Security expectations vs. limitations

In part 1 of this article series, we examine the security limits of various embedded applications. What kind of security levels can realistically be achieved by developers, integrators, and users of embedded systems.

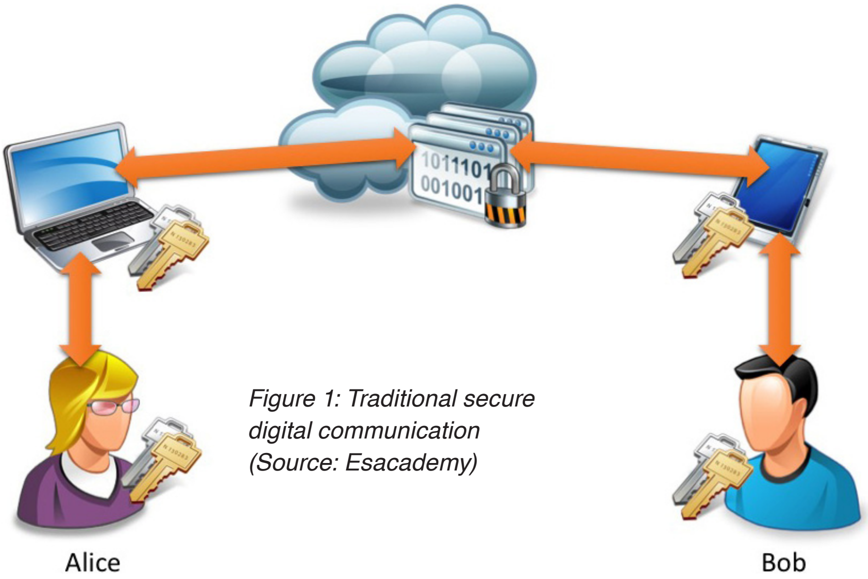

Figure 1: Traditional secure digital communication (Source: Esacademy)

Sometimes the perception about embedded security still seems to be that it is either "there" or "not there" but in fact, security is not a binary "on" or "off". There are various levels and if your customer expects 100 % security, then you first need to help them to review their expectations.

When it comes to the security of embedded systems we still see a lot of unrealistic expectations. With this article we would like to give developers, integrators, and users of embedded systems an overview about what kind of security levels can realistically be achieved. Let's look at traditional secure communication models as illustrated by the individuals Alice and Bob exchanging messages in Figure 1. The security goal here is to provide a private messaging system, which is both encrypted and authenticated in a way that no third party can read or manipulate the communication between them.

The attack vectors potentially available to a third party like Chuck (assuming he can make use of them) are illustrated in Figure 2 and include:

- intercepting the messages and trying to decrypt them
- accessing Alice's or Bob's computer or message device to extract keys or messages
- directly tricking Alice or Bob into revealing keys or messages

It shouldn't be a surprise that it is more difficult to maintain a fully private channel if the third party has easy access to multiple attack vectors. If Alice and Bob are embedded devices within the same machine and Chuck has unlimited physical access to it, then it means he has direct access to the entire communication channel including both the transmitting and receiving computing devices.

Before selecting any application-specific security method, one needs to review the possible attack vectors and draw a line between those attacks that we can protect a system from and those that are beyond our control, like attacks involving extortion, personal threats and such. For many embedded systems the "unlimited physical access" is an important criteria. If we can say that the attacker can never have physical access because the machine is locked inside some building, then the required security levels can focus on protecting remote access. By securing the internal communication between the devices in the machine against manipulation, an attacker who gains remote access to it, for example through some gateway, won't be successful in gaining further control. In this article series, we examine the security limits of various embedded applications. For part 1 we start by examining the fare calculation of a taxi.

Limits of embedded security

Can you ever be sure to pay the correct taxi fare?

Over the last years, we have been involved in various security projects and learned that for specific security expectations there doesn't seem to be a realistic solution available. In this article we take the application of taxi fare calculation to show that sometimes even sophisticated security methods can only provide a somewhat moderate security level overall.

Today, a taxi fare calculation is based on a wheel pulse counter. One could now engage in the discussion whether today's average phone with all its sensors wouldn't be better suited to do a more reliable calculation of the fare. Currently, the only method that has gained official approval from the governing bodies is calculations based on direct input from the wheels. Other methods such as ones based on GPS combined with acceleration measurements have not yet passed the approval process.

The security challenge in this application is: how can we as customers ensure that a rogue taxi driver or company does not manipulate the way fares are calculated or presented to the passenger? Can we ensure overcharging becomes impossible?

