LoRaWAN with HSM as a Security Improvement for Agriculture Applications

Reinhard Kloibhofer✉[1], Erwin Kristen[1], Luca Davoli[2]

[1] AIT Austrian Institute of Technology GmbH, Giefinggasse 4, 1210 Vienna, Austria
reinhard.kloibhofer@ait.ac.at✉, erwin.kristen@ait.ac.at
[2] Internet of Things (IoT) Lab, Department of Engineering and Architecture, University of Parma, Parco Area delle Scienze 181/A, 43124 Parma, Italy
luca.davoli@unipr.it

Abstract.
The digital future in agriculture has started a long time ago, with Smart Farming and Agriculture 4.0 being synonyms that describe the change in this domain. Digitalization stands for the needed technology to realize the transformation from conventional to modern agriculture. The continuously monitoring of all environmental data and the recording of all work parameters enables data collections, which are used for precise decision making and the planning of in-time missions. To guarantee secure and genuine data, appropriate data security measures must be provided.

This paper will present a research work in the EU AFarCloud project. It introduces the important LoRaWAN data communication technology for the transmission of sensor data and to present a concept for improving data security and protection of sensor nodes. Data and device protection are becoming increasingly important, particularly around LoRaWAN applications in agriculture.

In the first part, a general assessment of the security situation in modern agriculture, data encryption methods, and the LoRaWAN data communication technology, will be presented.

Then, the paper explains the security improvement concept by using a Hardware Secure Module (HSM), which not only improves the data security but also prevents device manipulations. A real system implementation (Security Evaluation Demonstrator, SED) helps to validate the correctness and the correct function of the advanced security improvement.

Finally, an outlook on necessary future works declares what should be done in order to make the digital agriculture safe and secure in the same extent as Industrial Control Systems (ICSs) will be today.

Keywords: LoRaWAN, Trusted Platform Module (TPM), Internet of Things (IoT), Cyber-Physical Systems (CPS), Safety & Security, Agriculture.

1 Introduction

This paper aims to bring a reader closer to the importance of Long Range Wide Area Network (LoRaWAN) technology for the transmission of digital data and to present a concept for improved data protection, which, in turn, is becoming increasingly important, particularly around LoRaWAN-based applications in agriculture.

LoRaWAN has triumphed in recent years when it comes to periodically transferring small amounts of data over long distances [1]. This type of short- and medium-range data transmissions is gaining more and more importance in data and commands' distribution for sensors and actuators at the field level and in Internet of Things (IoT)-oriented contexts [2]. A special application area will be the modern agriculture domain, where different sensors, directly installed on the field, deliver environmental data to support finding correct decisions for the exact mission planning in time.

However, the increasing digitalization in agriculture and the associated networking of machines and production systems increase the risk of cyber-attacks. Especially, by widely distributed production facilities (at field level) in agriculture and the network supported interaction with the Information Technology (IT) world, new points of attack have been disclosed. This technical progress allows an easier penetration of attackers to the production facility, manipulating it and even impairing safety (e.g., machinery safety).

A new important aspect of modern agriculture is the fact that the above-mentioned field level becomes more and more powerful. Today's field devices are highly integrated and powerful electronic processing systems, with high-performance computing capabilities, firmware updates and maintenance interfaces. The attractiveness for cyber security attacks and field device misusing is expected to rise.

IoT security deals with different types of threats. The following list summarizes the most critical vulnerabilities.

- Espionage: this vulnerability type focuses on collecting data from the cyber-attacked victim. The data is used to gain secret knowledge or to obtain information in order to prepare further attacks, e.g., theft of sensor data.
- Destruction and Exaction: the goal of this vulnerability is to perform data adulteration or produce system damage, e.g., falsification of the original sensor data.
- Sabotage: the goal of this vulnerability is to reduce or prevent correct system operations, e.g., shortening battery life by permanently activating the sensor node.
- Misuse: the goal of this vulnerability is the unauthorized use of system equipments to perform criminal actions, such as building botnets and kidnapping of foreign computers for Distributed Denial-of-Service (DDoS) attacks. An example of this kind of vulnerability is the installation of malware.

