© Softing Automotive

Security Challenges in Diagnostics 4.0

AUTHOR

Markus Steffelbauer
is Head of Product Management and Marketing at Softing Automotive Electronics GmbH in Haar near Munich (Germany).

Remote and cloud diagnostic methods will play an ever greater role in future vehicle repairs. Security, which needs to cover all areas of vehicle electronics in terms of access and communication with external systems, is of central importance when configuring corresponding concepts. Softing shows the challenges involved in practical implementation and which systems are already available today.

DIAGNOSTICS – BOTH SIMPLE AND DIFFICULT

The main task of diagnostics is actually quite simple: getting a vehicle that does not work to run again. What usually happens is that the error first has to be localized, then the repair has to be carried out, and finally the success of the repair has to be verified. The error will basically be one of two types: a hardware error in which for example parts have to be exchanged or connections have to be repaired, or a software error, in which the Electronic Control Unit (ECU) software has to be updated.

In detail, the challenges are quite different. The owner of a car will usually make an appointment with their repair shop if a problem occurs, where it will be analyzed and put right. In the case of trucks, it is a different story: The load has to be unloaded at a specific time and, on the whole, the truck has to be on the road as long as possible. The repair shop times thus depend on the available idle time because that is

when the truck can be taken to the repair shop, often one specializing in a specific area, for example the gear manufacturer's repair shop. In the case of working machinery, the situation is similar, although the machinery is often too difficult to move and the repair shop has to come to the machine.

REMOTE, CLOUD AND SECURITY

These challenges, in combination with increasingly complex E/E architectures, mean that the repair processes, as well as the creation of the associated diagnostic and test tools, are becoming increasingly difficult to master. One solution is remote diagnostics:

- New ECU software is installed on the vehicle over an Internet connection and programmed at a suitable time (software update "over the air").
- The vehicle status is checked regularly using a cloud-based diagnostic system and, in the case of foreseeable problems, an appointment in the repair shop is suggested (predictive maintenance).
- When a machine is checked by a technical center, the repair employee can do a call out with the appropriate part and correct a fault at short notice (remote diagnostics).

These are just some of the obvious possibilities of how diagnosis can become more effective and more efficient in remote access. Furthermore, there are numerous other use cases in the engineering and development field [1]. The entirety of current diagnostic methods is generally referred to under the collective term Diagnostics 4.0: remote and cloud diagnostics (remote) as well as diagnostics on the vehicle (proximity). In practice, there is a further use case. Since the requirements for latency, bandwidth and availability for remote diagnostics are not sufficiently secured due to today's network connections, part of a modern diagnostic system is moved to the vehicle anyway. For some use cases – particularly those in which user interaction is only necessary to a limited extent – the entire tester can be implemented there (so-called in-vehicle approach).

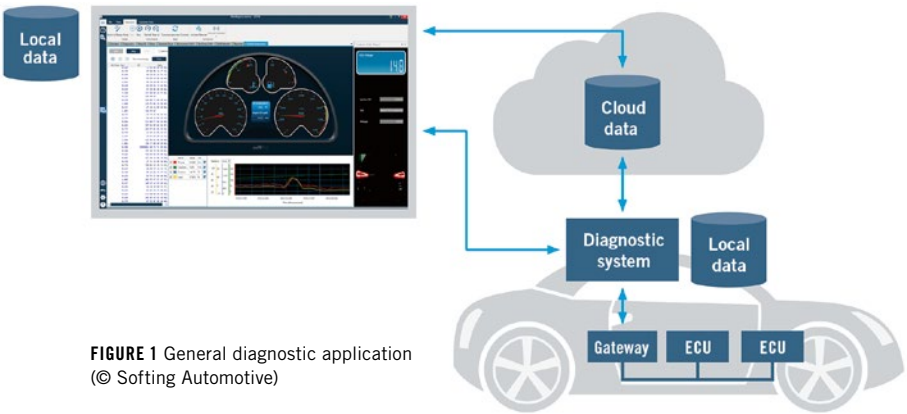

FIGURE 1 General diagnostic application (© Softing Automotive)

