CAN't track us: Adaptable Privacy for ISOBUS Controller Area Networks

Jan Bauer*, René Helmke, Alexander Bothe, Nils Aschenbruck

Institute of Computer Science, University of Osnabrück,
Wachsbleiche 27, 49090 Osnabrück, Germany

Abstract

Modern information technologies have revolutionized agriculture in many ways. Today, positioning systems along with Internet of Things technologies enable Smart Faming with site-specific applications and interconnected machinery. Current machines and implements are equipped with multiple sensors and actors that are embedded in the machine's Controller Area Network (CAN) using ISOBUS. In operation, such components continuously produce and exchange internal and environmental sensor information. However, CAN communication is not secure and privacy issues arise whenever different actors are involved in collaborative tasks. In this paper, we present *CAN't*, a modular privacy framework for collaborative Smart Farming. The core of the framework is a special proxy that allows to apply privacy filters to selected CAN/ISOBUS data streams. Moreover, it allows to agree on the accessible level of information content for each actor. We implemented a proof-of-concept prototype based on low-cost commercial off-the-shelf hardware. Its evaluation comprises technical and privacy aspects. By using real-world ISOBUS traces as well as a commercial CAN hardware simulator, we show the feasibility of our approach.

Keywords: ISO 11783, ISOBUS, Controller Area Network, Privacy, Data Sovereignty, Precision Agriculture.

1. Introduction

Nowadays, it is commonly agreed that information technology is the key for precision agriculture [1]. Advances in digitalization and modern information technologies have revolutionized industrial agriculture in many ways. They have already contributed significantly to process automation, yield increases, resource optimization, and thus sustainability. Nowadays, the global navigation satellite system (GNSS) along with sensor and Internet of Things (IoT) technologies (cf., [2, 3]) have enabled Smart Faming with site-specific applications,

*Corresponding author.

Email addresses: bauer@uos.de (Jan Bauer), rhelmke@uos.de (René Helmke), bothe@uos.de (Alexander Bothe), aschenbruck@uos.de (Nils Aschenbruck) resource-friendly management, and interconnected actors and machines [1, 4–6]. Moreover, so-called farm management information systems (FMISs) enable a remote management of farm activities that are carried out in the fields. Agricultural machinery can be integrated into such FMISs which in turn can additionally be complemented by external sensors, weather stations, and remote sensing [7].

Modern tractors and implements become more and more complex and effective. They are equipped with a variety of Electronic Control Units (ECUs) such as controllers for various machine and implement components as well as sensors and actors for machine and environmental data [1, 4, 8]. These components are connected using the machines' Controller Area Network (CAN), a reliable and robust internal vehicle binary unit system (bus) and protocol framework that is standardized by the ISO and well-established in the automotive industry [9, 10]. Especially for tractors and machinery for agriculture and forestry, the CAN standard is furthermore extended by the ISOBUS protocol [11, 12]. During agricultural tasks, the sensor and controller components continuously exchange internal and environmental information via CAN. They are controlled and supervised by a specific terminal that. is also responsible for recording all task-related information streams provided by ECUs and sensors, respectively. Hence, stored information may contain sensitive personal information of employees that are exposed to machine owners and other actors having access to the terminal. Moreover, in collaborative tasks, information needs to be exchanged: For instance, if a contractor is carrying out a task on a field, the machine often also inevitably records some sort of field-related data. Here, from the data sovereignty perspective of the farmer, an appropriate data protection would be desirable. Overall, serious privacy and data sovereignty issues arise already at machine level in digitalized and networked Smart Faming.

Currently, there is a lack of technical solutions to protect sensitive data at CAN and ISOBUS level. Thus, collaborating actors have to know and trust each other which complicates a free competition and makes misuse possible. Moreover, a growing skepticism of farmers against cloud-based FMISs provided by manufacturers can be observed as these systems require the upload of internal operational data such as machine data from terminals.

Therefore, we propose *CAN't*, a privacy and data sovereignty framework for CAN/ISOBUS-related information. *CAN't* consists of a hardware proxy that operates as man-in-the-middle in order to filter and manipulate data streams. It can be configured by a website-based user interface (UI) and comprises a modular privacy filter system.

The remainder of the paper is organized as follows: Section 2 provides background information on CAN and ISOBUS as well as the scenarios and privacy issues addressed by our framework. After a discussion of related work (Sec. 3), Section 4 presents our system architecture. Moreover, a modular privacy filter system is introduced by Section 5. The performance of our privacy proxy is evaluated in Section 6. Then, an extensive proof-of-concept evaluation is presented in Section 7 and subsequently discussed (Sec. 8). Finally, a brief conclusion is given in Section 9.

2. Background

2.1. Controller Area Network

CAN (ISO 11898) [9, 10] is a robust standard for in-vehicle communication that was developed by the Robert Bosch GmbH in the early 1980s. It specifies a serial multi-master and shared-medium bus system that significantly saves weight and costs by reducing the conventional wire harness. Nowadays, CAN is the de-facto standard in automotive industry and mandatory for vehicle diagnostics in the European Union since 2001[1]. The standard offers an efficient and flexible message-based communication for embedded automotive ECUs that is designed to be real-time capable and robust against electromagnetic interferences. Thus, it specifies the physical and data link layer of the ISO/OSI reference model. At the latter layer, it uses well-known techniques such as bit stuffing, Cyclic Redundancy Checks (CRCs), and an efficient Acknowledgment (ACK) method that acknowledges transmission success to at least one receiver. Furthermore, Carrier Sense Multiple Access/Collision Resolution (CSMA/CR) is used, i.e. a lossless bitwise arbitration method for contention resolution. For that purpose, the identifier (ID) part of the header that is included in each data frame is considered within the arbitration phase. CAN defines two basic types of its physical and data link layer, i.e. low-speed (up to 125 kbit/s) and high-speed CAN (up to 1 Mbit/s). Both of them provide a maximum payload size of 8 bytes, which are transmitted using fixed data rates. In addition to these types, two almost identical frame formats are specified at the data link layer: The Base Frame Format (BFF) (of CAN 2.0A) uses 11 bit object IDs whereas the Extended Frame Format (EFF) (of CAN 2.0B) provides 29 bit for identifying the content of a data frame.

2.2. ISOBUS

Based on CAN, ISOBUS[2] specifies the embedded communication for agricultural machinery, particularly between a tractor and an implement, as well as the integration into so-called FMISs. ISOBUS is the open and non-proprietary specification of the ISO 11783 standard [11] and managed by the Mechanical Engineering Industry Association (VDMA), cf. [12]. It uses high-speed CAN 2.0B with EFF and a nominal bit rate of 250 kbit/s and extends CAN by network and application layers. Because both, tractors and implements, consist of an increasing number of ECUs, but are usually manufactured by different manufacturers, the standardized communication is very crucial for a broad compatibility and interoperability.

