Facilitating a Secured Status Data Acquisition from Industrial Equipment via NFC

Christian Lesjak[1], Thomas Ruprechter[1], Holger Bock[1], Josef Haid[1], Eugen Brenner[2]
Design Center Graz Infineon Technologies Austria AG, Austria
Institute for Technical Informatics Graz University of Technology[2], Austria

Abstract

The advent of initiatives like Industry 4.0 promises increased operational efficiency through smart services and interconnected devices. To enable smart maintenance services for today's and future industrial equipment, regular status information must be transmitted from device customers to maintenance service providers over the Internet. However, simply attaching an industrial device to the Internet often leads to a security and privacy nightmare. Transparency about when and what data is being transmitted is of crucial interest to a customer. During transport, data must be protected against modifications and disclosure. A maintainer requires trust in the data's origin and integrity.

In this paper, we propose ESTADO, a system that enables smart services by providing the necessary connectivity from industrial equipment to service providers for device state tracking. Our system design focuses on the migration of current devices and the security aspect. Using a non-permanent NFC based connection, connectivity is only established ad-hoc on customer demand, and any data transmission is fully transparent to a customer. We study our design through a prototype implementation using an Infineon security controller and evaluate the security, usability and deployment aspects of our solution. Finally, we apply the STRIDE threat modelling technique on our design to discover unmitigated threats.

1. Introduction

To increase operational efficiency, collaborative automation of industrial devices is a key goal of initiatives such as the Industrial Internet, Industry 4.0 or the Smart Manufacturing Leadership Coalition [1] [2] [3]. In the context of the ARROWHEAD project, we specifically address smart services for maintenance, repair and overhaul (MRO).

Smart maintenance services aim to increase operational efficiency by reducing unplanned equipment downtime through predictive and optimal scheduling of maintenance tasks [4]. To facilitate the timely undertaking of servicing activities, the maintenance provider must centrally collect and analyze information about the condition of its equipment, which is deployed and located at customer sites. In this paper we investigate and propose a secured data acquisition system for equipment condition monitoring to enable such smart services. Thus, a solution for the secure transport of status information about industrial equipment (located at a customer facility) to the equipment maintainer (often located in another country or continent) is required.

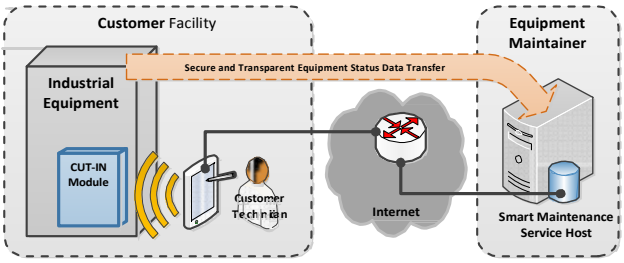

Figure 1. Equipment status data needs to be transmitted from the equipment deployed at customer site to the smart service at the equipment maintainer

The challenges in state of the art data acquisition systems are manifold. Wireless technologies promise advantages like reduced cost, faster deployment time and more flexibility [5]. Yet, a major issue to address is security. Not only the transport link from customer to maintainer needs to be properly protected, but also issues such as customer transparency about what data is being transmitted plays a critical role in sensitive industries. Furthermore, authenticity requirements on data received by a maintainer need to be fulfilled. Also, devices permanently connected to the Internet expose themselves to a great range of threats, therefore alternative approaches are of interest. But in order to propose such a system, the security requirements first need to be defined.

Figure 1 depicts the overall remote data acquisition scenario which facilitates smart

maintenance services by providing the necessary equipment status data to a central server. Our proposed system acquires this data from a so-called CUTIN module using a mobile NFC-enabled client, and then transmits the data to the central server at the equipment maintainer. Therefore, our paper makes the following contributions:

- We analyse and describe the requirements to enable smart maintenance services in Section III.
- We introduce and explain our novel concept ESTADO, which enables secured, transparent and ad-hoc status data transfer to online smart services in Section IV.
- We evaluate our pilot case implementation by means of a case study in Section V.
- We present a security analysis on our system design using the STRIDE threat modelling method in Sec- tion VI.

2. Background and Related Work

2.1. Near field communication (NFC)

NFC is a set of standards for wireless data and power transfer over very short distances up to 3-4 centimetres and based on inductive coupling [6]. A connection is established automatically when two NFC devices are brought into close proximity. Three characteristics substantially distinguish it from other wireless technologies such as WLAN, Bluetooth or ZigBee:

(1) Inherent proximity property (also known as "association by physical proximity"): On the one hand, communication with a fixed NFC endpoint is often used to infer location (or other contextual) information. On the other hand, holding a contactless card against an NFC reader proves possession and physical proximity to e.g., a door to be unlocked.

(2) Active initiation of a connection by a human operator: A connection is established by physically approaching another NFC target. In contrast, other wireless technologies, e.g., Bluetooth or Wi-Fi, may automatically re-establish a preconfigured connection as soon as the other device is in range of a communication partner (which can be up to a few hundred meters).

(3) Passive communication entities: Not all NFC devices require a dedicated power source. E.g., NFC tags and contactless cards are devices without a battery, and solely operate using the energy received from the other communication endpoint via the contactless link.

2.2. Data acquisition via NFC

Table I presents an overview of NFC based data acquisition systems in related work, which we discuss in this Section.

In [7] the "Smart NFC Interface" is presented, which provides a sensor interfacing module to read out data from a sensor via NFC. This NFC gateway is proposed as an industrial application to communicate with machines via mobile phones. However, the authors do not consider security aspects, and solely focus on the procedure of transferring data from their Smart NFC Interface to the smart phone.

In [8] the authors describe how to connect a sensor or programmable logic controller (PLC) via a mobile device to a back-end service. An NFC enabled phone is proposed as the link between sensor and service, providing both, a short range NFC, and a far range wireless interface. A comparison of back-end data links is given. Yet, security is only considered by mentioning the possible use of encryption.

[9] proposes to simplify and speed up error prone, manual monitor and control tasks. Therefore NFC is proposed for data acquisition from a measurement or sensor device. An NFC enabled mobile phone acts as the intermediary device to a remote central data acquisition server, where the data is displayed in a more presentable form. Heart rate monitoring is given as a specific application of the proposed system.