THE CHALLENGE OF SECURITY

Regardless of the method used, almost anything can be done on a vehicle using diagnostics. Parameters or the entire software can be modified, vehicle functions initiated, information read out from ECUs. This is already leading to considerable potential danger in today's diagnostic applications, but this potential is much higher in remote applications. This is why there has to be adequate protection in terms of confidentiality, integrity and authenticity:

- Third parties must not be able to read out any information from the vehicle. This particularly applies to personal data as it is subject to the General Data Protection Regulation with the familiar threats of punishment.
- Third parties must not be able to change either stored or communication data. This concerns vehicle configurations, for example, the changing of which can lead to increased warranty costs.
- There must be no misuse of the capabilities of the vehicle by third parties. In a worst-case scenario, any intervention in the vehicle while it is driving can lead to personal injury.

In a general diagnostic system, **FIGURE 1**, protection must be effective at numerous points. First of all, the applications (tester and diagnostic system) are protected against misuse. Users have to identify themselves for this purpose. Then, all local and cloud data is encrypted so that it can no longer be read from outside. The applications themselves only allow authorized users access after the first step. Finally, all communication connections have to be protected against tapping, which once again necessitates encryption. The goal of all measures must be end-to-end protection of the overall diagnosis function, with the user representing one end and the vehicle gateway usually the other. Within the vehicle, other protective mechanisms are necessary.

In encryption, a distinction is made between two basic procedures: symmetric and asymmetric encryption, **FIGURE 2**. In symmetric encryption, both parties involved know the secret key used to encrypt and decrypt. The procedure is already very secure with short keys 256 bit long and can also be calculated

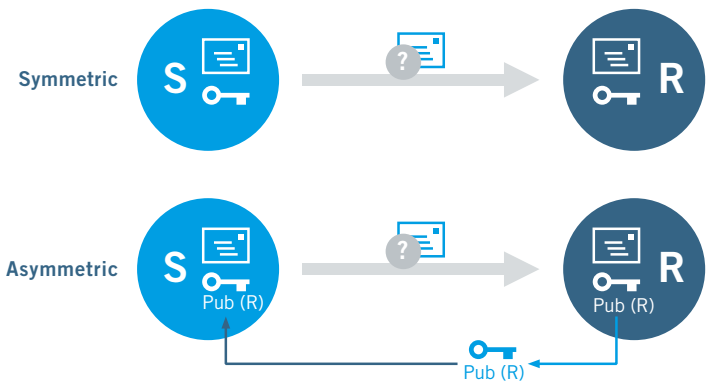

FIGURE 2 Symmetric and asymmetric encryption (© Softing Automotive)

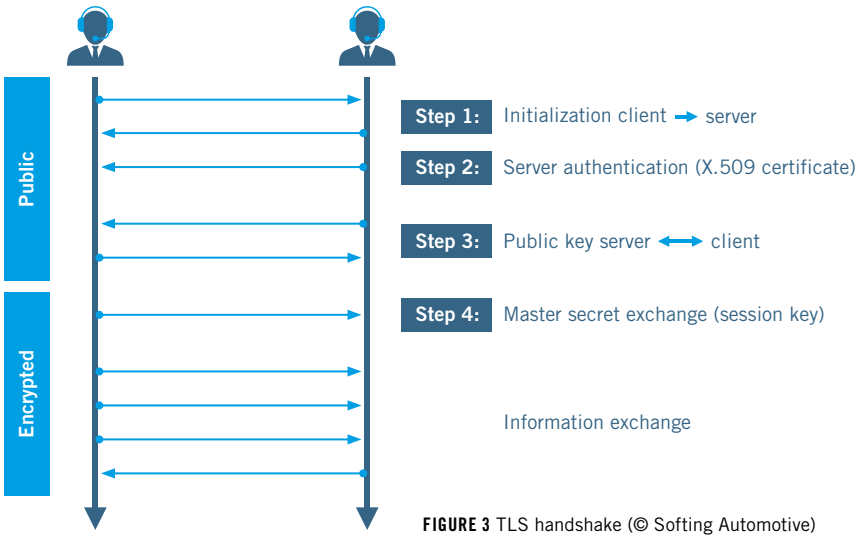

FIGURE 3 TLS handshake (© Softing Automotive)

efficiently. However, the encryption key must reach the parties involved securely and without a third party being able to corrupt it, for example in personal contact.

