CAN't – An ISOBUS Privacy Proxy for Collaborative Smart Farming

René Helmke, Jan Bauer, Alexander Bothe, Nils Aschenbruck
University of Osnabrück, Institute of Computer Science
Wachsbleiche 27, 49076 Osnabrück, Germany
Email: {rhelmke, bauer, bothe, aschenbruck}@uos.de

***Abstract*—Smart Farming is driven by the emergence of precise positioning systems and Internet of Things technologies which have already enabled site-specific applications, a sustainable resource management, and interconnected machinery. Nowadays, agricultural machines and implements are equipped with multiple embedded sensors and actors continuously producing extensive data streams. For data communication on such machinery, ISOBUS, an internal vehicle bus, is used. ISOBUS is based on the machine's Controller Area Network (CAN). However, neither CAN nor ISOBUS communication takes privacy or data sovereignty issues into account. With increasing interconnectivity of agricultural machines and their integration into farm management systems, those issues become more and more serious. In this paper, we briefly present the architecture of our modular privacy framework *CAN't*. Using off-the-shelf hardware, a special proxy is prototypically implemented that allows to purposefully filter and manipulate CAN data streams for the sake of privacy. The feasibility and possibilities of our approach are described in this paper. By means of a customized video game, a live demonstration will additionally show the effect of the proposed privacy filters.**

***Index Terms*—ISO 11783; ISOBUS; Controller Area Network; Privacy; Data Sovereignty; Smart Farming**

I. Introduction

Information technology is an integral part of modern agricultural processes to overcome rising challenges in an industrialized setting [2], [5]. Driven by a variety of technological innovations, agricultural processes have already been revolutionized in many ways, leading to yield increases, resource optimizations and, thus, to sustainability [13]. Modern agricultural machines are equipped with a plurality of Electronic Control Units (ECUs), ranging from controllers for various components to sensors and actors for machine and environmental information [2], [5], [10]. These ECUs are networked in the Controller Area Network (CAN) of the machine, a robust bus system [7], [8]. A basic CAN frame contains a unique identifier, a length field, and small payload field. Depending on application-specific requirements, data rates between 125 kbit/s and 1 Mbit/s can be achieved.

ISOBUS [1] is based on CAN and specifies embedded communication between ECUs in agricultural machinery [6]. Its purpose is to provide a versatile framework for complex applications in the areas of Precision Agriculture and Smart Farming while maintaining non-proprietary and manufacturer-independent interoperability between machinery and implements. ISOBUS standardizes behavioral and physical properties of various key entities within the network. Amongst these entities is a Virtual Terminal (VT) that is located, for example, in a tractor's cabin and provides user terminal functionality through a graphical user interface (GUI) [6, Part 6]. Some entities, including VTs, are capable of aggregating information sent from ECUs.

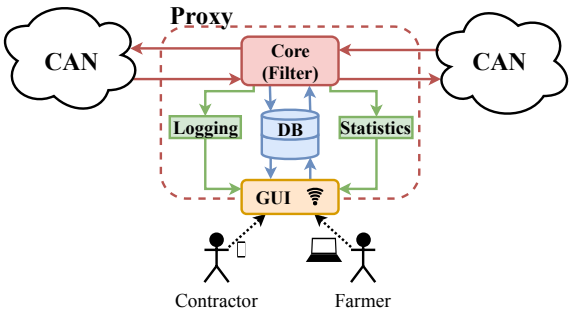


Fig. 1. System architecture of the privacy framework realized as intercepting proxy that separates the VT from the remaining machine CAN segment.