In a modern tractor, there are physically separated CAN segments: the traditional and closed tractor's internal CAN and an additional open ISOBUS segment that provides control for external tools and implements in agricultural

[1] https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:31998L0069&from=en

[2] https://www.isobus.net/isobus/

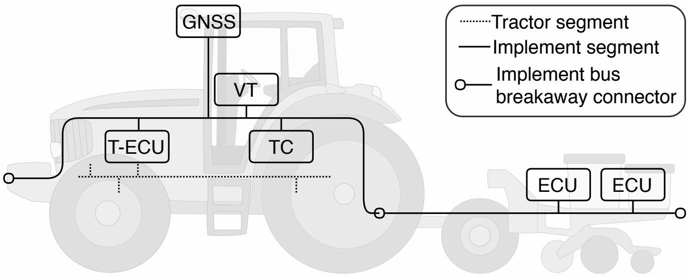


Figure 1: Simplified overview of the ISOBUS topology, according to [11, Part 4]. The implement bus connects a tractor via the T-ECU gateway with implements. It enables to merge required UIs in a single VT in the tractor cabin and also to integrate GNSS and FMIS components.

processes, i.e. data communication between the drive train and chassis of the tractor [11] as sketched in Figure 1. For that communication, there are four main entities defined by ISOBUS that are relevant for this work:

- The *Virtual Terminal* (VT) [11, Part 6] is a manufacturer independent, non-proprietary instance of a user terminal in a tractor's cabin with a graphical UI including various masks for different applications and tasks. During task execution, the VT records relevant process data.
- The *Tractor ECU* (T-ECU) [11, Part 9] is a gateway between the internal tractor CAN and the ISOBUS CAN and forwards relevant information and commands that are required to be shared between tractor and implement depending on concrete tasks.
- The *Task Controller* (TC) [11, Part 10] is the interface to the FMIS. It allows to define specific tasks that can be transmitted to the machine, supervised during task execution, and filled with information gathered during its execution.
- Another optional entity that can be considered as a special implement is the *GNSS Receiver*. If available, it provides position data information for navigation and documentation purposes. Otherwise, if there is no external receiver, the internal receiver of the tractor (if existing) can be made available via the T-ECU.

In order to identify a message's payload content, ISOBUS adopts the Parameter Group Number (PGN) format from the SAE J1939 standard. To minimize network load, closely related data types are bundled and transmitted using one or more messages identified by a single unique PGN which is used as CAN 2.0B ID, e.g., measured pressure values of individual tires. Various standards, including ISOBUS, define a payload structure and included data types for unique PGNs. Also, each type of data within a PGN has its own additional ID called Suspect Parameter Number (SPN). This ID is not transmitted, but used to associate a PGN with its bundled data types.

There are also technical development efforts beyond the existing ISOBUS standard. The Agricultural Industry Electronics Foundation (AEF), an organization that was founded by international agricultural equipment manufacturers and related associations, amongst others the VDMA, focuses on the further development of the ISOBUS, e.g., on a high-speed ISOBUS and also on wireless infield communication. However, such developments are currently still work in progress and have not yet reached the market. Moreover, existing machinery equipped with the original ISOBUS will certainly still be used for many more years, emphasizing the practical relevance of our work.

2.3. Data Exchange & Privacy Challenges

Agricultural collaboration often benefits from frequent data exchange between all participating actors and their machines. However, some data might be privacy sensitive and actors might be unwilling to share this data, particularly if new entrants in the market are involved. An exemplary motivating scenario that is taken from our previous work [13] is the process chain for wheat harvest. During the harvest, combine harvesters temporally store grain in their internal tanks. If the tank level exceeds a certain threshold, the grain has to be overloaded to transporters. Grain overloading has to be in time and often takes place while driving in order to save expensive process time. Hence, infield logistics of grain transportation is very important. Thus, for infield-planning and situational awareness, the positions and tank levels of harvesters and transporters' positions needs to be exchanged frequently.

From the motivating scenario, two major *privacy challenges* arise: (1) The personal privacy of employees driving machines that regularly transmit their position is not adequately protected in practice. Indeed, there is a tradeoff between operability and privacy. On the one hand, spatio-temporal accuracy of position data should be sufficient for infield planning, e.g., to arrange contacts for overloading phases. On the other hand, a boundless tracking must be prevented, particularly for employees in road traffic. (2) Internal operational data of farmers might not be protected adequately as well. The reason ist that for small and medium-sized farms, the purchase of specialized agricultural machinery such as combine harvesters is not profitable. Thus, farmers integrate external contractors into harvesting chains. As a consequence, a combine harvester of a contractor "senses" field-related information during the harvest, e.g., yield information with potentially high spatio-temporal context. However, contractors themselves do not really need this information, neither for operability nor for billing purposes. Hence, from a data sovereignty perspective, such information should not be gathered by external machines.

From a legal point of view, privacy and data sovereignty with regard to Smart Farming is very complex. An expert report states that an unlimited tracking of employees and a boundless generation of their movement profiles have to be organizationally prevented in order to protect employees from a permanent surveillance [14]. Hence, an adequate privacy policy should ensure that each entity gets only as much information as operationally required. This policy can be realized on several layers, e.g., at FMIS layer. Yet we believe that an early realization at machine layer, i.e. on CAN/ISOBUS, is more reasonable and effective to ensure privacy interests. However, both, CAN and ISOBUS have been designed focusing on robustness, reliability, and feasibility and not with IT security or privacy issues in mind. Note that, because our focus is on Smart Farming, we only consider ISOBUS-capable machines and ISOBUS communication in our work. Nevertheless, most of our results are not limited to the ISOBUS and, due to its modularity, our framework could also easily be adapted to universal CAN networks.

3. Related Work

The lack of security of both, CAN and ISOBUS is not a new insight and its vulnerability to malicious security threats has already been pointed out in the literature. A few solutions for different security issues have been proposed that essentially are based on Message Authentication Codes (MACs), i.e. they are developed in order to ensure integrity and authenticity of message delivery. Szilagyi and Koopman [15] assume pairwise symmetric keys that are preinstalled in all ECUs for their truncated MAC scheme. Moreover, as countermeasure against replay attacks, they propose a clock-based approach that links the temporal occurrence of events with messages triggered. However, global time synchronization cannot generally be presumed in CANs. Lin and Sangiovanni-Vincentelli [16] therefore present a comparable MAC approach that uses a message counter against reply attacks and, thus, do not rely on time synchronization. Moreover, they introduce grouped keys in order to mitigate the scalability problem of pairwise keys. Beyond integrity and authenticity, Wu et al. [17] propose AES-128 encryption with periodically renewed global keys for confidential CAN communication. They additionally use data compression for bus load reduction. Moreover, it is worth mentioning a complementary security approach that is presented in [18] which uses physical layer anomaly detection in order to mitigate unauthorized bus access. All these approaches confirm the importance of CAN security. In the agricultural context, we assume that unauthorized bus access and ECU compromises are rather unlikely because a potential attacker either requires physical access to machines or has to compromise the VT via its telemetry unit. Thus, our work is focused on privacy. We leave integrity, authenticity, and confidentiality approaches aside since they demand non-standard-compliant ECU modifications. Instead, we propose a privacy proxy that is realized as special ECU in the ISOBUS network. Our idea is inspired by the gateway role of the T-ECU defined in the ISOBUS standard itself. According to [11, Part 4], amongst others, the T-ECU has the task of filtering out critical engine data from the internal tractor CAN segment and prevent it from being forwarded to the implement segment. The technical requirements for such a proxy are also already specified in the standard and will be described in the following sections. With regard to the architecture and the practical implementation of our proxy, there are some conceptual similarities with [19] which presents a transparent data compression extension for bus load reduction in ISOBUS networks. However, to the best of our knowledge, there is no directly related work on CAN/ISOBUS privacy.

4. System Architecture

4.1. Concept & Requirements

Our main goal is to develop an ISOBUS privacy framework that is feasible to fully enforce the privacy policy derived from the motivational scenario in Section 2.3 at CAN/ISOBUS layer. Particularly in collaborative processes, but also farm-internal, each actor/employee must solely have full access to his/her own data, whereas the access to external data of other participants must be strictly limited the operational necessary level. That means, whenever the external forwarding of privacy sensitive data is required, our framework will offer the option to reduce the content of information.