Important countermeasures are the early detection of attacks, the encryption of transmitted data and the protection against unauthorized device access by using a login procedure (e.g., with usernames and passwords, as well as based on tokens).

2 Data Encryption with Symmetric and Asymmetric Keys

The data transmission from a sender to one or multiple receivers, as well as the data storage, must be protected against eavesdropping and manipulation. Therefore, there is the need of encryption algorithms, which are a set of mathematical procedures for performing encryption tasks on data. With the use of such algorithms, data will be transformed to ciphertext through a secret key, thus requiring the use of the same or another secret key to transforming data back into its original form. Moreover, through cryptography the data is transformed so that it cannot be read or understood by an eavesdropper, while only the trusted receiver, who has permissions, can transform the ciphertext to the original data (by using the secret key). This technique is old and used from the Roman times (e.g., through the Caesar cypher).

To encrypt and decrypt, it is possible to distinguish between (i) symmetric encryption and (ii) asymmetric encryption. The symmetric encryption represents the simpler way and is characterized by the fact that keys for encryption and decryption are identical, as shown in Fig. 1. In this case, both sender and receiver must have the same secret key, which, in turn, must be generated and exchanged at least at communication channel or storage's setup time. Another problem regards the protection of this key against unauthorized read-out or distribution. Examples of symmetric encryption algorithms are Blowfish, Advanced Encryption Standard (AES), and Data Encryption Standard (DES). The most commonly used algorithms are AES-128, AES-192, and AES-256, where the number denotes the key length in bits.

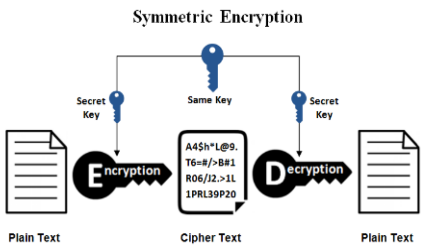

Fig. 1. Symmetric Encryption[1]

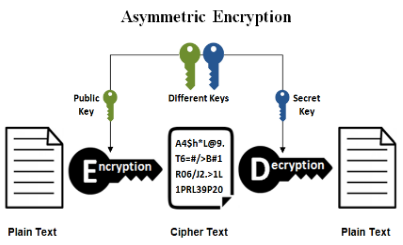

Fig. 2. Asymmetric Encryption[1]

A more complex encryption schema is represented by the usage of asymmetric encryption, which is a relatively new method. In this case, the keys for encryption and decryption are different, and denoted as private and public keys, as shown in Fig. 2. The public key, used for encryption purposes, is made freely available and can be distributed to everyone who wants to encrypt data for the receiver. Instead, the private key is known only by the receiver and will be never distributed to anyone else. A message that is encrypted using the public key can only be decrypted using its paired private key, while a message encrypted using the private key can be decrypted using the public key. Security of the public key is not required, it can be stored and sent

[1] Source: https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences

unsecured. Asymmetric keys improve the security of information transmitted during communication.

In current Internet communications, asymmetric encryption is the most commonly used technique for securing the data transfer.

3 Overview on LoRaWAN

LoRaWAN is a Media Access Control (MAC) protocol for Wide Area Networks (WANs) [3]. A focus in the design of LoRaWAN was to allow low-power devices to communicate with a LoRaWAN server, leading to the involvement in Low-Power WANs (LPWANs). LoRaWAN is implemented on top of the LoRa modulation in the Industrial, Scientific and Medical (ISM) radio bands. The specification can be found on the LoRa Alliance website,[2] while its network architecture is shown in Fig. 3.

