{nbg})*

In this section we discuss the simplest form TT communication networks, namely those without any mechanism to protect the communication from misbehaving network nodes. We denote this type of TT network as TT_{nbg}, with *nbg* standing for "no bus guardian". While TT_{nbg} has no independent protection mechanism, it still has a *linking interface subsystem* (LIFSS), as shown in Fig. 2(b). The LIFSS acts as a temporal firewall, i.e., it is responsible to synchronise the communication of the local node with the timing of the TT bus. Typically, the LIFSS buffers messages to be sent and only starts writing them to the TT bus as soon as the communication slot of the local node has started.

The case (A) of Fig. 3(a) shows an example of a normal message communication from a sender ECU to the receivers within its assigned timeslot (slot no. 2). In this example the communication round has five sender slots (slots 0–4). The case (B) of Fig. 3(a) shows the case where a message is sent by the sender ECU partically outside its assigned timeslot (slot no. 2), which results in an invalid communication, as it would interfere with messages sent by other ECUs.

The structured communication of TT communication networks has some useful security implications. For once, the sender of a message can be identified by the time slot when the message is sent (case A) of Fig. 3(a). However, with TT_{nbg} the risk is that compromised nodes can ignore their assigned sending slots, and cause problems (collisions) on the communication bus (cases B and C) of Fig. 3(a)).

3.3. TT with local bus guardian (TT_{lbg})

TT_{lbg} is an extension of TT_{nbg}, by adding a local bus guardian to each ECU, as shown in Fig. 2(c). The local bus guardian provides a temporal firewall, allowing messages from its local ECU to be only sent during its assigned time slot. For dependability reasons of the system, the local bus guardian is supposed to be separated from the ECU.

The case (A) of Fig. 3(b) shows an example of a normal message communication from a sender ECU to the receivers within its assigned timeslot (slot no. 2). The horizontal orange bars indicate the time where the LBG prevents transmission of messages. To do so, the LBG also knows the time slot(s) of its ECU. Case (B) of Fig. 3(b) shows the scenario where a messages was attempted to be sent partially outside the sender's time slot, where the LBG cuts off any communication outside it own time slot.

This separation is also relevant for security, as with a compromised ECU the attacker cannot bypass the temporal firewall of the local bus guardian, thus disabling interference outside the ECU's own sending slot.

3.4. TT with centralised bus guardian (TT_*)

TT_* is an extension of TT_{nbg}, by adding a central network guardian to the network, as shown in Fig. 2(d). Since the individual network nodes are connected in a star shape to the network guardian, this central network guardian is also called *star coupler*. The use of a central bus guardian has been motivated from a safety point of view, as it avoids vulnerability to slightly-off-specification failures that networks with a local bus guardian may exhibit [12].

Analogous to Fig. 3(b) for TT_{lbg}, the cases (A) and (B) of Fig. 3(c) show the normal communication of a message and the cut off transmission of the message when attempted to send outside its time slot (slot no. 2). The fundamental difference between TT_* and TT_{lbg} is shown in case (B), where for TT_{lbg} sending outside its time slot is cut off locally, while with TT_* the message still gets send to the CBG, which would then detect the invalid timing (outside of slot no. 2) of the message and terminates the sending.

From a security point of view, the central bus guardian provides the same benefits like the local bus guardian. However, the central bus guardian would have an additional advantage in case someone gets physical access to the network. Having physical access to a network node with local bus guardian would allow an attacker to replace the node with another variant without the bus guardian, thus removing the protection provided by the bus guardian. With a central bus guardian the bus guardian is physically separated from the network nodes, thus disabling jamming on the network even though a local ECU gets physically replaced by an attacker.

It should be noted here that if the centralised bus guardian itself can also be compromised by an attacker, this would give the attacker full control over all the messages sent in the network. If we assume an attack model with a compromised centralised bus guardian, then a way to mitigate this would be to duplicate the network links using a second bus guardian. However, a duplication of the network and star coupler would not be sufficient to identify modified messages sent out by the compromised star coupler.

3.5. Switched ethernet ($SwEth$)

Switched Ethernet ($SwEth$) allows to connect multiple Ethernet nodes via a network switch, as shown in Fig. 2(e). In vehicular systems a network technology based on that concept is *Time-Sensitive Networking* (TSN) [13]. TSN was designed with focus on supporting deterministic Ethernet communication. In $SwEth$ the deterministic behaviour comes from the network switches, allowing to combined standard Ethernet devices with other devices requiring deterministic communication behaviour. The use of switches in TSN allows to implement time-triggered communication behaviour for a subset of connected ECUs.

