Secure communication for CAN FD

Encrypted data transmission is not yet the norm in vehicle networks. Vector has conceived an implementation for secure communication over CAN. Protection goals were authentication and preventing replay attacks.

Author

Armin Happel

Principal Software Development Engineer
Vector Informatik GmbH
Ingersheimer Str. 24
DE-70499 Stuttgart
Tel.: +49-711-80670-0
Fax: +49-711-80670-111

Link
www.vector.com

In today's vehicle networks, data transmission is for the most part performed without any special security measures. Because of this, it is possible to read out the data transmitted in raw format or to even play it into the bus system in modified form if you have direct access to the vehicle bus. Encrypted data transmission would not only ensure that this information could only be evaluated by authorized recipients. At the very least, it would also make it much more difficult to intercept or alter the messages.

Media reports about vehicle manipulation [1], [2] raise the question of whether data in the vehicle network can actually be influenced by manipulation. Can a manipulated device or internally implanted device with a remote control function influence vehicle behavior? And what countermeasures can be taken to prevent such manipulations?

Today's vehicles are highly complex systems, which consist of networked sensors and actuators and continually transmit important data over bus systems. In the vast majority of cases, the information being transmitted is in raw data format. A plausibility check, if such a check is even possible, has limited effectiveness. The receiver is unable to verify whether the data was ▷

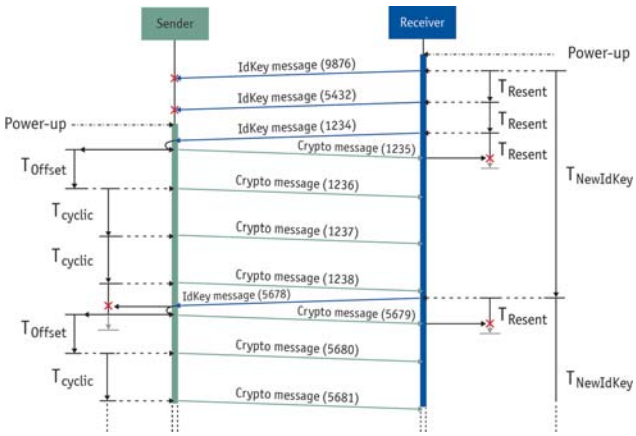

Figure 1: Message transmission and timing of encrypted communication

actually supplied by the desired sender or whether it was fed in by an outside electronic control unit, i.e. whether it is authentic data. The data is freely accessible as well, so an analysis of the bus information can be used to determine signal contents. The transmission is neither confidential nor authenticated.

This was the problem that engineers at Vector were confronted with. Their task was to come up with an implementation for secure communication over a CAN network which can be used flexibly and can also be integrated with Autosar-3.x basic software. Protection goals were authentication and preventing replay attacks. It was also desirable to implement communication that cannot be monitored

For the encryption method, the specialists chose the AES algorithm [3]. From today's perspective, this method is considered cryptographically secure. It involves symmetrical block encryption with a block length of 128 bits. It generates 16 bytes or a multiple of 16, which the sender transmits to the receiver. An additional advantage is that some microcontrollers already have very fast hardware-based implementations of this algorithm.

Since a CAN message can transmit a maximum of 8 data bytes per frame, a decision was made to utilize the ISO transport protocol (TP) that was already included in the communication stack for the transfer. To simplify the configuration and reduce protocol overhead, a unidirectional communication with a fixed 1:1 relation between sender and receiver was chosen.

Symmetrical encryption requires that both the sender and receiver have the same key. The software modules that are used permit dynamic allocation of the keys at runtime, so that the user or OEM can freely

