A New Standard: Redesigning Autonomous Vehicle Communication Using Multi-Party Computation

Sever Latyšov
Responsible Professor: Zekeriya Erkin
Cyber Security Group, Department of Intelligent Systems, Delft University of Technology

Abstract

Around the world millions of people get injured due to traffic accidents. Autonomous vehicles are expected to significantly reduce these numbers. To increase safety, autonomous vehicle communication can be used. Current vehicle communication networks called VANETs have security and privacy protection problems and the vehicle industry is reluctant to use them. Multi-Party Computation (MPC) is a cryptographic technique which allows a set of parties to compute the output of a function while not revealing the input. This paper investigates the potential and feasibility of multi-party computation to solve the present problems of autonomous vehicle communication. Two architectures are proposed and discussed on the basis of a literature study and interviews with experts in MPC and autonomous vehicle communication. The solutions seem to be feasible, but further experimentation is required to confirm this. The main obstacle for the implementation of MPC in autonomous vehicle communication is that at the moment, the autonomous vehicle industry is not concerned with improving autonomous vehicle communication.

1 Introduction

According to the World Health Organization, each year approximately 1.3 million people die as a result of road traffic accidents, and the amount of non-fatal injuries is even much higher [1]. As the world population grows and the roads become more crowded, these numbers will keep increasing. A study by the National Highway Traffic Safety Adminstration (NHTSA) shows that 94% of all car crashes are due to the error of the driver [2]. Driverless or autonomous vehicles therefore have the potential to be the safest transportation to date [3]. Autonomous vehicles have been studied since the 1980s but only in the last decade has the subject gathered massive popularity [4]. This development is due to autonomous vehicles increasingly becoming a reality across the world [5]. Many companies (e.g., Google and Waymo) have been field-testing their autonomous cars for many years. Nuro has recently become the first company allowed to launch commercial autonomous vehicles on public roads [6].

There is, however, little trust in current autonomous vehicles. A survey by the American Automobile Association shows that only 12% percent of people feel safe about riding in a self-driving car. Another 28% do not know how they feel towards the technology [7]. One of the reasons for this uncertainty is that people expect autonomous vehicles to be far safer than they perceive themselves to be [8]. Some incidents have caused distrust when it comes to autonomous vehicles. A very famous case is the death of Elaine Herzberg, who was hit by a self-driving Uber test vehicle in March 2018 and became the first pedestrian to die because of a self-driving car [9].

Besides safety concerns there are also concerns related to cybersecurity and privacy. Autonomous vehicles collect and process a vast amount of data. Some of the data is considered personal data under the definition of the General Data Protection Regulation (GDPR). The GDPR defines personal data as any information relating to an identified or identifiable natural person ('data subject') [10].

Connected vehicle techniques have been developed to improve the safety and mobility of self-driving cars in traffic. Vehicle-to-everything (V2X) communications are used to connect vehicles to infrastructures, pedestrians and other entities. Communication between vehicles is useful as it is much safer if a vehicle knows that the car in front is braking instead of predicting that the car in front is braking based on sensor input. V2X enables useful applications such as providing traffic condition information, electronic toll collection and parking payments and many more [11]. "Up to 80 percent of all unimpaired crashes' scenarios could potentially be addressed by V2V (vehicle-to-vehicle) and V2I (vehicle-to-infrastructure) technology combined" [12]. Connected entities form vehicular ad-hoc networks (VANETs).

For vehicle communication to work, the network has to be reliable and have minimum latency while at the same time data has to be transmitted securely and efficiently. However, VANETs are notoriously unreliable and are prone to malicious attacks. This unreliability jeopardizes the safety on the road. For example, if a vehicle receives tampered GPS (Global Position System) data of another autonomous vehicle, the vehicle may make incorrect decisions and spread

misinformation to other vehicles [13]. Furthermore, the data vehicles transmit can be personal data meaning that personal data will be stored on cars of different people and different companies. VANETs have serious security and privacy issues [14].

Secure Multi-Party Computation (MPC) is a cryptographic technique that has the potential to offer a solution to the security and privacy problems. The technique allows a set of parties to compute the output of a function while not revealing their input data to the other parties. The general theory of MPC was already created in the 1980s, but MPC has seldom been used in practice as the protocols were too slow for practical applications [15]. In the last decade, an interest in the practical use of MPC has been steadily growing and the efficiency and usability of the technique have been improved. MPC has already been successfully used for auctions, electronic voting, statistics gathering, and many other applications [16]. The industry domain of self-driving cars is mostly unexplored.

The aim of this work was to investigate the possibility of MPC as a secure and privacy-enhancing solution for autonomous vehicle communication. To find out to which extent MPC can improve communication between autonomous vehicles. The remainder of the paper is structured as follows: Section 2 explores further the topics of VANETs and MPC, and discusses relevant MPC research in the self-driving car domain. Section 3 presents the methodology of this research. Section 4 includes a design of a theoretical architecture that could improve autonomous vehicle communication. Section 5 discusses the limitations and the feasibility of this architecture are discussed. Section 6 reflects on the research in terms of ethics and reproducibility. Finally, section 7 concludes the paper and discusses potential future work.

2 Background

This section will illuminate the topics of VANETs and V2V communication as well as the challenges they face. Furthermore, the concepts of Multi-Party Computation, Garbled Circuits, Homomorphic Encryption and Additive Secret Sharing are explained and accompanied by the discussion of some related work.