However, as shown in Fig. 2(e), on a port of a switch in $SwEth$ one can either connect a single device (as shown on the right) or connect a sub-network consisting of multiple devices (as shown on the left). This increased flexibility of the network topology, however, comes with lower guaranteed security-related protections compared to TT_* networks. For example, for messages coming from a sub-network in $SwEth$ the switch has no possibility to provide temporal firewall guarantees within this sub-network in the same simple way as it is possible with the central bus guardian of TT_*.

4. Security analysis of different network types

In this section we compare the different communication networks that have been described in Section 3. This comparison focuses on their structural properties and how they help against the different attack types described in Section 2. This is a qualitative comparison, where we used fundamental behaviour principles of different communication networks to derive from that relevant cybersecurity implications.

The summary of the comparison is shown in Table 1. A horizontal double line in Table 1 separates different classes of vulnerability issues: 1. communication mechanisms, 2. concrete attack types, and 3. attack detection.

4.1. Identification of security-relevant of network properties

In the following we identify a number of security-relevant network properties feeding to our security analysis of TT communication networks.

Network segmentation requires some gateway to connect multiple networks. While to any of the discussed networks a gateway can be added, it is TT_* and $SwEth$ which have switches as standard components of their network. $CAN\ Bus$, TT_{nbg}, and TT_{lbg} do not inherently need such a network switch, this is why we did not list network segmentation in Table 1. With $SwEth$ it is quite likely that network segmentation in automotive systems is used, as this allows a better exploitation of the high data rates offered, for example, by TSN.

Message authentication is supported by the assignment of sending slots to each node in TT systems. Even with TT_{nbg} one would know the sender's identity. If a compromised nodes wants to send at the wrong slot, then this would cause a conflict with the node assigned to that slot. The same is also true for the TT variants with bus guardians, TT_{lbg} and TT_*. In $CAN\ Bus$ message authentication is not supported by the bus itself, as any node can fabricate a message with wrong sender id. With $SwEth$ a support of message authentication would depend on the implementation, but where each node is connected to a separate switch port the authentication is given to the switch based on the port.

Message non-repudiation is not supported by $CAN\ Bus$. With all TT variants (TT_{nbg}, TT_{lbg}, TT_*) message non-repudiation is not assured by itself. However, with TT networks one can add a trusted monitor on the network to log traffic with time stamps. Since TT networks assure message authentication, such a trusted monitor would then know that a valid message was sent from the supposed sender. But this only works in combination with a trusted monitor node. From the message alone there would be no complete mechanism of non-repudiation. With

Table 1
Security analysis of the different communication types: CAN Bus and Switched Ethernet ($SwEth$) vs. time-triggered communication (TT_{nbg}, TT_{lbg}, TT_*).

Vulnerability issues	*CAN*	TT_{nbg}	TT_{lbg}	TT_*	$SwEth$
Communication mechanisms:					
Network segmentation	none	none	none	yes (via switch)	yes
Message authentication	none	yes (via slot)	yes (via slot)	yes (via switch)	yes (port level)
Message Non-repudiation	none	supported (via slot)	supported (via slot)	supported (via switch)	limited (port)
Crash detectability	delayed	immediate	immediate	immediate	delayed
Data confinement	none	none	possible	possible	limited (subnet)
Data encryption	limited	possible	possible	possible	possible
Concrete network attack types:					
Bus jamming: bus domination	possible	impossible (conflict)	prevented	prevented	prevented
Bus jamming: protocol violation	possible	possible	prevented	prevented	limited (subnet)
Bus jamming:: access conflicts	impossible	impossible	impossible	impossible	limited (non-RT)
ECU-DoS via overload	possible	impossible	impossible	impossible	possible (preventable)
ECU-DoS via deactivation	possible	possible	possible	possible	possible
Network attack detection:					
DoS detection	delayed	immediate	immediate	immediate	delayed

$SwEth$ it would depend on the implementation, but it could follow the scheme described for TT networks.

