A Privacy-Preserving Infrastructure for Driver's Reputation Aware Automotive Services

Gianpiero Costantino[1], Fabio Martinelli[1], Ilaria Matteucci[1](✉), and Paolo Santi[1,2]

[1] Istituto di Informatica e Telematica, Consiglio Nazionale delle Ricerche, Pisa, Italy
{gianpiero.costantino,fabio.martinelli,ilaria.matteucci}@iit.cnr.it
[2] MIT Senseable City Laboratory, Cambridge, MA, USA

Abstract. Even though the introduction of ICT in transportation systems leads to several advantages in terms of efficiency of transport, mobility, traffic management, and in improved interfaces between different transport modes, it also brings some drawbacks in terms of increasing security challenges, also related to human behavior. For this reason, in the last decades, attempts to characterize drivers' behavior have been mostly targeted towards risk assessment and, more recently, to the training of machine learning software for autonomous driving. In this paper, we propose, for the first time, to use driver behavioral characterization to build a general *reputation profile*, that can be used to create innovative, reputation-aware automotive services. As a first step towards realizing this vision, we present guidelines for the design of a privacy preserving vehicular infrastructure that is capable of collecting information generated from vehicles sensors and the environment, and to compose the collected information into driver reputation profiles. In turn, these profiles are exchanged in a privacy preserving way within the infrastructure to realize reputation-aware automotive services, a sample of which are described in the paper. As a fundamental component of the infrastructure, we show that: *i*) multi-dimensional reputation profiles can be formed building upon the recently introduced notion of driver DNA; *ii*) multi-dimensional comparison of profiles can be achieved by means of a reputation lattice rooted in the notion of algebraic c-semiring; and *iii*) a secure two-party mechanism can used to provide services to drivers on the basis of their reputation and/or DNA's parameters.

Keywords: Drivers' reputation profile · Privacy preserving infrastructure · Vehicular network · Reputation-aware services

1 Introduction

Recalling the directive of the European Union 2010/40/EU, made on the 7th of July 2010 [20], Intelligent Transportation Systems (ITS) are "*advanced applications, which [...] aim to provide innovative services relating to different modes of*

T. Groß and T. Tryfonas (Eds.): STAST 2019, LNCS 11739, pp. 159–174, 2021.
https://doi.org/10.1007/978-3-030-55958-8_9

transport and traffic management and enable various users to be better informed and make safer, more coordinated and 'smarter' use of transport networks". In particular, the directive defines an ITS as a system in which Information and Communications Technology (ICT) is applied in the field of road transport, including infrastructures, like tunnels and vehicles. In fact, advances in both vehicle and personal communication technologies are creating increasing opportunities for collecting data within and around the car. With thousands of signals customarily generated by today's vehicles, a car can be considered a veritable mobile sensing platform that produces a few Gigabytes of data per hour [19].

Besides creating potential privacy and security issues when this immense amount of data is connected to the Internet, as implied by the transition to connected and autonomous vehicles, opportunities for unprecedented understanding and optimization of what happens in vehicular infrastructure arise. Within this context, a problem of particular interest is how to leverage vehicular and/or smartphone data to characterize driver behavior. Its characterization is especially interesting for the auto insurance industry, since it can be used to produce accurate risk profiles and personalized policy rates [22]. Characterizing driver behavior finds application also in the development of autonomous driving technologies, where "good" driving styles can be used to train the car control software and give a human feeling to autonomous driving.

This paper suggests a possible use of driver behavior characterization that substantially evolves its role beyond what currently considered in the insurance and autonomous vehicle industry. Building upon the recently proposed notion of Driver DNA [12], we herein propose that vehicle-collected data can be used to compute a "driver reputation" profile that synthetically summarizes a driver's reputation within the vehicular ecosystem. Reputation profiles of circulating drivers can, then, be exchanged in a privacy preserving way with surrounding vehicles or infrastructure to enable innovative management of road infrastructure and driver-aware ITS services, as described in Sect. 5. A key component of the envisioned notion of driver reputation is a framework that enables secure and private exchange of driver reputation profiles between vehicles and between a vehicle and the road infrastructure. The initial design of such a framework is the focus of the present paper, in which we introduce a privacy-preserving infrastructure able to evaluate the reputation of drivers and to provide them with customized services based on their reputation evaluation.

The paper is organized as following: Section 2 reports some literature about driver behavior characterization through vehicles parameters and possible applications and services designed accordingly. Section 3 presents a possible approach to profile a driver, estimate her reputation, and eventually compare different drivers' profile. Section 4 describes our proposed infrastructure able to collect information and to exchange reputation profiles in a privacy-preserving way. Section 5 proposes some ideas of possible services that the infrastructure can provide to "good " drivers, i.e., drivers with a high reputation while Sect. 6 describes our prototype of privacy preserving comparison functions with experimental results. Finally, Sect. 7 analyse the presented work and discuss some points regarding it while Sect. 8 draws the conclusion of the paper and outlines future research directions.