2.1 VANET

VANET (Vehicular Ad-hoc NETwork) is a special type of MANET (Mobile Ad-hoc NETwork). MANET is a decentralized type of wireless network where mobile devices acting as nodes create the network. In the case of VANETs the nodes are the vehicles [17]. VANETs were first introduced in 2001 as vehicular communication systems and have since then become a big part in the development of intelligent transportation systems (ITS) [18]. ITS were mainly used for on-board entertainment services, but VANETs showed that ITS could also be used to improve road safety and traffic conditions.

It is standard for VANETs to have the Wireless Access in Vehicular Environments (WAVE) architecture. The whole architecture is built on the IEEE 802.11p WLAN protocol which allows multiple vehicles to use wireless communication to form a local area network (LAN) and enables data exchange between the vehicles [19]. The emerging technology 5G C-V2X has proven to perform better than 802.11p in certain aspects and could become the standard in the future [20]. 802.11p forms the basis for dedicated short-range communications (DSRC) which is the standardization of frequencies in which VANETs operate [21]. DSRC consists of seven communication channels of 10 MHz which work in the approximate range of 300 meters. DSRC allows vehicles to send messages using their dedicated channels. For example, messages regarding critical safety are send over the Critical Safety of Life channel. To realize vehicle communication an On Board Unit (OBU) is installed inside the vehicles. This is an embedded hardware device with a memory, a processor and computational capabilities [19].

In a VANET network, two types of communication are most often used in practice: vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I). The challenges faced in V2V are slightly different to those in V2I, as the infrastructure nodes do not move [19]. V2V does not require any infrastructure support and is purely ad-hoc in nature. In this research we mainly focus on V2V communication.

VANETs and especially V2V communication are a unique type of ad-hoc network. V2V is characterized by the following challenges [14] [22]:

- **Dynamic topology:** It is expected that the speed and direction of vehicles is constantly changing. Nodes can move up to 200 km/h and still be able to connect with the network.
- **Node heterogeneity:** There are many different types of vehicles, such as emergency vehicles, authority vehicles, transport vehicles, and more.
- **Node density:** The amount of nodes in the network can strongly vary. Sometimes roads can be completely void of vehicles while at other times every spot on the road can be occupied. High node density means more messages to send and process.
- **Intermittent connectivity:** This is the direct result of high mobility. Connectivity between nodes is short and changes frequently. Nodes can disconnect at any time. High packet loss must be avoided.

Data that autonomous vehicles collect and process can be roughly divided into learning and decision-making data. Autonomous vehicles have an enormous amount of data, and almost all of it comes from sensors, e.g. radar, LiDAR, cameras, ultrasonic, and so forth [23]. Most of the sensor data gets processed in the learning and planning phases of self-driving cars. This data will not be used for communication in the foreseeable future. Only some sensor data (e.g., GPS and load) is used in the decision-making process. Other data needed for decision-making is input from the passengers (e.g., vehicle destination). Vehicles also have in-vehicle design information which companies would like

to keep hidden from their competitors. These are the data that are shared or could potentially be shared with V2V communication.

V2V transmits data directly to other vehicles. This is a major concern as VANET is an open-access environment where any node can become a part of the network and can receive or transmit data. Passive adversaries can learn the patterns and habits of car users while active adversaries can send malicious data to cause real harm. There have been many works on the topic of security and privacy and the following have been consulted for this paper [14] [13] [24] [25]. The proposed security models are often computationally heavy, inefficient or they require a lot of storage.

A problem with V2V in autonomous vehicles is that there is no standard for how V2V is used. Some cars only use V2V in groups while others only do one-to-one connections. At the moment the main focus in autonomous vehicle development is vehicle safety. V2V communication is a neglected area, as it often lacks any form of authentication and uses simple encryption techniques. Because of the security issues, only the most basic information is shared, namely speed and GPS location. Sending more information is a risk that companies do not find worth taking.

2.2 Multi-Party Computation

The concept of multi-party computation was introduced in 1982 by Yao as a framework for the Millionaire's cryptography problem [26]. Multi-party computation is a cryptographic technique which allows a set of parties $P_1, \ldots, P_n$ each with their own input $x_1, \ldots, x_n$ to compute some function $y = f(x_1, ..., x_n)$ based on each of the parties' input x_i [27]. The technique also takes into account that the parties do not trust each other, do not want to reveal their inputs to the other parties, and may even try to learn information about the others' inputs.

For MPC to be secure some properties need to be upheld. The ideal/real simulation paradigm is an effective method to find these properties [28]. The paradigm defines an *ideal world* where an independent party gets the private inputs, calculates the functions and returns to everyone the output. This party is assumed to be absolutely honest and trustworthy. The scenario in the ideal world is secure by definition. By comparing the ideal world computations with the computations done in reality, the paradigm shows that MPC has to uphold the following properties [15]:

- **Privacy:** The only input known to a party should be their own. The only information available about other parties' inputs is what can be derived from the output itself.
- **Correctness:** It is guaranteed that from the inputs a correct output will be calculated. The output will only be incorrect if the inputs are inct.
- **Independence of Inputs:** The honest and the corrupted parties must choose their inputs independently from each other.

In order to maintain these qualities, MPC computations get more complex under various adversary models.