Pure filtering that selectively passes and blocks messages can be implemented in different ways at CAN layer. Conceptually, due to the shared medium, jamming approaches well-known from wireless communication such as reactive or selective jamming using intentional physical layer interferences are possible. Existing rushing attack approaches at medium access layer could also be implemented by leveraging the simple priority mechanism of CAN in order to repress original messages and replace them with manipulated content. However, both approaches demand the implementation of a non-compliant ECU and might also be detectable by anomaly detectors. In contrast, we decided to realize a standard-compliant intercepting privacy proxy as regularly ECU that allows various configurable filter and manipulation options. Standard compliance is important to enable a feasible and cost-effective retrofitting of existing machines. According to [11, Part 4], this proxy should be designed as a network interconnection unit (NIU). Thus, we decided to implement it as NIU bridge type which is transparent in the network and works in promiscuous mode. This type is explicitly intended for suchlike purposes and also applied for the above mentioned T-ECU. From the standard, three mandatory requirements for the NIU bridge implementation arise, namely a defined *1) maximum transit-delay* with *2) message sequence and priority conservation* properties as well as *3) guaranteed filtering and forwarding rates* that must not be exceeded. These requirements are used in the following sections and referred to as *NIU requirements.*

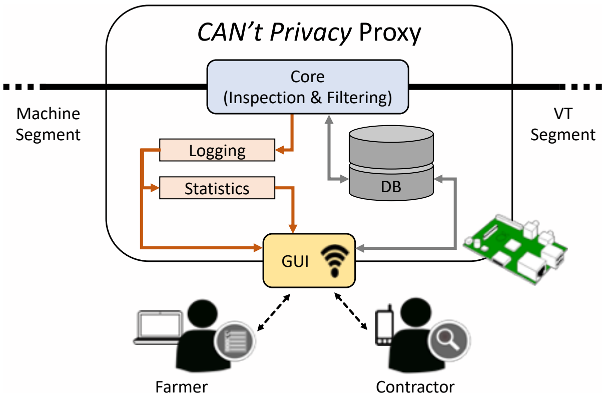


Figure 2: General system architecture of the *CAN't* privacy proxy realized as ISOBUS-compliant NIU that separates the VT from the remaining bus segment.

4.2. Architectural Design

Similar to the data compression extension in [19], our privacy proxy is architecturally placed directly between the implement bus and the VT (cf. Fig. 1), i.e. it separates the terminal from the remaining machine bus as depicted in Figure 2. Using a graphical UI and wireless Internet connectivity, the proxy should be dynamically configurable during the runtime. By this means, all actors, particularly those who are involved in collaborative process have the possibility to contractual agree on the detail of information that is going to be exchanged by jointly configuring the filters of the proxy. Those agreements in terms of corresponding filter configuration are stored in the proxy's database. The database also contains information about all relevant PGNs and SPNs that are needed for message identification and selective filter applications. Furthermore, a *Logging* component ensures that each activated filter and every conducted filtering is logged. This is particularly important for compliance checks in case of a joint privacy policy agreements. Also a *Statistics* component is designed for extracting ISOBUS load and other statistical network information and, thus, enables real-time monitoring of current proxy and network states.

Due to the specific ACK scheme of CAN, no special effort is necessary in case of frames are discarded by the proxy. The proxy itself is able to acknowledge the corresponding receive without any address spoofing. At the ISOBUS layer, there is no common ACK mechanism. However, for certain data types, a request-response scheme is used. Thus, in case suchlike request messages are discarded by our proxy, corresponding response messages need to be faked. On the other hand, if such a message is not discarded, but its payload is manipulated when being forwarded, the proxy has to deal with a possible re-manipulation. That means, in order to successfully deceive the request sender, the original manipulation has to be inverted again if a response to the manipulated message arrives.

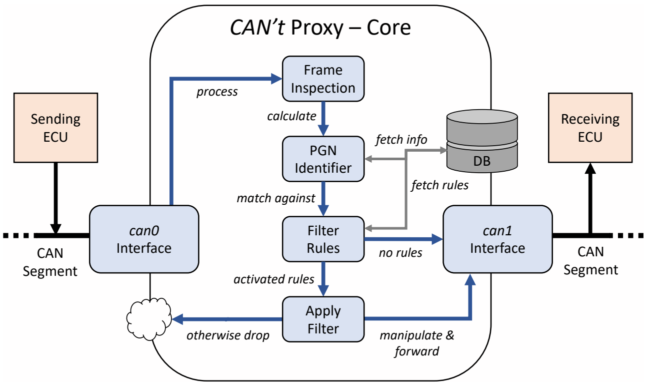


Figure 3: Conceptual representation of the core functionalities.

4.3. Implementation

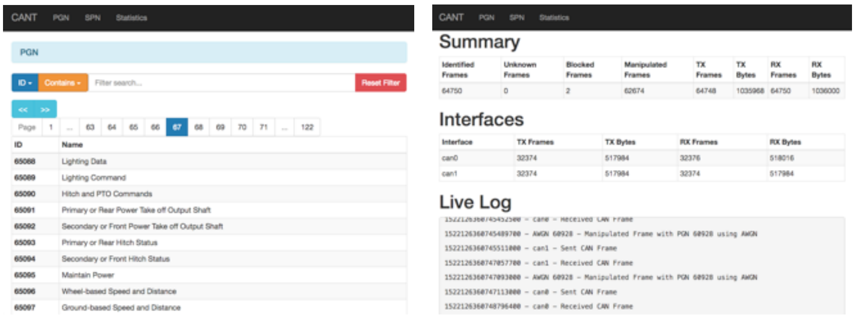
4.3.1. Hardware

For the prototype implementation of *CAN't*, low-cost developer hardware is utilized. We use a commercial off-the-shelf single-board computers, namely Raspberry Pis 3, and extended this platform by appropriate CAN boards (PiCAN 2[3]) with MCP2515[4] CAN controllers. These extension boards are attached via SPI using the GPIO interface. Two proxy versions were implemented. The first version is equipped with a PICAN 2 Duo that features two CAN interfaces which are used to directly bridge both ISOBUS segments according to the overview in Figure 2. The second version is more complex and consists of two Raspberry Pis, each with a single PiCAN 2. One device acts as interface to the VT segment, the other one as interface to the implement's bus segment. Data transmission is realized by bridging both devices via Ethernet or WiFi. Also a public land mobile networks (PLMN) bridge using corresponding modems on both devices is generally conceivable but might be challenging due to additional transmission delays.

4.3.2. Software

On the Raspberry Pi platform, a common Linux distribution (Raspbian Stretch) is used. No kernel modifications are required. The software implementation of *CAN't* is based on Go[5], an open source programming language created by Google. Go features cross compiling support and enables highly parallelized processes and, thus, is appropriated for the efficient implementation of a NIU-compliant proxy. We use SocketCAN[6] as a kernel module that fully supports our PiCAN hardware. It makes two CAN interfaces available, *can0*

[3] http://skpang.co.uk/catalog/images/raspberrypi/pi_2/PICAN2DSB.pdf
[4] https://www.microchip.com/wwwproducts/en/en010406
[5] https://golang.org/
[6] https://www.kernel.org/doc/Documentation/networking/can.txt



ID	Name
65088	Lighting Data
65089	Lighting Command
65090	Hitch and PTO Commands
65091	Primary or Rear Power Take off Output Shaft
65092	Secondary or Front Power Take off Output Shaft
65093	Primary or Rear Hitch Status
65094	Secondary or Front Hitch Status
65095	Maintain Power
65096	Wheel-based Speed and Distance
65097	Ground-based Speed and Distance

(a) Navigable and searchable PGN catalog.