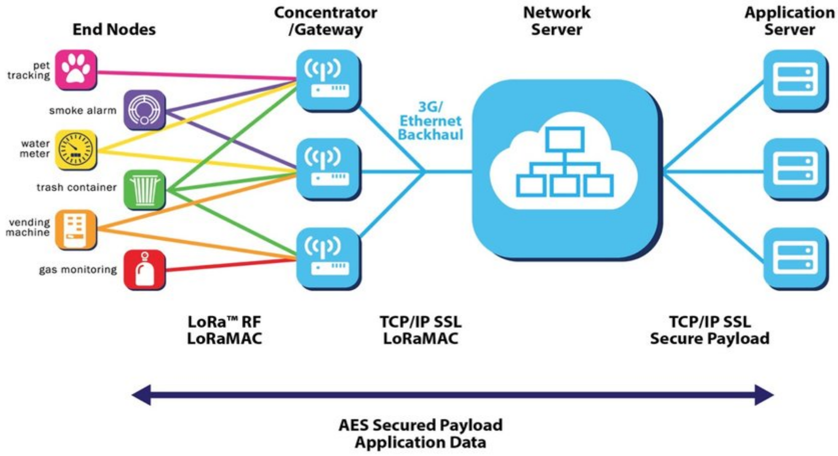

Fig. 3. LoRaWAN Network Architecture[3]

Instead, LoRa represents the physical layer of a communication protocol able to support long-range communication and is based on Chirp Spread Spectrum (CSS) modulation [4], which significantly increases the communication range, if compared to Frequency Shift Keying (FSK) modulation. CSS has been used in military applications for long time, but LoRa is the first low-cost implementation available for commercial use, allowing data transmissions over distances up to 10 km.

While many existing deployed networks follow the mesh network approach, where each end-node is also used to forward messages from other nodes to extend the transmission range, LoRaWAN uses a star topology. This saves battery life of the end-nodes because they do not act as gateways (GWs). Moreover, a LoRaWAN end-

[2] Source: https://lora-alliance.org/

[3] Source: https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf

node is not associated with a specific GW since, in a well-designed operating environment, uplink data from an end-device are received by multiple intermediate GWs and forwarded to the Network Server (NS). Hence, the NS can handle multiple copies of data, performs security checks, schedules acknowledgements, manages the back channel from the Application Server (AS) to the end-node, and decides over which GW the back communication will be performed.

End-nodes can work asynchronously, meaning that they can wake-up and communicate when they have new data. For receiving downlink data from the server, the end-node opens the receiver interface at pre-defined time windows. This operating mode, defined as Class A, helps to save battery power and enables an operation time for up to 10 years with a single battery cell. Other operating modes can set the end-node to continuously open the receiver interface to react faster to commands from the server, thus consuming more battery power and therefore lowering the battery lifetime.

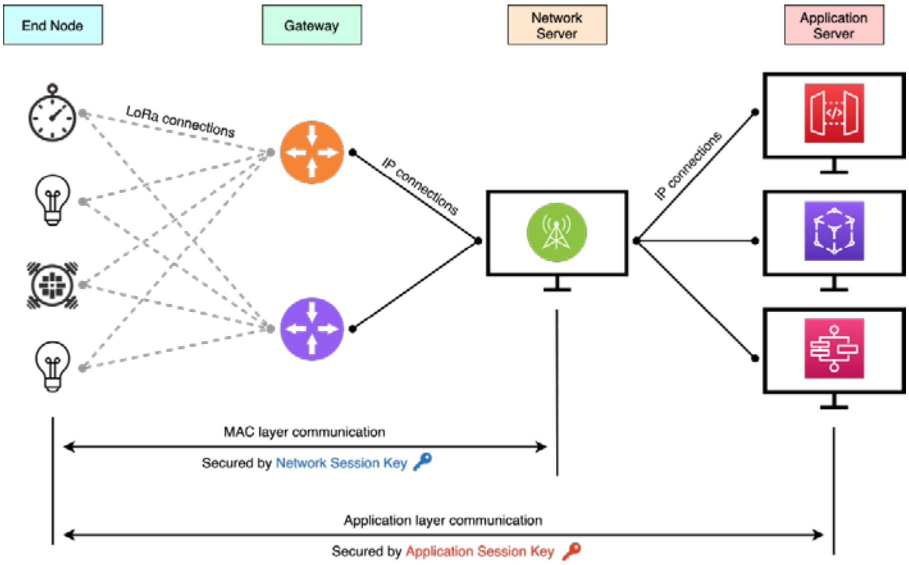

Fig. 4. LoRaWAN security with network and session keys

Regarding security and encryption, in LoRaWAN they are performed in both the network and application layers [5], as illustrated in Fig. 4. The network security enables authenticity of the end-node in the network, while the application security protects data between end-nodes and the AS. Moreover, the network layer does not have access to application data and, for both layers, AES-based symmetric encryption is used, being well-analysed, approved by the National Institute of Standards and Technology (NIST), and widely used.