Crash detectability is not directly supported by $CAN\ Bus$ and would require some application-specific timeout. In TT networks (TT_{nbg}, TT_{lbg}, TT_*) each node is required to send a message in each communication round, which provides a direct way to detect if a node has crashed. In $SwEth$ it would depend on the concrete implementation whether crash detection is supported by the communication system.

Data encryption is not directly built into the discussed network types ($CAN\ Bus$, TT_{nbg}, TT_{lbg}, TT_*, $SwEth$), and would need a solution at application level. With $CAN\ Bus$ there are limitations for the use of encryption, e.g., the small size message size of 8 bytes in classic CAN. However, CAN FD has got a larger message size of 64 bytes, lifting CAN's limitation in that respect. The other aspect that limits the use of encryption for the real-time control is that encryption introduces delays due to computing overhead.

Data confinement is not possible with CAN or TT_{nbg}, as they are buses with broadcast communication. TT networks with a bus guardian (TT_{lbg} or TT_*) in principle would allow the use of data confinement by building that functionality into the bus guardian. Generally, TT_* would provide stronger data confinement as the bus guardian is separated from the network nodes. The bus guardian would need to filter the data payload in real-time while forwarding the empty frames, as the TT protocols typically need to broadcast messages in order to synchronise the clocks of all nodes. With $SwEth$ the same type of data containment as in TT_* would be possible, limited to the configurations where only one ECU is connected to a port of the network switch.

4.2. Comparison of network attacks

Bus jamming via bus domination is a specific attack only possible on $CAN\ Bus$ among the network types we consider in this comparison, as none of the other network types exhibit bus domination. Bus domination is often mentioned as a particular vulnerability of $CAN\ Bus$. However, we want to provide a different view on that. Assuming that an attacker on a compromised node is not able to modify the network stack in order to implement the other bus jamming attacks discussed below, then bus domination can be also used as a security feature. The idea is that the nodes providing services of higher criticality get also a higher bus arbitration priority assigned. This way, a compromised node can only provide a bus jamming attack on nodes with lower priority, but not to those of higher priority.

Bus jamming via protocol violation would be possible in $CAN\ Bus$ and TT_{nbg}, as they lack a bus guardian mechanism. In Fig. 3(a) the case (C) shows that for TT_{nbg} there is no protection from jamming via protocol violation. With TT_{lbg} and TT_* bus jamming via protocol violation does not work due to the temporal firewall provided by the bus guardian. The case (C) in Figs. 3(b) and 3(c) shows that the bus guardian of TT_{lbg} (local) and TT_* (central) can protect against jamming via protocol violation. However, there is still a different level of protection between TT_{lbg} and TT_* in the special case that an attacker has physical access and is able to replace a computing node: with TT_* the CBG still protects the bus (still protected as shown with case C) in Fig. 3(c), while with TT_{lbg} the computing node would be replaced with an attack node without a LBG (Fig. 3(d)). In $SwEth$ it would be possible to implement a temporal firewall to avoid such an attack, limited to those configurations with only one network node per port of the network switch.

Bus jamming via access conflicts is not possible in $CAN\ Bus$ as it has a loss-less bus arbitration based on priorities. With TT_{nbg} bus jamming is possible, but it would have to be via protocol violation, as discussed above. With TT_{lbg} and TT_* no bus jamming is possible due to the bus guardians, though TT_{lbg} is vulnerable if the attacker can physically modifications to bypass the local bus guardian. In $SwEth$ it would be possible to implement a temporal firewall to avoid such an attack, again limited to those configurations with only one network node per port of the network switch.

ECU-DoS via overload could be possible in $CAN\ Bus$ as there is no mechanism to prevent it. In TT networks (TT_{nbg}, TT_{lbg}, TT_*) a ECU-DoS via overload attack is not possible, as each node is not able to send more often than the scheduled message slots per communication round. This is even true for TT_{nbg}, as additional sending attempts would cause a Bus jamming via protocol violation, but not valid messages would result out of that. In $SwEth$ it would be possible to implement a temporal firewall to avoid such an attack, again limited to those configurations with only one network node per port of the network switch. Subnetworks in $SwEth$ with nodes directly connected to the same bus cannot have a protection at the network level.

ECU-DoS via deactivation is an attack that is specific to the internals of an ECU and does not rely on achieving any abnormal network traffic. Thus, none of the discussed network types can provide a protection against that.