Summary

Identified Frames	Unknown Frames	Blocked Frames	Manipulated Frames	TX Frames	TX Bytes	RX Frames	RX Bytes
64750	0	2	62674	64748	1035968	64750	1036000

Interfaces

Interface	TX Frames	TX Bytes	RX Frames	RX Bytes
can0	32374	517984	32376	518016
can1	32374	517984	32374	517984

Live Log

```
1522126360745452500 - can0 - Received CAN Frame
1522126360745489700 - AWGN 60928 - Manipulated Frame with PGN 60928 using AWGN
1522126360745511000 - can1 - Sent CAN Frame
1522126360747057700 - can1 - Received CAN Frame
1522126360747093000 - AWGN 60928 - Manipulated Frame with PGN 60928 using AWGN
1522126360747113000 - can0 - Sent CAN Frame
1522126360748796400 - can0 - Received CAN Frame
```

(b) Statistic and event log summaries.

Figure 4: Screenshots of *CAN't* framework's graphical UI provided by the implemented proxy.

and *can1*. With the Berkeley socket API, the interfaces can be accessed as raw socket (*AF_ CAN* family).

The general procedure of CAN frame processing is visualized in Figure 3. It describes the selective bridging (i.e. forwarding or filtering of frames) in a single direction only, e.g., from the implement to the VT segment or vice versa. There is a permanent sniffing on the ingress CAN interface (here *can0*). Whenever a frame is transmitted via the corresponding bus segment, the proxy deserializes the received frame and hands it over to a *Frame Inspection* component for format parsing and PGN determination. In the *PGN* component, a database lookup is conducted in order to fetch additional information about the frame and included SPNs. Then, the *Filter Rules* component checks potentially activated filters and their configuration for the determined PGN. In case, no filter is activated, a simple passthrough forwarding to the egress CAN interface (here *can1*) is initiated by the *Apply Filter* component. Otherwise, the corresponding frame is discarded (i.e. block filter) or its payload is manipulated according to a particular filter strategy before being forwarded. Finally, the frame is appropriately serialized and sent to the opposite bus segment. Note that the original addressing of the frame is consciously not modified so that a receiver is technically not affected by the privacy proxy as a man-in-the-middle. The components mentioned above are implemented as autonomous processing block allowing process parallelization with mulit-threading in order to increase the proxy's performance. Each block is connected via channels that forward frames from one processing block to another in a kind of flow graph.

For the database that contains relevant PGN and SPN information, the relational SQL database MariaDB[7] is used. Corresponding data schemata are developed for overall 352 non-proprietary PGNs, 6412 manufacturer-independent SPNs (including their 1:N relationship), and several predefined filter strategies.

[7] https://mariadb.com/

Due to performance requirements, the database is only queried for UI-based interactions and initial filter configuration. For the actual proxy operation, relevant information is cached in the *PGN* and *Filter Rules* components.

Finally, for the purpose of configuration and monitoring of proxy and filters, a graphical UI is provided by the Raspberry Pi via a WiFi access point. It offers an intuitive HTML5-based web interface using ReactJS[8] and Twitter Bootstrap[9]. This web interface is implemented as a single page application in order to reduce server resources. According to Figure 2, the user has access to logging and statistics components and to the database. Therefore, detailed information about PGNs (cf. Fig. 4(a)), SPNs, and filter states can be obtained. Here, various filter rules can be individually activated and configured for each ID. A second website enables a fine-grained monitoring of the CAN interfaces, network statistics, and filter event logs as demonstrated by Figure. 4(b).

5. Privacy Filter Strategies

Our framework provides a modular set of predefined privacy filters and filter strategies. By *filter strategy* we mean a certain filter together with a specific configuration. Privacy filters and strategies exclusively serve as privacy and data sovereignty protection. They are not intended for deceiving other participants. Hence, the application of each privacy filter strategy is completely made transparent in the UI for all participants. As the concrete configuration of filters clearly depends on the type of payload information and is highly application- and event-specific, we only provide a limited set of modular basic filter mechanisms as functional building blocks.

The flow graph that represents the processing of filter strategies is sketched in Figure 5. All strategies can be assigned to one or multiple PGNs. Therefore, the PGN initially needs to be identified. Different basic filter strategies can be individually configured and easily be extended by users for specific tasks and agricultural processes. To refine strategies, they can also be combined which is technically implemented as a concatenation of individual strategies. Then, a universal filter component is responsible for asserting these strategies (cf. Fig.5). Note that, while being forwarded along the processing channels, frames are processed sequentially in a first in – first out (FIFO) manner, conserving the original message sequence.

Beyond pure message filtering that discards specific ISOBUS messages according to their PGNs, payload-depended manipulation filters and filters that artificially add synthetic messages are enabled. The latter can be used either to fake ACKs (responses) of previously discarded messages or to intentionally add fake messages for the sake of obfuscation. Generally, the information included in the payload of ISOBUS messages can be manipulated by modifying individual payload values or the entire value range of a message stream. Since machine and

[8] https://reactjs.org/

[9] https://getbootstrap.com/

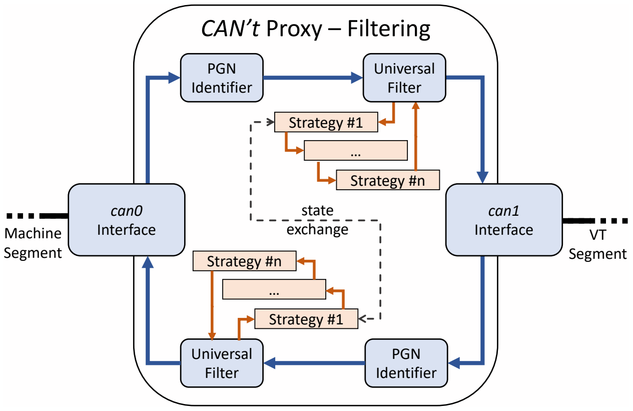


Figure 5: Flow graph representation of the modularly implemented filter processing.

implement data always contains spatio-temporal information, the manipulation can also be conducted on temporal and/or spatial dimension using perturbation approaches, for instance. However, in the context of agriculture, there is often a necessary link between spatial and temporal components. This is particularly the case, if controlled traffic farming (CTF) technology is used, a soil protection driving strategy restricted to permanent traffic lanes. Moreover, a temporal perturbation is restricted in live operation mode when messages are exchanged during a collaborative task. Thus, spatio-temporal manipulation strategies must be applied carefully. Nevertheless, an event-based activation of privacy filters is possible, e.g., message passing only within the spatial boundaries of a certain field or during the planned processing time of a certain task. Furthermore, also event-based dosing of delivered level of information content is conceivable. Regarding both privacy challenges in the motivational scenario (cf. Sec. 2.3), 1) employees' privacy could be protected, if accurate position data is solely available in the field whereas only coarse or obfuscated information is transmitted during road traffic and 2) data sovereignty of farmers could be improved if contractors harvesters would not continuously gather spatial yield information in terms of accurate tank level changes or only in case a critical threshold is exceeded, for instance.

An initial set of privacy filters strategies has been implemented within the *CAN't* framework. It comprises many basic functions that are not explicitly listed here. Instead, we only introduce the following four basic filters, the impacts of which are exemplarily evaluated in the next section. For data *perturbation*, additional random noise can be added, e.g., additive white Gaussian noise (*AWGN filter*). Furthermore, the level of information content of privacy sensitive data types (identified based on PGN or SPN) can be adjusted by mathematically non-injective functions. That might be a simple pruning of their value ranges (*Threshold filter*), a numerical rounding (*Rounding filter*), or a throttling of update rates (*Throttling filter*), for instance.