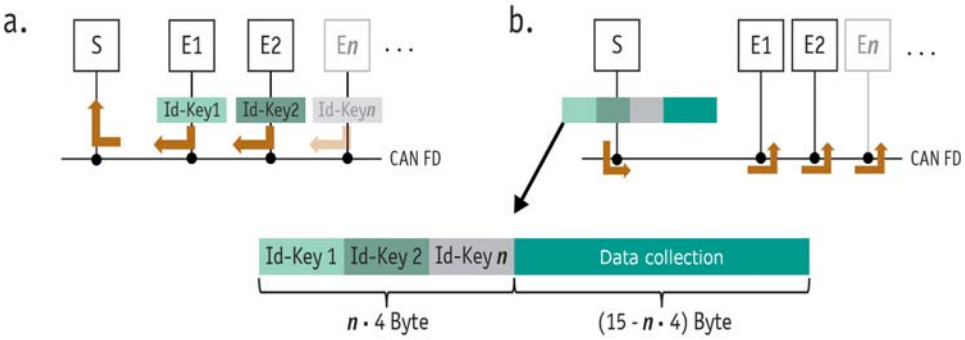

Figure 2: ID keys of multiple receivers in the use of CAN FD

choose them. A higher-level method such as a (asymmetrical) key exchange method might be implemented, or a static allocation might be made such as in end-of-line programming. Whenever an ECU is replaced and a vehicle specific key is used, the new ECU must be set up by an authorization method, which keeps the key confidential under all circumstances.

Preventing replay attacks

In this configuration, an encrypted transmission of messages is now possible, where the information is, however, still purely static, i.e. a unique key text can be assigned to the plain text signals. This means that replay attacks, i.e. recording excerpts of a desired communication and replaying it into the system at a later time, can still be made. That is because the receiver cannot check whether the message actually originates from the sender at this point in time. To make checking possible, at the start of communication the receiver generates a random value – which is referred to as the ID key in the following – and it communicates this to the sender. The sender increments the value with each transmission and appends it to the transmit message. When the message arrives, the receiver checks whether the ID key matches the expected value. If it does, it processes the message; otherwise it rejects it. To tolerate possible message losses, the receiver will also accept a slightly higher value. This means that the counter in the transmit message continually alters the encrypted data even if the signal contents remain the same (Figure 1).

Depending on the word width of the ID key and the frequency with which the message is sent, overruns of the counter value might be expected in the message, which would lead to repeated transmission of the encrypted message. To avoid this, the ID key is only valid for a certain time period. When this period expires, the receiver must generate a new value and communicate it to the sender. Immediately after receiving a new ID key, the sender transmits the encrypted message. This means that the receiver is also able to initiate repetition of a message, such as if the received ID key does not agree with the internal key, and this reduces latency times. Although the sending node receives and considers new ID key messages for a time T(offset), to avoid an overload of the bus system such messages do not immediately lead to resending of the encrypted message. To make the protocol more robust, the receiving side uses the timer T(Resent) to monitor the response of the sender with the new counter value. If it does not get an acknowledgment message from the sender, the receiver generates a new ID key and resends it. This makes it possible to detect even a brief failure of the sending ECU and shortens the time for resending. It also avoids storage of the ID key in non-volatile memory.

Data transmission without segmentation

There is a significant disadvantage associated with segmented data transmission in CAN over the ISO-15765 transport protocol. Transmission time is increased, and this method is restricted to a fixed 1:1 relationship, because segmented data transmission over ISO-15765 is very difficult to implement with multiple nodes. CAN FD on the other hand enables simultaneous transmission of the entire encrypted message to multiple receivers [4]. Each receiver needs the same symmetrical key to decrypt the encrypted message. Two variants of the ID key for authentication come into consideration: either all receivers agree on a commonly agreed value, or all receivers independently generate and send their ID key to the sender. The sender manages all counters and appends them to the data message. The positions of the counter values within the encrypted message must be uniquely assigned to the receivers.

Figure 2 shows data transmission for multiple receivers. First, the receivers transmit their randomly generated start values to the sender. The sender then increments all ID keys for each send cycle and insert them into the encrypted message at the predefined positions. The relevant