DoS detection is not supported by the mechanisms of $CAN\ Bus$: Network-DoS in case of bus jamming via bus domination does not cause any invalid messages and can only be observed by detecting an unusual pattern after multiple messages; ECU-DoS cannot be detected at the network level, as nodes are not expected to exhibit a period activity like a heartbeat signal. In TT networks (TT_{nbg}, TT_{lbg}, TT_*) it is possible to immediately (within a communication round) detect a Network-Dos and also a ECU-DoS, as there is a pre-established communication plan with periodic behaviour that all nodes must adhere to. In $SwEth$ it would be possible to implement such DoS detectability by using a communication pattern as in the TT networks.

4.3. Discussion

In this section we summarise the cybersecurity benefits of TT communication in Section 4.3.1. To balance this, Section 4.3.2 give an insight into known cybersecurity challenges of TT communication.

4.3.1. Cybersecurity benefits of TT communication

Our security analysis in Section 4.2 has shown that $CAN\ Bus$, and basically also other *symmetric communication buses* (SCB), can provide rather minimal security support at the network level. We have shown that TT networks (TT_{nbg}, TT_{lbg}, TT_*) provide (and with the guardian of TT_{lbg} and TT_* even enforce) a stronger structure of the communication, which also has a positive impact on the system security. $SwEth$, like TSN, provides a lot of flexibility on how to implement the network communication. Thus, to retain some of the security properties provided by TT networks, similar mechanisms would also be needed to be implemented in $SwEth$.

As an additional coincidental result of our security analysis, we found that regarding security the often criticised possibility of bus domination in $CAN\ Bus$ is not that bad after all, as it can be also used in an advantageous way to protect more critical services from Network-DoS attacks via bus domination by compromised network nodes of lower criticality.

4.3.2. Cybersecurity challenges of TT communication

The results of this cybersecurity analysis showed different beneficial security properties of TT communication networks. However, it has to be pointed out that these security properties are only uphold if the underlying implementation details of the protocol are robust enough against cybersecurity attacks. Skopik et al. discuss different critical services of TT communication that need to be designed in a cybersecurity-robust way, for example, the clock synchronisation service among computing nodes is of utmost importance [14].

Recently, Loveless et al. have presented PCSPOOF, a successful attack on the synchronisation protocol of TTEthernet, which is a TT network that uses central bus guardians (TT_*) [15]. TTEthernet allows to combine Ethernet devices with best-effort timing requirements and real-time requirements via star couples, which are central bus guardians (CBG) [16]. The authors were able to infer information of the real-time network from best-effort nodes, which they used to create malicious synchronisation messages. In addition, the authors inject high-voltage electrical noise on the Ethernet cable of the CBG, causing it to send these malicious synchronisation messages. As a result, TTEthernet devices could loose synchronisation for up to one second, causing the loss of tens of TTEthernet messages.

Another possible attack to TT communication systems is exploiting the predictable behaviour of the schedule, to produce timing attacks on critical tasks by consuming shared resources of these tasks when needed. The defence against such timing attacks on TT communication systems is the use of more complex TT scheduling policies, resulting in less predictable scheduling of TT activities [17–19].

Besides the TT-specific attacks, there are, of course, also generic network attacks that can also impact TT communication systems. For example, attacks based on intentional electromagnetic interference (IEMI) transmit fault signals from an antenna via radiation on the network cable to interfere with the network communication [20]. These IEMI-based attacks work usually only for unshielded communication cables. Another DoS attack for networks in general would be the destruction of network devices via high-voltage circuits [21].

5. Related work

El-Rewini et al. provide a security analysis of in-vehicle communications and external (V2X) communications [22].

Trawczynski et al. have done DoS attack detection by exploiting the predictable behaviour of TT systems and detect deviations from the expected deterministic behaviour [23].

A specific security concern of highly predictable time-triggered real-time systems is that their predictable behaviour allows for targeted security attacks. Yoon et al. have proposed *TaskShuffler*, a schedule randomisation generator to reduce the chances of timing-inference attacks [17]. Krüger et al. addressed the risks of timing-inference attacks specifically for time-triggered systems [18,19]. The authors focus on the issue that redundancy via task replication for safety–critical applications might not prove to be effective in case of joints attacks on all the replicas. Jointly targeted attacks on all task replicas would violate the fault-independence assumption such systems are built upon. The authors propose two runtime mitigation strategies to defend against timing-inference attacks, namely to do schedule randomisation at slot level and to do randomisation within a set of offline constructed schedules. While this research has focused on attack risk reduction via randomisation of whole systems, we focus on the security provisions from the communication network.

