Protected in Every Situation

Flexible HSM Firmware for Security-Related AUTOSAR Systems

With the continual increase of ECU networking, the requirements for data security also increase. Special ECU hardware units, such as programmable hardware security modules (HSMs), are progressively being used to efficiently protect data. An HSM must be able to support many different applications. Just how flexible and configurable does the HSM firmware need to be to allow for this?

Security is not a new topic in automotive electronics. Today's ECUs already check the authenticity of the application during startup and whenever a software update is made. Additionally, security-related diagnostic services are only enabled after successful authorization. However, along with the degree of networking in vehicle electronics, the number of external interfaces is growing, and consequently, so does the potential attack surface. In addition, networking enables new use cases which are inherently security-related. One example of such use cases is automatic billing in the electric mobility field.

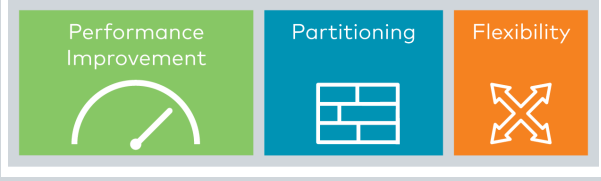

Figure 1: Typically, an HSM is used to pursue three goals.

Heightened sensitivity to security needs has led to a situation in which semiconductor manufacturers have further improved their hardware support for security. Hardware security modules (HSMs) are now available for many modern microcontrollers. The use of HSMs typically pursues three goals **(Figure 1)**:

- Increased performance: The use of special accelerators reduces the amount of time it takes to perform a cryptographic computation – such as encryption. This leads to shorter wait times and less load on the main processor.
- Partitioning: Partitioning the memory creates an area for storing confidential data, e.g. encryption keys.
- Flexibility: Programmability of the HSM enables coverage of different use cases and OEM-specific requirements.

Design and Functionality of an HSM

An HSM is a subsystem within a microcontroller, comparable to an additional processor core in a multicore processor.

It generally has its own RAM and flash memories which are protected from access by the rest of the system. In addition, an HSM is equipped with hardware accelerators for reducing the computation time for cryptographic algorithms. It should also be noted that just like all other processor cores, the HSM can execute any software – i.e., it is programmable. Essentially, it is this software that determines the functionality of the HSM and represents its interface to the rest of the system. It is referred to as "HSM firmware."

To highlight the advantages of an HSM, three approaches to implementing security functions need to be compared **(Figure 2)**:

> A pure software solution on the main processor
> A hardware-accelerated solution on the main processor
> A solution with HSM

Since cryptographic algorithms are generally computing-intensive, a pure software solution requires a lot of processing time, and so it is often not the ideal solution. The use of a hardware-accelerated computation on the main processor and accelerated computation in the HSM achieve comparable values. However, the HSM offers the advantage of improved concurrency if the computation is performed in software. This can relieve the load from the main processor.

The HSM offers the best solution for separating and protecting confidential contents. Partitioning the memories keeps protection-worthy contents encapsulated in the HSM and therefore separated from the rest of the application. For the hardware-accelerated solution on the main processor, this only applies to a limited number of symmetrical keys. In the case of the HSM, on the other hand, if its memory is sufficiently large, it can store a flexible number of keys, certificates and other contents.

In terms of flexibility, the HSM benefits from its programmability. Hardware support on the main processor has a fixed functional scope, and therefore it cannot react to changing requirements. Of course, this also applies to the hardware support on the HSM. But the added option of implementing requirements in the HSM firmware yields a decisive gain in flexibility.

Use Cases of an HSM

As mentioned, some ECUs check for the authenticity of the ECU application at startup. However, this method, known as "secure boot," extends the duration to start the system. The HSM can significantly reduce the time required by means of hardware support. It is even possible for the HSM to execute this check concurrent to the system start. However, this procedure requires a complex interaction between the HSM and the main processor and must be compatible with automotive OEM requirements.

Another use case is message authentication. Here the transmitted message is extended by appending a message authentication code (MAC). This allows the receiver to check the authenticity of the message. The process of computing and checking this code generates additional overhead which increases processor load on the ECUs. The use of CAN FD leads to a load profile in which relatively small data packets arrive in quick succession. The accelerated computation of the MACs happens very quickly due to the small data size. However, due to the high frequency, the additional overhead of the communication between the main processor and the HSM is especially important. This should be considered when implementing the HSM firmware and reduced as much as possible.