It is common in MPC to divide parties into three categories: *input parties*, *computation parties* and *output parties*. The input parties provide the computation with data. Computation parties carry out the privacy-preserving computation on the data provided. Apart from what is revealed through the architecture of the computation, these parties should not know anything about the input data. At last, there are result parties who receive the computation's outcomes. Parties can take on multiple roles [29]. In our research autonomous vehicles take on the roles of both the input and result parties.

The following encryption schemes are important for this research:

Garbled Circuits: The concept of Yao's garbled circuits is used in almost every MPC protocol that evaluate a Boolean circuit [30]. The basic idea is to let party A, called the *creator*, garble the function to be computed. The garbled circuit is sent to party B, called the *evaluator*. Then A gives B a key X_a that corresponds to A's input a. B requires the key Y_b associated to his input, but A should not learn what B's input bit b is [31]. This is done using an Oblivious Transfer (OT) in which A inputs Y_0, Y_1, and B inputs b and learns Y_b, but A learns nothing. If A wants to garble an AND gate, A computes four ciphertexts: $C_{00} = E_{X0,Y0}(Z0)$, $C_{01} = E_{X0,Y1}(Z0)$,$C_{10} = E_{X1,Y0}(Z0)$, $C_{11} = E_{X1,Y1}(Z1)$. They are permuted at random and transmitted to B, who can only decrypt the ciphertext encrypted using the keys X_a, Y_b. B learns the output $Z_{a \wedge b}$ and the parties exchange it [27].

Homomorphic Encryption: When some algebraic manipulation, such as addition or multiplication, acts consistently between the plaintext and the ciphertext, it is called homomorphic encryption. In other words, if we multiply or add the ciphertext, the plaintext changes as well. Let X and Y be two numbers, with encryption and decryption denoted by E and D, respectively. The homomorphic characteristics for X and Y can thus be expressed as: $D[E(X) + E(Y)] = D[E(X + Y)]$ or $D[E(X)xE(Y)] = D[E(XxY)]$. Schemes that only support either multiplication or addition, not both, are called partially homomorphic [32]. Fully homomorphic encryption supports both multiplication and addition simultaneously, but is much more computationally complex [33].

Additive secret sharing: A secret sharing scheme is a cryptographic tool that divides a secret into a number of shares, with one share sent to each computational party. In additive secret sharing, the secret can be reconstructed only when all shares are combined. This scheme is based on additive homomorphic encryption and can be formalized mathematically. A secret s to be shared among n parties is divided in shares $(r_1, \ldots, r_{n-1}, r_n)$, where r_i is random for $i \in \{1, ..., n-1\}$, and $r_n = s - \sum_{i=1}^{n-1} r_i$. To recover the secret all shares have to be combined: $s = r_n + \sum_{i=1}^{n-1} r_i$

[34].

2.3 Related Work

Little research has been done on MPC with vehicle communication and MPC for real-time applications. The reason for the latter being that MPC is still seen as insufficiently fast [35]. Some works related to this research are [36],[37], and [38].

[36] explored the possibility of using MPC for authentication in VANETs. The majority of authentication schemes in VANETs are designed on the basis of digital signature technology in public key infrastructure (PKI). The protocols require excessive computation costs while having low authentication efficiency as they have been simplified to work in VANETs. This problem has also been confirmed during our interviews. The paper suggests an authentication scheme based on the characteristics of the solutions of nonhomogeneous linear equations and by means of oblivious transfers. By theoretical analysis it is shown that the proposed MPC scheme has lower total overhead and less interactivity between the nodes than traditional public key algorithms.

[37] proposed an MPC solution to create a real-time privacy-preserving Consensus-based Speed Advisory System (CSAS). The idea is for vehicles to exchange data using secret sharing before sending it to CSAS, which does the computations. Using SUMO for simulation [39], the scheme is validated in a setting with two vehicles. It is shown that for a small amount of data, both the data sharing and data aggregation process can be done in less than 5ms.

3 Methodology

Two successive methods were employed during this research. The first method was to perform a literature study and the second was to conduct interviews with experts in the field of autonomous vehicle communication. After collecting sufficient information about the technique, a possible application in this field was studied. This section will discuss the methodology of the literature study and the expert interviews.

3.1 Literature study

The purpose of the literature study was to develop a thorough understanding of MPC, autonomous vehicle communication and the current state of MPC in vehicle communication. This was done by first researching the general theory and applications of MPC. After this, the current technology and problems in autonomous vehicle communication were researched. In this area the privacy and data security related problems deemed to be the most relevant. Following this, existing research of MPC use cases in vehicle communication were studied, with a focus on the previously found privacy and security issues. Sufficient research was available about MPC itself and vehicle communication itself, but research on the intersection of these subjects was sparse, as it is still in its early stages.

3.2 Expert Interviews

After the literature study, multiple interviews were conducted with experts in the field of autonomous vehicle communication to gain an even deeper understanding of the problems and considerations in the development of the communication systems and to discuss hypotheses about the proposed use cases. Interviews were held with researchers from the Intelligent Vehicles research group at the TU Delft, the Centre for Accident Research & Road Safety in Queensland (CARRS-Q), TNO, Electrical and Systems Engineering at the UPenn, and engineers from the Formula Student Team Delft. These organizations were chosen based on their experience in the field and their different perspectives on the issues. The interviews proved to be a valuable addition to the prior research as they shed light on the overlooked difficulties which the responsible parties are facing. The insights from the interviews were used to clarify the obstructions related to the proposed use cases and to understand the possibility of their implementation.