Püllen et al. discuss an extension of the time-triggered communication protocol FlexRay to provide message authentication based on the transmission of *Message Authentication Codes* (MACs) [24]. In contrast, our focus is on security provisions from the communication protocol, without relying on additional cryptographic methods.

The security challenges of $CAN\ Bus$ have been studied by many researchers. Jadhav and Kshirsagar compared the security challenges of the automotive protocols LIN, CAN, MOST, and FlexRay [25]. Aliwa et al. compared the security challenges of the automotive protocols LIN, CAN, and FlexRay [26]. Bozdal et al. studied attacks and potential solutions especially for the $CAN\ Bus$ protocol. Zhang et al. developed CANsec, a security evaluation tool to simulate attacks in CAN networks [27]. Nowdehi et al. compared extensions of $CAN\ Bus$ to provide message authentication [9].

Takahashi et al. did an analysis of security attacks and possible countermeasures on the LIN bus [28]. We did not include LIN Bus in our comparison, as it is a different class of network, focussing on cost-efficient automotive sensor networks.

Murvay and Groza have demonstrated the possibility of attacks based on discarding messages and message spoofing on the standard FlexRay protocol [29]. Their message spoofing is based on inserting adversarial frames and later discarding the genuine frames. Their attack analysis is based on the assumption that an attacker has the ability to modify the firmware, either via update through channels like ODB or OTA or by simply physically replacing a local node. While their attacks are demonstrated on a directly connected network bus, the authors also discuss specific weaknesses of the FlexRay star coupler that might also allow their attacks in a star-shaped network with a star coupler.

From our classification, the authors did use bus jamming via protocol violation as well as bus jamming via access conflicts.

TSN combines best-effort and real-time traffic, hence security is also very important. Feng et al. present a security analysis of TSN [30]. They show what different defence mechanisms against network attacks TSN has in place, for example, traffic filtering. Meyer at al. show that with TSN a metering-based traffic filtering can be implemented to counter DoS attacks [31]. Muguira et al. did an implementation of wire-speed cryptography on TSN for the electric sector [32]. Lin and Yu argue that to achieve safety and security in Ethernet-based automotive networks require tradeoffs [33]. Ergenc et al. present and discuss a list of 30 potential security issues in TSN [34]. Luo et al. worked on routing and security mechanisms for automotive TSN and $CAN\ Bus$ [35]. Liu et al. presented some work on a cybersecurity testbed to test the robustness of a TSN system against DoS attacks [35]. Current research on TSN security has provided evidence that TSN needs structured solutions to support security.

None of the described security analyses discusses the specific properties of TT communication networks in detail. This article specifically focuses on a security analysis of TT communication networks. These concepts could be useful for TSN, where the choice of a TT-based traffic filtering, as analysed in this article, would allow a robust jitter in case of DoS attacks.

6. Summary and conclusion

In this paper we provided a qualitative cybersecurity analysis of different time-triggered (TT) communication networks. In order to cover a wide scope relevant for future automotive communication networks, our analysis includes $CAN\ Bus$ as a baseline and also switched Ethernet ($SwEth$) like TSN as a candidate for future automotive communication networks. Furthermore, we include TT communication protocols with different levels of temporal behaviour assurance (see TT_{nbg}, TT_{lbg}, TT_{*}) into our security analysis.

In our security analysis we found that the structured communication behaviour of TT networks and their assurance via a so-called bus guardian provide beneficial security assurances. While $SwEth$ networks as future automotive networks allow for a very flexible network communication structure, we argue that for those services where security properties similar to TT networks are required, the $SwEth$ networks are recommended to implement the same structured communication and assurance that we identified for TT networks. However, as shown by concrete examples, it can be a challenge to make sure that the concrete implementation of the network protocol guarantees the identified security properties.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] Hermann Kopetz, Real-Time Systems - Design Principles for Distributed Embedded Applications, second ed., Springer, London, UK, ISBN: 978-1-4419-8236-0, 2011.

[2] Hermann Kopetz, Günther Bauer, The time-triggered architecture, Proc. IEEE 91 (1) (2003) 112–126.