The throttling of update rates could either be implemented by reducing the actual message rate, i.e. by discarding a certain percentage of CAN frames that belong to a particular data type, or by manipulating their payloads in a particular manner, while keeping the original rate. Therefore, the payload of those messages that are not supposed to update the data stream has to be accordingly overwritten with the outdated value, previously buffered from the latest non-manipulated message. The latter approached is used in our evaluation since it has two decisive advantages. First, it is simpler because it neither requires to adapt sequence numbers (SNs) when messages are discarded, nor to generate fake possible responses in case an ISOBUS request-response scheme is used (cf. Sec. 4). Second, it is less invasive as the original message rate is not changed. Thus, it is expected to be less conspicuous to error handling routines or Intrusion Detection Systems (IDSs). However, for request-response messages, the manipulations the proxy preformed for requests (overwritten payloads) have to be inverted again for response messages. For that purpose, there are two instances of the same filter for each direction, implement to VT and vice versa, as shown in Figure 5. In case a filter is configured for a request-response message stream (identified by certain PGNs), both instances have to share the state of performed manipulations for a latter inversion (*state exchange*).

6. Performance Evaluation

6.1. Measurement Setup

The performance of the implemented privacy proxy is crucial for the technical compliance with the ISOBUS standard (cf. *NIU requirements* in Sec. 4.1) and eventually for its practical feasibility. A low and constant latency in terms of the proxy's transit delay is very important for a seamless and failure-free integration of the proxy into existing ISOBUS-capable machines. But also message sequence and priority conservation is necessary, particularly regarding frames with different CAN priorities, as generally specified for NIUs [11, Part 4]. Hence, we evaluated the performance of our prototype and initially used a controlled and replicable laboratory setup. In a baseline setup, two Raspberry Pi devices were connected via CAN. One acts as a sender of ISOBUS messages, the other one as a receiver. We then generated synthetic ISOBUS traffic between both devices and monitored message delivery. In a second step, the connection between both devices was interrupted by our prototype so that the bus is divided into two separated segments, according to the general concept in Figure 2. This setup allowed a comparative evaluation.

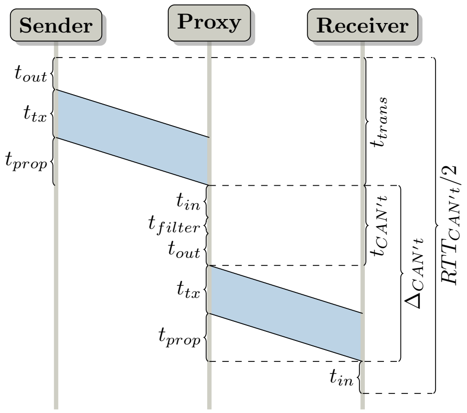


Figure 6: Time-sequence diagram and delaying phases of the message forwarding process including the intercepting privacy proxy.

6.2. Performance Metrics

Due to time synchronization issues between both Pis and the HAL of the CAN boards, the *round trip time (RTT)* is considered as main performance metric, substitutional for the transit delay. The RTT consists of the latencies induced by both transmissions, sender to receiver and vice versa, and, thus, quantifies the two-way delay of the CAN frame transmissions. For a reliable RTT determination at the sender device, each message transmitted sequentially (in stop & wait manner) and immediately sent back from the receiver to the sender.

According to the terminology of delay phases in [20], the RTT twice contains the processing delay of sending and receiving devices and in- and egress queueing delays (summarized as t_{in} and t_{out}), the propagation delay t_{prop} and the transmission delay t_{tx} (cf. Fig. 6) under which the transmission delay is expected to have the major impact in our setup. The transit-delay of the proxy $\Delta_{CAN't}$ can be described as

$$\Delta_{CAN't} = t_{in} + t_{filter} + t_{out} + t_{tx} + t_{prop} \tag{1}$$

With an overall proxy's processing delay $t_{CAN't}$ and due to the negligible signal propagation delay on short distances, this equation can be simplified as

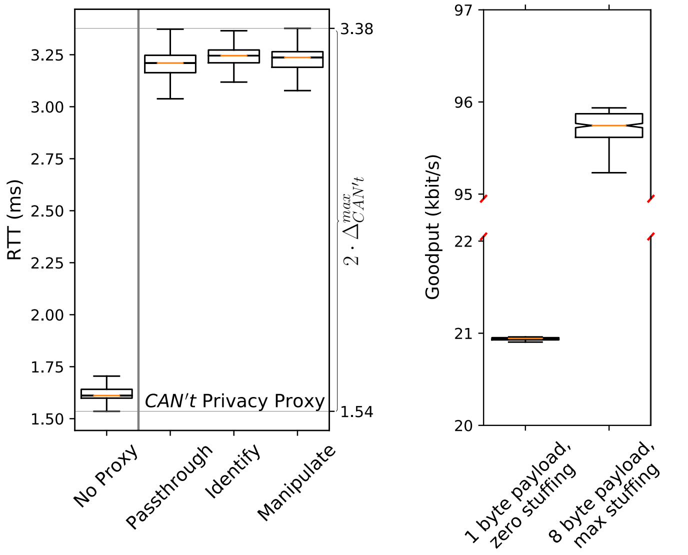
$$\Delta_{CAN't} = t_{CAN't} + t_{tx} \tag{2}$$

as sketched in Figure 6.

The transit-delay of the proxy can be derived from the RTT measurements by using the following estimation

$$\Delta_{CAN't} \approx \frac{(RTT_{CAN't} - RTT_{\neg CANt})}{2} \tag{3}$$

with $RTT_{CAN't}$ representing the RTT determined with and $RTT_{\neg CANt}$ without the privacy proxy as man-in-the-middle.



(a) Latency induced by the privacy proxy evaluated by RTT measurements.

(b) Maximal ISOBUS goodput achieved in a setup without privacy proxy.

Figure 7: Experimental measurement results.

Overall, the RTT evaluation comprises four experiments, namely the setup without a proxy and three variants with the intermediate proxy. Therefore, the proxy either ran in direct *passthrough* mode, in passthrough mode with PGN *identification*, or in message *manipulation* mode. For the latter mode, the AWGN filter was exemplarily applied. Each experiment includes 100.000 messages with maximum message size (8 byte payload with maximal number of stuff bits).

In addition to the RTT, *goodput* measurements were conducted in order to determine the maximal ISOBUS message delivery rate and to derive the message time (i.e. the complete time required to deliver a certain message) as well as possible filtering and forwarding rates. For the measurements, saturated traffic is generated and, for a duration of 5 min, directly transmitted from a sending to a receiving device without being intercepted by the privacy proxy. The goodput is measured every second and two cases are investigated differing in the size of messages, i.e. either minimal (1 byte payload with zero bit stuffing) or maximum message sizes (8 byte payload with maximal bit stuffing) are used.

6.3. Results

The results obtained in each experiment are shown in Figure 7. Regarding the latency in Figure 7(a), the boxplot representation confirms the overall reasonable constancy as RTT variances are relatively low. The RTT observed in

Table 1: ISOBUS message characteristics.

			analytical*: (conservative bit stuffing estimate)		empirical: (averaged from goodput experiments)		
Msg. Size	Payload (byte)	Bit Stuffing	Trans. Delay (μs)	Goodput (kbit/s)	Goodput (kbit/s)	Delivery Rate (messages/s)	Msg. Time (μs)
min	1	zero	288	27.78	20.94	2618	t_{trans}^{min}=382
max	8	max	t_{tx}^{max}=608	105.26	95.71	1496	668

*based on EFF (29-bit IDs), 64 bit CAN frame overhead, and 250 kbit/s bit rate (4 μs/bit).

the experiment without a proxy is roughly 1.6 ms. This is plausible since the pure two-way transmission delay (without any queueing and processing delays) is $2t_{tx}^{max} \approx 1.2$ ms (cf. Tab. 1). The integration of the proxy inherently adds an additional latency $\Delta_{CAN't}$ that again includes processing, queueing (in- and egress interfaces), propagation, and transmission delays, cf. Eq. 2. Consequently, the RTT is roughly doubled by the proxy to approximately 3.2 ms as can be observed in Figure 7(a). Moreover, the comparison of proxy modes suggests that the processing of frames (database lookup for PGN identification and payload manipulation) has a negligible impact on its transit-delay.