2 Related Work

In the last few years, interest about the characterization of driver behavior according to information collected from the vehicle has consistently increased. However, to the best of our knowledge, none of the existing work attempts to link driver behavior to the notion of reputation and trust as proposed herein.

One of the early works in this field is presented in [4], where the authors proposed a traffic simulation model incorporating assumptions about what a safe drivers' behavior should be. The main outcome of the paper is the comparison between results obtained in the simulation and the real world.

Other recent works [9,26] present approaches to identify reckless drivers based on a combination of speed and acceleration. Both measures are retrieved from different ICT systems present in the vehicle itself. Indeed, in [9], the authors used GPS-enabled mobile phones as a low-cost opportunity for collecting instantaneous vehicle speed and other information. In [26], the information was retrieved from SD Card and GPS on vehicle.

In [10], the driver is considered as part of the vehicle system (driver-in-the-loop), more specifically as the control unit of the entire system. In this way, the authors described three methods to identify driver behavior as a comparison with the actual and the expected behavior of the system by considering different aspects of the drive-in-the-loop vehicle system.

Works about how to link the driver behavior with traffic accidents, safety on roadside network, and possible rewarding are mostly related to the insurance world. For instance, reference [22] is about the risk of reckless drivers and how insurance reward can depend on the driver behavior. Adapting insurance fee to driver behavior is promoted as a method to incentivize drivers to drive more carefully and reduce accidents.

To our best knowledge, the idea of characterizing driver's behavior with the final aim of computing a comprehensive driver's reputation profile and to realize reputation-aware vehicular services is novel to this paper.

About reputation-aware vehicle service, several services for ITS have been introduced in the literature. Following the standardization work of European Telecommunications Standards Institute (ETSI), ITS applications (or service) have been categorized in a number of classes. While their requirements and operational constraints have been defined in ETSI, security specifications are not fully defined and mostly left to the single developers. For instance, secure and privacy aware versions of two representative classes of ITS applications are Driver Assistance – Road Hazard Warning, and Community Services. In case of road hazard warning, there is ample literature that studies under what conditions the communication network (V2V and V2I communication) is able to provide the adequate level of responsiveness necessary to enable early hazard detection [11]. Since security and privacy requirements as mandated by the proposed architecture will introduce significant communication/computational overhead, there is a need of carefully analyzing and testing the interplay between security level, communication performance, and achieved effectiveness in providing secure and early warning to the drivers.

3 Defining Driver Reputation

This section describes a possible way of defining the notion of driver reputation. We start by observing how to objectively quantify driver reputation starting from vehicle collected data is a very challenging problem by itself. While intuitively understandable by the human mind – it is relatively easy, when you sit beside a driver, to judge if she is "good" or "bad" at driving –, the notion of "good" driving style, which should be the basis for establishing a driver's reputation, is evasive from a quantitative viewpoint.

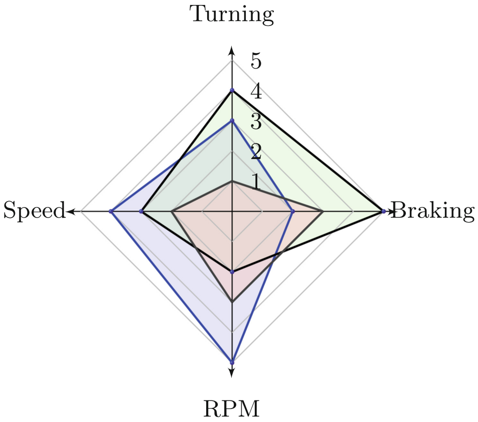


Fig. 1. Radar graph representation of a Driver's DNA.

Recently, the notion of Driver DNA [12] has been proposed to concisely represent a driver's driving style starting from car-collected data analysis, integration with road and weather information, and comparison with peer drivers. The Driver DNA, as defined in [12], is composed of four parameters which cannot be directly compared with each other. Individually, each parameter is measured with a rank ranging between 0 (lowest score) and 5 (highest score). The four parameters are: braking (b), turning (t), speeding (s), and RPM (rpm) (revolutions per minute). The first parameter (braking intensity) is used to quantify a driver's aggressiveness, the second (steering wheel angle) is used to quantify comfort in driving, the third parameter (driving above speed limit), which is also combined with weather information, is directly related to accident risk, while the fourth parameter (engine RPM) is used, when compared with values obtained by peer drivers, as a proxy of a driver's fuel efficiency.

Following [12], we represent the profile of each driver as a tuple of four elements (b_i, t_i, s_i, rpm_i), with $b_i, t_i, s_i, rpm_i \in [0, 5]$, one for each parameter we are going to consider to identify the driver's DNA. Using the profile, we associate to each driver a reputation value.