[3] Stefan Poledna, Wolfgang Ettlmayr, Markus Novak, Communication bus for automotive applications, in: Proc. 27th European Solid-State Circuits Conference, 2001, pp. 482–485.

[4] Stefan Poledna, Hermann Kopetz, Wilfried Steiner, Deterministic system design with time-triggered technology, in: Proc. Microelectronic Systems Symposium, MESS, 2014, pp. 1–4.

[5] ISO, ISO 17458-1:2013 Road Vehicles — FlexRay Communications System — Part 1: General Information and Use Case Definition, International Standards Organisation, New York, USA, 2013, Technical Committee, ISO/TC 22/SC 31 Data communication, this document is part of the FlexRay standard documents ISO 17458-1 to 17458-5, reviewed and confirmed in 2019.

[6] International Standards Organisation, ISO/IEC 21806-1:2020: Road vehicles — Media Oriented Systems Transport (MOST) — Part 1: General information and definitions, International Organization for Standardization, New York, 2020, Technical Committee: ISO/TC 22, Road vehicles, Subcommittee SC 31, Data communication.

[7] Lucia Lo-Bello, Wilfried Steiner, A perspective on ieee time-sensitive networking for industrial communication and automation systems, Proc. IEEE 107 (6) (2019) 1094–1120.

[8] Mehmet Bozdal, Mohammad Samie, Sohaib Aslam, Ian Jennions, Evaluation of CAN Bus security challenges, Sensors 20 (8) (2020).

[9] Nasser Nowdehi, Aljoscha Lautenbach, Tomas Olovsson, In-vehicle CAN message authentication: An evaluation based on industrial criteria, in: Proc. 86th IEEE Vehicular Technology Conference, VTC-Fall, IEEE, 2017.

[10] Raimund Kirner, Peter Puschner, A quantitative analysis of interfaces to time-triggered communication buses, IEEE/ACM Trans. Netw. 29 (4) (2021) 1786–1797.

[11] Peter Puschner, Raimund Kirner, Asynchronous vs. synchronous interfacing to time-triggered communication systems, J. Syst. Archit. 103 (2020) Special Issue on the 2019 IEEE Symposium on Real-time Computing ISORC (SI:ISORC19).

[12] Astrit Ademaj, Slightly-off-specification failures in the time-triggered architecture, in: Proc. 7th IEEE International Workshop on High Level Design Validation and Test, Cannes, France, 2002, pp. 7–12.

[13] Mohammad Ashjaei, Lucia Lo Bello, Masoud Daneshtalab, Gaetano Patti, Sergio Saponara, Saad Mubeen, Time-sensitive networking in automotive embedded systems: State of the art and research opportunities, J. Syst. Archit. 117 (2021) 102137.

[14] Florian Skopik, Albert Treytl, Arjan Geven, Bernd Hirschler, Thomas Bleier, Andreas Eckel, Christian El-Salloum, Armin Wasicek, Towards secure time-triggered systems, in: Proc. Computer Safety, Reliability, and Security, SAFECOMP'12 Workshops, in: Lecture Notes in Computer Science, vol. 7613, Springer, 2012, pp. 365–372.

[15] Andrew Loveless, Linh Thi Xuan Phan, Ronald Dreslinski, Baris Kasikci, PCSPOOF: Compromising the safety of time-triggered ethernet, in: Proc. 44th IEEE Symposium on Security and Privacy, SSP'23, IEEE, 2023.

[16] SAE, SAE Standard AS6802: Time-Triggered Ethernet, SAE International, Warrendale, Pennsylvania, USA, 2011.

[17] Man-Ki Yoon, Sibin Mohan, Chien-Ying Chen, Lui Sha, TaskShuffler: A schedule randomization protocol for obfuscation against timing inference attacks in real-time systems, in: Proc. IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS, IEEE, 2016, pp. 1–12.

[18] Kristin Krüger, Gerhard Fohler, Marcus Völp, Paulo Esteves-Veríssimo, Improving security for time-triggered real-time systems with task replication, in: Proc. 24th Int'l Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA'18, IEEE, 2018, pp. 232–233.

[19] Kristin Krüger, Nils Vreman, Richard Pates, Martina Maggio, Marcus Völp, Gerhard Fohler, Randomization as mitigation of directed timing inference based attacks on time-triggered real-time systems with task replication, Leibniz Trans. Embedded Syst. 7 (1) (2021).