[10] introduces a maintenance system based on NFC tags. The aim is to document maintenance tasks to control their fulfillment and making false claims by maintenance personnel easier to discover. Compared to NFC interfaces for sensors, the NFC tag is not connected to the system to be maintained. It is solely used to identify the actual point of maintenance where the technician is present, and to store a synchronized secret generated by the back-end server. Any data collected during a maintenance task must be manually input into the NFC enabled mobile phone, in order to be transferred to the back-end. The proposed security measure using synchronized secrets provides only limited security, but can be deployed with simple memory NFC tags, which do not need to support cryptographic operations.

[11] studies the application of NFC to ultra-low power wireless sensors. The authors highlight the advantage of NFC compared to other wireless technologies which are usability, price, batteryless operation of the NFC device to be read and its short range. This reduces intentional or unintentional interferences. Again, the "Smart NFC Interface" is presented, which can be used to equip sensors with an NFC interface, or which can act as an NFC reader that connects via Bluetooth to another

mobile device without NFC.

In conclusion, the related work on data acquisition from devices via NFC, using a mobile client, has been proposed many times. However, the main focus of research was to propose new use cases, explain system and interface designs, or discuss its usability. As far as we understand, no sufficient considerations of security aspects have been given.

2.3. Smart services

In [12], smart services are characterized as proactive instead of reactive service actions. This means a service action is triggered *before* actual demand for a service arises. To offer such preventive, or more generally pre-emptive, servicing, the service provider needs to be aware of a customer's current equipment status.

The case study "From Legacy to Connectivity" [4] analyses the migration of industrial devices into the world of smart services by a case study on measurement equipment in the automotive industry. The authors discuss the challenges of implementing smart services for long-time established products with usually resource constrained embedded controllers. Besides this legacy aspect, connectivity, security and privacy/transparency issues are elaborated, e.g., the customer wants to know what data was transmitted when to whom. Furthermore, usability and efficiency are a key concern. A proposed mediator concept, similar to our CUTIN module, outlines a possible migration strategy.

Conclusively, smart maintenance services "are provided on field intelligence that is provided by technology either embedded in a product/equipment or facilitated by the use of devices, sensors or any other technology based tools" [13].

2.4. NFC interfaces for embedded systems

Generally, NFC interfaces into embedded system have been technically studied in [14], [15], and [16]. The case study in [17] presents a system to use NFC for device ID authentication based on digital certificates.

3. Requirements

From related work in [4] as well as from a security analysis and input from project partners, we derived the following the requirements in order to enable secure smart services.

R1: Support migration from legacy devices: Collect maintenance relevant data from industrial devices with different interfaces and protocols, and provide filter and aggregation functionality before sending this information.

R2: Prevent leakage of sensitive information from the industrial device, if this data is not required for providing smart maintenance services.

R3: Protect the equipment from access via the Internet on the newly established maintenance interface.

R4: Protect maintenance data against manipulation by the customer, who might want to circumvent the service contract with a maintainer.

R5: Transparency for customer: Certain industries use measurement devices that operate with confidential data, e.g., measurement results on yet to be released engines in the automotive environment. Therefore, it is in a customer's best interest that this confidential data is not transmitted to unauthorized entities. Thus, a customer needs transparency about what data is being sent to the maintainer.

R6: Protect the confidentiality of maintenance status data during transport from customer to maintainer to prevent information leakage about what or how many devices a customer uses, from which a competitor could gain valuable information.

R7: Origin authentication for maintenance data: the maintainer needs to trust that the received status data originates from the stated device identity to allow an automated process to analyse the data and finally to automatically infer future service actions from.

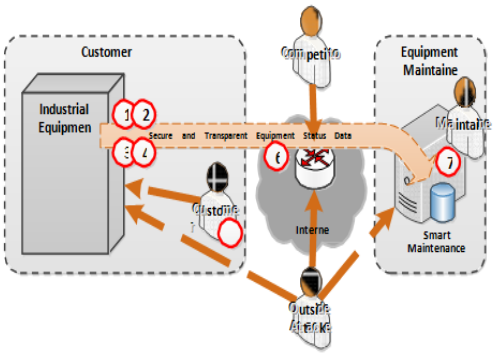

Figure 2. Seven requirements to enable secure smart maintenance services

4. ESTADO

In this Section, we describe the architecture and design details of our system, following the previously given requirements. A generic description of our concept is depicted in Figure 3. Three key ideas contribute to our novel idea:

Table 1. OVERVIEW OF RELATED WORK ON DATA ACQUISITION SYSTEMS USING NFC

Host	NFC interface	Reader	Security	Scenario/application	Back-end	Remarks
Ultra-low power sensors with NFC for mobile applications [11]:						
Ultra-low power sensors	Smart NFC Interface (NFC transmission module with SPI)	Mobile phone or Smart NFC Interface with Bluetooth as gateway to another Bluetooth device	None	Temperature sensor, energy consumption meter	None	Smart NFC Interface is a multi-purpose platform developed for evaluating NFC technology
A maintenance system based on NFC [10]:						
None (tag has only NFC interface)	NFC Forum Type 4 Tag (writable memory)	Mobile phone	Synchronized secret between server and tag	Central process control and documentation to track maintenance tasks.	Web services	-
Application scenario for NFC: mobile tool for industrial worker [7]:						
Sensors or machines in industrial environment	Smart NFC Interface (NFC transmission module with SPI)	Mobile phone or Smart NFC Interface	None	Industrial environments such as factories	None	Utilizes the Smart NFC Interface from [11]
A generic NFC-enabled measurement system for remote monitoring and control of client-side equipment [9]:						
General sensors and measurement devices	RFID tag with unknown link to host	Mobile phone	None	Heart rate monitoring (HRM)	Back-end server that displays or processes the data	Formulation of a concept for a complete user-friendly monitor and control system for general sensors and measurement devices
Using NFC-enabled phones for remote data acquisition and digital control [8]:						
Sensors or PLCs	No implementation details given	Mobile phone	Superficial discussion on potentially using shared, secret keys	Variety of sensors (medical, automotive, etc.)	Comparison of data connections (SMS, GPRS, HSPA, Wi-Fi, Bluetooth, Infrared, NFC)	No pilot case implemented