For data encryption and decryption, Network Session Key (NwkSKey) and Application Session Key (AppSKey) are used. These keys should be strongly protected against hacking and misuse in either end-node, NS, and AS.

For setting-up a LoRaWAN network and the application, NwkSKey and AppSKey must be generated and exchanged among the different network devices (end-node, NS, and AS) through two different techniques (defined in the LoRaWAN standard):

- Over-The-Air-Activation (OTAA): this is the preferred and most secure way, since an end-node communicates with the NS to perform the activation process, denoted as join procedure. According to the LoRaWAN specifications, the OTAA mode is used when an end-node is already deployed, or after a reset.
- Activation By Personalisation (ABP): in this mode, the session keys are pre-stored in the end-node and the servers (NS and AS). This activation might seem simpler, because the join procedure is skipped, but it has some disadvantages related to security aspects.

In both activation modes, root keys and session keys must be protected. On the server side, a Key Management (KM) system can be used [6], while on the end-node, the protection of the keys is more challenging. In order to furtherly improve the security, a periodical keys alteration is recommended, in order to prevent a successfully security key theft via brute force methods.

4 Security Module

A Trusted Platform Module (TPM, also known as ISO/IEC 11889-1:2015 [7]) is a device provided with a secure cryptographic processor, that is a dedicated microcontroller designed to secure hardware through integrated cryptographic keys. Once enabled on a system, the TPM can provide full disk encryption capabilities. Moreover, it becomes the "root of trust" for the system to provide integrity and authentication of the boot process, and keeps hard drives locked/sealed until the system completes a system verification, or authentication check. The TPM includes a unique Rivest-Shamir-Adleman (RSA)-based security key burned into it, used for asymmetric encryption. Additionally, it can generate, store, and protect other keys used in the encryption and decryption process. A TPM is normally integrated in the system hardware (HW) and cannot be removed; without the TPM, the system cannot work. A Hardware Security Module (HSM) is like a TPM, but it can also be added or connected later to the host system, by a connector, and it can perform the same security features as a TPM.

The security module adopted for the implementation purposes in this paper is an HSM called Zymkey 4i [8] and produced by Zymbit Corporation. As shown in Fig. 5, it is a small-scale module which is designed to work with Raspberry Pi (series 3 and 4) boards. However, it can also be connected to other microcontrollers or host systems.

Fig. 5. Zymkey 4i module[4]

The Zymkey HSM has multiple security layers to protect against cyber and physical threats. A secure element (SE), as part of the HSM, with micro-grid protected silicon stores sensitive resources, while a security supervisor isolates the SE from the host computer and provides additional functions of multi-factor identity/authentication for devices and physical security. The key features of the Zymkey module are the following.

- Multi Device Identification and Authentication: Zymkey enables remote confirmation of the HW configuration of the host device. It has a unique Identification (ID) token that was created with several device-specific parameters. Cryptographically-derived ID tokens are never made available to customers.
- Data Integrity, Encryption & Signing: the cryptographic engine uses some of the strongest encryption functions available on the market to encrypt, sign and authenticate data. These includes Elliptic Curve Digital Signature Algorithm (ECDSA), Elliptic-Curve Diffie–Hellman (ECDH), Advanced Encryption Standard (AES-256), Secure Hash Algorithm (SHA256). It also incorporates a True Random Number Generation (TRNG).
- Key Security, Generation & Storage: the module can store key pairs in tamper-resistant silicon to support different security services. Multiple key slots can be used. There are pre-defined and user slots available. Once generated, private keys are never exposed outside of the silicon and therefore cannot be copied, or keys can be stored in the module which will be erased depending on security policy.
- Physical Tamper Detection: the module monitors the physical environment for symptoms of physical tampering (Perimeter Detection). This includes the supervision of interrupting or break of two independent wire loops. A physical intrusion into a device (like a sensor) can be monitored. Optional accelerometers detect shock or fast orientation changes. Also, the quality of the power supply can be monitored.
- Real Time Clock (RTC): the Zymkey includes a battery-backed RTC to support off-grid applications.
- Ultra-Low Power Operation: the module delivers long-term autonomous security from a built-in battery.
- Secure Element Hardware Root of Trust: the Zymkey provides different layers of hardware security, having a dual secure-processor architecture in which it is hard to penetrate.