As the four parameters composing driver DNA cannot be directly compared – i.e., a score of 4 in braking cannot be compared to a score of 4 in speeding – the authors of [12] suggests graphically representing a driver's driving style as a radar graph of the four dimensions, where a relatively larger area of the radar graph indicates a relatively better driver.

Starting from this idea, we enhance the driver's characterization by adding the notion of *driver's reputation* as a unique value that identifies the goodness or recklessness of the driver. In fact, we consider as driver *reputation score* R_{D_i} the internal area identified by the radar graph derived by the four parameters of the driver's DNA. As seen from Fig. 1, the area of a radar graph can be calculated as the sum of the areas of the four triangles composing the graph, each having two of the parameters composing the profile as perpendicular sides. Hence, given the 4-tuple $P_{D_i} = (b_i, t_i, s_i, rpm_i)$ associated with driver D_i, her reputation R_{D_i} can be computed as follows:

$$R_{D_i} = \frac{(b_i \times t_i) + (t_i \times s_i) + (s_i \times rpm_i) + (rpm_i \times b_i)}{2}$$

Note that the order of parameters in the graph influenced the result of the area. Thus, considering the 4-tuple (b, t, s, rpm), we label the graph starting from the right-hand side with the first element of the tuple, i.e., b, and then we proceed counterclockwise to label the other directions with the remaining parameters, as in Fig. 1. To ensure consistency, the same order of parameters is used to compare different driver's profile.

As it will become clear later on, a single reputation score associated to a driver might not be sufficient to enable reputation-aware automotive services as described in the following. For this reason, we set forth the notion of *reputation profile* for a driver, which we define as:

$$RP_{D_i} = ((b_i, t_i, s_i, rpm_i), R_{D_i})$$

i.e., the profile and the synthetic reputation score.

According to this definition of reputation, we have to characterize "good" and "bad" driver behavior. Different strategies may be followed, e.g., the median value of each measure, i.e., 2.5, as threshold value to distinguish between good and bad. Hence, a driver has a *good* behavior when her reputation score is higher than 12.5, and a *bad* behavior, otherwise.

Once it has been calculated, the reputation score becomes part of the driver profile in addition to the other information in the profile. Hence, keeping also the information in the profile, that is richer than the single score R_{D_i} it is possible to allow services responsive to specific aspects of driving, such as, fuel-efficiency, accident-risk, etc.

Moreover, two drivers could be directly compared through their reputation score. However, it is possible that two drivers have the same reputation scores but for very different reasons. Indeed, being the parameters values independent and not comparable to one another, the results of the ordering of driver's profiles is a *lattice* as the one in Fig. 2. We refer to it as *reputational lattice* in which all the driver with the same reputation score are at the same level of the lattice. Having the same reputation score, are classified in the same way with respect to the ITS. In this case, we use the driver's information to distinguish among drivers. In fact, a better assessment of driver reputation can be achieved by accounting for the individual parameters that compose a driver's DNA.

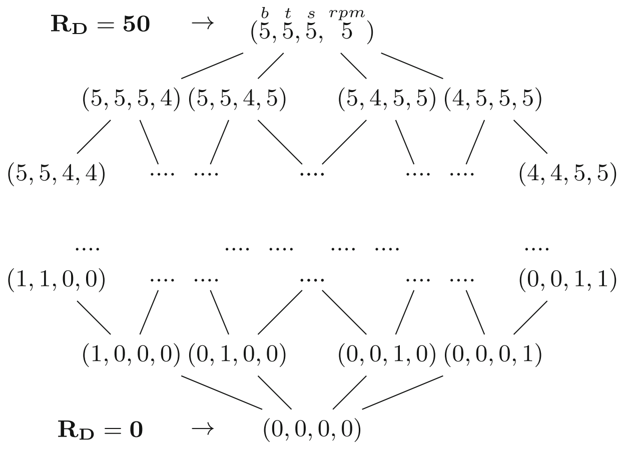


Fig. 2. Reputational lattice.

Using the lexicographic order on tuples of values, it is possible to prioritize one parameter over another (depending on the order of the components in the lexicographic order itself), and to compare different driver profiles to customize transportation services according to their reputation.

