Rights Management with NFC Smartphones and Electronic ID Cards: A Proof of Concept for Modern Car Sharing*

Timo Kasper, Alexander Kühn, David Oswald, Christian Zenger, Christof Paar

Horst Görtz Institute for IT Security, Ruhr-University Bochum, Germany
{Timo.Kasper,Alexander.Kuehn,David.Oswald,Christian.Zenger,Christof.Paar}@rub.de

Abstract. Numerous contactless smartcards (and the corresponding RFID readers) are compatible with NFC, e.g., Mifare cards and the governmental ID card in Germany called nPA. NFC-enabled smartphones and other NFC objects such as door locks have become widespread. Existing and future applications of the up-and-coming technology require a secure way of assigning and transporting user rights, e.g., for opening and starting a car or access control to a building. In this paper, we propose a scheme that securely identifies a customer on a website and creates a (personalized) credential containing the booked access permissions. This credential is safely transported via the Internet to the user's smartphone and finally grants access to an NFC-enabled object. In our proof-of-concept implementation, an application on a commercial smartphone is used for communicating with a web server of a car rental agency. During the booking process, the phone operates as an RFID reader to interrogate the nPA of the user and utilizes the security mechanisms of the nPA, including the PACE protocol, for identifying the customer. After having obtained the credential, the smartphone emulates a Mifare DESFire card that is read by the NFC door lock of a rental car to verify the validity of the access permission. We discuss security issues and limitations of our approach.

Keywords: German electronic identity card, user rights management, car sharing, smartphone, NFC, contactless smartcard emulation

1 Motivation

The ISO 14443 standard for contactless smartcards [32] was published around 2000. Today, many applications for ticketing, micro payments, access control, and identification rely on contactless cards that are compliant to this standard, e.g., NXP's Mifare family of cards, electronic IDentification (ID) cards, and passports. In these applications, the roles of the actively powered reader interrogating passive cards are clearly defined. A few years later, Near Field Communication (NFC) [33] emerged from ISO 14443 and is today implemented in many embedded devices, such as smartphones, door locks, and other objects. NFC is compatible to ISO 14443, but enables additional features: For instance, an NFC-enabled object can choose to appear as a reader in one moment and switch its role to appear as a contactless card in the next moment. Furthermore, an operating mode with an extended range is defined, whereas both participants function as active readers communicating with each other.

* This work was supported in part by the German Federal Ministry of Economics and Technology (Grant 01ME12025 SecMobil).

Major manufacturers of smartphones, e.g., Nokia, HTC, LG, and Samsung, offer NFC-enabled smartphones. The NFC-link is designed to operate at a distance of only a few centimeters and provides an additional communication interface to the ones already provided by the smartphone, e.g., Wireless Local Area Network (WLAN), Universal Mobile Telecommunications System (UMTS), Global System for Mobile Communications (GSM), Global Positioning System (GPS), or Bluetooth. Furthermore, the main processor of the smartphone can perform complex computations and can be employed to interconnect the communication interfaces of the smartphone.

Due to the compatibility with existing ISO 14443 Radio Frequency IDentification (RFID) readers and infrastructure, various other objects are already equipped with NFC technology, e.g., locks for hotel rooms and safe deposits, payment terminals, and ticketing solutions. Likewise, electronic passports and governmental ID cards, e.g., in Germany, as well as future electronic driver's licenses can be accessed with NFC. Recently, even cars — often already augmented with GPS receivers and a GSM or UMTS modem — additionally provide an NFC-enabled door lock. Thus, an excellent basis is given for realizing various new applications that combine services on the Internet with contactless cards and other NFC objects.

Outsourcing business and administrative tasks to the Internet potentially provides a significant gain regarding costs for companies. Therefore, the verification of a user's identity without a face-to-face meeting becomes more and more important. Modern car sharing solutions are emerging, e.g., in Germany DriveNow by BMW, Car2Go by Daimler, and Flinkster by Deutsche Bahn. Fleet management of vehicles, access to a safe deposit or post box, and opening doors of previously booked hotel rooms are some more possible business options that can be realized by means of NFC-enabled smartphones with Internet access: The issuer transfers an electronic key (or another right) via the Internet to the smartphone. Once stored on the smartphone, the electronic key can repeatedly be used to access a particular secured object via NFC.

The required components such as NFC-enabled locks and smartphones are available. However, for example for billing purposes, the identity of the customer must be verified securely. This is where governmental electronic IDs come into play, since they usually provide methods for an NFC-based, remote identification of the owner via the Internet. In this context, the protection of the individual's identity is of particular relevance. The new German electronic identity card (nPA) is designed to provide the required features, i.e., a mechanism for identification, authentication, and the protection of stored personal data, e.g., the identity of the owner. With the nPA, it is possible to identify an individual in passport controls, at airports, at governmental authorities, and on the Internet.

1.1 Contribution and Outline

Due to the complexity of the involved communication interfaces and technologies, for some system designers it is unclear how to address the security risks of NFC and how to realize the corresponding schemes for identifying the customer and transporting the booked rights from the issuer to the target object. Due to lack of publicly disclosed realizations of the mentioned applications, it is often uncertain, if according systems are practically feasible at all.

In this paper we aim to fill in this gap. To this end, in Sect. 3, we propose a scheme for *(i)* identifying a customer by combining the NFC interface and the Internet connection of the smartphone with the security capabilities of an electronic ID card, *(ii)* issuing a credential containing the rights assigned by the issuer to the customer and securely transferring the credential to the smartphone, *(iii)* assuring that the rights specified in the credential can be verified by the target object, e.g.,

an NFC-enabled door lock of a car, even in a scenario without a permanent Internet connection, and finally *(iv)* securely exchanging the credential via an NFC connection between the smartphone and the target object and decide whether to grant or deny access. As an alternative scenario for using the NFC phone, users shall be able to obtain user rights and execute granted rights on the basis of a standard contactless smartcard, i.e., compatibility to commercially available contacless cards shall be maintained, where possible.

As the main contribution, in Sect. 4, we illustrate the practical feasibility of the proposed scheme by realizing all relevant parts of the scheme as a proof-of-concept on a commercial smartphone and validating our implementation. We practically demonstrate that the smartphone is suitable for using the electronic identity (eID) function of modern governmental documents and realize the corresponding cryptographic schemes in Sect. 4.3, including the required computations on specific elliptic curves. We further implement the 3DES-based authentication scheme of Mifare DESFire cards in Sect. 4.4, and show that emulating a DESFire card with a commercial NFC smartphone is practical. In Sect. 4.5, we pinpoint implementation obstacles and finally discuss security issues in Sect. 5. In the following Sect. 2, the relevant fundamentals of the nPA, NFC, and Secure Elements (SEs) are described.