[4] Source: https://community.zymbit.com/t/getting-started-with-zymkey-4i/202

Each module has a unique Serial Number (SN). When the Zymkey module is paired to a host system, the host platform's SN and the Secure Digital (SD) card's SN will be stored together with the unique ID in the Zymkey crypto accelerator chip. After the pairing process (binding), the module is only linked to the host system. For development purposes, a temporary binding is possible. After cutting a lock tab on the module (as shown in Fig. 6), the binding is permanent, and the module cannot be used on another host. If the host's File System (FS) is encrypted with the module through Linux Unified Key Setup (LUKS), then the FS can only be read with the module connected to the host.

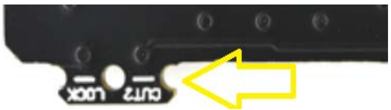

Fig. 6. Lock Tab of the Zymkey module

5 Implementation of HSM in a LoRaWAN End-Node

As explained in Section 1, security for digital data communication is a very important topic. Especially for agriculture sensor-based applications, where the physical space is not enclosed and protected as in industrial applications. Sensors are exposed in or near agriculture fields and can be stolen and then manipulated in a laboratory environment. For medium or wide agriculture environments, it is very important that both devices and data cannot be manipulated in any way, so that the whole agriculture output is not endangered.

About this, there are different types of vulnerabilities for deployed sensors, that can be improved with the use of a security element in the end-device:

1. move the sensor from the intended location;
2. physical integrity of the sensors;
3. reuse the HW for manipulation of the agriculture environment;
4. manipulate communication data.

A widely used protocol for semi-automated agriculture application is the LoRaWAN protocol, because this protocol is designed for long-range communication with small data amounts and for long-time battery use. LoRaWAN also has good security features using symmetric AES encryption and decryption on network and application layers. But, as explained in Section 2, encryption and decryption keys must be protected against read-out and manipulation from hackers.

5.1 Secure Evaluation Demonstrator

In the EU AFarCloud project,[5] one topic is the analysis and improvement of security in agriculture applications. Therefore, the Security Evaluation Demonstrator (SED) is under development to demonstrate how security improvements can be archived.