Example 1. Let us consider three drivers profiles:

$$P_{D_A} = (\overset{b}{2}, \overset{t}{3}, \overset{s}{4}, \overset{rpm}{5})$$

$$P_{D_B} = (\overset{b}{5}, \overset{t}{4}, \overset{s}{3}, \overset{rpm}{2})$$

$$P_{D_C} = (\overset{b}{3}, \overset{t}{1}, \overset{s}{2}, \overset{rpm}{3})$$

represented in Fig. 1. The reputation score of the three drivers is calculated as follows:

$$R_{D_A} = \frac{(2 \times 3 + 3 \times 4 + 4 \times 5 + 5 \times 2)}{2} = 24$$

$$R_{D_B} = \frac{(5 \times 4 + 4 \times 3 + 3 \times 2 + 2 \times 5)}{2} = 24$$

$$R_{D_C} = \frac{(3 \times 1 + 1 \times 2 + 2 \times 3 + 3 \times 3)}{2} = 10$$

Hence, driver D_A and driver D_B have the same reputation higher than 12,5, so they are considered as good drivers. Driver D_C is a reckless driver, since her reputation is less than 12,5, and consequently, less than the reputation of the other two drivers. However, if we want to compare the three drivers with respect to the *braking parameter*, we note that, the worst driver is D_A. Moreover, D_A is a good driver but the value of the braking parameter is less than 2.5 (it is 2), hence with respect to this parameter, it is considered an "aggressive" driver.

4 Our Privacy Preserving Infrastructure

We assume to work in an Intelligent Transportation Systems as the one depicted in Fig. 3. It is composed of three layers *Ground*, *Fog*, and *Cloud*. The infrastructure we have in mind is based on Fog [23–25] and Cloud computing. The *Ground layer* involves all vehicles that interact with the fog layer to manage and share in-vehicle information. Vehicles contain a large number of internal sensors, e.g., photonic sensors, LiDARs, and communication systems, that can be used, among other things, to sense the quality of the road, traffic, vehicle trajectories, weather conditions, and so on. The *Fog layer* is composed by fog nodes, that are smart components of the road infrastructure, and can be located, for instance, at a gas station, a smart traffic light, a pay toll station, and so on. The fog node is able to collect and exchange data with vehicles and other components of the infrastructure in a safe and secure way. In the same way, fog nodes communicate with the cloud to perform more complex calculation, in case there are required to provide a better service to the drivers. Once smart devices at the fog layer collect information from vehicles, the data can be forwarded to the *Cloud layer*. In this layer, all data coming from the different devices at the fog layer are collected, where upon some analytic operations are executed to obtain both new derived information able to improve the safety of the stakeholders, or to provide customized applications to the infrastructure nodes. The Cloud layer will also contribute to implement the security and privacy aspects [15].

We also assume that each driver in the ITS reported in Fig. 3 is characterized by a multi-dimensional reputation profile, which should be considered as a valuable and private information to the driver. Reputation profiles of drivers become a sort of passport in the infrastructure. Thus, they can be exchanged in a secure and private way with surrounding vehicles and roadside infrastructure to realize innovative reputation-aware vehicular services, a sample of which are described in Sect. 5.

4.1 Secure Two Party Computation

Given the importance of a driver's reputation profile in the envisioned scenario, the proposed infrastructure shall guarantee that such profiles are exchanged in a

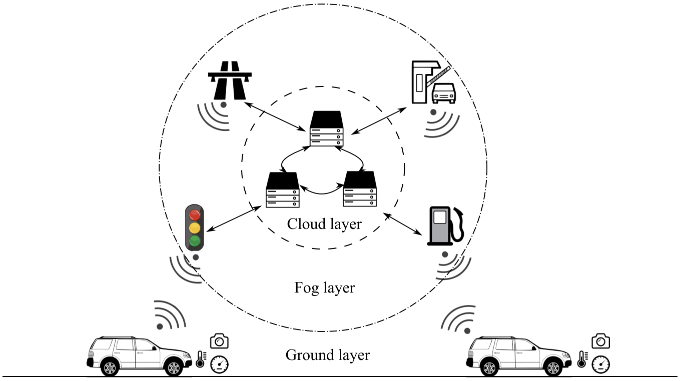


Fig. 3. The proposed three-layers infrastructure.

privacy preserving way. To this aim, we enhance each fog node with the ability of performing a simple algorithm for *secure two party computation* (2PC). The algorithm allows drivers to discover whether they fulfill the conditions to obtain a certain service provided by the fog layer of the infrastructure, without disclosing their profiles. The 2PC technique was first presented in [1] with the goal of solving the Millionaires' problem: two parties, Alice and Bob, each holding some private data x and y, want to discover whom of them is richer, i.e., whether $x > y$ or $x < y$, without disclosing out to the other party the amount of money and without using a Trusted Third Tarty (TTP).