▷

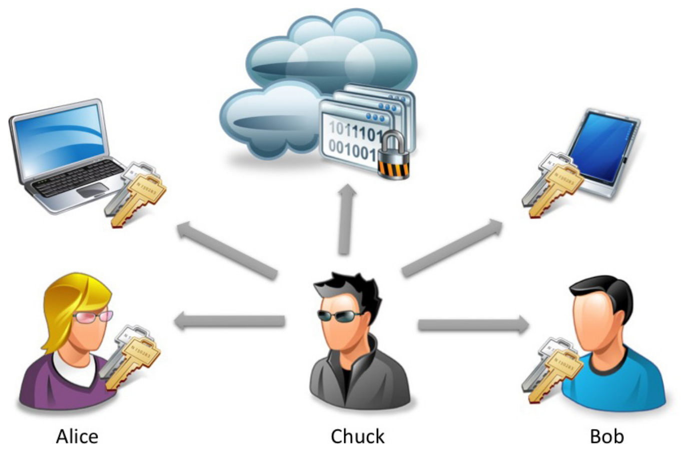

Figure 2: Possible attack vectors (Source: Esacademy)

On a technical level, the wheel pulse counter sensor detects magnets rotating/passing by its sensor. With every pass, the counter is incremented. How many pulses are counted per rotation can vary in different cars, this is one of the many parameters a meter needs to know when making its calculations. The wheel pulse counter value is transmitted via a Controller Area Network and, after passing through one or multiple bridges or gateways, finally reaches the meter where the fare is calculated.

One potential method for manipulating the system is illustrated in Figure 3. The rogue driver/owner connects a CAN device that implements a man-in-the-middle manipulation. The CAN cable is cut somewhere between the sensor and the meter. The manipulating device is then inserted in between the network branches and acts as a bridge. It passes on all CAN messages – but when passing on the wheel pulse counter value it inserts its own value that is always a certain percentage higher. This would result in a higher fare being calculated.

To prohibit the use of such a manipulating device, some say that end-to-end security from wheel pulse counter to the meter is needed. The idea is that the meter only accepts authenticated wheel pulse data. Then a man-in-the-middle device would only work if it knows the security methods and keys involved.

The security challenge here is, that potential manipulators of such a system (rogue taxi drivers or companies) have full, unlimited physical access to the entire system – from message producer (wheel pulse sensor) to the consumer (meter).

A security system that uses the same master key(s) shared among multiple vehicles would be unsuitable here. The rogue party could simply report a car as stolen and then send the sensor and the meter to a third party “data and code recovery or extraction service” (there are multiple companies offering such services starting at below one hundred Euro). The extraction would give them access to all code and data stored in these embedded devices. Even if keys are hidden and encrypted, with access to all code, hackers will be able to retrieve the keys eventually. If these keys are used in multiple cars, it is then easy to still build a man-in-the-middle attack device.

There are dedicated security hardware chips/micro-controllers that make such an extraction more difficult, but choice and price of these have not yet reached a level where they can easily replace common micro-controllers.

▷

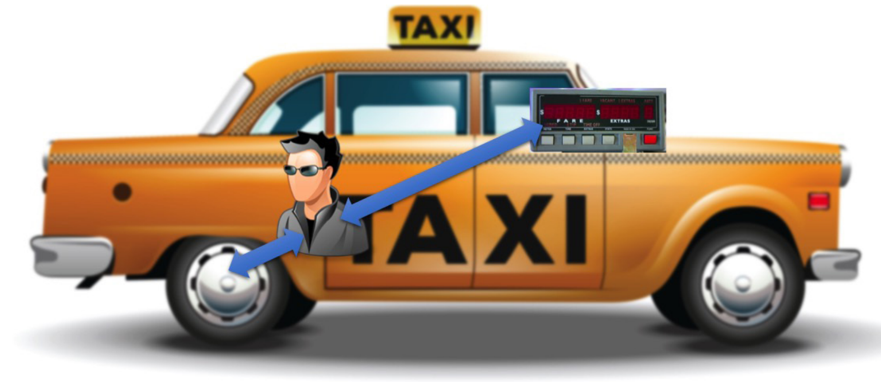

Figure 3: Man-in-the-middle attack between wheel and meter (Source: Esacademy)

A higher security level would require that keys are random and unique in every taxi. In Figure 4 this is illustrated with Alice, the manufacturer of the wheel pulse counter and Bob, the taxi meter manufacturer. Each wheel pulse counter and meter require their own individual pairing key (silver) – and a potential master key (golden) to reset/erase/revoke these keys.

The issue with such master keys is: they provide a back door that could also be used by hackers. Yes, also the back door would have security mechanisms. However, the motivations for hackers to break it grows with the number of installed devices.

As with many security systems, the key generation/distribution logistics remains one of the biggest challenges: whose responsibility would it be to generate and install keys or make the initial pairing? And when and where would this happen?

Even worse, looking at the entire system, manipulating the communication between the sensor and the meter is only one of several options available to manipulate the system. Let's do a review from wheel to meter:

- Wheel: A 3 % variation in the tire diameter results (multiply by Pi) in a 10 % variation of the measured distance. As far as we know, a 5 % variation in measured distance is therefore generally accepted as "within parameters" to allow for variations in tires.
- Communication: Today, manipulation of the communication between wheel pulse counter and meter could be achieved by inserting a man-in-the-middle CAN device. As the potentially rogue parties have full physical access to both producer and consumer, this could only be prohibited by using a security method based on individual, non-revocable keys.
- Meter: In the end, it is the meter that displays the fare. How difficult would it be to manipulate the meter itself to show a different fare? Well, as previously mentioned, code can potentially be extracted and in a next step could be manipulated to display a different fare, not impossible for a motivated hacker. To prohibit such manipulations, meters are sealed.