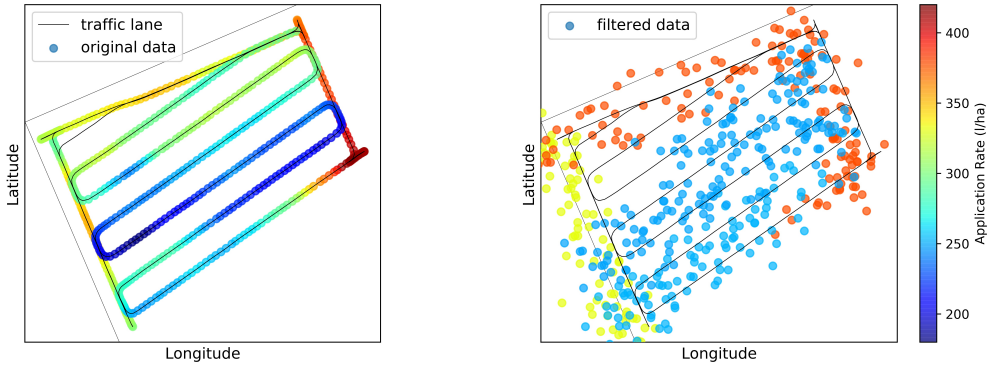
ISOBUS uses CAN 2.0B (29 bit identifiers) with a nominal bitrate of 250 kbit/s. Parameter Group Numbers (PGNs) are used as data type identifiers and structure a CAN frame's payload into a predefined number of closely related types of information with an internal identifier, called Suspect Parameter Number (SPN). For instance, there is a PGN for *wheel-based speed and distance* that includes SPNs for speed, distance, direction, implement start/stop operations, etc.

Due to the topological nature of ISOBUS, each frame is broadcasted within the network [7]. Yet, neither CAN nor ISOBUS defines countermeasures or mitigation options for common network-based attack vectors. In addition to serious security aspects, this also raises a large number of questions of privacy and data sovereignty, explicitly in the fulfillment of cooperation tasks in agriculture: As it is common for farmers to include contractors in their value-added chains [4], [11], the ECUs (e.g., VTs) of machines used collect and aggregate precise information about their environment. Since environmental information can include the fertility and yield of a field, farmers are forced to place great trust in contractors. Similarly, the aggregated speed and position of machinery in road transport can lead to further labor law issues.

There has been a considerable amount of related work to establish confidentiality and integrity in CAN (e.g., [9], [12]).

978-1-7281-1025-7/19/$31.00 ©2019 IEEE

 DOI: 10.1109/IPCCC47392.2019.8958765



(a) Original high-resolution application map.

(b) Filtered low-resolution application map.

Fig. 2. The impact of privacy filters exemplarily demonstrated by the accuracy of a crop protection application map. The original data (a) was modified by two filters in (b): GPS coordinates were perturbed with a random bias and a rounding filter was additionally applied to application data from the plant sprayer.

However, these solutions cannot be integrated into existing machines nor are they aimed at our use case in terms of privacy and data sovereignty. Therefore, we developed *CAN't*, an ISOBUS privacy proxy, and already showed its on-the-fly data perturbation and coarsening capabilities for sensitive data [3]. The remainder of this paper briefly explains the *CAN't* architecture and privacy filters (Sec. II), presents implementation details (Sec. III), and finally clarifies our poster and demo contribution (Sec. IV).

II. Architecture & Privacy Filters

CAN't is designed to be a small, cheap, and yet powerful device participating in the ISOBUS. By physically dividing the bus into two separate network segments and to use this device as a man-in-the-middle proxy, *CAN't* is able to relay, filter, and manipulate information between these networks. Thus, broadcasts between ECUs of different segments can be selectively suppressed while data privacy is established and the functionality of the machine is maintained. An optimal injection point for such a proxy can be located directly in front of a data logging ECU, i.e. the VT.

A general overview of the proxy's architecture is illustrated in Fig. 1. The main functionality of *CAN't*, that is forwarding, dropping, and manipulating data, is realized in its *core* component: The encapsulated information of each incoming frame is identified by extracting the frame's PGN, which is matched against a blacklist within the *database*. A blacklisted PGN, or SPN respectively, is associated with one or more privacy filters that are iteratively applied to the frame's content. Depending on the specific filter, a frame can either be simply dropped or manipulated before being forwarded to the target segment. For the on-site configuration of active filters, which are contractually agreed by all participants, e.g. for the sake of the privacy of the employees and the data sovereignty of the farmers, the proxy can be remote-controlled via a *GUI*. Furthermore, the *logging* and *statistics* components are used by the *GUI* to verify correct configuration and behavior of *CAN't* without exposing sensitive information. Detailed information on *CAN't*, its components, and a performance evaluation can be found in [3].