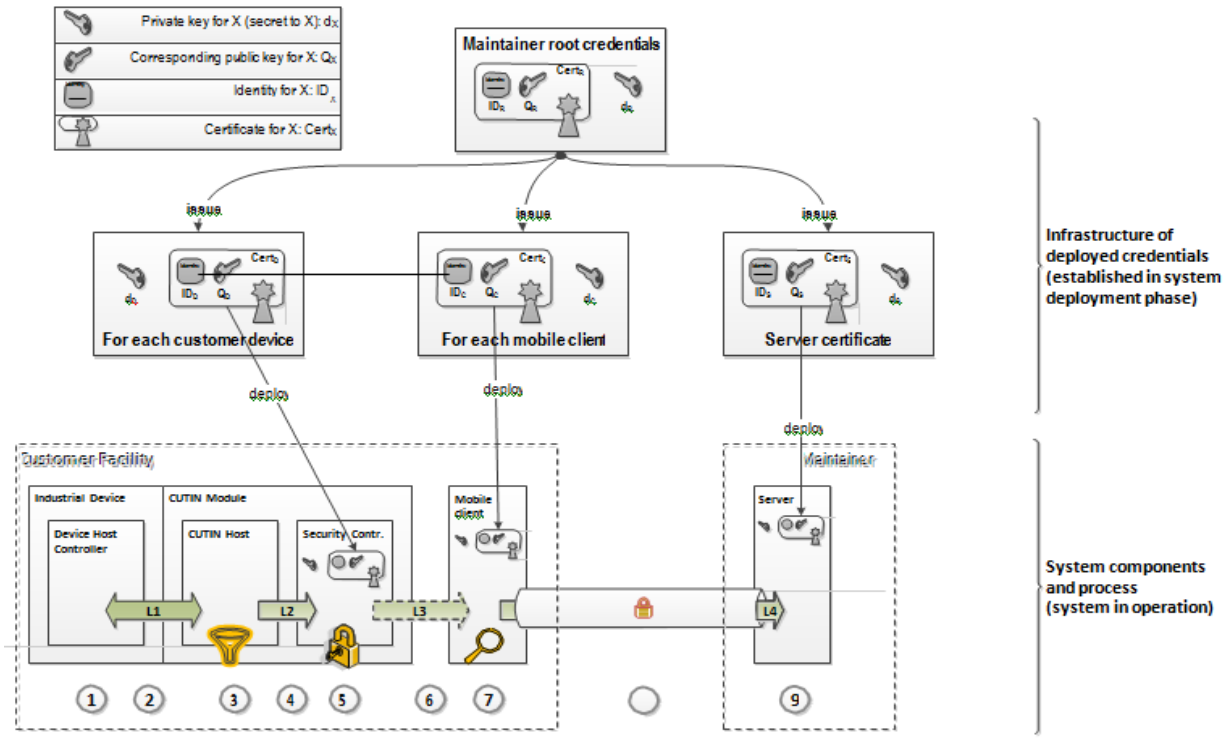

Figure 3. Illustration of system components, data flow (1-9) and the deployed security credentials including the certificate infrastructure

1) The concept of a dedicated module, the so called CUTIN module, allows the migration of current or legacy devices to the smart maintenance scenario. This module can be attached to any industrial device to enable it for smart services.
2) The split of the CUTIN module into two separate controllers provides both a powerful and interface- rich micro controller and a less powerful, but secure controller with NFC interface.
3) The NFC interface provides ad-hoc connectivity, and therefore substitutes a permanent Internet connection.

4.1. Data

The industrial device contains and generates different kinds of data. Static data like metadata (e.g., device ID, version data, etc.), firmware (e.g., operating system, software, etc.) and configuration is stored, and typically changed or updated during maintenance or configuration tasks. Dynamic data generated are operational data (e.g., generated measurement results, data required for process control, etc.) and maintenance status data (e.g., operating counters, switch rates, etc.). For the maintenance provider, only a subset of the metadata and the maintenance relevant data is of interest to fulfill its service. Therefore, we use two central data structures depicted in Figure 4:
A channel snapshot = (name, valu) represents the value of a specific device operating counter at a specific moment in time. Such a channel is a single, particular indicator for a device's maintenance status. Typically, a device has (far) more than one channel. A simple example for a channel is the total minutes of operation of a device since its last maintenance. Another example constitutes the value of a specific sensor that observes the condition of some component of the industrial device.

The device snapshot represents a device's maintenance state through a list of channel snapshots. For each channel, only the latest (most recent) channel value is stored. Further- more, a device snapshot is linked to a single device instance, indicated through the device ID field. Through a counter, all device snapshots can be chronologically sorted. A digital signature protects the content field by providing integrity, and origin authentication.

4.2. Procedure

1) The industrial device operates to fulfill its purpose in the customer plant, thereby generating two kinds of data: operational data, which is used by the customer, and maintenance status data, which is of interest to smart maintenance service provider.
2) The industrial device's host controller is permanently connected to the host controller of the CUTIN module via an interface supported by the industrial device.
3) Depending on the specific protocol used on L1, the CUTIN host regularly polls, or simply listens for data. It filters relevant information for maintenance, and possibly aggregates this information.
4) The CUTIN host regularly provides updates on channel snapshots to the security controller via another permanent connection.
5) The security controller stores this data in a secured non-volatile memory, and protects it using a digital signature, among other mechanisms, to provide authenticity and integrity.
6) The latest device snapshot can be read out at any time via a non-permanent, temporary NFC connection.
7) The mobile client displays the device snapshot to a customer technician, who is in charge of verifying the data as to whether it contains sensitive information from a company perspective. If not, he acknowledges the transfer of this data to the back-end of the maintainer.
8) A secure channel between the mobile client and the back-end is established, ensuring confidentiality of the data between the customer and the maintainer while moving through the Internet.
9) The server receives a device snapshot through the secure channel. By verifying the digital signature, it checks the snapshot integrity and origin, before storing and using this data for smart maintenance services.