In literature, there are several 2PC frameworks. Some examples are listed below:

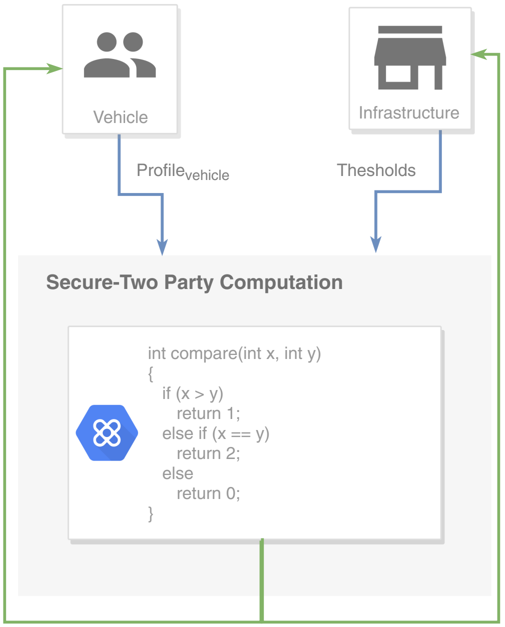
- FairPlay [18] can be considered the first influential 2PC framework. It allows users to write functions using a high level language, called SFDL, and to compile SFDL functions into garbled boolean circuit. A limit of Fairplay is given by the limited number of commands and operations that is possible to express through SFDL. FairPlay has strong security properties in the context of two-party computation. The framework is shown to be secure against a malicious party; in particular *i*) a malicious party cannot learn more information about the other party's input than it can learn from a TTP that computes the function; and *ii*) a malicious party cannot change the output of the computed function.
- A few years later, the same researchers have released FairplayMP [3], which is the extension of Fairplay that works with more than two parties.
- MobileFairplay [8] ports Fairplay to Android Smart-phones. In particular, MobileFairplay takes as input functions written and complied using the SFDL language and extends the application domains also to Android devices.
- MightBeEvil [14] and CBMC-GC [13], similarly to Fairplay, take as input functions written in high-level language, that can be run in a private way. In case of CBMC-GC, functions are written using the C language, then transformed into boolean circuits by the CBMC-GC compiler, and executed as

illustrated in [16]. A version for Android Smart-phones of CMCG-GC was presented in [7], showing much better performances compared with Fairplay for Android Smart-phones.

- *CBMC-GC* v2.0 [5,6] is a new optimized compiler to generate circuits for 2PC and Multi-Party Computation MPC) starting from ANSI-C source code.

5 Privacy-Preserving Reputation-Aware Vehicular Services

Vehicles in the considered infrastructure can ask for services, getting different quality and or prices depending on their driver's reputation profile. Typically, we can assume that to obtain, say, a special discount on a service, a driver must provide her profile to be compared with an access threshold used by the service provider. This comparison function hits the driver's privacy since the service provider will be able to know the entire profile in case of full profile disclosure, or at least a single parameter in the reputation profile. To protect the privacy of the drivers, we implemented the comparison function in a privacy-preserving manner that make use of the 2PC technique CBMC-GC v1.0. The presented method allows drivers to discover whether they meet the conditions for obtaining a certain service level without disclosing their profile (Fig. 4).



```
int compare(int x, int y)
{
  if (x > y)
    return 1;
  else if (x == y)
    return 2;
  else
    return 0;
}
```

Fig. 4. 2PC flow for profile comparison

Examples of innovative “reputation-aware” services enabled by the proposed infrastructure are described below:

Reputation-Aware Fuel Cost. Currently, fuel cost is decided at the level of the single gas station, and it is applied independently of the driver’s attitude to save or waste fuel while driving. In an effort to incentivize fuel-efficient driving style, one might think of a scenario where fuel cost is personalized to reflect a driver’s fuel efficiency. When entering a gas station, the vehicle onboard software sends driver’s reputation information – in this specific case, both her reputation score and her fuel efficiency score – to the fog node installed at the gas station. After proper authentication, the driver will be offered a personalized fuel price: a relatively lower price for drivers with relatively higher reputation, and vice versa.

Reputation-Aware Tolling. Similarly to the case of fuel price, also access to road infrastructure is currently oblivious to driving style, and is typically done based on the type of vehicle. However, a driver with a relatively higher risk profile (e.g., more aggressive, or speeding more frequently) might pose a relatively higher prospect cost to the infrastructure manager than a relatively more cautious driver, due to the higher risk of incurring accidents, damage road components, etc. One can then envision a scenario in which the price to access road infrastructure (highways, bridges, etc.) is personalized based on a driver’s reputation profile. Similarly to the gas station scenario, the vehicle onboard software shares driver’s reputation information with the fog node interfacing with the tolling system, and a driver is charged a variable amount that reflects her accident and damage risk profile.

6 Prototype of Privacy-Preserving Functions

To evaluate the feasibility of privacy-preserving, reputation-aware services as described in Sect. 5, we built a test-bed with a client-server paradigm where an Android Radio unit (Fig. 5) acts as client and represents the onboard computational unit of a vehicle, and a server that mimics as node of the infrastructure, i.e., a fog node. To achieve privacy-preserving comparison, we leveraged CBMC-GC on the fog node, while for the vehicles we use the Android porting of CBMC-GC. In our test-bed, the fog node runs on a Ubuntu 16.04.5 virtual machine with a dual core and 2 Gbyte of RAM, and the client on a Radio with Android 6.0, Quad-core at 1.2 GHZ and 1 Gbyte of RAM.