Asymmetric encryption works with one key pair at both parties. The public key is made available to the relevant partner who can use it for encryption. The private key is kept safely and is used to decrypt the message encrypted with the public key. In this way, the difficulty of transferring keys experienced in the symmetric procedure can be avoided, although longer key lengths are required here which can also lead to long computing times. Furthermore, the public keys must be obtained from a trustworthy source.

SUITABLE APPROACHES

For safe end-to-end safeguarding, the application has to be protected first. It must not be able to be compromised either by copying the entire application or by patching individual files. This usually takes place with licensing procedures and by "packing" (enveloping) the relevant application parts. Commercial tools are available for this purpose. Users also have to authenticate themselves. This is important because not everyone who happens to get hold of a tester should be allowed to access the vehicle – and not every authorized person should be able to program ECUs either. For this purpose, role models can be stored in the application and protected with a registration process – in the simplest case, a password.

The data is usually protected using symmetric encryption procedures – regardless of whether application-independent, locally on the vehicle or in the cloud. The corresponding key must be securely compiled into the program and kept in the memory during runtime to achieve very good backup levels. Examples of such encryption procedures are Blowfish or Advanced Encryption Standard (AES), which both represent block ciphers and therefore do not negatively influence the data size. Moreover, neither of them is patented and can be used in the public domain.

Communication connections are usually safeguarded using protocols which implement a hybrid of symmetric and asymmetric encryption. In this process, a code – or the components required for calculation – is exchanged in an initialization phase using asymmetric communication. The actual communication can then be carried out very efficiently with this information and symmetric encryption. The key is only valid for safe procedures during one communication session and is then discarded.

An example of this kind of procedure is Transport Layer Security (TLS), which is regularly used in Internet protocols, for example with https, in VPN implementations and in the Internet of Things (IoT) in the MQTT protocol, **FIGURE 3**. TLS implements a handshake at initialization which first generates the key using various procedures (RSA or Diffie-Hellman-Merkle key exchange). The authentication of the communication partner also takes place using a standardized procedure with trusted certificates (X.509 certificate). Subsequently, symmetric encryption is carried out using the procedure negotiated in the initialization. This is usually, as with data encryption, AES.

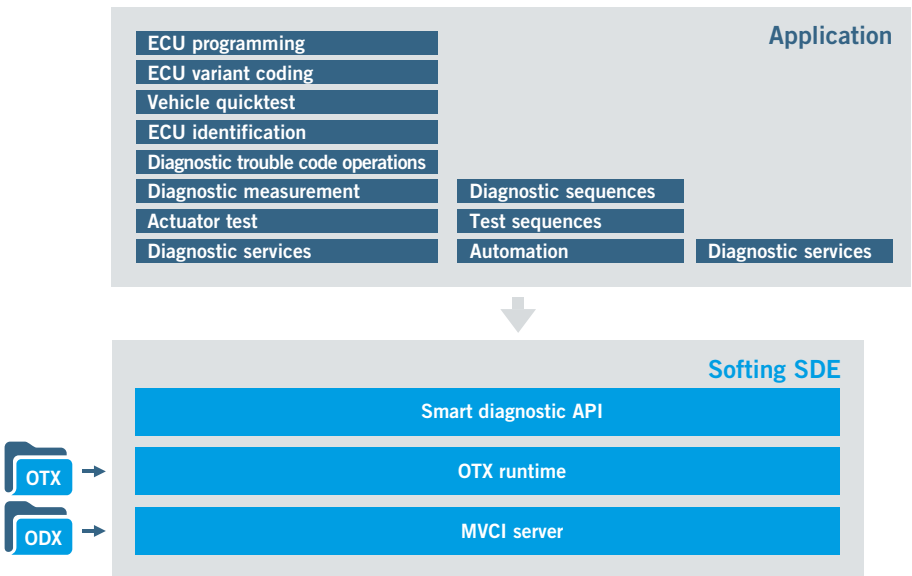

FIGURE 4 The diagnostic runtime system Softing SDE (© Softing Automotive)

IMPLEMENTATION IN PRACTICE

Anyone wanting to implement secure remote diagnostic systems has a second challenge to overcome, in addition to the matter of security: Today's diagnostic systems are usually diagnostic-service-based, in other words, they have to trigger a diagnostic service for each subtask.