4.3. Deployed security credentials

Figure 3 also illustrates an elementary certificate infrastructure to facilitate the cryptographic security measures. A maintainer root certificate CertR represents the root of trust. It may either be self-signed, or issued by a public root certificate authority (CA). During the deployment, this root certificate needs to be distributed to all mobile clients and the server for verifying certificate chains.

For each CUTIN module, a unique device identity IDD needs to be generated in order to later match device snapshots to actual devices. Therefore, each CUTIN security controller generates a private/public key pair, thereby securely storing the private key dD inside its secured memory. The corresponding public key Qd gets certified by the maintainers root private key and then deployed into the CUTIN security controller. The purpose of this credential is to provide authenticity and integrity for the device snapshot.

For the secured channel between mobile client and server, each mobile client and the server, need to generate their respective key pairs and get appropriate certificates issued and deployed.

In [17], a system to provision and verify industrial de- vice identities has been proposed. The authors differentiate a device's lifecycle into manufacturing, provisioning and operational phase. The generation and distribution of the aforementioned credentials takes place in the provisioning phase.

4.4. Detailed description of system components and links

The device host is the actual controller of the industrial equipment, required to fulfill its operational purpose at the customer (e.g., measurements in end of line (EoL) testing [4]). Depending on the kind of industrial device, this host controller can be a rather simple programmable logic controller (PLC), or as complex as a fully featured industrial PC. The device may also have further connections to some industrial control systems necessary to integrate the device into a customer's processes. As we assume such devices not to be reprogrammable (due to organizational or financial reasons), we intend to support the migration of these devices to smart services using a so called CUTIN module, which is permanently attached to an industrial device to enable it for smart services. In future devices, the components of this module can be integrated upfront.

Inside the CUTIN module, we combine an industrial micro controller (CUTIN host) with a security controller (CUTIN security controller). Through this design, we split the CUTIN module into two worlds: a powerful, general purpose controller for communication with the industrial device, and a secured, dedicated hardware module for protecting the acquired information.

The CUTIN host controller supports a variety of communication standards to acquire maintenance relevant status data from the device host controller via L1. L1 is a bidirectional communication link based on physical interfaces such as Ethernet, USB, a serial connection etc. On application level, any standardized or proprietary communication protocol support can be implemented in the CUTIN host, in order to filter and aggregate maintenance status data. The process might either follow a poll or a listen strategy, depending on the type of protocol to the device. To have a single CUTIN module support different kinds of (related) industrial devices, the CUTIN host might have access to a protocol description.

The CUTIN security controller enables a number of central security functions. Based on an Infineon security controller IC, it offers a trusted and tamper resistant platform to execute cryptographic operations. Its firmware is programmed and fixed, and protected from change or manipulation after deployment. Its secured credential storage contains the private key used for signing the device snapshot, which protects it against modifications, and provides origin authentication. The corresponding public key together with the device's ID is combined in a certificate issued to each individual device, and signed by the private key of the server. The device snapshot is updated each time a new channel snapshot is received from the CUTIN host. For each channel, only the latest (most recent) snapshot is saved in its secured memory. The link L2 can be any communication protocol supported by the security controller, such as I2C or SPI.

During a regular data acquisition procedure (e.g., weekly), a customer's employee visits all devices at a customer site. He connects to each CUTIN security controller using an NFC- enabled mobile client to retrieve the most current device snap- shot via the NFC interface (L3). The mobile client verifies the data's integrity and authenticity using the device's certificate containing the necessary public key for this device. The device certificate is further verified using the public key contained in the maintainer's trusted root certificate. The snapshot is displayed to the mobile client user, and optionally stored on the device. After the user has verified the data in respect to not containing customer sensitive information (e.g., measurement results), the user approves the snapshot to be transmitted to the maintainer's server. Therefore, the mobile client establishes a secure channel with the server over TLS, using the server's certificate. Client-side authentication is not necessary, as the data to be transmitted is already integrity and authenticity protected by the snapshot's signature, but has been added as an additional security layer.

The connection to the server (L4) can be any link over Internet, e.g., via a local WLAN access point or GSM. The data could also be cached for a limited time on the mobile client, to be transmitted after the technician has visited all relevant devices.

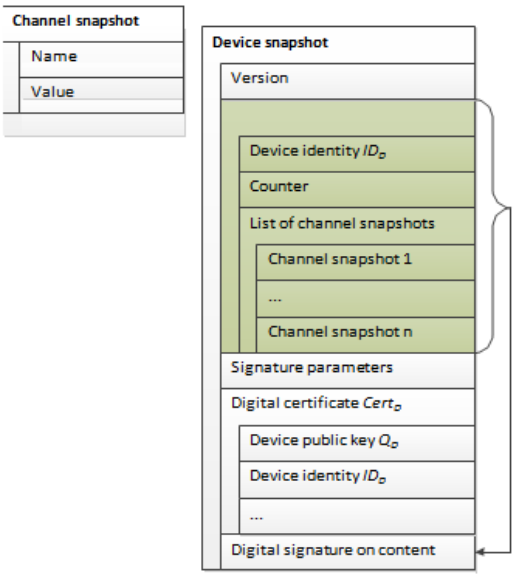

Figure 4. A device's current maintenance status is represented by a device snapshot which captures the device state at a specific time

The server receives the device snapshot via the secure channel. It verifies the integrity and authenticity using the digital signature. Via the device ID, it retrieves the associated digital certificate containing the device's individual public key. The snapshot is stored in its database, to eventually facilitate smart services such as

the prediction of future maintenance, repair and overhaul tasks.

4.5. Variation

The concept of a dedicated CUTIN module is proposed to facilitate the migration of today's devices to smart services. In future, newly developed industrial equipment, the central components of the CUTIN module (host and security controller), could also be directly part of the device.

Instead of using a secure channel between mobile client and maintainer's server, the device snapshot can also be encrypted already in the security controller. But this would no longer allow the verification of the data to be transmitted to be verified by the mobile client user (unless the device has decryption keys). We consider this system variant to be part of future research.

In the device snapshot depicted in Figure 4, the device certificate is labelled as optional. The certificate contains two essential elements: the device ID, which can also be embedded into the snapshot directly, and the public key for verifying the signature. Instead of extracting the public key from the certificate, it might be looked up from a database by using the device ID.