[5] Source: http://www.afarcloud.eu/

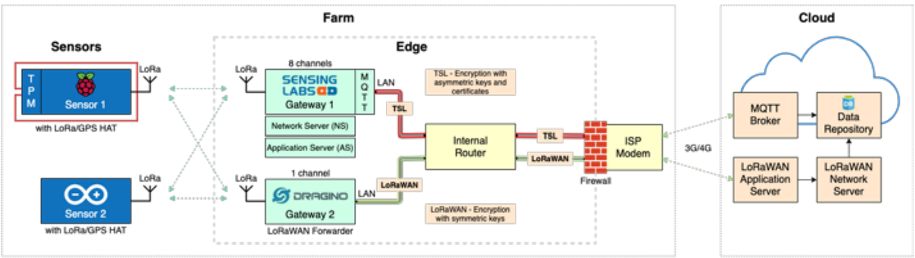

Fig. 7. Secure Evaluation Demonstrator

Fig. 7 illustrates the block diagram of the SED. There are two main blocks: the "Farm" block, which represents the farm environment, and the "Cloud" block, with processing and data repository services. The Farm can be further divided in "Sensors" and "Edge" sections. The first one comprises one LoRaWAN-oriented sensor with TPM, while the second sensor only provides LoRaWAN communications. Sensor data are received by Gateway 1, a GW with integrated LoRaWAN NS and AS, and Gateway 2, a GW acting as a LoRaWAN forwarder. Gateway 1 transfers sensor data over the internal router, while an Internet Service Provider (ISP) modem, via MQTT protocol, send them to a Cloud-based MQTT broker with Data Repository. Gateway 2 performs the data transfer to the Cloud-based Data Repository via the LoRaWAN protocol. In the second case, LoRaWAN NS and AS in the Cloud are used. While by the first data transmission to the Cloud, data are ciphered with asymmetric encryption, data for the second transmission are protected with symmetric encryption. Finally, a firewall (FW) serves as an additional farm protection shield, for example commands and firmware updates.

5.2 Hardware Implementation

The sensor nodes of the SED are both built-up with a Raspberry Pi [9] (Sensor 1) and an Arduino platform [10] (Sensor 2). The more powerful sensor node, Sensor 1, is built as a sandwich construction around a Raspberry Pi with a LoRa shield attached and equipped with an HSM on top. The use of two different add-on boards, sharing the same control lines, results in a problem in most cases. Thus, a physical separation of the chip select control lines on the Raspberry Pi board was necessary here, as shown in Fig. 8.

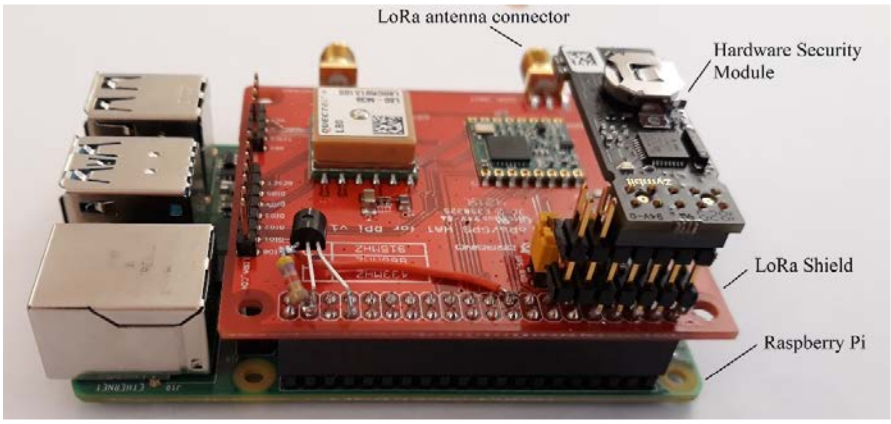

Fig. 8. Hardware implementation of the developed sensor

5.3 LoRaWAN Implementation

For the LoRaWAN software implementation, in C/C++ there are free available LoraMAC-in-C (LMIC) [11] software libraries useable for OTAA and ABP. In the proposed deployment, initially OTAA has been used, thus having three secret root keys before the activation is completed (AppKey, 64 bit; DevEUI, 64 bit; and ApPKey, 128 bit) and two additional secret session keys after a successful activation of the end-node in the LoRaWAN network (NwkSKey, 128 bit; and AppSKey, 128 bit).

In the free sample software codes, the root keys are hard-coded, meaning that, if the source code is stored in the end-device, an intruder can read these keys and use them in a duplicated device. If only the compiled code is on the end-device, these root keys are still in the code and can be extracted to use them in a duplicated device.

The session keys are generated during the activation process and are stored in RAM during the execution, and could be read out, too, but a much higher technical effort is needed.

5.4 HSM Integration

With the integration of a Zymkey HSM on the Raspberry Pi, the protection of the secret keys can be considerably increased. There are two possibilities for protecting keys with this module: (i) save the keys inside the module, or (ii) encrypt the secret keys using the HSM and store the encrypted keys on the Raspberry Pi. In both cases, the HSM must be firstly paired to the Raspberry Pi.