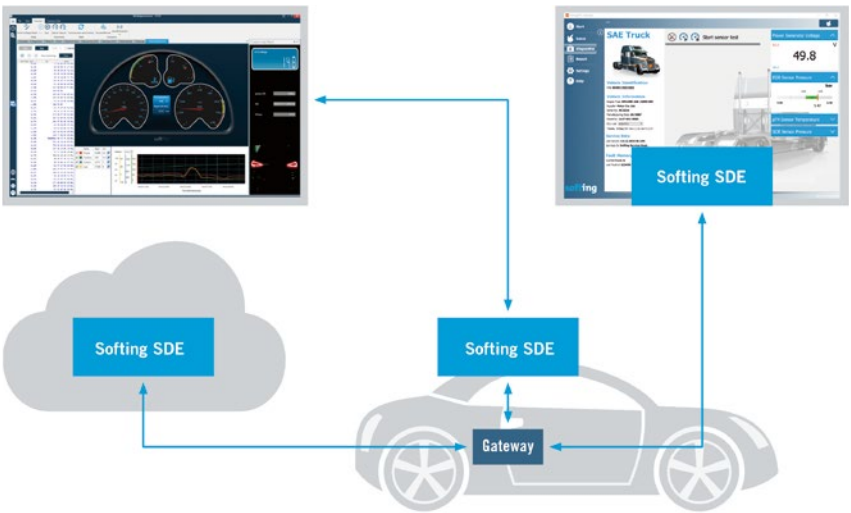

FIGURE 5 Use cases for Softing SDE (© Softing Automotive)

However, diagnostic tasks regularly require several substeps and comprise different ECUs. Over a remote connection, this becomes both slow and unstable which in turn necessitates a different system architecture. Practice shows that the combination of diagnostic services into self-contained diagnostic tasks can be abstracted very well. A good example of this is the task "ReadDtc": First of all, an error list has to be retrieved from the ECU and then an additional diagnostic service has to be carried out for additional information for each individual error. If this is run as an entire sequence in the vehicle, all the remote tester has to do is trigger the sequence and then retrieve the results once the sequence has been completed.

This principle has already been implemented in the diagnostic runtime system Softing SDE, **FIGURE 4**. The Application Programming Interface (API) offers functions for the most important diagnostic tasks, such as "ReadIdentification," "ReadDtc," "EcuVariantCoding" or "EcuProgramming." Communication is described via the Open Diagnostic Data eXchange (ODX) standard, sequences are either hard-coded or defined in the Open Test sequence eXchange (OTX) standard. The entire diagnostic system is platform-independent and can therefore be used under different operating systems. This means the same data record is always used and the runtime behavior is also identical. Regardless of whether in the engineering tester under Windows and without a remote connection, in manufacturing with a diagnostic runtime system integrated in a Vehicle Communication Interface (VCI) or integrated directly in the vehicle – Softing SDE enables a noticeable increase in both quality and efficiency, **FIGURE 5**.

CONCLUSION

Along with today's diagnosis directly on the vehicle, remote and cloud diagnostics – Diagnostics 4.0 – is moving into the repair landscape. However, the extended possibilities resulting from the use cases in the vehicle, on the vehicle and from a distance require considerable rethinking in terms of the architecture of the diagnostic system and the security requirements. Some of the diagnostic functions have to be moved into the vehicle to become independent of the network infrastructure and to be able to handle diagnostics autonomously in the vehicle independently of the tester. Security for a remote connection has to be planned holistically from user authentication through application and the connection routes to the vehicle gateway. This makes it possible to establish secure end-to-end diagnostics with the existing technologies. Softing SDE already shows how such diagnostics can work in the vehicle – in the VCI, but also in classical diagnostic systems.

REFERENCE

[1] Steffelbauer, M.: Biggest Possible Development Efficiency with Remote Engineering. In: ATZelectronics wordwide 10/2019, pp. 36-39