The results of the goodput experiments are visualized in Figure 7(b). As expected, the achieved goodput in the uninterrupted setup is very stable and, furthermore, strongly depends on the actual message size, i.e. the payload-to-CAN- overhead ratio. From the goodput, both the maximal message delivery rate and also the minimal message time are derived for minimal and maximal message sizes. The results are listet in Table 1 and compared with values that are analytically determined based on the corresponding frame sizes (including CAN overhead with induced stuff bits according to a conservative bit stuffing estimate) and a bit time of 4 μs.

6.3.1. Transit-Delay

With regard to the *NIU requirements*, a maximum transit delay of 10 ms is recommended. According to Equation 3, the worst-case maximum transit delay can be derived from the comparison of both setups as

$$\Delta_{CAN't}^{max} = \frac{(RTT_{CAN't}^{max} - RTT_{\neg CANt}^{min})}{2} \tag{4}$$

where $RTT_{CAN't}^{max}$ is the maximal observed RTT in the proxy setup whereas $RTT_{\neg CANt}^{min}$ presents the lower RTT bound in the setup without proxy. The evaluated results (cf. Fig. 7(a)) yield

$$\Delta_{CAN't}^{max} \approx \frac{3.38\,ms - 1.54\,ms}{2} \approx 0.92\,ms \overset{!}{\leq} 10\,ms \tag{5}$$

Hence, the transit-delay requirement is sufficiently met by our approach.

6.3.2. Message Sequence and Priority Conservation

Generally, ensuring the conservation of message sequences and priorities, i.e. the correct transmission order of messages that are processed and forwarded by the proxy, can be relatively complex. If processing times, however, can be guaranteed that are short enough so that the entire processing, including both a possible forwarding or discarding, is completed prior to the reception of the subsequent message, no internal message queueing will occur in the proxy. Thus, if suchlike short processing times would be achieved by our prototype, it can be argued that the simple FIFO queue in its current version is reasonable for message handling. The message delivery rate directly derived from the goodput corresponds to the highest possible rate of message arrival at the proxy. Thus, the previously derived minimal message time t_{trans}^{min} (cf. Tab. 1) at the same time represents the upper bound for the proxy's processing time. Hence, using Equation 2, the second *NIU requirement* can be defined as

$$t_{CAN't}^{max} = \Delta_{CAN't}^{max} - t_{tx}^{max} \leq t_{trans}^{min} \tag{6}$$

Inserting the results obtained by the RTT evaluation and assuming the analytical $t_{tx}^{max} = 608\,\mu$s, it can be concluded:

$$\Delta_{CAN't}^{max} - t_{tx}^{max} \approx 920\,\mu s - 608\,\mu s = 312\,\mu s \overset{!}{\leq} 382\,\mu s \tag{7}$$

That means, due to its high performance, sequence and priority conservation is guaranteed since there is no internal queueing of ISOBUS messages within the privacy proxy, at least not with the complexity of currently implemented basic filters. Indeed, the margin left in Equation 7 is rather tight. However, the inequation is quite conservative because both $\Delta_{CAN't}^{max}$ and t_{trans}^{min} are worst-case assumptions which are relatively unlikely for real ISOBUS communication that usually has lower bus utilizations.

6.3.3. Filtering and Forwarding Rates

The investigation of the proxy's processing delay in the previous section already reveals the non-existence of message queueing, independently of payload sizes and delivery rates. Consequently, the proxy does not represent a bottleneck with regard to forwarding and filtering rates. Thus, the maximum number of messages that can be guaranteed to be forwarded or filtered per second is not affected by message processing. Thereby, the proxy does not limit the maximal message delivery rate of ISOBUS.

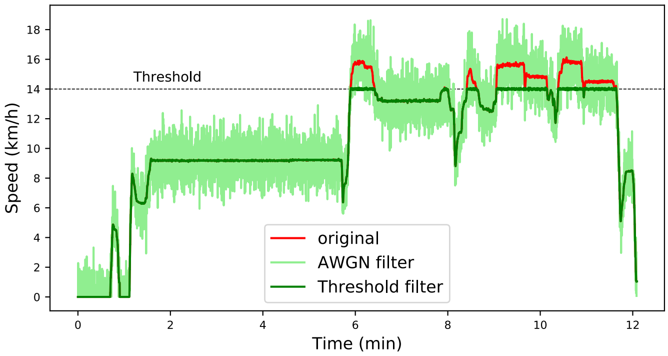


Figure 8: Impact of privacy filters on tractor's speed (evaluated using a real-world CAN trace).

7. Privacy Evaluation

7.1. Trace-base Simulation

7.1.1. Measurement Setup

The performance evaluation in the previous section already confirms the technical compliance of our prototype implementation with the NIU requirements. Now, we turn our attention to its practical feasibility and the impact of exemplarily demonstrated privacy filters. We therefore reused our previous setup and insert different real-world CAN/ISOBUS traces into our investigation that were recorded on agricultural machines. These traces were used to be replayed by the sender device into the CAN bus (with the original frame addressing). Without loss of generality, the evaluation presented here is limited to the effect of privacy filters on the (wheel-based) speed information obtained by a common tractor.

7.1.2. Results

After the successful transmission of the unfiltered data stream that was replayed by the sender, the received data was interpreted and the relevant information (wheel-based speed) was extracted. In Figure 8, the resulting speed curve (original) is visualized over time. In the next step, the transmission was intentionally manipulated by our proxy with two privacy filters, namely an AWGN and a threshold filter. The results are integrated in Figure 8 and show the general functioning of our approach. It can be observed that the AWGN filtering leads to data perturbation. However, this perturbation appears not to be sufficient to obfuscate the actual operation of drivers in practise and leaves room for further perturbation filters. Nevertheless, the threshold filter is able to restrict the maximum speed that is logged in this scenario. This can be very useful from the privacy perspective since, assuming a suitable configuration, speeding issues can no longer be documented in an FMIS.

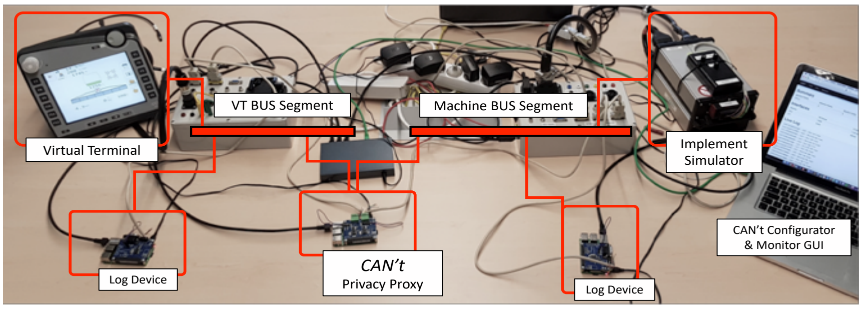


Figure 9: Simulation setup evaluating the prototype's operability and the impact of privacy filters between a simulated implement (trailed crop sprayer) and a real commercial terminal.