Because some transmitted information is critical to the functionality of the machine, it does not seem practical to completely drop all frames identified by certain PGNs. Instead, often a selective modification of the payload of those frames might be more reasonable. However, there can be no universal manipulating filter since each PGN defines an individual payload structure. Obscuring sensitive data, thus, highly depends on the type, frequency, and accuracy of information. Hence, we have decided to divide filters into three different categories, which must be adapted for each PGN: While *blocking* filters generally drop frames, there might be *value-centric* ones that apply some kind of threshold or rounding operation to original data sets. The third category is used for total or partial *perturbation* of data by the application of a strong noise source. With these filters the accuracy of agricultural application maps can be deliberately reduced, as visualized for example in Fig. 2.

III. Implementation

The privacy proxy is prototypically implemented on the Raspberry Pi platform, which is extended by appropriate CAN transceivers (MCP2551). The software is written in *Google Go* and based on *SocketCAN*, an open-source set of CAN drivers available in the Linux kernel. Our framework is available as *open source* software[1] under BSD license. It includes a set of predefined privacy filters as well as an extensive PGN database. Moreover, it provides monitoring and configuration a GUI implemented as an intuitive HTML5-based single page application using *ReactJS* and *TwitterBootstrap*. The framework was developed and tested using a commercial ISOBUS hardware infrastructure, consisting of a VT manufactured by Competence Center ISOBUS[2] and a specific implement simulator. See [3] for details.

In order to further develop and validate additional filters and to demonstrate the privacy proxy easily, a CAN/ISOBUS simulator is realized. This simulator uses SuperTuxKart 1.0[3],

[1] https://sys.cs.uos.de/cant
[2] https://www.cc-isobus.com
[3] https://supertuxkart.net

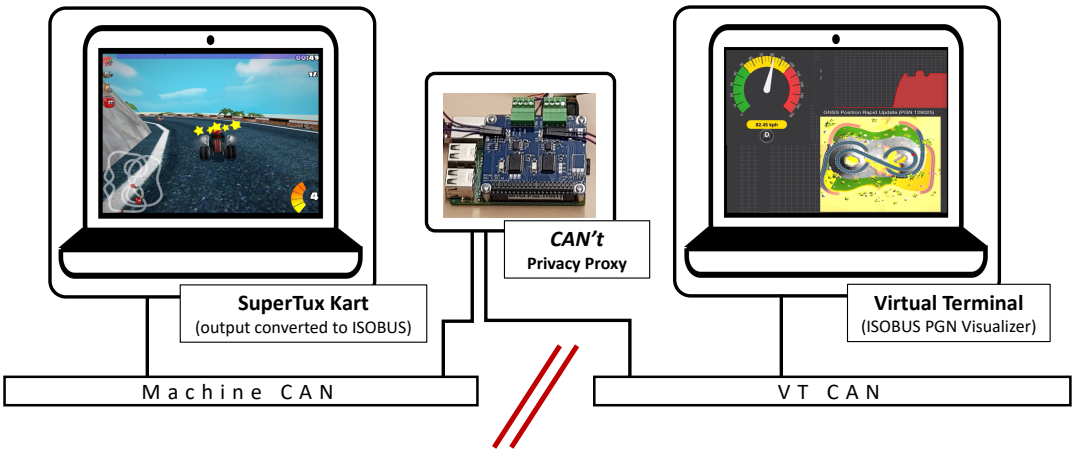


Fig. 3. Demonstration setup used to show the technical feasibility of the realized CAN/ISOBUS privacy proxy and the impact of privacy filters.

an open-source and cross-platform 3D arcade racer video game as data input. For this purpose, a simple modification was implemented that extracts the player's telemetry data from the game, i.e., the speed and the 3D coordinates of the vehicle. During the game, this data is streamed via UDP to an ISOBUS converter, which is also implemented in *Go* on the basis of the Raspberry Pi platform. The converter translates the UDP payload into ISOBUS messages by creating regular frames with corresponding PGN and SPN identifiers, which are subsequently forwarded via its CAN interface.