4 Design

This section discusses the proposed architectures that use multi-party computation for communication between autonomous vehicles.

4.1 Requirements

Based on the interviews conducted, we mapped out the most important requirements needed for functional data sharing between autonomous vehicles. Certain requirements, like privacy, correctness and independence of inputs, are assumed to be sufficient within MPC, as has already been discussed in Section 2.2.

- **Real-time:** Due to the nature of V2V communication the information sharing has to be as close to real-time as possible. The implementation needs to be time efficient and have low computational complexity. This means that the chosen protocols have to be compatible with the computed functions.
- **Reliability:** Information transfer has to be reliable and the loss of packets should be minimized. If a packet gets dropped it should preferably have a low impact on the transmission. The packet drop should be detected and in case the error correction does not work, the sender should retransmit the packet.
- **Cost:** Self-driving cars are part of the commercial industry and so the cost of the product should be minimized. This concern brings many questions: if the vehicles are going to do the computations is there space for hardware in the vehicles? Moreover, is it better to use servers for computations if they are cheaper, even if the overall communication process becomes slower?
- **Communication:** By reducing how much has to be communicated, the risks of the process going awry decrease. If the interactivity is low and communication rounds are minimized there will be less transmission delays. Minimizing the number of bytes transmitted increases reliability.

- **Trust:** Each car company wants to make as much profit as possible. It is difficult to work with competitors, especially on a subject such as data sharing. Also at the moment, the priority lies with increasing safety, vehicular communication is seen as optimization instead of as a key feature. For companies to use MPC, the technology will have to prove its benefits to the industry.

4.2 Stakeholder Analysis

The main stakeholders in the architecture proposals are autonomous vehicle companies, government institutions, cloud computing services and autonomous vehicle users. All of these parties have different roles in the architecture and consequently different expectations.

Users of self-driving cars want to provide as little personal information as possible while expecting the vehicle to be absolutely safe. The personal data that is collected must be kept secure. Autonomous vehicle companies want to use personal data to improve autonomous vehicle communication and make their technology safer. At the same time they want to protect sensitive data in order to comply with regulations and satisfy users. The companies want to make a profit while avoiding lawsuits, fines and a bad reputation. Government institutions want everything to be done according to regulations and they want to make the roads as safe as possible. Similar to the car companies, cloud computing services are interested in profits, avoiding fines and maintaining the trust of their users.

4.3 Architectures

We propose two MPC architectures for the use in vehicle communication:

Cloud-Aided Two-Party Computation (CA-2PC): Two vehicles in range of each other act as the input parties, computing parties and output parties. The computation uses garbled circuits and is based on a standard two-party computation [40]. Both parties are assisted by cloud servers, called *dealers*, which can be used for preprocessing. The cloud can be a server provided by either the autonomous vehicle company, or by a cloud computing service (e.g., Amazon Web Services). The structure of CA-2PC is illustrated in Figure 1. Information is send over using V2V communication. When two vehicles want to exchange information they generate the garbled circuits, download the preprocessed data, send the data to each other and share the output after computing it.

Three-Party Computation (3PC): Two vehicles in range of each other act as the input parties when they want to exchange information. The input parties distribute their data among the computational parties using additive secret sharing. Each input party splits their data into three random shares of data and sends each share to one computational party. The computational parties consist of three servers, two of which are from autonomous vehicle companies and the third from a government institution for traffic management. It is possible for two different car companies to participate

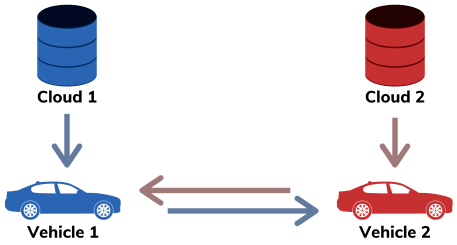

Figure 1: Proposed cloud-aided two-party computation architecture.

in this. Principally, three-party computation using secret sharing has been shown to be highly efficient for additive and boolean functions [41] [42]. After finishing the computation, the servers return the output back to the vehicles. In this case, they are also the output parties. The structure of 3PC is illustrated in Figure 2. Within 3PC the computation does not happen through V2V.

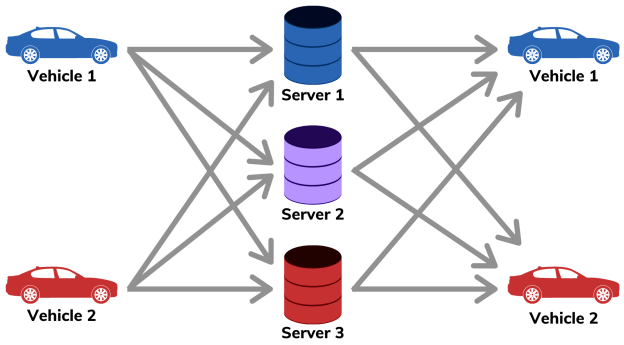

Figure 2: Proposed three-party computation architecture.

4.4 Security Assumptions

This section will explain the security assumptions for CA-2PC and 3PC.