2 Fundamentals

In this section, we present the eID service of the nPA that provides a method for the online identification of customers. For the secure transmission of personal data, the Password Authenticated Connection Establishment (PACE) and Extended Access Control (EAC) security protocols of the nPA are described. Furthermore, important aspects of SEs and NFC technology are shown.

2.1 The German Electronic Identity Card (nPA)

With the introduction of the nPA in Germany on November 1st, 2010, a new system for the administration of electronic identities has been established. The so-called eID service is a service supported by the nPA for verifying the correctness of an electronic identity on the Internet. For using the secure authentication on the Internet, e.g., to open a bank account, for online shopping, or administrative tasks, an nPA, and a Personal Computer (PC) with an RFID reader [18] and Internet access are required. For verifying the owner's identity, a multi-factor authentication based on *knowing* a Personal Identification Number (PIN) and *owning* the nPA is employed [19].

eID Service The eID service [17] of the nPA enables to verify the identity of the passport owner. As a trust anchor, an eID server, e.g., operated by Bundesdruckerei GmbH in Germany, participates in the identification process. The required personal customer data, stored on the nPA and signed by the Federal Ministry of the Interior of the German Federal Republic, is redirected to the service provider by the eID service. For securing the personal data, various cryptographic mechanisms, e.g., Advanced Encryption Standard (AES), Elliptic Curve Cryptography (ECC), RSA, and SHA-1 are implemented on the nPA.

The communication between an nPA, an RFID reader, and the participating eID server is secured by the PACE protocol and the EAC protocol. For validating the authenticity of the personal nPA data, its signature has to be verified with the corresponding public key. The authenticity of this public key can in turn be verified with a certificate issued by a Root Certificate Authority (RCA) belonging to the Public Key Infrastructure (PKI) [9] of the Federal Office for Information Security [19].

Password Authenticated Connection Establishment The PACE protocol is the base security protocol of the nPA. It establishes a secured communication channel between the nPA and the RFID reader using the Diffie-Hellman Key Exchange (DHKE) or the Elliptic Curve Diffie-Hellman Key Exchange (ECDHKE). For a successful execution of the protocol, a PIN, e.g., the number printed on the ID card or an individual, secret eID key only known to the nPA holder, has to be entered.

During the PACE protocol, two shared session keys are generated, namely an encryption key $\mathtt{K}_{\mathtt{enc}}$ and a key $\mathtt{K}_{\mathtt{MAC}}$ for authentication with a Cipher-based MAC (CMAC) [14]. After a successful execution of the PACE protocol, mutual authentication between RFID reader and nPA is established. More information about the PACE protocol can be found in [20,31].

Extended Access Control The EAC protocol relies on a successful PACE execution and uses the generated key pairs. EAC consists of two parts, the Terminal Authentication (TA) and the Chip Authentication (CA). The TA uses a challenge-response protocol for enabling the nPA to verify the authentication of the terminal and to define the terminal's access permissions to the personal data. In the CA, the nPA proves its authenticity and the correctness of the stored data to the terminal. While the PACE protocol establishes a secure communication channel between the nPA and the RFID reader, the EAC protocol secures the communication from end to end between the nPA and a remote server. The Secure Messaging (SM) of the EAC employs newly generated session keys of the current protocol run, as further detailed in [20,21].

2.2 Operating Modes of Near Field Communication

NFC is a wireless communication technology, defined in the ISO/IEC 18092 standard [33] and is compatible to ISO/IEC 14443, i.e., based on RFID communication at a frequency of 13.56 MHz. The standard specifies three modes of operation in which NFC can be communicate:

Reader mode (active) An NFC-enabled device in active mode (Proximity Coupling Device (PCD)) can access a passive NFC tag, e.g., a contactless card (Proximity Integrated Circuit Card (PICC)) or an NFC device in passive mode.

Card emulation mode (passive) An NFC-enabled device in passive mode emulates a contactless smartcard and thus acts as a PICC.

Peer-to-peer mode Two NFC-enabled reader devices both operate in active mode (as PCDs), i.e., actively transmit their data.

The first two operation modes are used in our concept and its implementation: The NFC smartphone acts in the reader mode for the communication with the nPA, while the card emulation mode is used for the communication with the NFC-enabled door lock of the car.

2.3 Secure Elements

An attacker can gain unauthorized access to the Operating System (OS), e.g., by means of a Trojan horse, and get hold of secret data. Thus, all data held in the OS of the smartphone and all computations carried out by the main processor can be possibly read out by an attacker using malware. Thus, for securely storing (small amounts of) security-sensitive data and executing security-related algorithms with low computational demands an SE should be used: Besides tamper-proof volatile

and non-volatile memories, it contains a secure microprocessor (often with an 8-bit architecture) that usually runs a Javacard OS. Three distinct types of SEs are common in NFC smartphones:

1. an embedded Secure Element (eSE) that is realized by the manufacturer of the smartphone, either as a separate chip connected to the NFC circuitry or directly integrated into the NFC chip itself
2. a Subscriber Identity Module (SIM) card issued by the mobile communications provider
3. an SE embedded into an (micro)SD card can be used, if the smartphone provides a corresponding slot

For using the SE, access must be granted to the developer, which often poses a problem in practice. Maybe, trusted zones in main processors, such as the ARM TrustZone [45], constitute an alternative to SE: TrustZone is a software-based security solution that is used in mobile devices with integrated ARM processor for securing the access to confidential data and secure online transactions.

3 Concept of Right Management

In this section, our concept for the secure management of user rights for NFC applications is presented in the context of a car sharing scenario. After introducing the basic idea, the involved participants and the concept goals, the three major parts of our concept, registration, credential request, and credential usage, are illustrated in detail.

3.1 Introduction to Mobile Rights Management

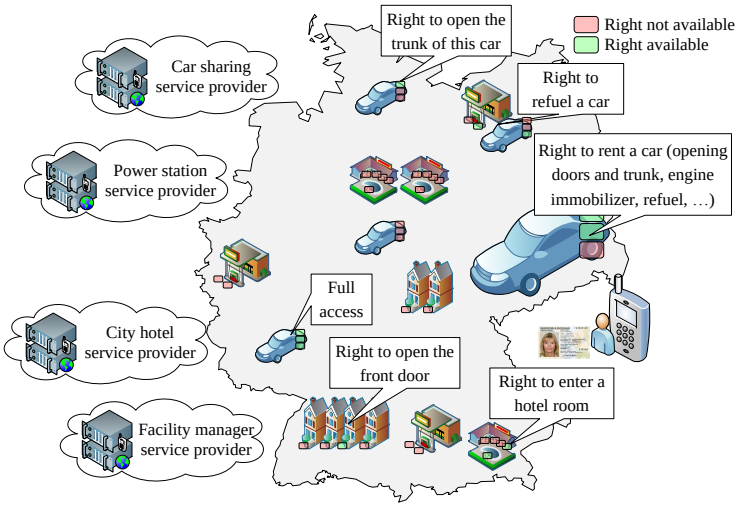

Fig. 1: Examples for actions and corresponding rights that can be assigned by our proposed scheme

In the first quarter of 2013, 210 million smartphones were sold worldwide [25]. The more than one billion smartphones currently in use can thus serve as an economical, flexible, and mobile alternative to existing infrastructures. A key advantage of

NFC-enabled smartphones is that the mobile service provider does not have to hand out additional (electronic) tokens, but can instead provide a software-application that can be downloaded to the user's phone.