[20] Matthias Kreitlow, Frank Sabath, Heyno Garbe, Analysis of IEMI effects on a computer network in a realistic environment, in: Proc. IEEE Int'l Symposium on Electromagnetic Compatibility, EMC'15, 2015.

[21] Etherkiller — Coming Soon to a NIC Near You, Technical Report, 2022, available online at https://etherkiller.org. (Accessed 28 November 2022).

[22] Zeinab El-Rewini, Karthikeyan Sadatsharan, Daisy Flora Selvaraj, Siby Jose Plathottam, Prakash Ranganathan, Cybersecurity challenges in vehicular communications, Veh. Commun. 23 (2020) 100214.

[23] Dawid Trawczynski, Janusz Zalewski, Janusz Sosnowski, Design of reactive security mechanisms in time-triggered embedded systems, SAE Int. J. Passeng. Cars - Electron. Electr. Syst. 7 (2) (2014) 527–535, SAE World Congress 2014.

[24] Dominik Püllen, Nikolaos Athanasios Anagnostopoulos, Tolga Arul, Stefan Katzenbeisser, Security and safety co-engineering of the FlexRay bus in vehicular networks, in: Proc. Int'l Conference on Omni-Layer Intelligent Systems, COINS'19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 31–37.

[25] Shriram Jadhav, Deepak Kshirsagar, A survey on security in automotive networks, in: Proc. 4th Int'L Conference on Computing Communication Control and Automation, ICCUBEA'18, 2018, pp. 1–6.

[26] Emad Aliwa, Omer Rana, Charith Perera, Peter Burnap, Cyberattacks and countermeasures for in-vehicle networks, ACM Comput. Surv. 54 (1) (2021).

[27] Haichun Zhang, Xu Meng, Xiong Zhang, Zhenglin Liu, CANsec: A practical in-vehicle controller area network security evaluation tool, Sensors 20 (17) (2020).

[28] Junko Takahashi, Yosuke Aragane, Toshiyuki Miyazawa, Hitoshi Fuji, Hirofumi Yamashita, Keita Hayakawa, Shintarou Ukai, Hiroshi Hayakawa, Automotive attacks and countermeasures on LIN-Bus, J. Inf. Process. 25 (2017) 220–228.

[29] Pal-Stefan Murvay, Bogdan Groza, Practical security exploits of the flexray in-vehicle communication protocol, in: Akka Zemmari, Mohamed Mosbah, Nora Cuppens-Boulahia, Frédéric Cuppens (Eds.), Risks and Security of Internet and Systems, Springer, 2019, pp. 172–187.

[30] Feng Luo, Bowen Wang, Zihao Fang, Zhenyu Yang, Yifan Jiang, Security analysis of the tsn backbone architecture and anomaly detection system design based on IEEE 802.1Qci, Secur. Commun. Netw. (2021).

[31] Philipp Meyer, Preventing Dos Attacks in Time Sensitive Networking in-Car Networks Through Credit Based Ingress Metering, Technical Report, CoRE Research Group, Hochschule für Angewandte Wissenschaften Hamburg, 2017.

[32] Leire Muguira, Jesús Lázaro, Sara Alonso, Armando Astarloa, Mikel Rodriguez, Secure critical traffic of the electric sector over time-sensitive networking, in: Proc. 35th Conference on Design of Circuits and Integrated Systems, DCIS, IEEE, 2020, pp. 1–6.

[33] Chung-Wei Lin, Huafeng Yu, Invited: Cooperation or competition? Coexistence of safety and security in next-generation ethernet-based automotive networks, in: Proc. 53rd ACM/EDAC/IEEE Design Automation Conference, DAC, IEEE, 2016.

[34] Doğanalp Ergenç, Cornelia Brülhart, Jens Neumann, Leo Krüger, Mathias Fischer, On the security of IEEE 802.1 Time-Sensitive Networking, in: Proc. IEEE Int'L Conference on Communications Workshops, ICC Workshops, IEEE, 2021, pp. 1–6.

[35] Feng Luo, Zhenyu Yang, Zitong Wang, Jiajia Wang, Routing and security mechanisms design for automotive tsn/can fd security gateway, in: SAE Technical Paper 2022-01-0113, WCX SAE World Congress Experience, SAE, 2022, pp. 1–9.