Moreover, a visualizer is implemented using JavaScript and the same platform, which receives the telemetry data via its CAN interface. It offers a graphical representation as HTML dashboard including a map visualization, speed-gauge, and speed time series.

IV. Poster & Demonstration

Our demonstration will show a fully functional proof-of-concept prototype of *CAN't*. With the SuperTuxKart-based simulator presented in the previous section and the corresponding ISOBUS visualizer, the main contributions of *CAN't* will be demonstrated. To this end, the original CAN/ISOBUS network is separated into two bus segments. These segments are bridged by the privacy proxy as illustrated in Fig. 3. This setup makes it possible to apply various privacy filters, e.g., selective forwarding of ISOBUS messages or application-specific modifications of their payload.

The major goal of *CAN't* is privacy. Hence, the *impact of privacy filters* that are already implemented in our framework will be exemplarily shown by a comparative visualization of unfiltered vs. filtered information. The practical relevance is highlighted by means of typical use cases. In order to involve the audience, an interactive live configuration of privacy filters will be enabled and users will be able to observe their effects. Finally, the limitations of our prototype, use cases, its potential impact on the automotive sector, as well as promising approaches for future work will also be discussed.

Acknowledgments

The authors would like to thank Michel Löpmeier of the Competence Center ISOBUS e.V. (CCI) for supporting the development of *CAN't* with CAN/ISOBUS 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" (02K14A194) and managed by the Project Management Agency Karlsruhe (PTKA). The authors are responsible for the contents of this publication.

References

[1] Agricultural Machinery Association, "ISOBUS 11783 Online Data Base," German Engineering Federation (VMDA). [Online]. Available: https://www.isobus.net
[2] H. Auernhammer, "Precision farming – the environmental challenge," *Comput. Electron. Agr.*, vol. 30, no. 1–3, pp. 31–43, 2001.
[3] J. Bauer, R. Helmke, A. Bothe, and N. Aschenbruck, "CAN't track us: Adaptable Privacy for ISOBUS Controller Area Networks," *Comput. Stand. Inter.*, vol. 66, p. 103344, 2019.
[4] J. Bauer and 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.
[5] S. Cox, "Information technology: the global key to precision agriculture and sustainability," *Comput. Electron. Agr.*, vol. 36, no. 2–3, pp. 93–111, 2002.
[6] Int. 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.
[7] ——, "Road vehicles - Controller area network (CAN) – Part 1: Data link layer and physical signalling," ISO 11898-1:2015, 2015.
[8] U. Kiencke, S. Dais, and M. Litschel, "Automotive Serial Controller Area Network," SAE Int., SAE technical paper 860391, 1986.
[9] C. W. Lin and 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.
[10] A. J. Scarlett, "Integrated control of agricultural tractors and implements: a review of potential opportunities relating to cultivation and crop establishment machinery," *Comput. Electron. Agr.*, vol. 30, pp. 167–191, 2001.
[11] S. Scheuren, S. Stiene, R. Hartanto, J. Hertzberg, and M. Reinecke, "Spatio-Temporally Constrained Planning for Cooperative Vehicles in a Harvesting Scenario," *KI - Künstliche Intelligenz, German Journal on Artificial Intelligence*, vol. 27, no. 4, 2013.
[12] Y. Wu, Y.-J. Kim, Z. Piao, J. G. Chung, and 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.
[13] N. Zhang, M. Wang, and N. Wang, "Precision agriculture – a worldwide overview," *Comput. Electron. Agr.*, vol. 36, no. 2–3, pp. 113–132, 2002.