But now think about the manipulations already performed today to banking machines. Additional keyboards and card readers can be tacked-on to banking machines in a way that users don't recognize the difference. In the same way a meter-like display could be designed to clip onto or fully around an existing meter. The original meter "vanishes" inside a fake meter that can display whatever the taxi driver would like it to display.

Reviewing this list, prohibiting any type of manipulation becomes very challenging. Adding authentication and possibly encryption to the communication between sensor and meter only addresses one of many attack vectors. When the possible attacker has full physical access to the car, the keys used to protect the communication and possibly a back door for a system reset must all be individual and may not be shared, requiring a complex key infrastructure.

In the long-term, the entire process on how taxi fares are calculated needs a review. Busses, trains, ferries, and planes primarily charge by destination. Can you imagine an airline or train transport system trying to surcharge their passengers on detours or delays? Somehow that is common practice with taxi cabs – with a taxi you may pay more, when you get into a traffic jam or detours are taken.

I fully understand that historically these rules have been established to protect the drivers and cab owners: make a passenger pay if they want to make stops or want specific routes to be taken. However, other industries, too, have faced re-structuring challenges or even (industrial) revolutions in the past and survived.

Even without being a fortune teller it looks obvious to us that self-driving vehicles will arrive sooner or later. If these are used for passenger transportation, possibly even with some hop-on/hop-off support, then the current (taxi) fare calculation can hardly be maintained.

An intermediate solution might be to remove the manipulation incentive: If the base fare is calculated on the theoretical distance (based on a current map, start and end coordinates), then this is a fixed, transparent value that all parties: owner, driver, and the passenger can verify. If the map used includes current road closures, then most detour reasons are already taken into account. If ▷

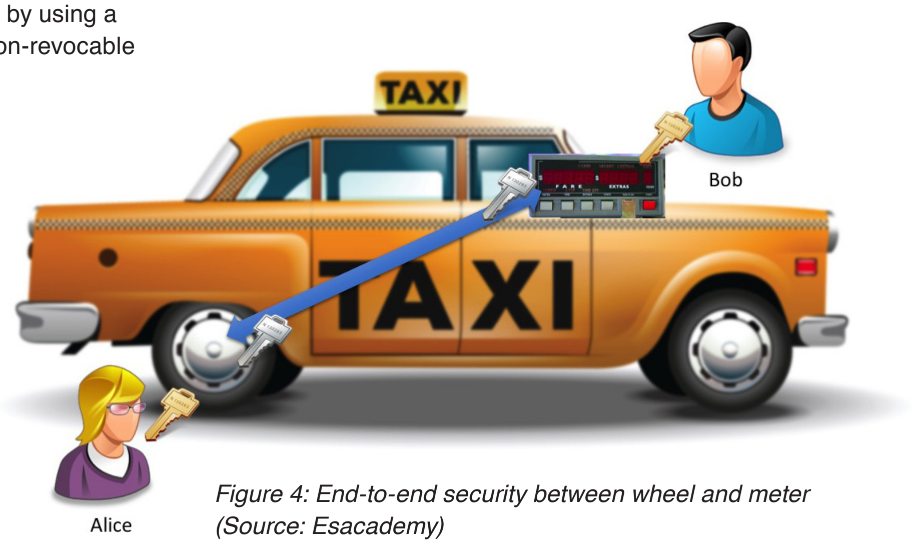

Figure 4: End-to-end security between wheel and meter (Source: Esacademy)

needed, there could still be surcharges added for passenger-caused waiting time.

In summary, reviewing this application has taken an interesting twist: we started off with the request of adding security features to the CAN communication used for the fare calculation of a taxi meter. And in the end, the recommendation for this specific application is: do not use that very CAN communication at all to calculate the fare. Sometimes being a CAN security consultant requires to also recommend solutions outside of CAN. ◀

Preview to the next article of the series

In the next issue, the authors will have a look at security requirements for applications that are being discussed, driven by now-possible scenarios that could be taken out of a Hollywood blockbuster: A swarm of flying drones intersecting planes and trucks or ships with hazardous goods on collision course. The self-driving car of a prosecutor is taken over by a hacker on the payroll of organized crime and driven straight off the cliff. One can expect that lawmakers won't stand by while such scenarios become a reality.

Regulations that mandate security in place "at all levels" of such machines and devices are likely. Will they be technically detailed and allow exceptions for less-vulnerable levels of communication within the machine? Probably not. The long-held argument that a CAN bus is often within a closed system not accessible from the outside andtherefore does not need security might become moot once tough regulations become law.

Authors

Olaf Pfeiffer
Christian Keydel
EmSA (Embedded Systems Academy)
opfeiffer@esacademy.de
ckeydel@esacademy.de
www.esacademy.de