In CA-2PC the malicious adversary security model is assumed. The adversary can make the corrupted parties deviate from the protocol specification. The malicious adversary is also called *active* and can for example refuse to execute the protocol, suspend the protocol at any time or try to learn information about the other parties [43]. In a two-party setting that means that one of the two parties is corrupted. In CA-2PC there is no real reason to assume that parties will act honestly as any vehicle can communicate and there is no telling who is controlling the vehicle. To protect data from the adversary, the protocol needs higher security. This adds a communication and computation overhead, which is not ideal when the protocol has to be as close to real-time as possible. There are many protocols for garbled circuits against malicious attackers, such as cut-and-choose, LEGO and DUPLO. For CA-2PC we propose the Dual Execution protocol from

this paper [44]. This protocol is efficient, optimized and allows for preprocessing.

It does not matter if the clouds in CA-2PC are trusted or not. If the clouds are from the vehicle's own car company there should be no concern. Even if they are not, the computations for preprocessing which are done in the cloud are independent of any input data.

In 3PC a semi-honest security model is assumed. The corrupted parties follow the protocol and do not deviate from it. The adversaries keep track of the inputs and outputs to try and learn more about the other parties. In contrast with the malicious adversary, this adversary is called *passive*. The security model guarantees that there is no inadvertent data leakage during the computations [43]. This weak security model works for 3PC because as we saw in the stakeholder analysis, the risk for the parties is too big to act dishonestly. One of the computing parties is chosen to be a government institution to further increase the trust for the data providers.

If the two autonomous vehicles in 3PC are from the same company, two out of three computing parties are from the same organization. More than half the parties belongs to one group and the semi-honest model does not hold anymore. But as the data from both vehicles already belongs to the company the vehicle users gave consent to have their personal data. There is no problem of parties colluding. The privacy between the cars holds up as vehicle users will not reveal any data to each other.

4.5 Standardization and optimization

There are currently no clear standards for autonomous vehicle communication. Self-driving car companies use their own communication formats or do not use vehicle communication at all. For communication between different vehicle companies to work, there has to be an agreement between the companies. This agreement has to include what data is shared and what functions the MPC protocols calculate, and therefore will decide the message types and what bandwidths of VANETs will be used.

The first versions of the garbled circuit, the homomorphic encryption and the secret sharing protocols were really inefficient. They were even thought to be impossible to use in practice. Since then many optimization techniques have been discovered, making the protocols very efficient if used correctly. For example, the FreeXOR [45] and the oblivious transfer (OT) extension [46] are some important optimizations for garbled circuits.

Optimizations are especially important in CA-2PC. Because the computation takes place in a V2V environment, the online period must be as brief as possible. This is what the clouds are for, as they handle the preprocessing or offline phase. The clouds generate *raw material* based on randomness [47]. This is information independent from the input data and the function that was calculated using MPC, e.g. oblivious transfers. Preprocessing can speed up the online phase more than 50 times [48].

A framework that could be used for the implementation of these architectures is MP-SPDZ [49]. The framework can use many important protocols and use them in a semi-honest or malicious setting. It is up to date with many optimization techniques and allows for preprocessing. The code for the framework is open-source which builds trust with vehicle users. Sharemind [41] is a well-established framework that could also potentially implement the 3PC architecture. Both frameworks use virtual machines, which could add overhead to the computations.

5 Discussion

This section discusses the result for the MPC architecture in V2V communication that was obtained in the previous chapter. The feasibility of the architectures will be examined as well as the potential problems using insights from the interviews with experts. Up until now, most MPC use cases were related to massive data aggregation. In this paper we explored the possibility of using MPC in a real-time scenario. Significant progress toward real-time MPC has been made in recent years, but is not yet fully realized [35]. It is uncertain exactly how efficient MPC is at this moment, and experimentation is required to determine this.

The results consist of two architecture proposals and in both cases only two vehicles are involved. From the interviews it became clear that the scenario of just two vehicles communicating should be the first step in developing the MPC structure. If one-to-one vehicle communication works flawlessly, scaling to multiple vehicles should be straightforward.

The advantage of multiple vehicles communicating simultaneously is that certain functions become more precise and more vehicles can start directly implementing the result. For example, it becomes possible to compute the best driving speed for the environment at that moment. With only one-to-one communication a vehicle will have to communicate multiple times with different vehicles before coming to the same conclusion.

The disadvantage of multiple vehicle communication is that companies have to agree on a lot more message formats and if the communication is implemented using VANETs, disconnections mid-computation will occur more frequently.

Two different architectures were proposed because they both appear to be feasible whilst having vastly different benefits and shortcomings. The first architecture which we called CA-2PC requires no intermediate parties and therefore acts more independently, which is valuable for a self-driving car.

A drawback of CA-2PC is that the car communicates through VANETs and communication can get disrupted because of the rapidly changing node topology.

The second architecture which we called 3PC is much more reliable, as the returning output can be received even after the vehicles separate. A drawback of 3PC is that it depends on commercial companies working together.

MPC will utilize functions which take communication data

from autonomous vehicles as input and give output that would aid the autonomous driving. The protocol used in CA-2PC is, to a greater extent than 3PC, influenced by the types of functions which must be computed. However, these functions do not yet exist, and will likely be a combination of arithmetic functions and for the most part comparison functions, according to the interviewees.

Protocols based on garbled circuits perform well with comparison functions while homomorphic encryption and additive secret sharing perform well with arithmetic functions [27]. The protocol should not be chosen based on performance alone, but also based on how well they satisfy the requirements elicited in Section 4.1.

Considering these requirements, secret sharing has a high interactivity cost. The number of communication rounds is linear in the depth of the circuit being computed, while homomorphic encryption and garbled circuits have constant number of rounds [50].