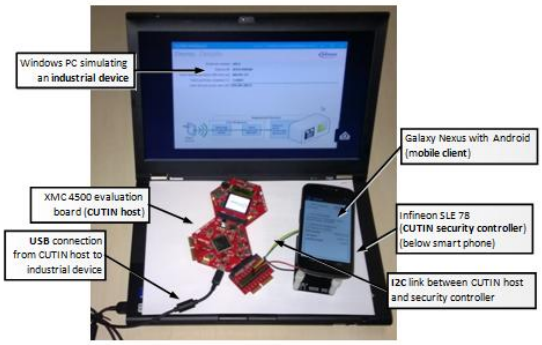

Figure 5. Illustration of our pilot case implementation comprising a Windows PC, an XMC 4500, a security controller and a mobile phone

To what extent the cryptographic parameters are contained inside a device snapshot is up to the system implementation. The information can be implicitly inferred from the version field (e.g., version X uses parameters A, B and C), or could be explicitly included in the snapshot. In practice, explicit parameter declaration might be in favor of a more open and flexible implementation.

The NFC interface does not currently require authentication of the mobile client. Any reader client implementing the communication protocol with the security controller may read out the maintenance data. Assuming physical access control already in place on customer sites, this might be sufficient. Yet, in future extensions of this work, only mobile clients that have authenticated against the security controller might be permitted to read out data.

5. Case Study

This Section explains implementation details of our pilot case and evaluates it in regard to the aspects proposed in [4].

5.1. Experimental prototype

Our pilot case implementation depicted in Figure 5 emphasizes the link from the device host controller to the mobile client, as we assume the implementation of a secure channel between the mobile client and a back-end server is a state-of-the art task and we do not consider this the focus of our research. Therefore, our pilot case focuses on the CUTIN module concept and retrieving the data via NFC.

CUTIN host and data fetch from device host: As host controller for the CUTIN module an Infineon XMC 4500 with evaluation board constitutes the more powerful and connection-rich maintenance data fetcher. It provides a wide range of interfaces, from which we use USB and Ethernet for our purposes, as it can fetch data from two types of device hosts. First, a Windows PC simulates an industrial device with randomly changing operating counters, from which data is acquired via USB. Second, an actual "AK protocol" implementation on the CUTIN host alternatively polls data from real world measurement devices. The host is connected via I2C to the security controller. In our implementation, a single channel snapshot can be transmitted at once. A snapshot is composed of two strings, one containing the channel name, and the other the channel value encoded as string.

Cryptographic parameters: The CUTIN security controller uses an EAL5+ certified Infineon security controller. Each time the security controller receives a new channel snapshot, it incorporates it into the device snapshot. If a channel name already exists, its value is updated accordingly instead. Then the counter inside the snapshot is incremented, and a new signature calculated. The fingerprint is stored in the non-volatile memory and ready to be read at any time via NFC. The device snapshot signature is calculated using the elliptic curve digital signature algorithm (ECDSA), using the secp192r1 curve together with a SHA-2 hash. The hardware for the CUTIN security controller is based on an EAL5+ [18] certified Infineon product, which provides secured code execution and data storage using a self checking dual CPU concept, integrity checks for data transfer and caches, encrypted memory and encrypted calculation in the CPU itself. Therefore, our hardware platform allows for secured storage of the necessary credentials, and the device snapshot.

Mobile client: As mobile client, we chose Android as

platform. Any NFC-enabled Android device with operating system version 4 or higher is supported. For NFC data exchange, we use the NFC Forum reader/writer mode [6], where the first NDEF message read indicates our protocol and is used to launch the application. Subsequent messages are used to exchange the actual device snapshot. For digital signature verification, we use SpongyCastle.

5.2. Deployability: legacy and connectivity

The deployment of our solutions has multiple aspects.

First, considering requirement R1, which demands support for legacy devices (without need for replacing them), we provide compatibility to these devices by design, through our CUTIN module concept. This module can be attached to any industrial equipment to connect it to smart services. Inside this module, the CUTIN host provides a number of communication interfaces and sufficient processing power to acquire, filter and aggregate maintenance data from various types of industrial devices.

Second, from a security perspective, each device requires a certificate including the device's ID and public key, and a corresponding private key, which are stored in the memory of the security controller in the CUTIN module. The deployment (provisioning) of these credentials has been discussed in [17].

5.3. Usability and efficiency

Compared to other wireless technologies such as WLAN or Bluetooth, NFC does not require explicit device pairing or a selection of which target to connect to. Due to its short range, the user of the mobile client implicitly selects the desired target device by bringing the mobile client in very close proximity. However, the amount of data transferable via NFC is limited by usability and transfer speed. Albeit its data rate on the air interface is 106 kbit/s, we observed actual transfer speeds of about 2 kB/s. This poses a limit on the amount of data that can be transferred within a time window that is still comfortable for the user of the mobile client, as the device must stay in practically the same position in order not to disconnect.

By using an NDEF message of a custom type (NFC Forum external type, [6]) on the security controller, the required smart phone application to read and verify the device snapshot can be automatically started, requiring no further, potentially time consuming, user interaction. In case of failure of the industrial device, or of the CUTIN host, the most recent device snapshot can still be acquired from the security controller, which can be solely powered from the energy provided via the NFC link wirelessly.

5.4. Security, privacy and transparency

On the one hand, the firmware programmed into the security controllers' memory is protected from being changed after deployment, by both authorized and malicious entities. The firmware is not specific to a certain industrial device type and model, but generic. On the other hand, the host controller of the CUTIN module is more flexible in order to support different devices. It might dynamically load a configuration to support a dedicated interface and protocol in order to communicate with, and acquire data from, different industrial devices. The limited interface of the security controller to the CUTIN host constitutes a first measure against unwanted data leaks. Furthermore, the NFC interface of the security controller only serves read-only data to the mobile client. The mobile client cannot write data to it, nor send any commands. As there is no other Internet connection to neither the CUTIN host nor the industrial device itself, the only connection from device as well as CUTIN host to the Internet leads through the security controller and its NFC interface. Thus, we effectively prevent any access into the industrial device from the Internet (cf. requirement R3). Furthermore, unintended leakage of non-maintenance status data (R2) is prevented on multiple stages. First, the limited interface between CUTIN host and security controller only allows the exchange of channel snapshots to (but not from) the security controller. Second, the customer can always check the data to be sent, before it leaves the mobile client, and potentially not confirm the transmission to the back-end. Therefore, the security controller acts as a firewall between the device (and CUTIN host) and the outer world.