In the case of electric vehicles, communications between the vehicle and the charging station (vehicle to grid) are governed by the ISO 15118 standard. Challenging cryptographic algorithms are used here, specifically those used to enable billing operations between the vehicle owner and the network operator. Communications are secured by Transport Layer Security (TLS), a method that is widely used on the Internet. In addition to the protection of the

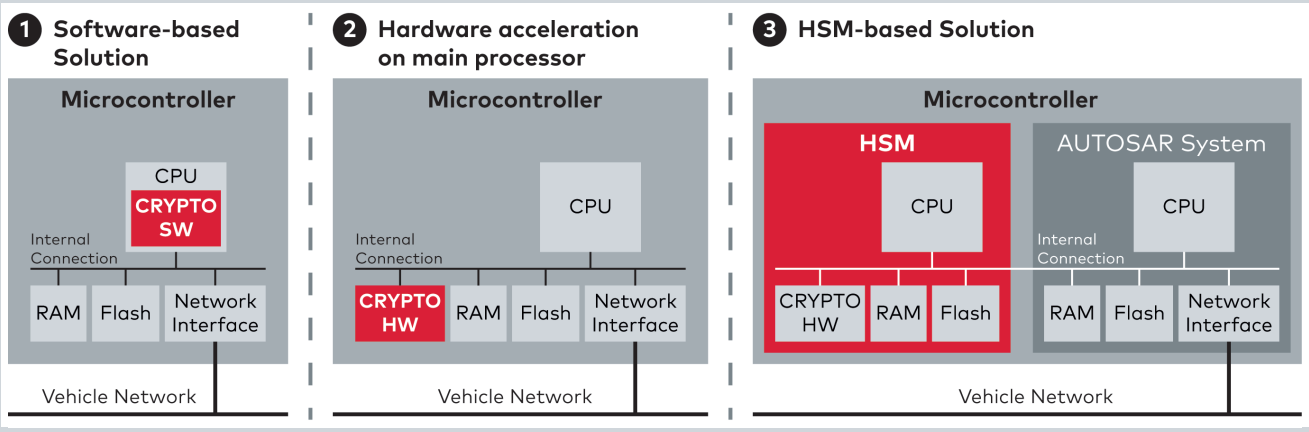

Figure 2: Different approaches for implementing security functions

communication, a process is defined for safe storage and installation of certificates. An HSM can reduce loads of both the main processor and the time-intensive process for establishing a TLS connection, and it can separate protection-worthy contents such as private keys from the rest of the system.

The mentioned application areas illustrate just how multifaceted the requirements for HSM firmware already are today. However, further applications can already be expected in the near future such as diagnostic communication over IP (DoIP) validated with TLS, a new ISO-standardized, certificate-based diagnostic service for enabling security-related diagnostic services and validation of data traffic via Internet Protocol Security (IPsec).

Linking an HSM to an AUTOSAR System

The software architecture of an AUTOSAR system differentiates between the ECU's application software and the basic software. The basic software provides various background services to the application software – such as those for bus communications, diagnostics and memory management as well as cryptographic services for security-related functions. The cryptographic services are provided by the Crypto Service Manager (CSM) module. The services offered include symmetrical and asymmetrical encryption, computation of cryptographic checksums, generation and verification of MACs or signatures, as well as generation of random numbers.

In addition, the CSM offers access to a database for storing security-relevant information such as cryptographic keys, certificates and application data. Depending on the configuration, individual contents may be read or written or they may be exchanged using a cryptographic protocol. Furthermore, operations are available for deriving new cryptographic keys from confidential data that has already been stored. To ensure that protection-worthy contents remain in the database, the provided cryptographic services access the database contents directly.

The cryptographic operations for implementing services are provided by so-called crypto drivers (CRYPTO). A general distinction is made between software drivers and hardware drivers. Software drivers (CRYPTO SW) rely on a software library with cryptographic algorithms. Hardware drivers, on the other hand, enable the integration of cryptographic hardware accelerators as well as the integration of an HSM into an AUTOSAR system. Since the CSM addresses the drivers via an intermediate layer known as a crypto interface (CRYIF), the software and hardware solutions can be operated simultaneously.

The central task of an HSM crypto driver (CRYPTO HSM) is to rapidly route operational instructions to the HSM firmware. The driver and the HSM firmware communicate via the microcontroller's shared memory. Multiple channels (HSM channels) can be created in memory, via which the driver passes operational instructions to the HSM **(Figure 3)**.