We present our concept for the secure managing of user rights exemplarily for mobile car sharing. As shown in Fig. 1, our approach is not limited to this single use-case, but can rather be applied to various other applications in which a smartphone user is required to securely obtain credentials. In our scheme, a credential can contain any kind of rights, e.g., the right to access a building or to refill the battery of an electronic vehicle at a charging station.

3.2 Participants

Our proposed concept distinguishes three participants. *Customers* want to use mobile services. *Service providers* are — besides providing the services — responsible for the distribution of different permissions for using the services, i.e., manage the credentials. The *trusted eID service*, e.g., the Bundesdruckerei GmbH, ensures the identity of the customer to the service providers.

The customer possesses an NFC-enabled smartphone with Internet access and an nPA. She knows the secret PIN for the eID application of the nPA. She wants to flexibly use different services on the go, for example, renting a car or booking a hotel room without face-to-face meetings with the vendor.

The service provider offers a smartphone application that serves as a reservation system and gives access to other service-related information to the customer. Further, the application contains the public keys of the service provider that can be used to secure the communication. The service provider operates an Internet server for the issuance and administration of credentials. Each customer who wants to use one of the services, e.g., rent a specific car at a specific time, has to send a request via the smartphone application to obtain a valid credential for this service. Thus, the task of the service provider is to administrate the generation of service permissions on request and securely distribute them to the correct customer's smartphone.

3.3 Registration

Before using a car sharing service, customers have to be registered once for billing purposes and to ensure that the customer possesses a valid driver's license. For that reason, in current realistic scenarios the customer has to visit one of the service provider's shops and present her driver's license[1].

For the registration, the customer has to enter a secure, confidential password that is used for the generation of an individual customer public key pair, consisting of a public key $\mathtt{pk_C}$, a secret key $\mathtt{sk_C}$, and a certificate $\mathtt{cert_{pkC}}$ for the public key of the customer $\mathtt{pk_C}$. This data is transferred to the SE. The public keys $\mathtt{pk_C}$ of all customers are stored in a data base of the service provider in combination with the Machine Readable Zone (MRZ) of the customer's nPA for binding the public key pair to the nPA. Furthermore, the smartphone stores $\mathtt{pk_{SP}}$.

Likewise, each NFC service object securely creates and stores an individual public key pair ($\mathtt{pk_{SO}}$, $\mathtt{sk_{SO}}$), the public key of the service provider $\mathtt{pk_{SP}}$, and unique service object information. The latter is also stored, together with $\mathtt{pk_{SO}}$, in the database of the service provider. Each service object and the service provider know and can securely execute a key derivation function, using an `Access Variable`, the service object information, and the public key $\mathtt{pk_{SO}}$ as inputs, for generating an `Authentication Key` (see Sect. 3.5).

[1] In future scenarios in which an electronic driver's license is available, this visit could probably be omitted, as described in Sect. 3.7.

3.4 Booking a Right

For obtaining the right for a particular service via the Internet, a request has to be issued to the service provider. Afterwards, the corresponding credential needs to be created and transferred to the smartphone of the customer. All communication via the Internet is secured with the Transport Layer Security (TLS) protocol [13]. The booking process consists of four main steps, as illustrated in Fig. 2.

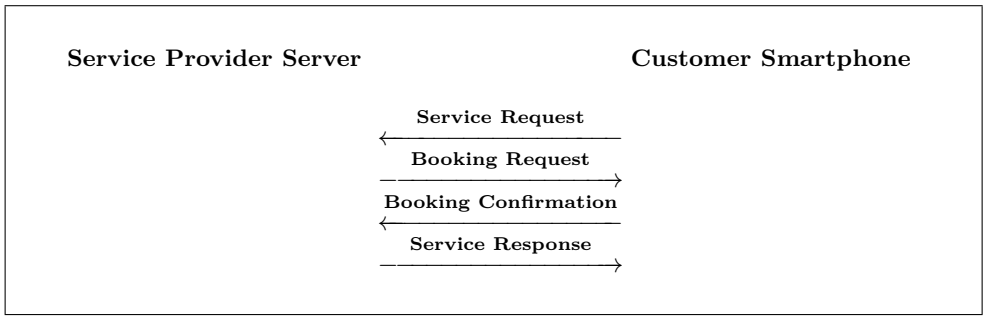

Fig. 2: Exchanged messages between service provider server and customer smartphone

Service Request If the customer wants to use a service, e.g., book a rental car as close as possible to her current position, she starts the application on her smartphone that sends a request containing service-relevant information $\mathtt{I}_{\mathtt{SReq}}$, e.g., the GPS coordinates of the smartphone and the desired time of departure, to the service provider. In addition to the service information, the request contains the customer certificate $\mathtt{cert}_{\mathtt{pkC}}$. A random nonce $\mathtt{N}_{\mathtt{C}}$ and a timestamp $\mathtt{ts}_{\mathtt{SReq}}$, both generated on the SE, are added as security values.

The service information $\mathtt{I}_{\mathtt{SReq}}$ and the security values are concatenated and hashed by a secure hash function, e.g., SHA-512 [3,16]. Afterwards, the cryptographic security token $\mathtt{t}_{\mathtt{SReq}}$ is generated signing the hash value with the customer's private key $\mathtt{sk}_{\mathtt{C}}$ on the SE of the smartphone. The nonce $\mathtt{N}_{\mathtt{C}}$ is stored on the SE. The service information, the security values, the the customer certificate $\mathtt{cert}_{\mathtt{pkC}}$, and the cryptographic security token are concatenated and encrypted with the public key $\mathtt{pk}_{\mathtt{SP}}$ of the service provider. The encrypted data packet $\mathtt{p}_{\mathtt{SReq}}$ is sent as the service request to the service provider. In summary, the following computations are executed:

$$\mathtt{h}_{\mathtt{SReq}} := \mathrm{hash}(\mathtt{I}_{\mathtt{SReq}} \,||\, \mathtt{N}_{\mathtt{C}} \,||\, \mathtt{ts}_{\mathtt{SReq}})$$
$$\mathtt{t}_{\mathtt{SReq}} := \mathrm{sign}_{\mathtt{sk}_{\mathtt{C}}}(\mathtt{h}_{\mathtt{SReq}})$$
$$\mathtt{p}_{\mathtt{SReq}} := \mathrm{encrypt}_{\mathtt{pk}_{\mathtt{SP}}}(\mathtt{I}_{\mathtt{SReq}} \,||\, \mathtt{N}_{\mathtt{C}} \,||\, \mathtt{ts}_{\mathtt{SReq}} \,||\, \mathtt{cert}_{\mathtt{pkC}} \,||\, \mathtt{t}_{\mathtt{SReq}})$$

