SECURECONFIG: NFC and QR-Code based Hybrid Approach for Smart Sensor Configuration

Thomas Ulz, Thomas Pieber, Christian Steger
Institute for Technical Informatics
Graz University of Technology
Graz, Austria
{thomas.ulz, thomas.pieber, steger}@tugraz.at

Christian Lesjak, Holger Bock, Rainer Matischek
Development Center Graz
Infineon Technologies Austria AG
Graz, Austria
{christian.lesjak, holger.bock, rainer.matischek}@infineon.com

***Abstract*—In *smart factories* and *smart homes*, devices such as *smart sensors* are connected to the Internet. Independent of the context in which such a smart sensor is deployed, the possibility to change its configuration parameters in a secure way is essential. Existing solutions do provide only minimal security or do not allow to transfer arbitrary configuration data. In this paper, we present an NFC- and QR-code based configuration interface for smart sensors which improves the security and practicability of the configuration altering process while introducing as little overhead as possible. We present a protocol for configuration as well as a hardware extension including a dedicated security controller (SC) for smart sensors. For customers, no additional hardware other than a commercially available smartphone will be necessary which makes the proposed approach highly applicable for smart factory and smart home contexts alike.**

***Index Terms*—Near Field Communication; Internet of Things; smart sensor; configuration; security controller.**

I. INTRODUCTION

For *smart sensors* [1] that are connected to the Internet it is crucial that their configuration and firmware can be updated in a secured and efficient way. Such smart sensors can be deployed in a wide range of fields such as in a *smart factory* or in a *smart home*.

Smart Factory [2]: In smart factories it is essential to perform maintenance operations of sensors involved in the production prcoess. By introducing a secured and easy to use configuration interface, even untrained staff can perform firmware updates or configuration changes. However, it is very important to protect the confidentiality and authenticity of configuration data as employees applying the configuration updates could be potential adversaries. By enabling any employee or external person to perform configuration operations, the flexibility of the already deployed sensors will be increased while the associated maintenance costs will be decreased [3].

Smart Home [4]: Also in a smart home context, configuration and firmware updates for devices need to be performed using a secured and easy to use configuration interface. Devices not only include smart sensors but also other electronic devices such as WiFi routers. Similar to the smart factory use-case, also in a smart home context the configuration data must be secured against various attacks for sustaining the proper functionality of the configured devices. A configuration interface included into smart home devices enables any customer to perform firmware and configuration updates. These updates could, for instance, even be provided by a vendor's helpdesk.

By including such configuration interfaces into smart sensors, also the *Bring Your Own Key (BYOK)* principle [5] can easily be applied in both the smart factory and smart home context. BYOK would allow customers to change vendor supplied cryptographic keys, and thus, give them the certainty that no third party is able to access their data.

The approach presented in this paper not only is able to transfer cryptography keys but also arbitrary configuration data and firmware updates. To transfer data, NFC technology is chosen for three reasons. (i) NFC offers some security advantages compared to other wireless technologies [6]. Also, certain kinds of attacks such as man-in-the-middle are harder to conduct due to the limited communication range of NFC. (ii) The update process can be performed without an internal power source, if the necessary hardware is powered by the NFC field. (iii) NFC is easy and intuitive to use. Humans easily understand the principle of bringing one device near to another to transfer data [7].

If NFC is used to transfer data from a backend to a mobile device and from the mobile device to smart sensors, at least three NFC-enabled devices would be necessary. While smart sensor and mobile device must be equipped with an NFC interface in any case, needing an additional NFC-enabled device such as an NFC reader for the backend is inconvenient at least in a smart home context. Therefore, a combination of NFC and QR-Codes is used in the approach presented in this paper. The presented approach also relies on the functionalities provided by a security controller (SC). We propose to use a SC to protect the confidentiality and authenticity of configuration data that is stored on the SC. To the best knowledge of the authors, no other publication described a combination of these techniques to perform updates for smart sensors. The main contributions of this paper are:

1) The presented configuration approach allows arbitrary configuration data including firmware updates to be transferred in a secured manner.
2) The presented approach therefore is suitable for industrial as well as smart home use-case scenarios.

3) A hardware extension and cryptographic methods are presented that are applicable also for legacy sensors.