Signing the device snapshot, which is transferred to the mobile client, protects any data leaving the security controller against modifications by the customer. Any signature verifier can detect both unintentional and intentional changes (R4).

The customer or one of its employees checks the data contained before the device snapshot leaves the mobile client (and thus the customer facility). This step makes the process of what data leaves the device fully transparent to the customer Furthermore, the customer has full control over when the data is transmitted - whenever a mobile client reads the data in very close proximity and the operator of the mobile client has inspected and approved the data to be sent (R5).

To protect customer privacy and the confidentiality (R6) of the maintenance status data on the transport to the back-end of the maintainer, a secure channel between mobile client and maintainer server is established. This secure channel is initiated using a pre-installed maintainer root certificate on the mobile client. Client authentication may be neglected, as the transmitted data is signed (and therefore integrity and authenticity protected).

Finally, the server verifies the signature of a

received device snapshot to detect modifications and verify the origin's authenticity. This allows the server to trust from which device a snapshot stems, and that no changes were made since the snapshot's generation.

6. Threat Modelling With Stride

In this section we further investigate the security of our system design using threat modelling. This is an analysis method at design level to uncover, evaluate and mitigate threats on a software system. The aim of threat modelling is to produce secure software by design, and it can be asset-driven, attacker-driven, developer-driven or software design-driven. STRIDE [19], [20], is a design-driven threat modelling process which is part of Microsoft's software development lifecycle (SDL), and it is based on either an available architectural design or the decomposition of an existing application. As a starting point, a visual representation of a software system using a data flow diagram (DFD) is generated. A DFD includes processes (DLLs, EXEs, components, services, web services, etc.), data stores (database, file, registry, etc.), data flows (function call, network traffic, etc.) and external entities (people, other systems, etc.). Furthermore, trust boundaries constitute points or surfaces where an attacker can interject, e.g., network connections, machine boundaries or between privilege levels. Based on the DFD and a set of analysis rules, STRIDE generates a list of potential threats. There are six threat categories, and each category relates to a distinct security property:

1) Spoofing: impersonating something or someone else, e.g., by pretending to be another user or the website of some bank. The security property against spoofing is authentication.
2) Tampering: modifying data or code, e.g., information sent via a network or files stored on a disk. Integrity mechanisms are necessary to prevent or detect tampering.
3) Repudiation: denying to have performed an action, e.g., not having signed a document. The desired security property is non-repudiation.
4) Information disclosure: unauthorized access to information, e.g., eavesdropping a private conversation. Mechanisms to establish confidentiality are required.
5) Denial of service: denying or degrading service availability, e.g., crashing a process. The related security property is availability.
6) Elevation of privilege: gain higher-ranked, unauthorized access to something, e.g., gain administrative rights as limited user. Authorization mechanisms are required.

In the following mitigation process, threats are addressed and alleviated by redesigning the system, by applying standard or new mitigations, by accepting the vulnerability by design, or by declaring a threat as not applicable.

6.1. Assumptions

Before starting the threat modelling process, we state a number of assumptions that apply to our system design.

As stated in Section IV-D, we introduce the concept of the CUTIN module to support legacy devices, as we cannot modify existing industrial equipment. Therefore, we cannot add any security measures to these legacy devices. An adversary with physical access to an industrial equipment could manipulate any maintenance relevant data this device generates and sends to the CUTIN host. Therefore, any data protection mechanisms are applied starting with the CUTIN security controller.

Furthermore, we chose a hardware security module which provides secured code execution and secured credential storage, making manipulations or key extraction practically quite hard. By that, we aim to eliminate most threads against this system component. Additionally, we assume that the firmware for this security controller will be developed using secure coding practices, and will be certified in order to attest the software's security.

6.2. Data flow diagram design

The data flow diagram depicted in Figure 6 models our system concept with data flows between processes. The diagram comprises three external entities. The industrial device and the CUTIN host are represented by a sole node data aggregation (CUTIN host), as both units provide no security measures and solely serve as the initial data source for channel updates. The external entity maintenance technician operates the mobile client. The data sink is the server, which finally stores all received device snapshots for further processing in the context of smart services.

The security controller trust boundary relates to the physical security controller entity. It comprises a credential store, a snapshot generation process and a snapshot store. The credential store holds the private key for signing device snapshots. The process snapshot generation, which receives channel updates from the CUTIN host, protects the data using a digital signature (among other mechanisms) and eventually stores the resulting device snapshot in the snapshot store.

The mobile client trust boundary relates to the mobile client operated by a human operator. Here, the gateway process reads data via NFC from a snapshot store, displays it to the user, and relays the data via a TLS channel to the server. In order to verify the snapshot, and to establish the secured TLS channel, cryptographic credentials are stored in the mobile client credential store. The maintenance technician initiates two essential data flows. First, the NFC touch data flow is the user action

