White Paper

Securing your IP and Protecting Sensitive Data

Markus Vomfelde, Sr. Manager, IoT and Infrastructure Business Unit, Renesas Electronics Corp.

Brad Rex, Sr. Product Marketing Manager, IoT and Infrastructure Business Unit, Renesas Electronics Corp.

Zachary Ellis, Sr. Marketing Specialist, IoT and Infrastructure Business Unit, Renesas Electronics Corp.

January 2020

Abstract

In the first white paper of this series, Renesas gave an overview about security for the connected world. This white paper will give more details about the demand and the techniques – why and how – to protect stored data in the MCU or application. Based on an artificial example it will guide you through the consideration of different levels of security implementation based on the demand from the application and potential attack scenarios. This should help you to prepare a security plan for data stored in the device and find the best security level based on your demand.

Why protect local stored data?

There are two different kinds of local stored data, the application program which will be executed while operating, and the local data which is used during operation. The application program includes the knowledge or IP of the manufacturer, and therefore the device manufacturer protects its knowledge from being stolen, reused, or copied. Data is normally stored on a device. Data can be transferred and updated on a separate, identical device since all devices have the same attributes. The local data is stored during setup of the device in the final environment, or during operation of the device. All devices have different data inside and the updates are usually more frequent than the application code. The motivation to protect this data is more from the user of the device, as the data might include sensitive information for their environment. Even though the motivation of securing the data is rather different, the demand to protect the data to be accessed externally is mandatory for both types of data.

Is your device connected?

This is a very important question for the amount of security implementation. For a device running standalone without any connection, possible attacks can be reduced to ones which have direct physical access. This is still an issue for the manufacturer to protect the IP inside the device, but for the user data, this is much more unlikely as an attacker usually doesn't have physical access to the device. The next level of connection is a device operating in a local closed network without any connection to the Internet. Here an attacker has to get access to the network before an attack on the device can start, but it also needs to be protected against external access and not become the entryway to the closed network. Finally, a device with direct connection to the Internet demands the highest amount of security

implementation as the number of potential attackers is no longer limited to local connectivity but can be done globally and with nearly endless computing power. Also, this kind of attack will increase to get access to data stored inside the device.

Example Application and Security Demands

To make the complete topic more concrete, let us consider an example application, which is fully artificial, but realistic enough to reflect the security demand for real applications.

As an example, we have a door lock with a fingerprint sensor to give access to restricted areas of a company building. This sensor has a very clever algorithm to store the fingerprints of the 50 most common users inside the device with a very low memory size. This feature makes it most attractive for customers in the market. For other users, the device connects via a company Wi-Fi network to a server and makes the comparison with stored fingerprints, as this takes more time to grant access, the internal stored data is at a real advantage. The Wi-Fi network also has access to the Internet to be ready for over-the-air updates to the device from the manufacturer.

As a device manufacturer, you have to devise a plan to implement security for your application. In this white paper, we will just focus on the data stored in the device and ignore the data which will be exchanged via operation or program update (data in flight).

The first kind of data to secure is the IP of the fingerprint algorithm. This is the value of the device itself and should be protected against any access an attacker can get to the device, either direct or via a data connection. As the device is connected to a network, it is not enough just to protect the MCU in the device from read out, copy, or reprogramming. In addition, you must secure the IP to be dumped from the memory via the connection to the attacker.

The second kind of data you must keep in mind is the user data, the stored fingerprints and the network access data in our example. As explained above, the physical access to the device will become more difficult for an attacker to get user data. The access via an Internet connection is more likely, and therefore needs better security protection against attacks. This is partially in the hand of the user and their protection of the network; however, inside the device the security must be implemented to complete the security setup.

Secure your IP

Based on the given example, there are several security parts necessary to protect stored data. To focus on data security, we assume that we are using a device with a secure device identity and can be trusted. The next white paper in this series will explain what kind of MCU is necessary.

Regarding protection of the IP, there are several levels of protection to be implemented which depend on the security plan you've selected, and your defined scope of protection. As a first step toward implementation, the MCU you choose has to offer protection against unwanted debugger access and reprogramming. There are a variety of ways to achieve this crucial protection and you have to judge by comparing the different ways of implementation. Different vendors use various protection methods which also have varying security capabilities. You must make sure that this implementation is recommended for security and not just for preventing unintended modification of the device. The next level is to use an MCU with implementation to support different access areas, which are either trusted or non-trusted. This will avoid the MCU core from having direct access to the IP and therefore an easy dump of data cannot be performed. Here you will also find different solutions. Most common is the implementation of a Memory Protection Unit (MPU), which could be used for purposes described above, or the TrustZone® implementation of an ARM® based microcontroller. Finally, you can store the IP in an encrypted way on the device. This will make it much more resistant against physical attacks as there is no non-volatile

memory where the IP is stored as readable data and cannot be read out via encapsulation or analysis with an electron microscope. Consequently, the key for the encryption, which is stored in the MCU, must also protect against readout, direct access from CPU, and must be stored securely to avoid the readout of the key and the encrypted IP to get access to the secret information. If you store the algorithm encrypted, you have to decrypt it in the RAM of the device and execute from there. This is the most secure way to store the IP, but it will also be mandatory to include the RAM portion where the algorithm is stored into the trusted area of the MPU.

Secure the Data at Rest

In the second step you must decide on the data the end customer will store in the device. In our example, the fingerprint data was stored to have fast access to the area, and also the access to the network of the customer to allow connection to the server where all fingerprint data is stored. This will also allow the manufacturer to make future firmware updates. Basically, the same security measures can be applied as this action was performed for the IP stored in the device. We want to have a closer look and decide on the mandatory level of security implementation in the operation. The device should be protected against read out or reprogramming, even partially, to avoid the installation of any kind of malware which could provide data over the network to the attacker. Also, the implementation of trusted and non-trusted memory areas is very meaningful, as this will limit the possibility of the MCU to access the stored data. This will make any attack more difficult and provides improved protection with limited performance degradation.

Finally, the encryption of the data is a mandatory measure, as this will give a negative effect on the performance. All stored fingerprints must be decrypted before the algorithm can start its operation, so this add on in performance has to be considered in advance. On the other hand, the physical access to the device inside the customers' building might be rather difficult and therefore needs to be considered if this add on becomes mandatory. What can an attacker do with the stored fingerprint data as long as the algorithm of making the comparison is not accessible? For the network access data, this is different. Here, the negative impact for the performance is almost zero, as this needs to be done once or twice a day, but if somebody can get hold of a device and can read out the network access code as unencrypted data, they will have the full access to the customer network and this might become more dangerous and unpredictable. Again, it needs to be highlighted that the storage of the key for the encryption has to be done with more security than the data itself to avoid any unwanted access to the encrypted data. A very effective way to do so is a uniquely wrapped key on each MCU, but the topic 'Key Management' will be discussed in one of the following white papers in this series.

Conclusion

The decision of how, and to which level, to implement security always depends on the application, the expected attackers, and their access to the device or data to be secured. This means that for each security implementation, the development team has to consider this at the start of the project to make the important decision for the MCU, which is fitting to all needs of the security implementation. The example here shows the wide variety of security for local stored data and it will increase with additional functions for data in flight, or secure programming over the air. Further white papers of this series will provide you this information and support your design of a secure product for the connected world.

Renesas offers multiple MCUs[1] that address the concerns discussed in this white paper. Please visit our website to learn more.

References

[1] RA Family of 32-bit Arm Cortex-M MCUs
RX Family of 32-bit MCUs
Synergy Platform of 32-bit Arm Cortex-M MCUs + qualified software