After receiving the service request, the service provider decrypts it with his secret key $\mathtt{sk}_{\mathtt{SP}}$. The authenticity is ensured by means of the security values and the cryptographic token: The signature is verified with the customer's public key $\mathtt{pk}_{\mathtt{C}}$ that authenticity can be validated using the certificate $\mathtt{cert}_{\mathtt{pkC}}$. The integrity of the received data packet is checked by computing the hash value of the received service information and security values and comparing it with the received hash value. The freshness of the received service request is guaranteed by means of the nonce and the timestamp $\mathtt{ts}_{\mathtt{SReq}}$. For the time verification by verifying the timestamps and the

execution of user rights at the granted time period, the clocks of all participants have to be synchronous. If an error occurs, e.g., because the timestamp is out of time, a signed and encrypted error message is sent to the customer smartphone and the service request is rejected.

Identity Verification The service provider uses the eID functionality of the nPA, described in Sect. 2.1, for reading out data of the customer's nPA and verifying the customer's identity, as illustrated in Fig. 3.

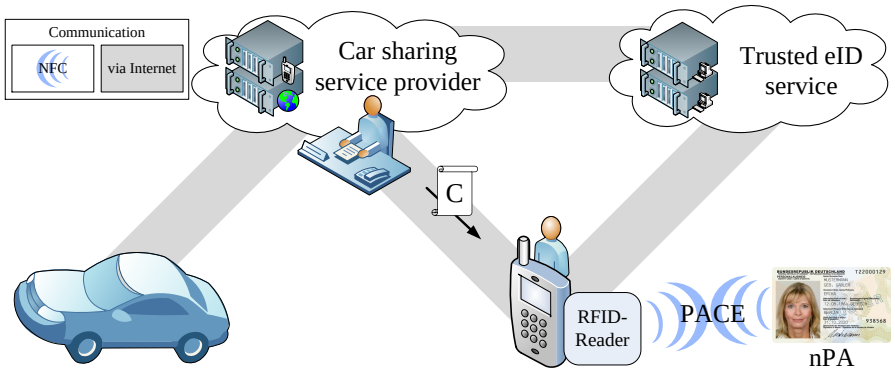

Fig. 3: Using the eID function of an electronic identity card (nPA) for booking a car and obtaining a credential $\mathtt{C}$ containing the assigned right: The smartphone acts as an RFID reader and communicates with the nPA using the PACE protocol. The authenticity of the nPA and the identity of the customer is then securely checked via Internet with the help of a trusted eID service and forwarded to the car sharing service provider. Furthermore, certain security parameters of the cars in the field can updated on a regular basis, e.g., via GSM or UMTS.

The NFC communication between nPA and smartphone is secured by means of the PACE protocol, during which the eID PIN — only known by the customer — has to be entered. The communication between smartphone, trusted eID service, and the service provider is secured by the TLS protocol. If the identity verification fails, a signed and encrypted error message is sent to the customer smartphone and the service request is rejected.

Service Credential Generation and Service Response After a successful authentication and identification of the customer, the service provider modifies the nonce $\mathtt{N_C}$ to $\mathtt{N_C'}$ in a defined way, which is also known to the SE in the smartphone. Afterwards, a booking request is generated by concatenating the service information $\mathtt{I_{BReq}}$ (e.g., coordinates of the car and the `User Right Validity Period`), the unique service object information $\mathtt{UI_{BReq}}$ (e.g., an unique car ID), the modified nonce $\mathtt{N_C'}$, and the timestamp $\mathtt{ts_{BReq}}$. This data is hashed and the hash value is signed with the private key $\mathtt{sk_{SP}}$ of the service provider. The data and the signed hash value are again concatenated with the customer certificate $\mathtt{cert_{pkC}}$, encrypted with the customer's public key $\mathtt{pk_C}$, and sent as a booking request $\mathtt{p_{BReq}}$ to the smartphone

of the customer over a TLS-secured communication channel.

$$\mathtt{h_{BReq}} := \mathrm{hash}(\mathtt{I_{BReq}} \,||\, \mathtt{UI_{BReq}} \,||\, \mathtt{N_C'} \,||\, \mathtt{ts_{BReq}})$$
$$\mathtt{t_{BReq}} := \mathrm{sign}_{\mathtt{sk_{SP}}}(\mathtt{h_{BReq}})$$
$$\mathtt{p_{BReq}} := \mathrm{encrypt}_{\mathtt{pk_C}}(\mathtt{I_{BReq}} \,||\, \mathtt{UI_{BReq}} \,||\, \mathtt{N_C'} \,||\, \mathtt{ts_{BReq}} \,||\, \mathtt{cert_{pkC}} \,||\, \mathtt{t_{BReq}})$$

The customer smartphone decrypts the received booking request on the SE using the stored $\mathtt{sk_C}$. Then, the signature and the hash value are verified. The received modified nonce $\mathtt{N_C'}$ is re-calculated on the basis of the original nonce $\mathtt{N_C}$ stored on the SE to ensure that the booking request of the provider belongs to the corresponding service request of the smartphone. The contained timestamp proves the freshness of the booking request. In case of an error, a signed and encrypted error message is sent to the service provider and the booking request of the service provider will be sent a second time. If the second try also fails, the current service request is canceled and a new service request has to be sent.