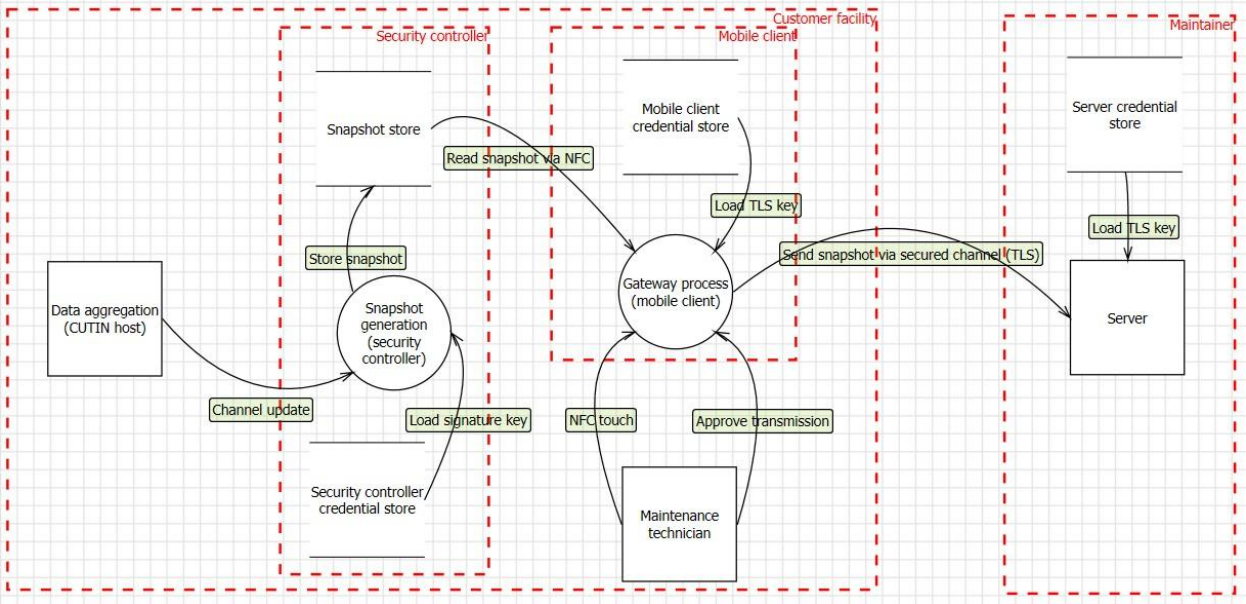

Figure 6. Data flow diagram of the ESTADO concept for the STRIDE analysis

of bringing the mobile client in close proximity of the security controller in order to retrieve the device snapshot via NFC. Second, the approve transmission data flow denotes the maintenance technicians explicit action to allow the device snapshot to be subsequently sent via the secured channel to the maintainer's server.

As both, mobile client and security controller, reside inside the customer facilities, they both also share the common trust boundary customer facility.

The remotely located server of the maintainer constitutes another dedicated trust boundary, which is accessible via the Internet. A credential store supplies the necessary cryptographic keys for snapshot signature verification and for establishing the secured TLS channel with the mobile client.

6.3. Threat identification and mitigation

Although the Microsoft Threat Modeling Tool 2014 allows to already specify certain security measures which were taken into account during system design (such as authenticated data flows), a total of 49 potential threats were identified by the tool. In this section, we will classify these threats into already mitigated ones, not applicable ones and threats that need further consideration. All threats discovered relate to one of the following nine data flows from the DFD:

Table 2. Data Flows in ESTADO

#	Data flow
(D1)	Channel update
(D2)	Load signature key
(D3)	Store snapshot
(D4)	NFC touch (user action that brings the mobile client in proximity of a security controller to retrieve its latest device snapshot)
(D5)	Read snapshot via NFC
(D6)	Approve transmission (user action)
(D7)	Load TLS key
(D8)	Send snapshot via secured channel (TLS)
(D9)	Load TLS key

1) Mitigated threats: From the total of 49 threats, we classified 23 as already mitigated by our design. Most of them are concerned with data flows inside the security controller trust domain (D2, D3). The security controller hardware already provides a secured execution environment as well as secured storage for data and credentials, therefore mitigating most threats there. Furthermore, as we assume it is rather small code base to be certified, we do not expect implementation errors leading to elevation of privilege threats from data flows D1 and D5. Also, data repudiation (D1), e.g., repudiation of a device snapshot, is mitigated by cryptographically signing the snapshot using a private key stored inside the security controller, therefore making it impossible to reject having signed a snapshot. Impersonation of a security controller (D5) is mitigated using the digital signature on the device snapshot, which provides authenticity using the secret, private signature key in the security controller.

2) Not applicable threats: We identified 10 threats as not applicable, mostly due to design decisions.

The interruption of data flow from technician to the gate

way process (D4, D6) is a user interaction and therefore an interruption would mean physically attacking the maintenance technician. A channel update data flow (D1) interruption, modification or sniffing is out of focus, as an attacker could access the industrial device directly on its unprotected interface. Spoofing of the snapshot generation or data aggregation processes (D1) is neglected, as the industrial device and CUTIN host do not implement any security mechanisms by design (legacy support). Interruption of the read snapshot via NFC data flow (D5) is easily detectable due to short distance of NFC, where an adversary would need to be directly next to the maintenance technician. Information disclosure from the snapshot store (D5), or spoofing of a gateway process in order to read out snapshots (D5) is not applicable, as the protection against unauthorized readout is by design the physical access control at customer facilities. And if this happens, an adversary could inspect the devices anyway. Tampering (D5) is prevented by the digital signature on the device snapshot.

a) Threats that need further investigation: For 16 threats, we do not explicitly provide mitigation mechanisms, yet their alleviation is a related to dedicated and comprehensive research field.

The security of the mobile client platform is dependent on its respective hardware and operating system. Different operating systems provide stronger levels of process isolation and against modification of its software. Hardware based security, such as the use of a trusted platform module (TPM) further increases the protection of credentials stored on mobile clients (D7). It is up to a system integrator to chose a right platform and configure it appropriately.

The implementation of secure applications, such as the gateway process, is subject to a secured development process, the programming language and execution environment, the selection of algorithms and protocols and much more [21].

To protect a user from impersonation by a malicious gateway process (D4, D6), a suitable mean of user authentication against the mobile client is necessary. A malicious gateway process should not be able to acquire a user's credentials.

On the server side, there are threats related to data flows inside the server (D9) and the protection of the credential store, as well as establishing the secured channel via TLS. They need to be addressed by an appropriate server platform, and the potential use of a hardware security module (HSM) to protect key material. Potential denial of service attacks against data flow D8 need to be alleviated by applying suitable DoS countermeasures.

6.4. Validation and conclusion

The thread modelling process using a data flow diagram and then identifying threads using STRIDE led to a potential number of 49 threats, of which 10 are not applicable and 23 were already mitigated. From the 16 threats left, they relate to four areas and need to be addressed not only on design, but also on implementation level. These areas are:

1) Mobile client platform security
2) Implementation aspects of the application on the mobile client
3) Secure user authentication methods of a user against the mobile client
4) Server-side security measures for availability and credential storage

We see these aspects as dedicated research works on their own. Our ESTADO concept addresses the protection of device snapshot data at device level, and the secured transport to a maintainer's central server.