Security Consideration of CBMC-GC. The authors claim that their framework provides security in the *honest-but-curious* attacker model in which an attacker follows all the steps of the protocol as per specifications. However, attacker’s goal is to get information on the other party during the message exchanging phase, with the purpose of acquiring at least part of the private profile.

Another situation to point out is that there is an asymmetry on the provided security guarantees as customary in 2PC. This makes very difficult to prevent one

Fig. 5. Our android radio used in the test-bed.

party from terminating the protocol prematurely, and not sending the outcome of the computation to the other party. This situation can be discovered by the weak party, but cannot be recovered from.

6.1 Evaluation

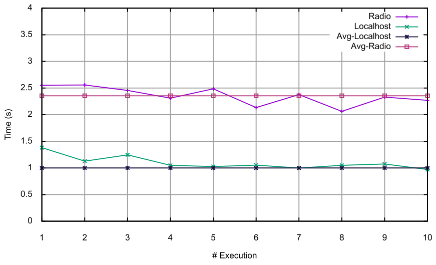
As first step, we compare the driver reputation with the reputation threshold fixed on the fog node to discriminate between "good" and "bad" drivers. If the value of the vehicle reputation is larger than the threshold, the vehicle gets 1, otherwise 0. In case of equal values, the output is 2. The source code for this function is reported in Listing 1.1.

Listing 1.1. C function to compare driver's reputation

```c
int compare(int x, int y) {
  if (x > y)
    return 0;
  else if (x == y)
    return 2;
  else
    return 1;
}

void comparerep(int INPUT_A_thr, int INPUT_B_rep) {
  int OUTPUT_rep = compare(INPUT_A_thr, INPUT_B_rep);
}
```

Figure 6a shows the time needed to execute the function listed in the code 1.1 using CBMC-GC and grouped in the table represented in Fig. 6b. In particular, we compare the running time when executing the vehicle part on the Android Radio, and when executing it on the same place of the fog node. This comparison is labelled as *Radio* and *Localhost* on the figure and table. The reported times are obtained by running the code in Listing 1.1 ten times. The two lines in the figure represent the average calculated for each of the ten executions. So, the



(a) Times to compute function in code 1.1

Localhost (sec)	Radio (sec)
1,381	2,553
1,127	2,558
1,246	2,456
1,049	2,311
1,026	2,486
1,052	2,134
0,999	2,377
1,049	2,063
1,074	2,330
0,971	2,270

(b) Performances of execution of code 1.1

Fig. 6. Global reputation comparison function

average time to execute the function in the privacy-preserving manner using the radio is of $2,354$ s. Instead, when the STC protocol is run in localhost the average time is of ∼1s.

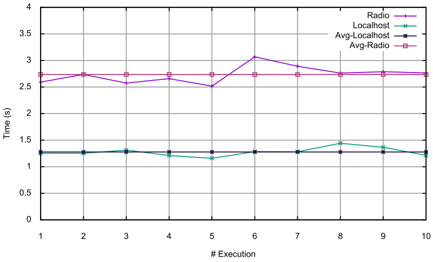
If the driver is considered “good”, then, a finer comparison on parameter is made. In fact, the code in Listing 1.2 illustrates the *C* function written to make the comparison for each of the considered parameters:

Listing 1.2. C function to compare driver’s parameters

```
int compare(int x, int y) {
   if (x > y)
      return 1;
   else if (x == y)
      return 2;
   else
      return 0;
}

void profile(int INPUT_A_brake, int INPUT_B_brake, int INPUT_A_turn, int
     INPUT_B_turn, int INPUT_A_speed, int INPUT_B_speed, int INPUT_A_rpm,
     int INPUT_B_rpm) {
   int OUTPUT_brake = compare(INPUT_A_brake, INPUT_B_brake);
   int OUTPUT_turn = compare(INPUT_A_turn, INPUT_B_turn);
   int OUTPUT_speed = compare(INPUT_A_speed, INPUT_B_speed);
   int OUTPUT_rpm = compare(INPUT_A_rpm, INPUT_B_rpm);
}
```

The main function *profile* takes as input the driver’s profile and four different thresholds of each service provided by the infrastructure. The driver’s input has as prefix *INPUT_A_*, instead the thresholds have as prefix *INPUT_B_*. These numbers are simple integer and in the profile are numbers that range from 0 to 5. Same interval is given to the threshold number.



(a) Times to compute function in code 1.2

Server (sec)	Radio (sec)
1.254	2.593
1,253	2,737
1,313	2,575
1,211	2,658
1,158	2,518
1,285	3,068
1,281	2,891
1,442	2,762
1,366	2,789
1,213	2,766

(b) Performances of execution of code 1.2

Fig. 7. Paramater comparison function

Then, the code contains a single function that simply compares each reputation parameter with the corresponding threshold. In particular, the *compare* function provides as output three different states, which are:

- *1*: if the value of the driver is higher than the threshold;
- *0*: if the value of the driver is lower than the threshold;
- *2*: if the values are the same.