The interviewees indicated that homomorphic encryption has a lower reliability than garbled circuits. In the first communication round homomorphic encryption needs to send over ciphertexts which contain large amounts of data and therefore need to be split up in packets. If one of the packets drops, the whole ciphertext has to be send again. Garbled circuits do not have this problem.

Therefore, based on the requirements as well, garbled circuits proved the most viable protocol to use in CA-2PC. If it turns out that the vast majority of functions are arithmetic, then homomorphic encryption is the right protocol for CA-2PC. Homomorphic encryption reaps great benefits from preprocessing, just as it does for garbled circuits [47].

In 3PC, it can be assumed that communication and interactivity costs have a negligible effect on the computation as the servers can be assumed to always be available. As such, secret sharing was deemed to be more feasible by the interviewees.

The feasibility of both architectures running in real-time depends on the size and depth of the functions' circuits. In CA-2PC the malicious adversary model creates the biggest overhead. [51] showed 2PC to be able to compute 1.3 billion gates in a few seconds using a semi-honest model. Additionally, [44] demonstrated hundreds of thousands of gates within seconds using the malicious adversary protocol we suggested in Section 4.4.

A model similar to 3PC calculates 1 billion gates in a single second under the right conditions while assuming a malicious adversary model [50]. [37] showed that real-time vehicle communication is achievable with very small amounts of data. From the interviews, it became clear that the functions our architectures are going to compute will not be as big as those in the mentioned papers. This is promising, but experimentation is necessary to prove the feasibility of both architectures. In light of our research 3PC is more feasible than CA-2PC, as the model is more reliable and three dedicated servers will always be more powerful than computations run on vehicles (even if they are assisted by cloud computing).

The autonomous vehicle industry is at the moment not concerned with improving autonomous vehicle communication. It has been sidelined until the safety of self-driving cars is fully guaranteed. For MPC to be used in the industry more interest in vehicle communication needs to develop first. Apart from this MPC is up till now largely unknown within the autonomous vehicle industry, as MPC is only recently becoming efficient enough to be used in the industry.

Despite the previously mentioned setbacks, MPC shows many promises in the autonomous vehicle industry. Using MPC vehicle communication can be made secure and privacy-preserving. This allows autonomous vehicles to share more data, achieve better situational awareness and thereby make the roads safer and more environmentally friendly.

6 Responsible Research

This section discusses the topics involving responsible research and their application to this paper. The limits of investigation are covered, the ethics regarding MPC in autonomous vehicle communication are discussed, and the reproducibility of the research is described.

6.1 Limits of Investigation

There are various limits to the results of this research that must be mentioned in order to understand their relevance.

The most significant limitation is the absence of experimentation in existing vehicle communication systems. Most experts in autonomous vehicles technology are not familiar with multi-party computation techniques. As the autonomous vehicle technologies are only in their early stages, the engineers are mostly occupied with the development of core functionalities instead of succeeding issues such as privacy handling. Also, since public use of autonomous vehicles is still restricted, privacy is not yet an immediate problem for related organizations. For that reason this research was limited to theory only. Nevertheless, the experts were realistic about the future importance of privacy in vehicle communication and were able to provide great insights about later development.

Another limit was the restricted time period available for this research. There was sufficient time to gain proficiency in the topics of MPC and autonomous vehicle communication and to explore the proposed architecture theoretically. Unfortunately time constraints obstructed the research from conducting practical experiments, which would have given important insights on the technical elements of the architecture. Some important factors in the application of MPC to autonomous vehicle communication can only be found by experimentation, e.g. the computation time or the error count. Since this was not possible, assumptions had to be made about these factors based on secondary resources. To finalize research on MPC in autonomous vehicle communication practical work must be done.

6.2 Ethics

MPC could solve privacy issues that are currently unsolved in autonomous vehicle communication. GPS-locations, destinations and parameters which reveal engineering secrets are currently communicated directly between autonomous vehicles in order for them to function. This is often privacy and security sensitive information from which ethical concerns and various risks may arise. For example, organizations could exploit location tracking for influence or for profit, and if the data falls in the wrong hands it could be used for malevolent activities.

Normally when a system requires someone's sensitive data to function, the person has to accept terms allowing the information to be used. For example, to make use of a navigation app one is asked to consent to their GPS location being used, but is also offered the choice of rejecting the location-based navigation. In an autonomous vehicle system this is different because one can only participate in traffic if the person's location is shared with the system. MPC could allow people to participate in the system without sharing their sensitive data with others.

Privacy is only a small part of the ethical discussion regarding autonomous vehicle technology. The most prominent topic in autonomous vehicle ethics is the preservation of human life in threatening situations. The discussed architectures touch both privacy and the human life. By using MPC techniques the computation time is affected, meaning cars will communicate at a slower rate. Thus the question arises whether the application of MPC can affect the safety of the autonomous vehicle system by impairing its processing speed. Before applying the architectures in dangerous situations it is therefore important to first research the effect of MPC on the safety of the system. After that, a better decision can be made about which parts of vehicle communication should be done through MPC.

6.3 Reproducibility

As was described in the methodology section, this research is based on a thorough literature study of available research on autonomous vehicle communications and MPC techniques, followed by a set of interviews with technology experts. To attain the knowledge that was used for this research the referenced articles should be studied and experts in the field of autonomous vehicle communication should be interviewed. Perhaps in the future more research will be available and experts will be more informed on MPC leading to a more accurate result.