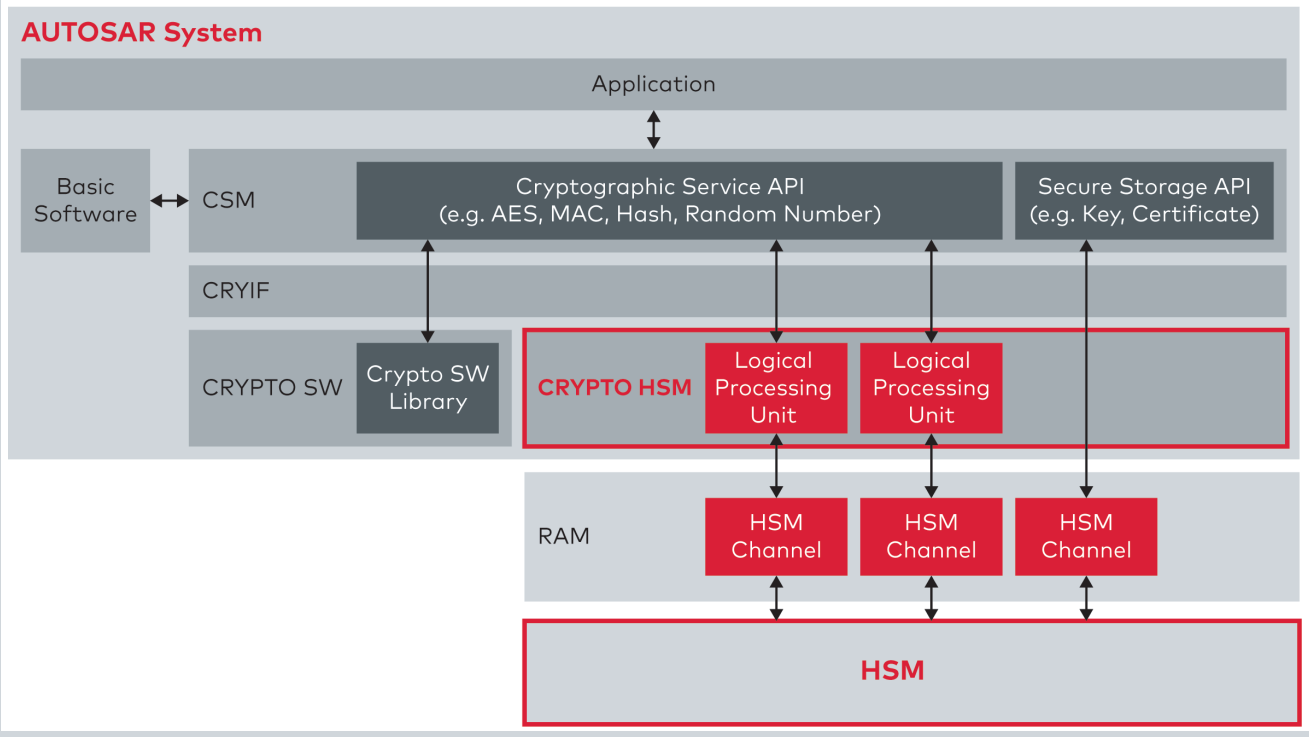

Figure 3: Linking an HSM to an AUTOSAR system

If the firmware has a suitable design, it is also possible to address the HSM from multiple cores of the main processor. This type of communication enables secure partitioning between the AUTOSAR system and the HSM firmware. The driver offers different logical processing units so that the different operations of the HSM can be addressed flexibly. They are created according to the specific functionality and configuration of the HSM firmware to address the functions of the HSM based on specific needs.

Software Architecture of a Flexible HSM Firmware Implementation

The software architecture of an HSM firmware must be modular and configurable to enable implementation of many different use cases. Therefore, the HSM firmware can be constructed in a way that is similar to the AUTOSAR architecture. This offers the following advantages:

> The HSM applies the AUTOSAR concept of crypto drivers as cryptographic extension modules.
> AUTOSAR modules for memory management can be reused.
> The HSM firmware is configured with familiar AUTOSAR tools.

A crypto driver is a modular processing unit with an AUTOSAR-standardized interface. A distributor component (Job Dispatcher) receives pending tasks and, like AUTOSAR, it distributes them to the relevant drivers in the HSM via an intermediate layer (CRYIF) **(Figure 4)**. The processing units in the HSM can be classified into three classes: hardware, software and special functions. The first class contains all hardware-accelerated operations. These are generally Advanced Encryption Standard (AES) computations and MAC computations, as well as the generation of random numbers. The second class consists of algorithms implemented in software such as the asymmetrical cryptographic method RSA and operations on elliptical curves. The third class – the special functions – consist of application-specific operations. These enable the implementation of OEM-specific or ECU-specific requirements. This design guarantees a high degree of portability of the HSM firmware, because hardware-specific modules are encapsulated. By integrating software libraries, the HSM firmware can be flexibly supplemented by other cryptographic operations.

Memory management of the HSM must also be flexible. Typically, small but many symmetrical keys are stored in the HSM for message authentication. For protected communication via TLS to a charging station or a diagnostic tester, however, just a few but larger certificates are needed. The database – known as Secure Storage – utilizes existing basic software modules for memory management to securely store contents in the HSM's nonvolatile memory. It includes the options of redundant data storage and memory partitioning.