If the verification of data authenticity and integrity was successful, the received service information is displayed to the customer and in case of correctness and customer acceptance, the customer finally confirms the displayed booking offer by sending of a booking confirmation to the service provider. The SE of the customer smartphone modifies the received modified nonce $\mathtt{N_C'}$, again in a predefined way known to both the phone and the provider to obtain $\mathtt{N_C"}$. Then, the booking confirmation is generated with the service information $\mathtt{I_{BReq}}$, the unique service object information $\mathtt{UI_{BReq}}$, the modified nonce $\mathtt{N_C"}$ and the timestamp $\mathtt{ts_{BCon}}$ that is hashed, signed, encrypted, and sent back to the service provider.

$$\mathtt{h_{BCon}} := \mathrm{hash}(\mathtt{I_{BReq}} \,||\, \mathtt{UI_{BReq}} \,||\, \mathtt{N_C"} \,||\, \mathtt{ts_{BCon}})$$
$$\mathtt{t_{BCon}} := \mathrm{sign}_{\mathtt{sk_C}}(\mathtt{h_{BCon}})$$
$$\mathtt{p_{BCon}} := \mathrm{encrypt}_{\mathtt{pk_{SP}}}(\mathtt{I_{BReq}} \,||\, \mathtt{UI_{BReq}} \,||\, \mathtt{N_C"} \,||\, \mathtt{ts_{BCon}} \,||\, \mathtt{t_{BCon}})$$

The service provider decrypts the received data and verifies the correctness of the signature, the hash value, and the timestamp. The nonce $\mathtt{N_C"}$ is checked by means of $\mathtt{N_C'}$ known to the provider. If the received nonce is correct, the service provider can conclude that the service credential has been successfully decrypted with the private key $\mathtt{sk_C}$ of the customer and that the booking confirmation belongs to the previously transferred response.

If the service information $\mathtt{I_{BReq}}$ and the unique service object information $\mathtt{UI_{BReq}}$ of the booking confirmation matches the data of the service provider response, the service provider generates the `Service Credential`. It contains the service information $\mathtt{I_{SC}}$, the unique service object information $\mathtt{UI_{SC}}$, an `Authentication Key` used for securing the communication with the NFC object, e.g., the car's door lock, the encrypted `User Rights Credential`, the again modified nonce, and the timestamp $\mathtt{ts_{SC}}$. The `Authentication Key` is generated using the key derivation function described in Sect. 3.3. The `Service Credential` is hashed, signed, encrypted, and sent to the customer smartphone.

$$\mathtt{h_{SC}} := \mathrm{hash}(\texttt{Service Credential})$$
$$\mathtt{t_{SC}} := \mathrm{sign}_{\mathtt{sk_{SP}}}(\mathtt{h_{SC}})$$
$$\mathtt{p_{SC}} := \mathrm{encrypt}_{\mathtt{pk_C}}(\texttt{Service Credential} \,||\, \mathtt{t_{SC}})$$

After sending out the `Service Credential` as service response, the online ticketing service is successfully completed and the service provider can issue a bill to the correct customer. On the side of the smartphone, the received service response with the `Service Credential` is decrypted, its signature and hash value are verified, and the credential is stored in the SE of the card.

3.5 Executing the Granted Rights

For executing the obtained right at the booked NFC-enabled service object, the corresponding `User Rights Credential` — stored in the SE of the customer — needs to be securely transferred via NFC. For this purpose, the smartphone emulates a contactless smartcard with a suitable mutual authentication protocol for opening an NFC-enabled car, as illustrated in Fig. 4. In our realization of the scheme, a Mifare DESFire card [39] is exemplarily emulated[2], however, other cryptographic contactless smartcards could serve for the same task.

Fig. 4: For executing the granted rights at an NFC object, the smartphone emulates a contactless cryptographic smartcard (here, a Mifare DESFire card). The NFC reader in the door lock of the car detects the presence of the smartphone and initiates the communication. After establishing a secure channel (with 3DES), the credential `C` is transferred from the phone to the car.

For the emulation of a Mifare DESFire card, the NFC smartphone uses the 112-bit `Authentication Key` of the NFC object (contained in the `Service Credential`) for the mutual authentication and for establishing a secure channel with 3DES according to the proprietary protocol [36]. From the point of view of the NFC object, e.g., the car to be opened, the encrypted `User Rights Credential` appears to be stored in a data file of the DESFire card.

If the RFID reader of an NFC object detects an emulated contactless smartcard (or a real contactless smartcard) in its vicinity, the NFC object initiates the communication: It generates the required `Authentication Key` by means of the key derivation function detailed in Sect. 3.3. Next, the `Authenticate` procedure of the Mifare DESFire smartcard is executed between the smartphone and the NFC object. As an outcome of the mutual authentication, smartphone and object agree on a `Session Key` for enciphering the current communication session with 3DES in Cipher Block Chaining (CBC) mode.

The service object reads out the encrypted `User Rights Credential`, protected from eavesdropping attacks, decrypts it with the private key $\mathtt{sk}_{\mathtt{SO}}$, and verifies the contained information by means of the signature of the service provider. If the unique service object information of the received `User Rights Credential` matches the service object, the access rights are granted and the service object can be used by the customer. For example, a car can be opened, and/or its engine can be started during a defined time slot. This service activation can be repeatedly executed as defined in the car sharing information, e.g., during the booking period.

3.6 Updating the Service Objects

Each service object securely and confidentially stores an own intern variable, the `Access Variable`, on the SE that is used for the derivation of the `Authentication`

[2] We opted to emulate a real-world card, because this enables additional use cases in which a real contactless card executes a (pre-paid) right instead of the smartphone.

Key. The Authentication Key is newly generated in the SE after each User Right Validity Period using a service object-specific key derivation function with the Access Variable and the unique service object information as input.

After each User Right Validity Period on service object the service provider tries to securely exchange the confidential Access Variable via Internet. In case of missing Internet connection, the service object changes the Access Variable in a predefined way, also known to the service provider. The Access Variable is again modified after each completed User Right Validity Period and used for the calculation of the next Authentication Keys until the service object has Internet access for exchanging the intern Access Variable at a specified time.

The User Right Validity Period is specified as completed, if the authentication between service object and (emulated) smartcard was successful, the User Right Credential could be read out and was correct and the User Right Validity Period ran out.

The next time the service object has Internet connectivity, the Access Variable can be securely exchanged in the SE of the service object. For the refresh of the Access Variable the service provider sends a message to the service object encrypted with the public key pk_{SO} of the service object. The message contains the new Access Variable, the time to replace the old Access Variable, a nonce N_{SP} and a timestamp ts_{SP}. This data is hashed and the hash value is signed by the service provider. The use of the service object public key pk_{SO} binds the Access Variable to the one service object. After receiving the message, the service object exchanges the Access Variable at the specified time. The service object modifies the nonce N_{SP} in a predefined way, also known to the service provider, to N_{SP}' and generates a response message containing the changed nonce N_{SP}' and a timestamp ts_{SO}, hashed and signed with the service object private key sk_{SO}. The entire data is encrypted with pk_{SP} and sent back to the service provider. This way is ensured that the message for exchanging the Access Variable has reached the SE of the service object.

The important point of exchanging the Access Variable only at the specified time is founded in the requirement that already booked services for the service object have to remain valid after exchanging the Access Variable. If a User Right Validity Period is too far in the future that a variable exchange would be too late related to the security, the service provider can send a new Service Credential with a new Authentication Key to this customer.

