Computer security used to protect drivers

Joao Quinta – Joao.Costa@etu.unige.ch

800 words – DSR

A new way of establishing communication between vehicles could help prevent crashes as well as improve traffic in dense areas.

Humans have notoriously slow reaction times, especially if a person is fatigued or isn't completely focused on a task. Imagine you are leaving work, you are tired, and just want to get home, while driving during rush hours in a city full of traffic. It is quite easy to lose focus and cause a small accident. Best case scenario no one get's hurt, but that's not always the case! This could be avoided if your car, or rather a computer inside your car, could brake when required. The same idea could be applied to create a synchronous car start when the traffic light turns green, rather than the current system where each car starts driving after the one in front has already started. If we were able to implement these technologies, more drivers would be able to get home safely as well as faster.

Now obviously, we have to be smart about this, think of this simple scenario, two cars are following each other, if the car in front communicates a sudden change in speed, we want the car following to act accordingly. However, we must be certain that both entities are real cars, that are broadcasting real data. This is important to prevent an hacker – an ill-intentioned individual – to send fake messages to a car. If this were to happen, the hacker, might be able to take control of certain actions in the car like breaking. To avoid this problem, we must make these communications between cars secure! This can be done by applying encryption to the messages being sent between them. Encryption will ensure the receiver that the messages are being sent from a trustworthy source.

There are other solutions available in newer cars that try to solve the same problem, however they don't use communication between cars. They simply add sensors to the car, if the car detects that the car in front is braking, it will decide to brake to prevent a crash. The problem is that sensors may malfunction and unfortunately there is no way to know what is happening if it is not working correctly. Our system solves this problem by making sure that if a message isn't correctly received, it will be sent again. This way you are sure that critical information will be read. Furthermore, current systems only attempt to prevent crashes, they can't help you reduce traffic in the same way a secure communication system would.

To achieve this communication between unauthenticated entities – that can't yet trust each other – we thought of using a trusted third party. This third party would authenticate every entity and assure other entities that they are talking with authenticated users. For example, a car would constantly broadcast its current speed and acceleration. This information would need to be encrypted with a signature. The signature would be authenticated by the receiver with the help of the already, set up trusted third party. This is a good proof of concept. The final solution might achieve authentication without the use of a trusted third party.

Right now, we are working on the authentication side of the solution, making sure that each entity is capable of self-authenticating itself. This is achieved by using a zero-trust security model[1], such models assume that any user is untrustworthy until they prove otherwise. This is even true for a user that has previously been authenticated, if it is a new connection, we want to make sure it's a real entity.

Zero trust security models use authentication factors[2] to identify users, these factors can be knowledge factors, for instance a password. They can be ownership factors, for example a cell phone, a software token, or even a built-in hardware token. Finally, we can use inheritance factors, a fingerprint, or maybe a retinal pattern.

In our case we can either rely on a built-in hardware token, meaning an unique code that a car would own, and only the car would know. There are other options, for example using the phone of the driver to identify a car.

Right now, we want to make sure each user likes this solution. So, we must be certain that no one can attack it, and take control of a car's functionalities, as this could easily be life threatening to the driver or occupants of the vehicle. Afterwards the goal will be to start implementing the broadcast of information. Finally, we will create an algorithm that takes as input broadcasted data, and as output computes which actions must be taken.

[1] 'What Is Zero Trust Security? Principles of the Zero Trust Model', *Crowdstrike.Com* <https://www.crowdstrike.com/cybersecurity-101/zero-trust-security/> [accessed 21 December 2022].
[2] 'Authentication', *Wikipedia*, 2022 <https://en.wikipedia.org/w/index.php?title=Authentication&oldid=1128426960> [accessed 21 December 2022].