The remainder of this paper will be structured as follows. In Section II, background information on the used technologies as well as related work will be given. Section III presents the proposed NFC- and QR-Code based configuration interface for smart sensors. This interface will be evaluated with respect to security and the imposed overhead in Section IV. This paper will then be concluded with Section V where possible future work is stated as well.

II. Background and Related Work

A. Near Field Communication (NFC)

NFC is a contactless communication technology that is based on RFID standards. It operates at a radio frequency of 13.56 MHz, up to a range of approximately 10 cm with supported bit rates of 106, 212, 424 and 848 kbits per second (kbps). Also, NFC devices are compatible with many existing RFID devices and tags as the NFC standard comprises various RFID standards [8]. NFC is used in a diverse range of businesses. Today, the most well-known application of NFC is in the mobile payment sector [9]. Coskun et al. [10] note that NFC is also widely used in mobile ticketing applications. Another prominent field for NFC is the Internet of Things (IoT). Atzori et al. [11] state that *NFC [...] together with RFID [...] will link the real world with the digital world.*

B. Quick Response (QR) Code

A QR code is a two-dimensional code that offers various advantages over traditional (linear) barcodes such as a much higher data density or the possibility to read QR codes from all directions. The higher density allows a maximum capacity of 2953 bytes. Although the encryption of a QR code's content is possible, encrypted QR codes can be rarely found [12]. Therefore, Conde-Lagoa et al. [13] suggest to encrypt the content using symmetric cryptography. Soon [14] lists sample applications such as ticketing or identification of all sorts of items.

C. Security Controller (SC)

SCs are used to provide protected processing and storage of data. The key differentiator when compared to traditional processing units is the *tamper resistance* [15] of such SCs which even includes invasive attacks utilizing physical access to the hardware. However, as SCs are not as powerful as general-purpose controllers, splitting the execution environment into a *secured world* and into a *normal world* is suggested for instance by Vasudevan et al. [16]. This splitting principle, also called *security by isolation* or *dual-execution* [17], is realized by implementing SCs as external hardware modules.

D. Configuration via NFC

Configuring devices via an NFC interface is quite a novel topic. Wu et al. [18] discuss the possibility to reprogram computational RFIDs (CRFIDs) over the air. In their approach, the firmware of passive CRFID tags is reprogrammed using the Electronic Product Code (EPC) protocol. Haase et al. [19] present an NFC based configuration solution for sensors and actuators in the home automation context. The authors propose to extend existing hardware with an NFC module that can then be used as a configuration interface for standard smart phones. The authors present a hardware concept and prototype as well as a mobile application for Android smart phones. Serfass and Yoshigoe [20] present a framework for NFC communication in wireless sensor networks. This Android based framework, according to the authors, would allow P2P transport of arbitrary data. In contrast, a more widely used approach is to use NFC to speed up the pairing of wireless devices [21], [22]. A comparison of related work to the approach presented in this paper is given in Table I. As can be seen there, all but one approach do not consider security at all. The approach presented by Haase et al. [19] mentions encryption but only encrypts the data while it is transferred via NFC. The data, however, can be read and changed on the Android smartphone. Also, no solution to transfer the configuration data to the smartphone is given in that work.

TABLE I
Comparison with related work

Related Work	Necessary Hardware	Supported Payload	Security Considerations
[18]	RFID Card Reader, CRFIDs	Firmware Only	None
[19]	NFC-enabled Phone, and Sensor	Arbitrary Data	Encryption used except for initial update; No encryption on mobile device
[20]	At least 2 Android Devices for P2P	Arbitrary Data	None
[21], [22]	RFID tags, 2 NFC devices to pair	Pairing information	None
This Work	NFC-enabled phone and sensor	Arbitrary Data	Discussed in Section III

III. SecureConfig

Configuring smart sensors in a smart factory as well as in a smart home context is desirable. For a solution that is suitable for both contexts, a couple of requirements need to be fulfilled. To be usable in a smart factory context, a central instance that manages all active configurations is needed. In a smart home context, no additional hardware besides a mobile device and sensors should be necessary to make the proposed approach feasible for many users. Therefore, the system architecture shown in Fig. 1 is proposed. It comprises three components.

1) *Backend:* Configurations are created, updated and securely stored at the backend.
2) *Mobile Device:* The mobile device is used to transfer configuration data provided by the backend to the smart sensor. In our prototype we used a smartphone.
3) *Smart Sensor:* The smart sensor receives the provided configuration update.