3.7 Future Concept Optimization

Sweden has developed and tested Electronic Driving Licenses (EDLs) [15], Arizona has efforts to introduce new EDLs [5] and Australia in cooperation with Gemalto issued the EDLs 2010 that shall replace the laminated driver licenses step by step [1,26]. With widespread distribution of EDLs in the future, the personal registration in stores would be unnecessary, if the driving licenses store additionally the address of the EDL owner. Instead of using the nPA, EDLs can be used for verification of identity and driver permissions at the same time.

4 Proof-of-Concept Implementation

We have fully realized a working Java implementation of our concept on an NFC smartphone in a lab setup and verified the functionality with different real nPAs. Due to lack of a car with an NFC-enabled door lock, a PC connected to a standard NFC reader serves as a replacement for the car. Our realization is implemented with the BlackBerry Eclipse Java plug-in 1.5.2 for Eclipse version 3.7.0

(Eclipse Indigo) that supports `JRE 6 System Libary` offering many NFC functions and cryptographic classes, but on the other hand lacks support for the nPA Elliptic Curve (EC); this problem is further treated in Sect. 4.2. In the following, we present the target platform and selected parts of the implementation with a focus on the time-critical operations.

4.1 NFC Smartphone Blackberry Bold 9900

As a target platform for our demonstrator, we opted for the NFC-enabled smartphone Blackberry Bold 9900 [6] produced by Research In Motion (RIM). It provides a Qualcomm MSM8655 processor with an ARMv7 instruction set running at 1.2 GHz and 768 MB RAM. The SECUREAD NFC Solution Module by Inside Secure [30] realizes the NFC hardware of the phone. The usage of the smartphone's NFC capabilities is well documented, e.g., in [7,41,42]. In addition to NFC, the phone supports various other wireless communication methods, such as UMTS, WLAN, GSM, Bluetooth, and GPS. The phone furthermore offers a slot for microSD cards that might be used as an alternative to the built-in eSE of the NFC interface and the SIM card.

The OS of the Bold 9900 natively supports the passive NFC mode, i.e., emulating of a contactless smartcard. This is a key advantage in comparison to current Android-based smartphones, in which the original OS has to be modified, e.g., by means of CyanogenMod [12], in order to emulate a contactless smartcard [2].

4.2 ECC for PACE

The PACE protocol for communicating with the nPA requires an ECC implementation on the smartphone. Even though ECC is the most efficient established public-key scheme, it is still computationally intensive, and thus a bottleneck of the performance of our concept.

The elliptic curve used by the nPA is based on a 256-bit prime field $\mathbb{Z}_p$. The modulus of $\mathbb{Z}_p$ is a generalized Mersenne prime which allows for an efficient implementation of the reduction, and thus enables an efficient implementation of curve arithmetic. Unfortunately, the Application Programming Interface (API) provided by RIM does not support arithmetics on the required elliptic curve `brainpoolP256r1` [37], hence, we had to implement our own ECC arithmetics.

Point multiplication using affine coordinates, analogous to the exponentiation in multiplicative groups, turned out to be the most efficient approach for our demands. In order to efficiently compute the point multiplication, we used the right-to-left binary algorithm [43], a variant of the double-and-add algorithm. Our implementation on average requires 360 ms for one point multiplication.

4.3 NFC in Active Reader Mode: nPA Communication

For verifying the identity of the customer via the eID service, the NFC smartphone was set up to communicate with the nPA in active reader mode (cf. Sect. 2.2). We implemented the complete PACE protocol with Elliptic Curve Diffie-Hellman (ECDH) (cf. Sect. 2.1), employing our ECC implementation (cf. Sect. 4.2) for the ECC computations. Furthermore, we realized the required encryption and decryption with 3DES in CBC mode.

The correctness of the implementation has been successfully verified by means of test vectors [8] and practical tests with different nPAs. The achieved performance, in terms of the average runtime for 1000 PACE protocol runs is given in Tab. 1. A complete PACE protocol run requires 3.5 s on average.

Table 1: Average runtime of our PACE implementation over 1000 protocol runs

Time	Average runtime (ms)
up to General Authentication	262
Encrypt nonce	105
Map nonce	1695
Perform key agreement	1291
Mutual authentication	148
Total	**3501**

4.4 NFC in Card Emulation Mode: A Virtual Mifare DESFire Card

Our realization of a virtualized Mifare DESFire MF3ICD40 card employs the card emulation mode of the NFC smartphone (cf. Sect. 2.2).

We implemented the `Authenticate` procedure of the DESFire card for the initialization of the communication, the authentication, and the establishment of a secure channel between reader and smartphone. All further communication is then encrypted with 3DES in CBC mode, as done by genuine DESFire cards. After an authentication with key 0, the `ReadData` command is used for reading out the encrypted `User Rights Credential`. The correctness of the implementation has been validated with an ACG HF Multi ISO RFID Reader [4] that natively supports Mifare DESFire. According to our measurements, the complete process of communicating with the door lock and verifying the granted user rights takes less than one second.

4.5 Implementation Obstacles

For accessing an SE of the BlackBerry Bold 9900 smartphone, RIM can grant special programmer rights by means of two signing keys that have to be integrated into the development environment. One of these keys is the `RESE` key, required for accessing the eSE. The second key, termed `NFCR` key, is required for accessing the SE in the SIM card. Unfortunately, all our requests to RIM to get the permission for the programming part of any SE have been denied[3]. For accessing the NFC interface, we thus had to bypass the integrated SE and in consequence all computations were performed on the main processor of the smartphone — a fact which constitutes a severe security risk (cf. Sect. 5.3). Furthermore, our requests to program the main processor efficiently in C or assembly were fruitless (although technically possible), thus all programs (including the computations on elliptic curves) had to be implemented in Java.

For a practically secure realization, the required access rights to the SE are not given, the emulation of contactless cards is restricted by the hardware, the OS, and drivers, and the possibility of efficiently implementing in C (instead of slow Java implementations) is not enabled by the manufacturers and carriers. This situation is not limited to the smartphone used by us. On the contrary, it is the only NFC phone we found on which a (Mifare DESFire) card emulation is practically feasible without modifying the OS.

5 Security Analysis

In this section, we address different security aspects of the developed concept for user rights managements presented in Sect. 3, namely, the security of using the nPA

[3] The same applies to various other manufacturers of smartphones: No vendor was willing to give us access to a secure element.

without a dedicated reader, protecting the NFC interface, and performing security-relevant computations in an unprotected environment.