Hence, each reputation-based service will apply the discount on the basis of the comparison results. For instance, the **Reputation-aware fuel cost** service will apply the discount when the result of the comparison function is 1 for the RPM parameter, while the service **Reputation-aware tolling** will apply the discount when the comparison function output is 1 for the speed parameter.

So, each time that the *compare* function is called, the output of the comparison is given to the $OUTPUT\_*$ variable that can be read at the end of the STC execution.

Figure 7a shows the time needed to execute the function listed in the code 1.2 using the CBMC-GC framework. Times are reported in the table in Fig. 7b. Also in this case, we compare the running time when executing the vehicle part on the Android Radio, and when executing it on the same place of the server. The average time to execute the function in the privacy-preserving manner using the radio is of $2,736$ s. Instead, when the STC protocol is run in localhost the average time is of $1,278$ s.

Summarizing, the results of our prototype evaluation clearly shows the feasibility of the proposed privacy-preserving framework, as the running times of the related secure functions are below 3 s in all the considered scenarios and hardware configurations.

7 Discussion

In the current paper, we based on the driver DNA on the radar graph introduced in [12] and we considered it as initial starting and studying step for our reputation calculations. However, the type of parameters as well as the number of those considered in the reputation formula may be extended or replaced with others. In addition, parameters selection depends on the support of the vehicles architecture. In fact, values are gathered from the internal network of a vehicle, for instance the CAN bus, and, to properly read this information, a hardware support may be needed to get accurate values for each parameter involved in the reputation formula. In our experiments, the adoption of our Android Infotainment system allowed us to get the needed parameters that were directly decoded from the information derived from the CAN bus of the vehicle.

Another aspect that can be considered but it is not part of the present work, is the manipulation of the information coming from the vehicle internal network to calculate the reputation formula. The presence of a physical attacker on the vehicle is not taken under consideration and this will not alter the calculus of the driver's reputation. To prevent this issue, different actions should be taken under consideration. However, it is not in the scope of this paper. For instance, the adoption of a secure internal vehicular protocol may reduce or avoid the presence of attacks on the physical bus to alter the transmitted content. Works on this topic are [2,17,21].

8 Conclusion and Future Work

In this paper, the notion of driver's reputation profile is introduced as a unique, multi-dimensional information associated to a driver's behavior. As an example, we have described how reputation profiles can be built starting from the *driver DNA* and a synthetic *reputation score*, respectively. Moreover, we propose a private vehicular infrastructure based on both fog and cloud networks, which is able to both collect the information needed to compute driver reputation profiles, and to provide reputation-aware services to the driver themselves. The proposed infrastructure and related reputation-aware automotive services have the potential to stimulate drivers to behave correctly to a much larger extent than what achieved by current practice based on risk profiling and personalized insurance rates. Through prototype implementation, we have tested and positively assessed the feasibility of a privacy-preserving implementation of the framework.

This work is intended to open more avenues for future research, rather than to present a fully developed system. In particular, we plan to assess the proposed framework on real test cases considering design and development of vehicles' evolution. This may require a more complex calculation of the reputation formulas that, for instance, consider benefits from the adoption of an hybrid engine that gets the support of the electric power. It could also require to move form a secure two party computation approach to a multi-party one in which reputation values are exchanged among more than two entities.

Another interesting points that we will investigate as future work is the multi-user driving scenario. Nowadays, modern vehicles support different driving styles that come out from different drivers. So, based on the current management of multiple-users, like traditional computers that supports different logins, the reputation of a user may be more accurate considering this additional feature.

Acknowledgement. This work has been partially supported by the GAUSS national research project (MIUR, PRIN 2015, n 2015KWREMX) and by H2020 EU-funded projects C3ISP (GA n700294).