Due to the relatively small amount of interviewees for this research, there is a significant chance that some stakeholders were left out from considerations. However, solving privacy issues in vehicle communication is mostly a technical issue and the expert opinions were also mostly on the technical side. More interviews will therefore likely improve the research on the possibility of using the MPC architectures.

7 Conclusions and Future Work

From this research it became clear that MPC is far from being implemented in autonomous vehicle communication.

Autonomous vehicle companies are primarily working on the core functionalities of the technology of autonomous driving and are mainly focused on increasing safety and efficiency, and less on private and secure V2X communication. Even though MPC could contribute to further increase safety through secure communication, it might take time for an MPC architecture to find its place as it needs more research and experimentation to determine its valuation.

Autonomous vehicle communication is currently a rather unregulated field, which means that CA-2PC will be quicker to implement as it requires no cooperation between intermediate (company) servers as opposed to 3PC. Furthermore, MPC has seen no obvious real-time applications thus far, and these are absolutely necessary within autonomous driving. Further experimentation is required to determine if real-time MPC for the application in V2V communication is possible.

However, MPC could potentially be used in the future precisely because there is no great secure communication standard for autonomous vehicles at this moment. If this happens, CA-2PC and 3PC could both prove to be viable architectures, but 3PC has more potential as it is a more reliable and more powerful than CA-2PC.

Future work can include the experimentation and research as mentioned above.

References

[1] World Health Organization. Road traffic injuries, 2021. *URL https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries Accessed June 2021.*

[2] Santokh Singh. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. Technical report, 2015.

[3] Nidhi Kalra. *Challenges and approaches to realizing autonomous vehicle safety*. RAND, 2017.

[4] Kambria. The History and Evolution of Self-Driving Cars, 2019. *URL https://kambria.io/blog/the-history-and-evolution-of-self-driving-cars/ Accessed June 2021.*

[5] Raul Fernandez-Rojas, Anthony Perry, Hemant Singh, Benjamin Campbell, Saber Elsayed, Robert Hunjet, and Hussein A Abbass. Contextual awareness in human-advanced-vehicle systems: A survey. *IEEE Access*, 7:33304–33328, 2019.

[6] Ryan Daws. Nuro receives first permit to operate self-driving cars commercially in california, 2020. *IoTnews, URL https://iottechnews.com/news/2020/dec/24/nuro-first-permit-operate-self-driving-cars-commercially-california/AccessedJune2021.*

[7] American Automobile Association. Self-driving cars stuck in neutral on the road to acceptance, 2020. *URL https://newsroom.aaa.com/2020/03/self-driving-*

cars-stuck-in-neutral-on-the-road-to-acceptance/ Accessed June 2021.

[8] Michael A Nees. Safer than the average human driver (who is less safe than me)? examining a popular safety benchmark for self-driving cars. *Journal of safety research*, 69:61–68, 2019.

[9] Zhi Quan Zhou and Liqun Sun. Metamorphic testing of driverless cars. *Communications of the ACM*, 62(3):61–67, 2019.

[10] European Commission. 2018 reform of EU data protection rules, 2018. *URL https://gdpr-info.eu/ Accessed June 2021*.

[11] Steven E Shladover. Connected and automated vehicle systems: Introduction and overview. *Journal of Intelligent Transportation Systems*, 22(3):190–200, 2018.

[12] Siva RK Narla. The evolution of connected vehicle technology: From smart drivers to smart cars to... self-driving cars. *Ite Journal*, 83(7):22–26, 2013.

[13] Jin Cui, Lin Shen Liew, Giedre Sabaliauskaite, and Fengjun Zhou. A review on safety failures, security attacks, and available countermeasures for autonomous vehicles. *Ad Hoc Networks*, 90:101823, 2019.

[14] Yeka Joseph Abueh and Hong Liu. Message authentication in driverless cars. In *2016 IEEE Symposium on Technologies for homeland security (HST)*, pages 1–6. IEEE, 2016.

[15] Yehuda Lindell. Secure multiparty computation (mpc). *IACR Cryptol. ePrint Arch.*, 2020:300, 2020.

[16] Ueli Maurer. Secure multi-party computation made simple. *Discrete Applied Mathematics*, 154(2):370–381, 2006.

[17] Ravi Tomar, Manish Prateek, and GH Sastry. Vehicular adhoc network (vanet)-an introduction. *International Journal of Control Theory and Applications*, 9(18):8883–8888, 2016.

[18] Chai K Toh. *Ad hoc mobile wireless networks: protocols and systems*. Pearson Education, 2001.

[19] Mohamed Nidhal Mejri, Jalel Ben-Othman, and Mohamed Hamdi. Survey on vanet security challenges and possible cryptographic solutions. *Vehicular Communications*, 1(2):53–66, 2014.

[20] YA Dahou Djilali, Y Bakhtil, B Kouninef, and B Senouci. Performances evaluation study of vanet communication technologies for smart and autonomous vehicles. In *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 79–84. IEEE, 2018.

[21] Qing Xu, Tony Mak, Jeff Ko, and Raja Sengupta. Vehicle-to-vehicle safety messaging in dsrc. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 19–28, 2004.

[22] Sabih ur Rehman, M Arif Khan, Tanveer A Zia, and Lihong Zheng. Vehicular ad-hoc networks (vanets)-an overview and challenges. *Journal of Wireless Networking and Communications*, 3(3):29–38, 2013.