In the proposed deployment, the second method has been adopted, meaning the encryption of the three root keys. The decryption of these keys is only possible if there is an access to the HSM. The executing software has access to the HSM and the keys, but not a hacker without authentication. For the implementation of the encryption and decryption of secret keys, the HSM provides libraries for C/C++ and Python. With the C/C++ library, the key can be encrypted (locked) and decrypted (unlocked). The session keys are not encrypted in the first version, but in an advanced version these keys should also be encrypted or stored in the HSM and not in RAM.

An additional (and higher) protection of the end-device is performed by encrypting the overall file system of the Raspberry Pi on the SD card with LUKS. With the Zymkey HSM, the key for the encryption of the file system will be stored directly inside the module and can’t be read-out by an intruder.

For protection against end-device movement, the integrated GPS module of the LoRa shield has been used. In an outdoor scenario (like an agricultural environment), the GPS position will be measured at fixed time intervals. If the GPS position changed because the normally fixed sensor is moved, the software will block the wireless LoRa communication. The server should trigger an alarm if the communication of a sensor is lost for a longer time.

A physical security feature for the end-device is the module’s tamper detection, implemented by using two physical wire loops which are arranged around the inner side of the sensor case. A physical manipulation of the case will interrupt one of the loops, which triggers an alert, while the software continuously monitors the status of the two wire loops.

6 Results and Expectations in Agriculture Applications

The SED is currently under construction and already well-advanced. Many implemented functions already work to full satisfaction. The following four tests are planned and in preparation, in order to verify the extended security functionalities.

1. Move the end-device from the intended location.
   In this test, the sensor is exposed outdoor with the GPS function activated. If the sensor is moved more than 100 m (location change event), or if the GPS position is not available, then the sensor triggers an alarm and interrupt the communication.
2. Physical integrity of end-nodes.
   The sensor is packed in a case together with a battery pack. Two wire loops are connected to the HSM and placed in a way that the wire loop will be destroyed if the case is opened. By cutting the tab of the HSM the device will be “armed” (thus recognizing a “close-then-open” event on one or two of the perimeter wire loops). Then, the “zymkey_perimeter_event_action” parameter will be set to “self_destruct”. If the device will be opened, which is only possible by cutting one of the wire loops, the HSM will be irrevocable destroyed and the sensor cannot be used any more.
3. Reuse the hardware for manipulation of the agriculture environment.
   Once at least one of the wire loops is opened, the keys in the module are destroyed. No more access to the Raspberry Pi or the FS is possible after extracting the coin cell of the module and restarting the sensor. Then, it is not possible to read the SD card because it is encrypted with the (destroyed) keys of the HSM.
4. Manipulate the communication data.

> This test must be done before destroying the keys in the module. The LoRaWAN sensor is set-up and powered. The OTAA procedure is executed automatically and the sensor sends data to the LoRaWAN server. Without authentication to the Raspberry Pi it is not possible to read any root key or session key. The session keys are never transferred via the LoRa link but generated on the end-device and on the server side from other keys plus some parameters generated by the LoRaWAN server. It is not possible to extract the key from the communication.

Modern agriculture systems are more and more essentially software-driven automation systems, including farm management centres, data storages, powerfully edge computers for the interaction with the devices in the field domain. The field domain is in the most cases located in the open field, far away and outside of protected areas.

In the field there are several networked vehicles, which are supported by a continuously data stream of commands and control data for guidance and assistance, and a collection of different smart sensors, which register all relevant environmental parameters and data for decision finding. These smart sensors are becoming increasingly complex, supporting a great number of features. Hence, sensors not only measure the environment around, but also perform data pre-processing, data forwarding, battery monitoring, firmware update and many other functions [12]. These powerful mini-PCs in the field are very attractive for cyber-attacks in the future.

Today's field elements have already reached a high level of technical complexity and must consequently be protected in future applications, and it is hoped that security protection proposals and concepts, as described in this paper, will become more and more important.