References

1. Andrew, C., Yao, C.: Protocols for secure computations. In 23rd IEEE Symposium on FOCS, pp. 160–164 (1982)
2. Bella, G., Biondi, P., Costantino, G., Matteucci, I.: TOUCAN: a protocol to secure controller area network. In: Proceedings of the ACM Workshop on Automotive Cybersecurity, AutoSec@CODASPY 2019, Richardson, TX, USA, 27 March 2019, pp. 3–8 (2019)
3. Ben-David, A., Nisan, N., Pinkas, B.: FairplayMP: a system for secure multi-party computation. In Proceedings of the CCS Conference, pp. 257–266. ACM, New York (2008)
4. Bonsall, P., Liu, R., Young, W.: Modelling safety-related driving behaviour-impact of parameter values. Transp. Res. Part A Policy Pract. **39**(5), 425–444 (2005)
5. Buescher, N.: CBMC-GC-2 (2018). https://gitlab.com/securityengineering/CBMC-GC-2
6. Buescher, N., Holzer, A., Weber, A., Katzenbeisser, S.: Compiling low depth circuits for practical secure computation. In: Askoxylakis, I., Ioannidis, S., Katsikas, S., Meadows, C. (eds.) ESORICS 2016. LNCS, vol. 9879, pp. 80–98. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45741-3_5
7. Costantino, G., Maiti, R.R., Martinelli, F., Santi, P.: Private mobility-cast for opportunistic networks. Comput. Netw. **120**, 28–42 (2017)
8. Costantino, G., Martinelli, F., Santi, P., Amoruso, D.: An implementation of secure two-party computation for smartphones with application to privacy-preserving interest-cast. In: Proceedings of the 18th International Conference Mobicom, pp. 447–450. ACM (2012)
9. Eboli, L., Mazzulla, G., Pungillo, G.: Combining speed and acceleration to define car users' safe or unsafe driving behaviour. Transp. Res. Part C Emerg. Technol. **68**, 113–125 (2016)
10. Filev, D., Lu, J., Tseng, F., Prakah-Asante, K.: Real-time driver characterization during car following using stochastic evolving models. In: 2011 IEEE International Conference on Systems Man and Cybernetics (SMC), pp. 1031–103 (2011)
11. Fracchia, R., Meo, M.: Analysis and design of warning delivery service in intervehicular networks. IEEE Trans. Mobile Comput. **7**(7), 832–845 (2008)
12. Fugiglando, U., Santi, P., Milardo, S., Abida, K., Ratti, C.: Characterizing the "driver DNA" through can bus data analysis. In: Proceedings of the 2nd ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services, CarSys 2017, pp. 37–41. ACM, New York (2017)
13. Holzer, A., Franz, M., Katzenbeisser, S., Veith, H.: Secure two-party computations in ANSI C. In: Proceedings of the CCS Conference, CCS 2012, NY, USA, pp. 772–783 (2012)
14. Huang, Y., Chapman, P., Evans, D.: Privacy-preserving applications on smartphones. In: Proceedings of the 6th USENIX conference on Hot topics in security, HotSec 2011, Berkeley, CA, USA, p. 4 (2011). USENIX Association

15. European Telecommunications Standards Institute: ETSI TS 102 940: Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management, September 2010. https://www.etsi.org/deliver/etsi_ts/102900_102999/102940/01.01.01_60/ts_102940v010101p.pdf
16. Kolesnikov, V., Schneider, T.: Improved garbled circuit: free XOR gates and applications. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008. LNCS, vol. 5126, pp. 486–498. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-70583-3_40
17. Kurachi, R., Matsubara, Y., Takada, H., Adachi, N., Miyashita, Y., Horihata, S.: CaCAN-centralized authentication system in CAN (controller area network). In: 14th International Conference on Embedded Security in Cars, ESCAR 2014 (2014)
18. Malkhi, D., Nisan, N., Pinkas, B., Sella, Y.: Fairplay—a secure two-party computation system. In: Proceedings of the 13th conference on USENIX Security Symposium, SSYM 2004, Berkeley, CA, USA, vol. 13, p. 20 (2004). USENIX Association
19. Massaro, E., et al.: The car as an ambient sensing platform. Proc. IEEE **105**(1), 1–5 (2017)
20. The European Parliament and of the Council. Directive 2010/40/eu. eur-lex.europa.eu. 7 July 2010
21. You, I., Jung, E.-S.: A light weight authentication protocol for digital home networks. In: Gavrilova, M.L., et al. (eds.) ICCSA 2006. LNCS, vol. 3983, pp. 416–423. Springer, Heidelberg (2006). https://doi.org/10.1007/11751632_45
22. Rønning, A.: Rewarding safe drivers could make roads safer. ScienceNordic. 12 September 2013. http://sciencenordic.com/rewarding-safe-drivers-could-make-roads-safer. Accessed 22 March 2021
23. Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N.: The case for VM-based cloudlets in mobile computing. IEEE Pervasive Comput. **8**(4), 14–23 (2009)
24. Stojmenovic, I., Wen, S.: The fog computing paradigm: scenarios and security issues. In: Federated Conference on Computer Science and Information Systems, 7–10 September 2014, pp. 1–8 (2014)
25. Vaquero, L.M., Rodero-Merino, L.: Finding your way in the fog: towards a comprehensive definition of fog computing. SIGCOMM Comput. Commun. Rev. **44**(5), 27–32 (2014)
26. Zeeman, A.S., Booysen, M.J.: Combining speed and acceleration to detect reckless driving in the informal public transport industry. In 16th International IEEE Conference on Intelligent Transportation Systems, ITSC 2013, The Hague, The Netherlands, 6–9 October 2013, pp. 756–761 (2013)