7.2. ISOBUS Simulator

7.2.1. Simulation Setup

Since an additional proof-of-concept evaluation using commercial hardware would increase the credibility and has an added value concerning practical feasibility, also special ISOBUS simulation hardware was used. We tested and evaluated our prototype on suchlike hardware that is equally embedded in real agricultural machinery and used for industrial ISOBUS development and compliance testing. The simulator comprise the entire bus communication of an ISOBUS-capable machine with its attached implement and simulates a certain task execution. In our case, the simulated implement is a trailed crop protection sprayer (manufactured by AMAZONE H. Dreyer GmbH & Co. KG) attached to a tractor that also provides accurate position data. The simulator is connected to a commercial terminal (CCI-200 developed by Competence Center ISOBUS e.V.) which implements a VT. In our setup, the proxy prototype is placed as man-in-the-middle between both devices, isolating the VT from the machine bus as illustrated in Figure 9. In addition, we add two Raspberry Pis for logging purposes, one for the machine segment and the other one for the separated VT segment. Again, we compare this setup with the original setup (without proxy) concerning its operability. Furthermore, as mentioned above, we exemplarily show the impact of illustrative privacy filters while, without loss of generality, focusing on positioning-related information. The following two scenarios are considered in the evaluation:

- Road traffic: For the sake of drivers' privacy, the information content of position data from machines on the road is intended to be reduced/perturbed.
- Crop protection: For the sake of data sovereignty of farmers, precise application maps should not be producible or reconstructable by contractors that are involved in farming activities, e.g., when applying crop protection on the field owned by a farmer. Therefore, in this scenario, the information content of tank level data should be purposefully reduced to a coarse

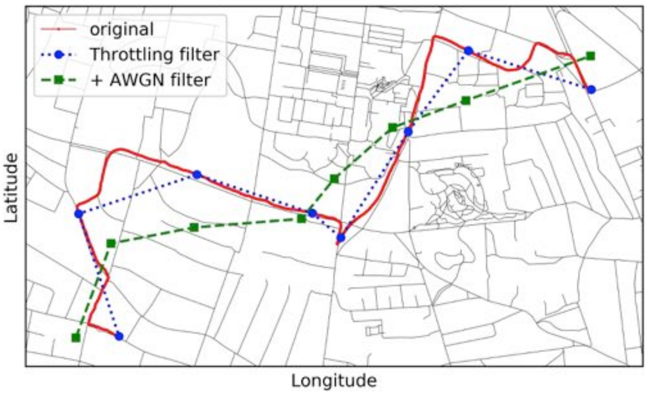


Figure 10: Impact of privacy filters on the accuracy of a GPS track transmitted from a machine to the VT in the road traffic scenario (evaluated with a CAN simulator).

level. Note that the scenario is selected due to the type of the available simulator (i.e. sprayer implement), but could be equally transferred to the motivational collaborative yield scenario.

7.2.2. Results

The simulative evaluation reveals an undisturbed operation of the system when the privacy proxy was integrated to the setup. No abnormal behavior or triggering of possible error handling routines in the both TC or VT were observed, neither in its passthrough mode nor in identification and manipulation modes.

In the road traffic scenario, two filters were demonstrated as shown in Figure 10. First, a throttling filter was used on the temporal dimension, i.e. in order to significantly reduce the original update frequency of position data. Here, the original rate of precise GNSS information from the simulated machine is reduced by updating only every 600th message with fresh positions. The effect is a coarsened tracking which successfully reduces the information content. The risk of exact reconstruction of the original track gets mitigated. Also the privacy sensitive driving speed is deliberately perturbed. In case an AWGN filter is additionally applied to the spatial dimension, the perturbation is further increased. Thereby, also the possibility to reconstruct the original driving route is apparently prevented, as illustrated in the figure. However, an operational necessary level of position information is still available allowing a situational overview and estimated arrival times for logistical planning.

In the second scenario, a rounding filter was used in order to mathematically round the tank level information of the crop protection sprayer to 25 % steps of the initial tank level at the beginning of a certain task (i.e. relative tank level). It is assumed that the entire tank is planned to be emptied in that task and the initial tank level value is known and considered for the configuration of the

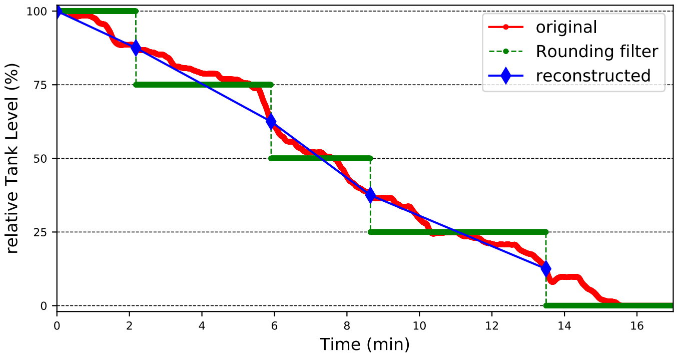
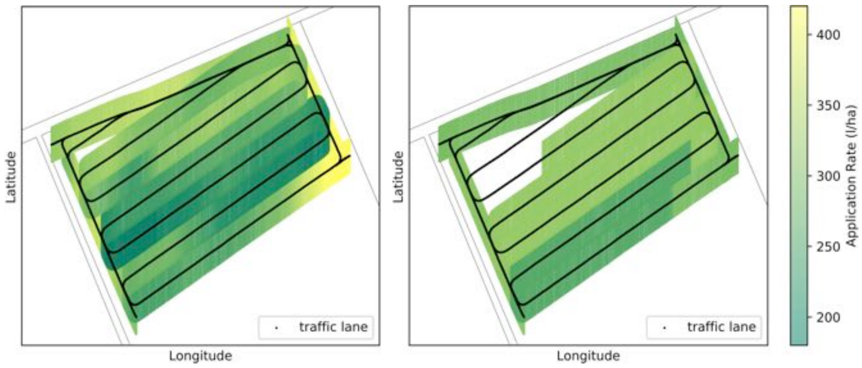


Figure 11: Impact of the rounding filter on the accuracy of the relative tank level of a sprayer implement (evaluated with a CAN simulator). Here, the precise original tank level decrease is exemplarily rounded to 25 % steps which prevents an accurate reconstruction.

filter. Figure 11 shows the effect of suchlike filtering on an exemplarily evaluated crop protection task. Despite the driving speed information is missing in the figure, crop protection is apparently not equally applied across the field. This variation is just the "site-specific" information that is reduced by the rounding filter. The average tank level decrease can however still be easily derived from the rounded information per 25 % section using linear approximations between the transitions of 25 % steps, except for last section (blue curve in Fig. 11). The rationale is that the exact point in time when the tank is actually empty is not reliably derivable.

Leveraging the application rates derived from the tank level decreasing process in combination with linked position information, application maps can be created as shown in Figure 12. In this scenario, the position information is not perturbed and describes the original traffic lane of the tractor/sprayer in the field. The comparison of both application maps (without and with privacy filters) emphasizes the impact of the rounding filter in suchlike contexts. The original unfiltered application map in Figure 12(a) shows a fine coloured gradation which represents the site-specific information of applied crop protection across the field. This site-specific information is only insufficiently reconstructable using the filtered tank level values. The reconstruction with linear approximation (cf. Fig.11) results in averaged application rates per 25 % section and, thus, also in a coarse application map with a rough coloured gradation as in Figure 12(b). Here, the gap in the reconstructed map is caused by the above mentioned missing approximation in the last 25 % section.



(a) High-resolution application map (original tank level values).

(b) Low-resolution application map (filtered tank level values).

Figure 12: Impact of the rounding filter on crop protection application maps derived from the tank level information (visualized in Fig. 11). The creation of (a) accurate maps with site-specific application rates, represented by the fine coloured gradation, can be prevented by filtering so that only (b) coarse maps with averaged rates are reconstructable.

8. Discussion & Future Work

The evaluation of the realized privacy proxy prototype showed the formal compliance with the NIU requirements of the ISOBUS standard and successfully demonstrated its practical feasibility in selected scenarios. With regard to privacy, the introduced filter strategies significantly mitigate the information content that is disclosed to third parties. However, our evaluation is limited to single data types (selected by certain SPN) and neglects a potential linking of related SPN and PGNs, respectively. For instance, there is an inherent link between position information and wheel-based speed, wheel-based distance, or wheel-based direction, but also between tank level and mass flow rate information. A holistic privacy framework would thus require a comprehensive application logic which is out of scope of our work but planned for the future.