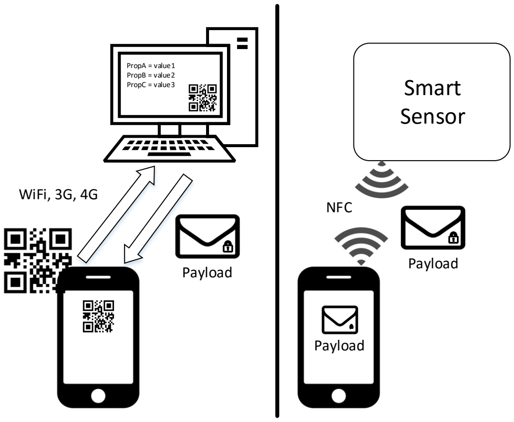

Fig. 1. Hybrid communication approach: On the left hand side, a configuration package is fetched from the backend using a QR code. On the right hand side, the configuration of a smart sensor using NFC is shown.

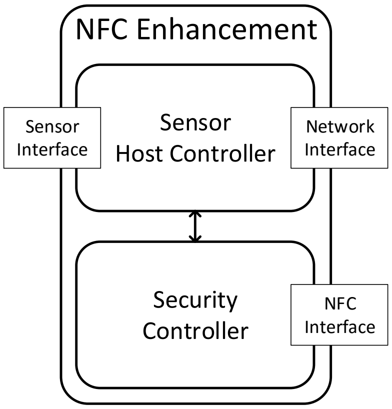

Fig. 2. NFC Enhancement component which can be connected to any conventional sensor via the *Sensor Interface* and thus making it a smart sensor.

In the context of this paper, it is assumed that the data stored on the *backend* is secured by appropriate security mechanisms such as a hardware security module (HSM), and thus, data stored at the backend is efficiently protected from loss or manipulation.

A. NFC Enhancement

To equip any arbitrary sensor with NFC capability and a SC, we propose a hardware component named *NFC Enhancement*. The component is shown in Fig. 2. As can be seen there, it comprises two controllers and various interfaces. The reasons for suggesting a dedicated NFC enhancement module are:

1) By designing a dedicated hardware module with an explicit interface to sensors, currently available (legacy) sensors can easily be transformed into a smart sensor. The NFC enhancement module can easily be offered as a single PCB which is easy to integrate for sensor vendors.
2) By including two controllers, responsibilities can be split perfectly according to the properties of both controllers. The sensor host controller provides interfaces to the sensor and optionally to a network while also offering computational power and memory for any kind of application. The less powerful but energy efficient SC on the other hand offers a secured execution environment and protected storage for configuration data as well as an NFC interface.
3) The NFC interface connected to the SC allows for ad-hoc connectivity instead of opening the configuration interface to a potential network connection. Also, the SC can be powered through the NFC field which allows for configuration updates independent of the sensor's and host controller's power supply.

B. Hybrid Communication Approach

As shown in Fig. 1, different technologies for data transfer are proposed in our approach. To transfer configuration data from the backend to the mobile device, QR codes are used. The configurations stored on the mobile device are then transferred to the smart sensor using NFC. The reasons for using this hybrid approach are:

1) By using QR codes to transfer configurations to the mobile device, no additional hardware (aside from the mobile device) such as an NFC reader is needed by customers. Configurations are imported by simply scanning the QR codes. This makes our approach especially suitable for smart home contexts while not limiting its usefulness in industrial contexts. Configurations could be printed for maintenance workers or displayed in web based configuration interfaces for customers.
2) NFC is suggested to transfer configuration data from the mobile device to the smart sensor. Reasons for using NFC for this data transmission are the additional security resulting from the limited communication range as well as the possibility to also configure sensors that are disconnected from their power supply. This also allows the initial configuration of sensors by the vendor during their assembly where no power supply is available. This property adds additional usefulness to our approach.

As a result of using two different technologies for data transfer, two separate data structures and methodologies need to be applied which are discussed for both variants.

C. Quick Response Code

Due to the size limitations of a QR code's maximum payload, two different modes for transferring configuration data to the mobile device are suggested.