[23] Jaycil Z Varghese, Randy G Boone, et al. Overview of autonomous vehicle sensors and systems. In *International Conference on Operations Excellence and Service Engineering*, pages 178–191, 2015.

[24] Xiaonan Liu, Zhiyi Fang, and Lijun Shi. Securing vehicular ad hoc networks. In *2007 2nd International Conference on Pervasive Computing and Applications*, pages 424–429. IEEE, 2007.

[25] Maxim Raya and Jean-Pierre Hubaux. The security of vehicular ad hoc networks. In *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, pages 11–21, 2005.

[26] Andrew C Yao. Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*, pages 160–164. IEEE, 1982.

[27] Claudio Orlandi. Is multiparty computation any good in practice? In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5848–5851. IEEE, 2011.

[28] Chuan Zhao, Shengnan Zhao, Bo Zhang, Zhongtian Jia, Zhenxiang Chen, and Mauro Conti. Secure comparison under ideal/real simulation paradigm. *IEEE Access*, 6:31236–31248, 2018.

[29] Peeter Laud and Alisa Pankova. Privacy-preserving record linkage in large databases using secure multiparty computation. *BMC medical genomics*, 11(4):33–46, 2018.

[30] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 162–167. IEEE, 1986.

[31] Thomas Schneider and Michael Zohner. Gmw vs. yao? efficient secure two-party computation with low depth circuits. In *International Conference on Financial Cryptography and Data Security*, pages 275–292. Springer, 2013.

[32] Mohammed Golam Kaosar, Russell Paulet, and Xun Yi. Fully homomorphic encryption based two-party association rule mining. *Data & Knowledge Engineering*, 76:1–15, 2012.

[33] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009.

[34] Mahir Can Doganay, Thomas B Pedersen, Yücel Saygin, Erkay Savaş, and Albert Levi. Distributed privacy preserving k-means clustering with additive secret sharing. In *Proceedings of the 2008 international workshop on Privacy and anonymity in information society*, pages 3–11, 2008.

[35] Joseph I Choi and Kevin RB Butler. Secure multiparty computation and trusted hardware: Examining adoption

challenges and opportunities. *Security and Communication Networks*, 2019, 2019.

[36] Cheng Song, Mingyue Zhang, and Wei-Ping Peng. Research on secure and privacy-preserving scheme based on secure multi-party computation for vanet. *J. Inf. Hiding Multim. Signal Process.*, 9(1):99–107, 2018.

[37] Mingming Liu, Long Cheng, Yingqi Gu, Ying Wang, Qingzhi Liu, and Noel E O'Connor. Mpc-csas: Multi-party computation for real-time privacy-preserving speed advisory systems. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[38] Tao Li, Lei Lin, and Siyuan Gong. Autompc: Efficient multi-party computation for secure and privacy-preserving cooperative control of connected autonomous vehicles. In *SafeAI@ AAAI*, 2019.

[39] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. Sumo–simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind, 2011.

[40] Benny Pinkas, Thomas Schneider, Nigel P Smart, and Stephen C Williams. Secure two-party computation is practical. In *International conference on the theory and application of cryptology and information security*, pages 250–267. Springer, 2009.

[41] Dan Bogdanov, Sven Laur, and Jan Willemson. Sharemind: A framework for fast privacy-preserving computations. In *European Symposium on Research in Computer Security*, pages 192–206. Springer, 2008.

[42] Dan Bogdanov, Margus Niitsoo, Tomas Toft, and Jan Willemson. High-performance secure multi-party computation for data mining applications. *International Journal of Information Security*, 11(6):403–418, 2012.

[43] Zheng Cao, Chenlin Huang, and Yun Li. A study on the improvement of computation, communication and security in garbled circuits. In *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*, pages 609–617. IEEE, 2021.

[44] Peter Rindal and Mike Rosulek. Faster malicious 2-party secure computation with online/offline dual execution. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 297–314, 2016.

[45] Vladimir Kolesnikov and Thomas Schneider. Improved garbled circuit: Free xor gates and applications. In *International Colloquium on Automata, Languages, and Programming*, pages 486–498. Springer, 2008.

[46] Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. More efficient oblivious transfer extensions. *Journal of Cryptology*, 30(3):805–858, 2017.

[47] Carsten Baum, Ivan Damgård, Tomas Toft, and Rasmus Zakarias. Better preprocessing for secure multiparty computation. In *International Conference on Applied Cryptography and Network Security*, pages 327–345. Springer, 2016.

[48] Yan Huang, David Evans, Jonathan Katz, and Lior Malka. Faster secure two-party computation using garbled circuits. In *USENIX Security Symposium*, volume 201, pages 331–335, 2011.

[49] Marcel Keller. Mp-spdz: A versatile framework for multi-party computation. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1575–1590, 2020.

[50] Toshinori Araki, Assi Barak, Jun Furukawa, Tamar Lichter, Yehuda Lindell, Ariel Nof, Kazuma Ohara, Adi Watzman, and Or Weinstein. Optimized honest-majority mpc for malicious adversaries — breaking the 1 billion-gate per second barrier. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 843–862, 2017.

[51] Shay Gueron, Yehuda Lindell, Ariel Nof, and Benny Pinkas. Fast garbling of circuits under standard assumptions. *Journal of Cryptology*, 31(3):798–844, 2018.