7 Outlook

There is currently a need to define cyber-security guidelines for modern agriculture (Agriculture 4.0), such as those already developed for industrial control systems in the European Union (EU). While in the USA the United States Department of Homeland Security (DHS) has carried out research during the last years to identify potential cyber-security vulnerabilities for agriculture, in Europe, however, a similar investigation does not appear to have taken place. Authors in [13] focus on many industries to show the risks and the need of monitoring support to ensure cyber-security; but the modern agriculture domain is not included. Even in the EU publication [14] from Q4 2017, smart farming and cyber-security are not addressed.

Acknowledgments
This work has received funding from AFarCloud which is an ECSEL Joint Undertaking (JU) project under grant agreement No 783221. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and Austria, Belgium, Czech Republic, Finland, Germany, Greece, Italy, Latvia, Norway, Poland, Portugal, Spain, Sweden.

Parts of this work were funded by the Austrian Research Promotion Agency (FFG) and BMK (Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology).

The work of Luca Davoli is partially funded by the University of Parma, under “Iniziative di Sostegno alla Ricerca di Ateneo” program, “Multi-interface IoT sYstems for Multi-layer Information Processing” (MIoTYMIP) project.

References

1. Alexander Grunwald, Marco Schaarschmidt, Clemens Westerkamp (2019). LoRaWAN in rural context: Use cases and opportunities for agricultural businesses, Mobile Communication - Technologies and Applications; 24. ITG-Symposium, Osnabrueck, Germany. Accessed on 2020-06-03.
Online: https://ieeexplore.ieee.org/abstract/document/8731787
2. Jeetendra Shenoy. Yogesh Pingle (2016). IoT in Agriculture; 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India. Accessed on 2020-06-03
Online: https://ieeexplore.ieee.org/abstract/document/7724508.
3. LoRa Alliance. LoRaWAN 1.1 Specification. Accessed on 2020-05-02.
Online: http://lora-alliance.org/lorawan-for-developers.
4. B. Reynders and S. Pollin, "Chirp spread spectrum as a modulation technique for long range communication," 2016 Symposium on Communications and Vehicular Technologies (SCVT), Mons, 2016, pp. 1-5, doi: 10.1109/SCVT.2016.7797659.
5. Eldefrawy, Mohamed & Butun, Ismail & Pereira, Nuno & Gidlund, Mikael. Formal security analysis of LoRaWAN. Computer Networks, 2018, vol. 148, pp. 328-339. doi:10.1016/j.comnet.2018.11.017.
6. Naoui, Sarra & Elhdhili, Mohamed & Saidane, Leila. (2017). Trusted Third Party Based Key Management for Enhancing LoRaWAN Security. 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), Hammamet, 2017, pp. 1306-1313. doi: 10.1109/AICCSA.2017.73.
7. ISO/IEC 11889. Information technology — Trusted platform module library. Accessed on 2020-06-03. Online: https://www.iso.org/standard/66510.html.
8. Zymkey 4i, Hardware security module for Raspberry-Pi. Accessed on 2020-06-03 Online: https://www.zymbit.com/wp-content/uploads/2018/12/Zymbit-Data-Sheet-Zymkey-4i-DATA-SHEET-04100910A2.pdf.
9. Raspberry Pi, Single Board computer, developed in the United Kingdom by the Raspberry Pi Foundation. Accessed on 2020-06-03. Online: https://www.raspberrypi.org/.
10. Arduino platform, Open-source platform for single-board microcontroller kits. Accessed on 2020-06-03. Online: https://www.arduino.cc/.
11. LoraMAC libraries. A LoRaWAN end-device stack implementation. Accessed on 2020-06-03. Online: https://github.com/Lora-net/LoRaMac-node.

12. Codeluppi, Gaia, et al. (2020). LoRaFarM: A LoRaWAN-Based Smart Farming Modular IoT Architecture. Sensors, 2020, vol. 20, issue 7, no. 2028, pp. 1-24. doi: 10.3390/s20072028.
13. Nai-Fovino, Igor, et al. European Cybersecurity Centres of Expertise Map - Definitions and Taxonomy. doi:10.2760/622400.
14. Directorate-General for Agriculture and Rural Development (European Commission), ECORYS, Wageningen Economic Research. Study on risk management in EU agriculture. doi:10.2762/08778.