1) The whole configuration payload is stored in the QR code, which allows to store about 2900 bytes of data. We denote this type as *inline* QR code. Inline QR codes do not require the mobile device to have an active network connection, thus, those codes can be distributed, for instance, to a maintenance worker without restrictions.
2) If the configuration data is larger than the size limit of 2900 bytes, only an URL pointing to the backend is

included in the QR code. The mobile device then needs to fetch the configuration data from the backend using a secure channel (TLS). This type is denoted as *URL* QR code. For the download process, the mobile device needs to have a network connection through which the backend can be reached.

The type that is used to transport configuration data however, does not solely depend on the configuration data's size. A second factor is the desired security level, as no active connection to the backend is needed for the *inline* type. Therefore, some of the security measures mentioned in Section III-E can not be applied.

D. Near Field Communication

To transfer configuration data via NFC, the *NFC Data Exchange Format (NDEF)* [23] that uses the NFC Forum reader/writer mode is used. NDEF abstracts the contactless communication and is supported by mobile platforms such as Android [24]. The proposed structure for NDEF packages is shown in Fig. 3. As can be seen there, various security related fields are included in addition to the (encrypted) configuration data.

E. Security Measures

To provide confidentiality, integrity, and authenticity of the transferred configuration data, *authenticated encryption (AE)* [25] is used. As can be seen in the NDEF packet structure shown in Fig. 3, the transferred packet comprises a couple of security related fields as well as the encrypted payload, and a MAC; the later two are calculated by applying AE. The AE method of operation considered as having the best security properties is *encrypt-then-MAC* [26] which is the reason why this approach is used in our work. When using AE it is also important to not use the same key for both encryption and hashing; therefore, a cryptographic key derivation [27] is applied to generate separate encryption and hashing keys from a master key.

In addition to the aforementioned cryptographic principles, additional information regarding the configuration data is included in the NDEF message (see Fig. 3). This information is used by the SC at the smart sensor to decide if a configuration update is rejected or accepted and consequently applied. As the confidentiality, integrity, and authenticity of this information also needs to be protected, all but two fields are included in the encryption process. The two unencrypted fields are:

- Realtime: The time in milliseconds since the mobile device was started.
- Cipher Spec: Specifies the applied cryptographic algorithms for the authenticated encryption.

The fields which are included in the encrypted payload are:

- Version: The configurations version number. A smart sensor will reject configuration updates with a configuration number less or equal to the currently applied configuration.
- Validity: If the transmitted *realtime* is later than the specified *validity*, a configuration update will be rejected.

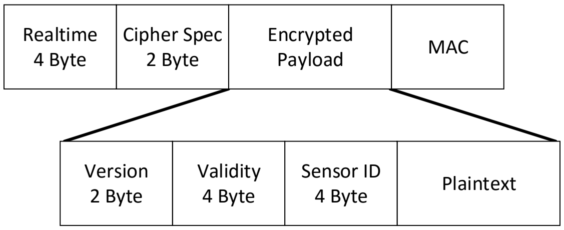

Fig. 3. NDEF packet structure. *Realtime* and *Cipher Specs* are transferred unencrypted. The size of the attached MAC depends on the cipher specs.

- Sensor ID: If the specified sensor ID does not match, the configuration update is rejected.

As there is no time synchronization between the backend and the smart sensor, the process of verifying the configuration's *validity* needs to be discussed in detail. Whenever a configuration is fetched from the backend, the following steps are performed:

1) For each configuration, a validity period Δ needs to be specified at the backend.
2) The mobile device sends a request to the backend, containing the current realtime ϑ.
3) Upon encrypting the configuration data, the included *validity* $\nu = \vartheta + \Delta$ is calculated.
4) The encrypted configuration data is sent to the mobile device.

For our approach to function properly, we assume a secured time source in the mobile device. In the case of an *inline* QR code, no connection to the backend is established; therefore, no validity can be specified for the included configuration data. Due to this, the *inline* mode needs to be considered as less secure than the *URL* mode.

IV. Evaluation

A prototype was realized to evaluate the feasibility, usability, and functionality of the presented approach. This prototype, pictured in Fig. 4, contains the following components:

1) *Sensor:* An air pressure sensor is used in this prototype to demonstrate the configuration update process.
2) *NFC Enhancement:* The NFC enhancement prototype that was realized is based on a concept presented by Lesjak et al. [3] that uses an Infineon XMC4500 microcontroller (Cortex M4 family) as the general purpose controller. This controller offers connection interfaces such as USB and Ethernet, as well as I2C. Via this I2C interface, a common criteria [28] EAL5+ certified SC by Infineon is connected to the XMC4500. This SC provides security features such as secured data storage and code execution by using a self-checking dual CPU concept, integrity checks for data transfers and caches, and encrypted memory and calculations in the CPU. Furthermore, this SC also includes a contactless interface capable of NFC communication. The NFC antenna is integrated into the NFC enhancement module as well.