In the proof-of-concept architecture, the privacy proxy enables a configurable filtering of information content that reaches the VT and, thus, eventually leaves a machine. However, it is currently not possible to split data streams generated by a particular machine according to their "owner" (e.g., farmer or contractor) since only a single VT is used. The split functionality could easily be realized by our approach using two VTs and two inversely configured proxies that adequately filter data streams that are forwarded to those VTs. However, due to the lack of space, this specific setup is not considered in our evaluation.

Moreover, due to the proof-of-concept nature and the focus on feasibility, only simple yet effective basic filters are included in the current framework. As part of our future work, existing state-of-the-art data perturbation and anonymization approaches will be evaluated with regard to their general applicability in our context and, if necessary, application-specific adaptions will be designed. Our work is also restricted to single frame transmissions (for wheel-based speed and tank level messages) and the Fast Packet Protocol (FPP) (for position messages in NMEA 2000 format). However, there are three other ISOBUS transport protocols (cf. [11, Part 3]) that need to be considered for a holistic framework.

From a technical perspective, also a spatial bridging of ISOBUS messaging is possible. ISOBUS segments could therefore be transparently bridged using WLAN or PLMN connections between two proxies. That could potentially allow further applications such as machines that remotely control an implement that is physically attached to another machine.

Despite of its agriculture- and ISOBUS-driven focus, our approach is in the end technically not limited to ISOBUS networks but generally applicable to vehicles or systems that use CAN technology. Hence, the privacy features enabled by our approach are interesting for other domains as well. Particularly due to modern information and communication technology in the automotive sector, privacy and data sovereignty issues have nowadays already become increasingly relevant.

9. Conclusion

In this paper, a modular privacy framework for CAN/ISOBUS communication was presented. Its focus is on privacy and data sovereignty issues of individual actors in the agricultural domain, particularly during collaborative tasks. Based on low-cost hardware, the core of the framework was prototypically realized by an intercepting privacy proxy that allows application-specific filtering at ISOBUS level. By adequately configuring such filters, an adaptive control for the accessible level of privacy sensitive information that leaves the machine is enabled. The technical performance and the framework's practical feasibility were successfully evaluated using real-world ISOBUS traces as well as industrial CAN simulation hardware. Moreover, an impact evaluation of meaningful privacy filters showed promising results and highlighted the opportunity of our approach.

Acknowledgment

The authors would like to thank the Competence Center ISOBUS e.V. (CCI) and CLAAS E-Systems KGaA mbH & Co KG for kindly providing real-world CAN traces. In particular, we thank Michel Löpmeier and Felix Wierling of CCI for helpful insight on CAN/ISOBUS issues and for supporting us with simulation hardware.

This work was partially funded by the German Federal Ministry of Education and Research (BMBF) within the Program "Innovations for Tomorrow's Production, Services, and Work" (02K14A19K) and managed by the Project Management Agency Karlsruhe (PTKA). The authors are responsible for the contents of this publication.

References

[1] S. Cox, Information technology: the global key to precision agriculture and sustainability, Computers and Electronics in Agriculture 36 (2–3) (2002) 93–111. doi:10.1016/S0168-1699(02)00095-9.

[2] Beecham Research Ltd., Enabling The Smart Agriculture Revolution – The Future of Farming through the IoT Perspective, Tech. rep. (2016).

[3] D. Vasisht, Z. Kapetanovic, J.-h. Won, X. Jin, R. Chandra, A. Kapoor, S. N. Sinha, M. Sudarshan, S. Stratman, Farmbeats: An IoT Platform for Data-driven Agriculture, in: Proc. of the 14th USENIX Conference on Networked Systems Design and Implementation (NSDI), Boston, MA, USA, 2017, pp. 515–528.

[4] H. Auernhammer, Precision farming — the environmental challenge, Computers and Electronics in Agriculture 30 (1–3) (2001) 31–43. doi:10.1016/S0168-1699(00)00153-8.

[5] N. Zhang, M. Wang, N. Wang, Precision agriculture – a worldwide overview, Computers and Electronics in Agriculture 36 (2–3) (2002) 113–132. doi:10.1016/S0168-1699(02)00096-0.

[6] J. M. Lowenberg-DeBoer, The precision agriculture revolution: Making the modern farmer, Foreign affairs 94 (2015) 105–112.

[7] D. J. Mulla, Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps, Biosystems Engineering 114 (4) (2013) 358–371. doi:10.1016/j.biosystemseng.2012.08.009.

[8] A. J. Scarlett, Integrated control of agricultural tractors and implements: a review of potential opportunities relating to cultivation and crop establishment machinery, Computers and Electronics in Agriculture 30 (2001) 167–191. doi:10.1016/S0168-1699(00)00163-0.

[9] International Organization for Standardization, Road vehicles - Controller area network (CAN) – Part 1: Data link layer and physical signalling, ISO 11898-1:2015 (2015).

[10] U. Kiencke, S. Dais, M. Litschel, Automotive Serial Controller Area Network, SAE technical paper 860391, SAE Int. (1986).

[11] International Organization for Standardization, Tractors and machinery for agriculture and forestry – Serial control and communications data network – Parts 1–14, ISO 11783-{1–14}:2007–17 (2007).

[12] T. Oksanen, M. Öhman, M. Miettinen, A. Visala, ISO 11783 – Standard and Its Implementation, IFAC Proceedings Volumes 38 (1) (2005) 69–74. doi:10.3182/20050703-6-CZ-1902.02102.

[13] J. Bauer, N. Aschenbruck, Measuring and Adapting MQTT in Cellular Networks for Collaborative Smart Farming, in: Proc. of the 42nd IEEE Conference on Local Computer Networks (LCN), Singapore, 2017, pp. 294–302. doi:10.1109/LCN.2017.81.

[14] P. Vogel, A. Klaus, Zulässigkeit der Verarbeitung von GPS-Daten im Arbeitsverhältnis (in german), Digivation-Sammelband (2018) 1–10.

[15] C. Szilagyi, P. Koopman, Flexible multicast authentication for time-triggered embedded control network applications, in: Proc. of the IEEE/IFIP Int. Conference on Dependable Systems & Networks (DSN), 2009, pp. 165–174. doi:10.1109/DSN.2009.5270342.

[16] C. W. Lin, A. Sangiovanni-Vincentelli, Cyber-Security for the Controller Area Network (CAN) Communication Protocol, in: Proc. of 2012 Int. Conference on Cyber Security, 2012, pp. 1–7. doi:10.1109/CyberSecurity.2012.7.

[17] Y. Wu, Y.-J. Kim, Z. Piao, J. G. Chung, Y.-E. Kim, Security protocol for controller area network using ECANDC compression algorithm, in: Proc. of the IEEE Int. Conference on Signal Processing, Communications and Computing (ICSPCC), 2016, pp. 1–4. doi:10.1109/ICSPCC.2016.7753631.

[18] P. S. Murvay, B. Groza, Source identification using signal characteristics in controller area networks, IEEE Signal Processing Letters 21 (4) (2014) 395–399. doi:10.1109/LSP.2014.2304139.

[19] N. Iglesias, P. Bulacio, E. Tapia, Enabling powerful GUIs in ISOBUS networks by transparent data compression, Computer Standards & Interfaces 36 (5) (2014) 801–807. doi:10.1016/j.csi.2014.01.007.

[20] R. Baumann, S. Heimlicher, M. Strasser, A. Weibel, A Survey on Routing Metrics, Tik report 262 (2007).