7. Conclusion

In this paper we proposed how to enable smart service connectivity for current and future industrial equipment. We derived our requirements from a migration case study [4] and reviewed related work on NFC based data acquisition systems. Based on this, we designed ESTADO to enable secured, trans- parent and ad-hoc online data transmission to centrally collect device maintenance status data. We presented a pilot case implementation utilizing an Infineon security controller and an Android smart phone. An evaluation in respect to deployability, usability and security indicates a strong suitability for the illustrated smart service use case. Especially in industries with sensitive information, the transparent and ad-hoc aspect seems promising to us. An exhaustive threat modelling applied to our ESTADO design generated 49 threats using the STRIDE method. While most of them are not applicable or already mitigated by our design, 16 need further investigation in four dedicated research fields.

In future research we want to consider system variants discussed in Section IV, e.g., which require the mobile client to authenticate in order to read out the maintenance data. Furthermore, we want to address the four dedicated research fields to further increase overall system security.

8. Acknowledgement

The authors would like to thank the Austrian Federal Ministry for Transport, Innovation and Technology as well as the ARTEMIS Joint Undertaking, which funded the ARROW-HEAD project (file number ART-010000-2013-3) under the FP7-JTI programme.

9. References

[1] Industrial Internet Consortium, "Introductory white paper," IIC, White paper, March 2014, http://www.iiconsortium.org/docs/ (accessed on 2014-

04-01).
[2] H. Kagermann, W. Wahlster, and J. Helbi, "Deutschlands Zukunft als Produktionsstandort sichern – Umsetzungsempfehlungen fu¨r das Zukun- ftsprojekt Industrie 4.0. Abschlussbericht des Arbeitskreises Indus-trie 4.0," 2013, http://www.bmbf.de/pubRD/Umsetzungsempfehlungen\ Industrie4\ 0.pdf (accessed on 2014-11-01).
[3] Smart manufactoring leadership coalition, "About SMLC," 2014, https: //smartmanufacturingcoalition.org/about (accessed on 2014-04-01).
[4] P. Priller, A. Aldrian, and T. Ebner, "Case study: from legacy to connectivity - migrating industrial devices into the world of smart services," in Emerging Technologies and Factory Automation (ETFA), 2014 IEEE International Conference on. IEEE, 2014.
[5] J. Akerberg, M. Gidlund, and M. Bjorkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automa- tion," in Industrial Informatics (INDIN), 2011 9th IEEE International Conference on. IEEE, 2011, pp. 410–415.
[6] NFC Forum, "NFC Forum technical specifications," http://nfc-forum. org/ (accessed on 2014-09-01), 2014.
[7] M. Sallinen, E. Strommer, and A. Ylisaukko-oja, "Application scenario for NFC: mobile tool for industrial worker," in Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Confer- ence on. IEEE, 2008, pp. 586–591.
[8] C. A. Opperman and G. P. Hancke, "Using NFC-enabled phones for remote data acquisition and digital control," in AFRICON, 2011. IEEE, 2011, pp. 1–6.
[9] ——, "A generic NFC-enabled measurement system for remote moni- toring and control of client-side equipment," in Near Field Communi- cation (NFC), 2011 3rd International Workshop on. IEEE, 2011, pp. 44–49.
[10] S. Karpischek, F. Michahelles, A. Bereuter, and E. Fleisch, "A mainte- nance system based on near field communication," in 3rd International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST 2009), 2009, pp. 15–18.
[11] E. Stro¨mmer, M. Hillukkala, and A. Ylisaukko-oja, "Ultra-low power sensors with near field communication for mobile applications," in Wireless Sensor and Actor Networks. Springer, 2007, pp. 131–142.
[12] H. Aschbacher, "Framework fuer das agile Entwickeln von IKT- basierten Dienstleistungen unter Nutzung von Smart Services," Ph.D. dissertation, Graz University of Technology, 2014.
[13] M. Holgado and M. Macchi, "Exploring the role of E-maintenance for value creation in service provision," in Engineering, Technology and Innovation (ICE), 2014 International ICE Conference on. IEEE, 2014, pp. 1–10.
[14] N. Druml, M. Menghin, R. Basagic, C. Steger, R. Weiss, H. Bock, and
J. Haid, "NIZE – a near field communication interface enabling zero energy standby for everyday electronic devices," in Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on. IEEE, 2012, pp. 261–267.
[15] N. Druml, M. Menghin, C. Steger, H. Bock, and J. Haid, "A secure zero-energy NFC solution for everyday electronic devices," e & i Elektrotechnik und Informationstechnik, vol. 130, no. 7, pp. 224–229, 2013.
[16] M. Menghin, N. Druml, M. T. Fioriello, C. Steger, R. Weiss, H. Bock, and J. Haid, "PtNBridge – a power-aware and trustworthy near field communication bridge to embedded systems," in Digital System Design (DSD), 2013 Euromicro Conference on. IEEE, 2013, pp. 907–914.
[17] C. Lesjak, T. Ruprechter, J. Haid, H. Bock, and E. Brenner, "A secure hardware module and system concept for local and remote industrial embedded system identification," in Emerging Technologies and Factory Automation (ETFA), 2014 IEEE International Conference on. IEEE, 2014.
[18] T. Boswell, "Security evaluation and common criteria," in Secure Smart Embedded Devices, Platforms and Applications. Springer, 2014, pp. 407–427.
[19] S. Hernan, S. Lambert, T. Ostwald, and A. Shostack, "Threat modeling- uncover security design flaws using the STRIDE approach," MSDN Magazine-Louisville, pp. 68–75, 2006.
[20] Microsoft, "Introduction to Microsoft security development lifecycle (SDL) threat modeling," 2006, http://download.microsoft.com/ download/9/3/5/935520EC-D9E2-413E-BEA7-0B865A79B18C/ Introduction to Threat Modeling.ppsx (accessed on 2015-01-08).
[21] D. D. Hwang, P. Schaumont, K. Tiri, and I. Verbauwhede, "Securing embedded systems," IEEE SECURITY & PRIVACY, vol. 4, no. 2, pp. 0040–49, 2006