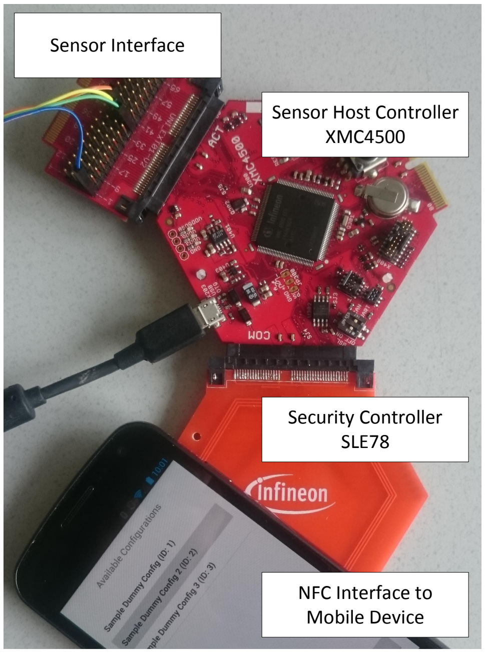

Fig. 4. NFC enhancement prototype.

3) *Mobile Device:* A Nexus S smartphone was used as NFC-enabled device in the presented prototype. On this device, Android 4.1.2 Jelly Bean was installed to use the latest NDEF functionality included with API level 14.
4) *Backend:* The backend was realized on a standard Windows PC in this prototype.

This prototype then was used to measure the time necessary for an update process. A configuration update containing 64 byte of data, for example, took roughly 200ms on average which is similar to the time a TLS handshake would need on such hardware.

A. Threat Analysis

To demonstrate the achieved security level, a threat analysis which highlights **Entities (E)**, **Assets (A)**, **Threats (T)**, applied **Countermeasures (C)**, and **Residual Risks (R)** is conducted. Due to the higher security offered by the *URL* QR code, this mode is discussed in this threat analysis. An overview of the threat analysis in *goal structure notation (GSN)* is shown in Fig. 5. The attack possibilities that are analysed are the smart sensor interface as well as the mobile device which is seen as a data channel. The backend is assumed to be properly secured by measures such as an appropriate firewall and a HSM, the SC at the smart sensor is assumed to be certified to the security level EAL5+ according to the common criteria. The assets that need to be protected are **configuration data (A1)** and **sensor functionality (A2)**. Threats can be posted by the **NFC enhancement vendor (E1)**, **customer (E2)**, **mobile device user (E3)** and an **external adversary (E4)**.

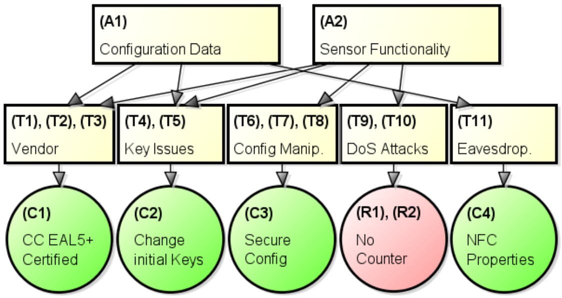

Fig. 5. Overview of threat analysis in GSN.

Threats resulting from **intentional or unintentional backdoors (T1)**, **weak cryptographic algorithms (T2)** and **bugs in cryptographic algorithms (T3)** by the **vendor (E1)** are investigated in the **common criteria EAL5+ certification process (C1)** for the included SC. The initial encryption keys specified for each SC could be **lost in a security breach (T4)** or even **disclosed in any form (T5)** by the **vendor (E1)**. This can be mitigated by **changing the initial key (C2)** as part of a configuration by the **customer (E2)**. Any malicious **mobile device user (E3)** could try to **manipulate configuration data (T6)**, **try to apply outdated configurations (T7)** or **try to apply configurations to wrong sensors (T8)**. **The presented security measures (C3)**, however, provide efficient mitigation of these threats. If the person responsible to update configurations **(E3) does not apply the configuration at all (T9)**, a potential denial of service attack results if the sensor's functionality is influenced by the missing configuration. There currently is no security measure implemented to **counteract missing updates (R1)**. If the malicious **mobile device user (E3)** or an **adversary (E4)** with physical access to the sensor continuously tries to change a configuration which is rejected by the SC, a possible **DoS attack (T10)** could result. There is currently no security measure implemented against this kind of attack **(R2)**. Attacks that **passively try to eavesdrop (T11)** configuration data are efficiently mitigated by the **implemented security measures (C3)** and the **security features of NFC (C4)**.