The available computing and memory resources of the specific HSM hardware are limited. To be able to implement all application cases with available resources, the HSM firmware must be tailored to the application cases. In the sim-

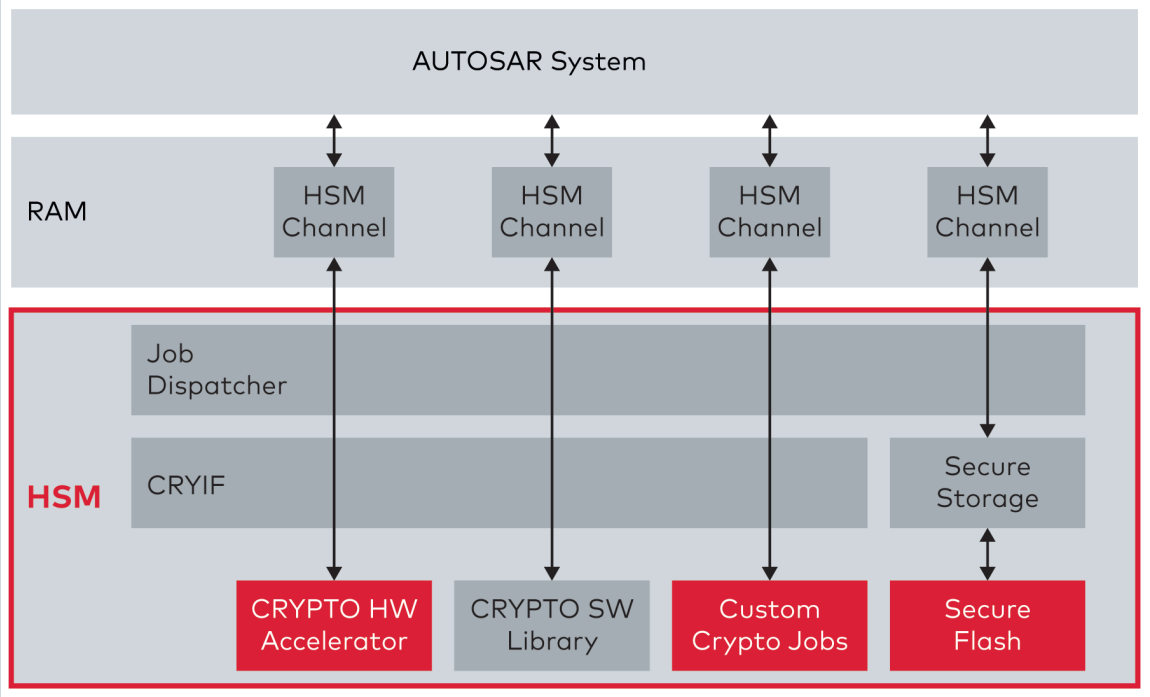

Figure 4: Simplified architecture of an HSM firmware implementation

plest case, this involves activating or deactivating cryptographic algorithms during the configuration process. In addition, the database's memory layout must be optimized. Existing configuration tools for AUTOSAR systems are used to make these settings in a user-friendly way.

Outlook

Today's hardware security modules already need to cover a large bandwidth of use cases. However, continued growth can be expected in their variety, scope and with respect to application-specific parts. For this reason, an HSM firmware must flexible and configurable to be viable for the future. In addition, it can be expected that ECU-specific and OEM-specific applications will increasingly be offloaded to the HSM whenever these applications work with confidential contents or need to be executed with protection from the rest of the system. A solution is being sought that can handle the defined requirements as efficiently and flexibly as possible. This affects both the HSM firmware and the HSM's interface to the application. The vHSM solution from Vector offers both and is based on the AUTOSAR design. This simplifies integration and configuration of the software.

Dr. Bastian Zimmer
is head of the Solution Management Team in the Embedded Software Department of Vector Informatik GmbH. He and his team are responsible for establishing innovative themes and technologies such as Ethernet, multicore, security and safety. After earning his doctorate degree at the Fraunhofer IESE, he joined Vector Informatik GmbH in 2015. At Vector, he was Solution Manager for Gateway Controllers until 2016, when he took on his current role.

Max-Ferdinand Suffel
works as a software developer on the Solution Management Team in the Embedded Software Department at Vector Informatik GmbH. He works on new technical concepts and solutions in the security area. He joined Vector Informatik GmbH in 2015 after graduating in computer science from the University of the Saarland which included a semester at the University of Washington in Seattle.

Translation of a German publication in Automobil Elektronik, issue 07-08/2018