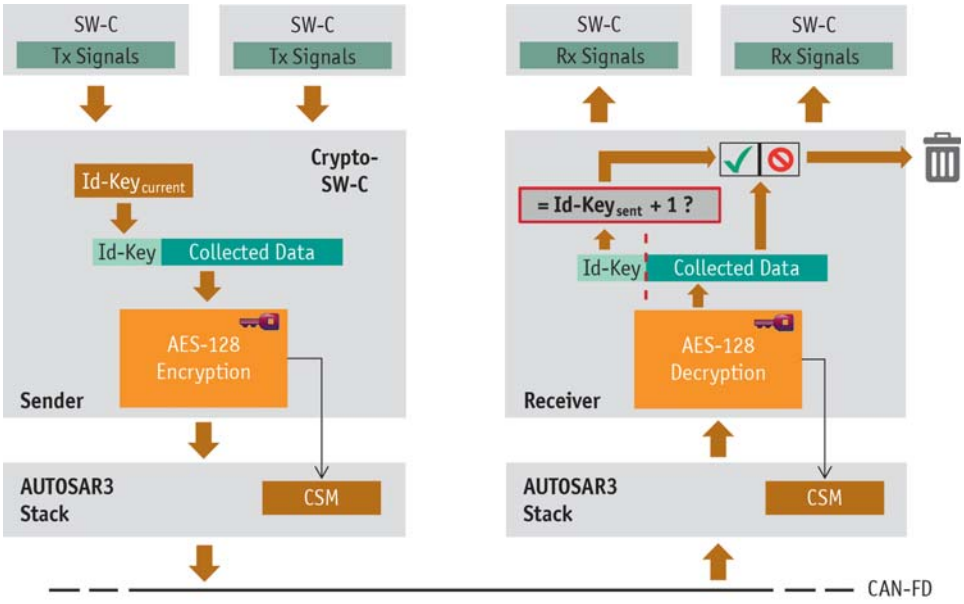

Figure 3: Software components for encrypted transmission

Literature

[1] http://www.chip.de/news/CAN-Hacking-Tool-Autos-hacken-fuer-20-Dollar_67066892.html [only German]

[2] http://www.can-newsletter.org/engineering/engineering-miscellaneous/140822_list-of-potentially-vulnerable-cars_blackhat/

[3] Advanced Encryption Standard (AES), FIPS PUB 197

[4] CAN with Flexible Data Rate – Specification Version 1.0, Robert Bosch, GmbH; April, 2012 http://www.bosch-semiconductors.de/en/ubk_semiconductors/safe/ip_modules/can_fd/can.html

receiver then checks its ID key and accepts the data or rejects it (Figure 2).

However, as the number of receivers increases, this reduces the message space that remains for useful data. The number of useful data bytes is also highly dependent on the selected word width of the ID key. The communication timing illustrated in Figure 1 was applied. It only required a modification for the sender in receiving the ID key. Instead of immediately transmitting the encrypted message, the sender waits for a configurable time T(IdKeyReply) to allow time for any other ID key messages from other receivers. The special case T(IdKeyReply)=0 covers the original method.

Vector implemented the protocol for CAN FD in a CANoe environment. The specialists subjected the protocol to extensive tests using this software tool for development, simulation, and testing of ECUs and networks. Along with the required robustness against replay attacks, another focus was to study message losses, failure, and re-entry of sender and receiver as well as timing errors and burst attacks. In all of these cases, the encryption system provided a stable transmission.

Summary and Outlook

In CAN FD, in particular, it took relatively little effort to implement robust transmission of encrypted data with multiple nodes, and this method can also fit into an existing Autosar environment. One disadvantage is the serialization and deserialization of the data on the application level (Figure 3), which means that modeling properties of the RTE cannot be used any longer for individual signals. The classic points of attack on such systems must still be kept in mind. They include, for example, weak random number generators for the ID keys (at startup) or spying the symmetrical keys.

In the security technology world, the AES-128 algorithm is considered secure for the near future, and its implementations are mature or will even be supported by hardware accelerators. The method presented here makes attacks on the CAN (FD) communication much more difficult, and manipulation is hardly possible without “insider knowledge”. It has already been in production use for several years, and it also has led to favorable classification of the relevant vehicle for insurance premiums. In this case, security not only protects data; it even offers a direct cost advantage to the end user.

In the near future, remote connections such as Car2x communication, WLAN, Bluetooth and Internet will continue to grow and will necessitate much more stringent requirements for IT security. These access modes must be made secure against attacks and must not permit any remote manipulation. This is especially true of information to driver assistance systems, which rely on reliable messages from other traffic participants and/or the infrastructure. ◀