B. Overhead

The overhead resulting from the implemented security measures can be split into a static and into a variable part. The static overhead, resulting from the information added to the encrypted configuration data and MAC, can easily be calculated by summing up all fields with specified sizes in Fig. 3. The resulting static overhead is $O_{static} = 16$ bytes. The variable overhead depends on the chosen cryptographic algorithms. For this evaluation, *HMAC-SHA256* is assumed as hashing algorithm which adds an additional overhead of $O_{variable} = 32$ bytes. The resulting total overhead in that case would be $O = O_{static} + O_{variable} = 48$ bytes. An overview of the overhead relative to the configuration data size up to 4 kB of data is shown in Fig. 6. As can be seen there, when transferring configuration data of about 300 bytes, less than 15% of the transferred data will be security imposed overhead.

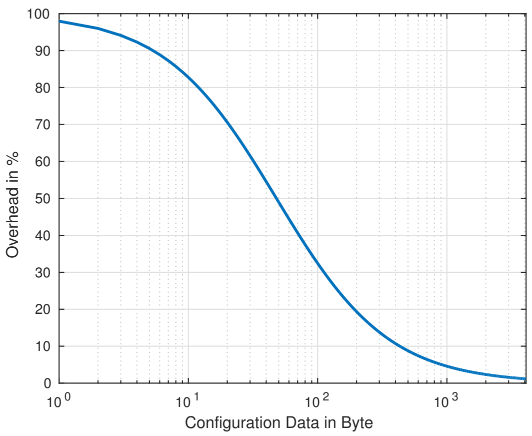

Fig. 6. Overhead in percent relative to transferred configuration data.

V. Conclusion and Future Work

In this paper we present an NFC- and QR-code based hybrid configuration approach for smart sensors which is suitable for smart factory and smart home use cases. To provide the necessary functionality for sensors, an *NFC enhancement* module is presented. To mitigate potential security challenges imposed by such an additional configuration interface, appropriate security measures are included in our approach. It is also shown that by including those security measures, an acceptable amount of overhead is imposed. The feasibility of our approach is demonstrated as a prototype which is presented in this work. As future work we plan to include a password based key exchange protocol such as SPAKE [29] to require user authentication when applying updates. Authenticated users could then read configuration parameters from a smart sensor, directly modify them on their mobile device and update the smart sensor's configuration.

Acknowledgment

This project has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 692480. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and Germany, Netherlands, Spain, Austria, Belgium, Slovakia.

References

[1] G. Meijer, K. Makinwa, and M. Pertijs, *Smart Sensor Systems: Emerging Technologies and Applications*. John Wiley & Sons, 2014.

[2] D. Lucke, C. Constantinescu, and E. Westkämper, "Smart Factory - A Step towards the Next Generation of Manufacturing," in *Manufacturing Systems and Technologies for the New Frontier*. Springer, 2008, pp. 115–118.

[3] C. Lesjak, T. Ruprechter, H. Bock, J. Haid, and E. Brenner, "Facilitating a Secured Status Data Acquisition from Industrial Equipment via NFC," *Journal of Internet Technology and Secured Transactions (JITST)*, 2014.

[4] R. Harper, *Inside the Smart Home*. Springer Science & Business Media, 2006.

[5] H. Zhang, "Bring your own encryption: balancing security with practicality," *Network Security*, vol. 2015, no. 1, pp. 18–20, 2015.

[6] E. Haselsteiner and K. Breitfuß, "Security in Near Field Communication (NFC)," in *Workshop on RFID security*, 2006, pp. 12–14.

[7] D. López-de Ipiña, J. I. Vazquez, and I. Jamardo, "Touch Computing: Simplifying Human to Environment Interaction through NFC Technology," *1as Jornadas Científicas sobre RFID*, vol. 21, 2007.