5.1 Security of using the nPA with an NFC Smartphone

The security of the nPA communication depends on the implemented PACE protocol that is analyzed in [34,11] and on the EAC protocol for which security analyses can be found in [46,11]. However, a security vulnerability related to using the nPA in our implementation remains: As demonstrated in [10], for a secure implementation of the PACE protocol, the PIN has to be entered using the keypad of a dedicated passport reader [18]. In our implementation, the customer provides the PIN by typing it into the user interface of an application running on the OS of the smartphone. The OS is prone to malware, e.g., a Trojan horse could let an attacker access and modify user inputs and other data. An attacker could obtain the secret PIN and, if the smartphone is situated close to the nPA[4] and NFC is enabled, the customer could be impersonated by the attacker. Despite the impersonation attack, the attacker has no access to the private key of the customer, if it was stored in the smartphone's SE. This private key is required for sending a service request to the service provider, thus the attacker cannot fraudulently obtain credentials. Nonetheless, typing a secret into the the OS poses a severe security risk for nPA applications in general, and should never be implemented in a practical application. Securely using eID cards like the nPA with an NFC smartphone thus needs to be further researched — the currently available hardware and software cannot ensure a secure usage.

5.2 NFC Communication

Eavesdropping and manipulation of any wireless communication interface is relatively easy. This of course also applies to ISO 14443 and thus, the NFC interface [40,28,22]. In [38], the NFC attack surface is summarized. [23,29] describe several well-known security threats of NFC. The NFC channel is secured by reliable cryptographic mechanisms in all parts of our concept, namely, those of the nPA and of Mifare DESFire cards that prevent these attacks.

A Man-In-The-Middle or relay attack is feasible for contactless smartcards and NFC [27,35], even with cryptographically secured messaging. However, the achievable ranges are well below 30 cm. An explanation of a practical relay attack on contactless transactions using NFC mobile phones is given in [24]. In the context of real-world-applications, e.g., our car sharing scenario, it is unlikely that an attacker gets into the required direct vicinity of the victim for unauthorizedly using the rental car. However, our implementation makes such an attack highly unlikely, since a user interaction via the application on the smartphone is required for initiating the communication to the car.

5.3 Missing Access to Secure Elements

The lack of an SE in our implementation results in serious consequences for the security of the user rights management. In principle, all relevant calculations and confidential data can be monitored by an attacker infiltrating the smartphone with malware. In our proof-of-concept implementation, an attacker could thus obtain the private key of the customer's smartphone and the credential, and use a booked car instead of the legitimate user. Nonetheless the attacker can not get newly requested credentials by herself; the attacker can send service requests on the behalf of the

[4] It is conceivable that a smartphone resides close to the nPA in a wallet of the owner.

customer, but is not able to correctly authenticate with nPA and PIN, if the PIN entering is secured.

The worst case for user right management system presented in this paper would occur, if the smartphone SE access is denied by the manufacturer, the smartphone is infiltrated by attackers malware, so that all credentials, the secure customer key $\mathtt{sk_C}$ and the nPA PIN can be spied out and the nPA is in NFC communication distance. The attacker can use old credentials, send new service requests and can impersonate the customer using the nPA and the nPA PIN.

6 Conclusion and Discussion

The two main goals of this work were to develop a concept for the secure management of rights and the design and realization of a mobile car sharing application on a smartphone. We presented a concept using a state-of-the-art NFC smartphone and the nPA that is also suitable for scenarios in which no permanent Internet connection is available. We combined and extended standard protocols and implemented the scheme in practice.

In our implementation, the smartphone in one moment acts as an RFID reader to interact with other NFC-enabled objects, such as electronic ID cards and NFC door locks, and in the next moment emulates a virtualized contactless card, in our case the Mifare DESFire card. Various other existing smartphone applications and future visions in the context of NFC can benefit from the presented work, e.g., smart metering for electronic vehicles, safe deposits of virtual warehouses, managing fleets of vehicles, or booking electronic keys for hotel rooms.

We showed that the developed concept and the practical performance of the implementation is suitable for real-world applications and discussed attacks and other security issues. We further pointed out the difficulties we experienced with commercial smartphones, SEs, and other implementation obstacles that hinder the development of secure NFC implementations. These are probably important reasons why many years after its invention, NFC is still not used to its full potential.

6.1 Future Work

The concept presented in this paper offers a lot of further extensions and possibilities for future solutions. As treated in Sect. 5 some difficulties in the implementation of parts of the concept, because of missing access permissions, etc., lead to attack vectors that have to be prevented.

The missing access to the smartphone eSEs for developers is a highly relevant problem in practice that can be result in security gaps, as discussed in Sect. 5.3. So, it is highly recommended that manufacturers of smartphones, restricting the access to secure elements or the NFC interface, provide a way to flexibly use their platforms to enable the development of secure NFC solutions in the scientific community.

Another security issue of the presented concept for right management is caused by the unsecured PIN entered into the customer's smartphone. So, another very important point on future work is the development of a system for secure smartphone inputs. This can be realized for example using a Trusted Platform Module (TPM), e.g., developed by the Trusted Computing Group [44], that provides special security features, e.g., a secure boot process. This way the PIN entering can be possibly secured from malware and OS manipulation on the smartphone, while Internet access is provided nonetheless.

References

1. *Biometrics, chips coming to Qld driver licenses.* http://www.itnews.com.au/News/174501,biometrics-chips-coming-to-qld-driver-licenses.aspx.
2. *Emulating a PKI smart card with CyanogenMod 9.1.* http://nelenkov.blogspot.it/2012/10/emulating-pki-smart-card-with-cm91.html.
3. D. E. 3rd and T. Hansen. *RFC 4634 - US Secure Hash Algorithms (SHA and HMAC-SHA).* Motorola Labs and AT&T Labs, Internet Engineering Task Force (IETF), July 2006.
4. ACG id. HF Multi ISO RFID Reader User Manual, 2006. http://www.rfid-webshop.com/shop/download/Reader/HF%2013.56%20MHz/ACG/Multi%20ISO/TAGnology_acg_contactless_flyer_neu_ACG_1356_HFMultiManual.pdf.
5. American Association of Motor Vehicle Administrators (AAMVA). *Electronic Driver's License (EDL).* http://www.aamva.org/EDL/.
6. BlackBerry. *BlackBerry Bold 9900/9930 Smartphones - Safety and Product Information.*
7. BlackBerry Support Community. *Java Development - NFC Primer for Developers.* http://supportforums.blackberry.com/t5/Java-Development/NFC-Primer-for-Developers/ta-p/1334857.
8. Bundesamt für Sicherheit in der Informationstechnik. *Worked Example for Extended Access Control (EAC) (PACE, Chip Authentication and Terminal Authentication)*, 2010.
9. S. L. Carlisle Adams. *Understanding PKI: Concepts, Standards, and Deployment Considerations.* Pearson Education, Inc., 2003.
10. Chaos Computer Club. *Praktische Demonstration erheblicher Sicherheitsprobleme bei Schweizer SuisseID und deutschem elektronischen Personalausweis*, 2010.
11. M. F. Christina Brzuska, Özgür Dagdelen. *TLS, PACE, and EAC: A Cryptographic View at Modern Key Exchange Protocols.*
12. CyanogenMod. *CyanogenMod.* http://www.cyanogenmod.org/.
13. T. Dierks and E. Rescorla. *RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2.* RTFM, Internet Engineering Task Force (IETF), August 2008.
14. M. Dworkin. *NIST Special Publication 800-38B (Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication).* National Institute of Standards and Technology, 2005.
15. European Commission. *Electronic driving licences.* http://ec.europa.eu/transport/road_safety/specialist/knowledge/esave/esafety_measures_unknown_safety_effects/electronic_driving_licences.htm.
16. Federal Information Processing Standards. *Federal Information Processing Standards Publication 180-2 - Secure Hash Standard*, February 2004.
17. Federal Ministry of the Interior, German Federal Republic. *White Paper - Neuer Personalausweis - eID-Server und eID-Service*, 2011.
18. Federal Office for Information Security, German Federal Republic. *Technical Guideline TR-03119 (Requirements for Smart Card Readers Supporting eID and eSign Based on Extended Access Control)*, 2011.
19. Federal Office for Information Security, German Federal Republic. *Technical Guideline TR-03127 (Architecture electronic Identity Card and electronic Resident Permitl)*, 2012.
20. Federal Office for Information Security, German Federal Republic. *TR-03110-2 (Advanced Security Mechanisms for Machine Readable Travel Documents – Part 2 – Extended Access Control Version 2 (EACv2), Password Authenticated Connection Establishment (PACE), and Restricted Identification (RI))*, 2012.
21. Federal Office for Information Security, German Federal Republic. *TR-03110-3 (Advanced Security Mechanisms for Machine Readable Travel Documents – Part 3 – Common Specifications)*, 2012.
22. T. Finke and H. Kelter. Radio Frequency Identification - Abhörmöglichkeiten der Kommunikation zwischen Lesegerät und Transponder am Beispiel eines ISO14443-Systems. http://www.bsi.de/fachthem/rfid/Abh_RFID.pdf. BSI - Bundesamt für Sicherheit in der Informationstechnik https://www.bsi.bund.de/ContentBSI/Themen/Elekausweise/rfid/Whitepaper/whitepaper.html.

23. K. Finkenzeller. *RFID-Handbuch.* Hanser Fachbuchverlag, Third edition, October 2002.
24. L. Francis, G. Hancke, K. Mayes, and K. Markantonakis. Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones. *IACR Cryptology ePrint Archive*, 2012.
25. Gartner. *Gartner Says Asia/Pacific Led Worldwide Mobile Phone Sales to Growth in First Quarter of 2013.* `http://www.gartner.com/newsroom/id/2482816`.
26. Gemalto NV. *Gemalto to provide a new Queensland Driver License.* `http://www.gemalto.com/govt/customer_cases/australia.html`.
27. G. Hancke. A practical relay attack on ISO 14443 proximity cards, 2005. `http://www.cl.cam.ac.uk/~gh275/relay.pdf`.
28. G. Hancke. Eavesdropping Attacks on High-Frequency RFID Tokens. In *Workshop on RFID Security*, 2008.
29. E. Haselsteiner and K. Breitfuss. Security in Near Field Communication (NFC) - Strengths and Weaknesses. *Workshop on RFID Security (RFIDSec 2006)*, Graz, Austria (2006).
30. INSIDE Secure. *SecureRead.* `http://www.insidesecure.com/eng/Products/NFC-Products/SecuRead`.
31. International Civil Aviation Organization. *JTC1 SC17 WG3/TF5 (Supplemental Access Control for Machine Readable Travel Documents)*, 2010.
32. International Organization for Standardization (ISO). ISO/IEC 14443: Identification Cards - Contactless Integrated Circuit(s) Cards - Proximity Cards - Part 1-4, 2001. `www.iso.ch`.
33. International Organization for Standardization/International Electrotechnical Commission. *ISO/IEC 18092 (Information technology — Telecommunications and information exchange between systems — Near Field Communication — Interface and Protocol (NFCIP-1))*, 2004.
34. D. K. Jens Bender, Marc Fischlin. *Security Analysis of the PACE Key-Agreement Protocol*, 2009.
35. T. Kasper, D. Carluccio, and C. Paar. An Embedded System for Practical Security Analysis of Contactless Smartcards. In *Workshop in Information Security Theory and Practice (WISTP 2007)*, volume 4462 of *Lecture Notes in Computer Science*, pages 150–160. Springer, 2007. Customized RFID Reader — GNU General Public License project `http://sourceforge.net/projects/reader14443`.
36. T. Kasper, I. von Maurich, D. Oswald, and C. Paar. Chameleon: A Versatile Emulator for Contactless Smartcards. In *ICISC 2010, Seoul, Korea*, volume 6829 of *Lecture Notes in Computer Science*, pages 189–206. Springer, 2011. Chameleon — GNU General Public License project `http://sourceforge.net/projects/chameleon14443`.
37. M. Lochter and J. Merkle. *RFC 5639 - Elliptic Curve Cryptography (ECC) Brainpool Standard - Curves and Curve Generation.* Federal Office for Information Security of the German Federal Republic, secunet Security Networks, Internet Engineering Task Force (IETF), March 2010.
38. C. Miller. Exploring the NFC Attack Surface. *Black Hat*, 2012.
39. NXP. *Mifare DESFire Short Form Specification MF3 IC D40*, 2004. `http://www.nxp.com/acrobat_download/other/identification/SFS075530.pdf`.
40. NXP. AN200701: ISO/IEC 14443 Eavesdropping and Activation Distance. Technical report, 2007.
41. Research In Motion (RIM). *BlackBerry JDE 7.0.0 API Reference.* `http://www.blackberry.com/developers/docs/7.0.0api`.
42. Research In Motion (RIM) - BlackBerry Support Community. *Java Development - NFC - Virtual Target Emulation.* `http://supportforums.blackberry.com/t5/Java-Development/NFC-Virtual-Target-Emulation/ta-p/1509687`.
43. M. Rivain. Fast and regular algorithms for scalar multiplication over elliptic curves. *IACR Cryptology ePrint Archive*, 338, 2011.
44. Trusted Computing Group. *TPM MOBILE with Trusted Execution Environment for Comprehensive Mobile Device Security*, 2012.
45. J. Winter, P. Wiegele, M. Pirker, and R. Toegl. A flexible software development and emulation framework for ARM TrustZone.
46. M. F. Özgür Dagdelen. *Security Analysis of the Extended Access Control Protocol for Machine Readable Travel Documents*, 2010.