[8] G. Van Damme, K. Wouters, and B. Preneel, "Practical Experiences with NFC Security on mobile Phones," *Proceedings of the RFIDSec*, vol. 9, 2009.

[9] J. Ondrus and Y. Pigneur, "An Assessment of NFC for Future Mobile Payment Systems," in *Management of Mobile Business, 2007. ICMB 2007. International Conference on the*. IEEE, 2007.

[10] V. Coskun, B. Ozdenizci, and K. Ok, "A Survey on Near Field Communication (NFC) Technology," *Wireless personal communications*, vol. 71, no. 3, pp. 2259–2294, 2013.

[11] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[12] V. Sharma, "A Study of Malicious QR Codes," *International Journal of Computational Intelligence and Information Security*, vol. 3, no. 5, pp. 21–26, 2012.

[13] D. Conde-Lagoa, E. Costa-Montenegro, F. Gonzalez-Castao, and F. Gil-Castieira, "Secure eTickets Based on QR-Codes with User-Encrypted Content," in *2010 Digest of Technical Papers International Conference on Consumer Electronics (ICCE)*, 2010.

[14] T. J. Soon, "QR Code," *Synthesis Journal*, vol. 2008, pp. 59–78, 2008.

[15] S. Ravi, A. Raghunathan, and S. Chakradhar, "Tamper Resistance Mechanisms for Secure Embedded Systems," in *VLSI Design, 2004. Proceedings. 17th International Conference on*. IEEE, 2004, pp. 605–611.

[16] A. Vasudevan, E. Owusu, Z. Zhou, J. Newsome, and J. M. McCune, "Trustworthy Execution on Mobile Devices: What Security Properties Can My Mobile Platform Give Me?" in *International Conference on Trust and Trustworthy Computing*. Springer, 2012, pp. 159–178.

[17] M. Sabt, M. Achemlal, and A. Bouabdallah, "The Dual-Execution-Environment Approach: Analysis and Comparative Evaluation," in *IFIP International Information Security Conference*. Springer, 2015, pp. 557–570.

[18] D. Wu, M. J. Hussain, S. Li, and L. Lu, "R2: Over-the-Air Reprogramming on Computational RFIDs," in *RFID (RFID), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–8.

[19] J. Haase, D. Meyer, M. Eckert, and B. Klauer, "Wireless sensor/actuator device configuration by NFC," in *2016 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2016, pp. 1336–1340.

[20] D. Serfass and K. Yoshigoe, "Wireless Sensor Networks Using Android Virtual Devices and Near Field Communication Peer-To-Peer Emulation," in *Southeastcon, 2012 Proceedings of IEEE*. IEEE, 2012, pp. 1–6.

[21] A. Matos, D. Romao, and P. Trezentos, "Secure Hotspot Authentication through a Near Field Communication Side-Channel," in *2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2012, pp. 807–814.

[22] M. Jung, J. G. Park, J. H. Kim, and J. Cho, "Interoperability between Medical Devices using Near Field Communication," in *2013 International Conference on Information Science and Applications (ICISA)*. IEEE, 2013, pp. 1–4.

[23] "NFC Data Exchange Format (NDEF)," NFC Forum, Tech. Rep. NDEF 1.0, 07 2006.

[24] A. Lotito and D. Mazzocchi, "OPEN-NPP: an open source library to enable P2P over NFC," in *Near Field Communication (NFC), 2012 4th International Workshop on*. IEEE, 2012, pp. 57–62.

[25] M. Bellare and C. Namprempre, "Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2000, pp. 531–545.

[26] H. Krawczyk, "The Order of Encryption and Authentication for Protecting Communications (or: How Secure Is SSL?))," in *Annual International Cryptology Conference*. Springer, 2001, pp. 310–331.

[27] ——, "Cryptographic Extraction and Key Derivation: The HKDF Scheme," in *Annual Cryptology Conference*. Springer, 2010, pp. 631–648.

[28] D. Mellado, E. Fernández-Medina, and M. Piattini, "A common criteria based security requirements engineering process for the development of secure information systems," *Computer standards & interfaces*, vol. 29, no. 2, pp. 244–253, 2007.

[29] M. Abdalla and D. Pointcheval, "Simple Password-Based Encrypted Key Exchange Protocols," in *Cryptographers Track at the RSA Conference*. Springer, 2005, pp. 191–208.