Digital Signature Records for the NFC Data Exchange Format

Michael Roland
NFC Research Lab Hagenberg
Upper Austria University of Applied Sciences
michael.roland@fh-hagenberg.at

Josef Langer
NFC Research Lab Hagenberg
Upper Austria University of Applied Sciences
josef.langer@fh-hagenberg.at

***Abstract*—The NFC Data Exchange Format (NDEF) is a standardized format for storing formatted data on NFC (Near Field Communication) tags and for transporting data across a peer-to-peer NFC link. Through NDEF and its various record types, events can be triggered on an NFC device by simply touching an NFC-enabled object. The number of use cases and real applications around NFC and NDEF technology increases continuously. However, existing applications provide hardly any protection against (malicious) manipulation of NDEF data. Digital signatures are a means of providing authenticity and integrity of NDEF data. Therefore, the NFC Forum – which is responsible for the specification of data formats, protocols and applications in regard to the NFC technology – is working on adding digital signatures to their NDEF format. While their signature record type is still in draft status and has not been released to the public, this paper discusses the various aspects of digitally signing NDEF records. First, we introduce the readers to the NFC Data Exchange Format, its use cases and its potential security threats. After that, we describe the potential of digital signatures for NDEF messages. Finally, we discuss the advantages and disadvantages of various ways to digitally sign an NDEF message.**

I. Introduction

Near Field Communication (NFC) is a contactless communication technology standardized in [1], [2]. It is an advancement of inductively coupled proximity Radio Frequency Identification (RFID) technology. Therefore, NFC supports contactless smartcard systems based on the standards ISO/IEC 14443 and FeliCa. Besides standardization through normative bodies like ISO/IEC and Ecma International, further specification of data formats, protocols and NFC applications is driven by the NFC Forum[1].

NFC has three operating modes: peer-to-peer mode, card emulation mode and reader/writer mode. The peer-to-peer mode is an operating mode specific to NFC and allows two NFC devices to communicate directly with each other. In card emulation mode, an NFC device emulates a contactless smartcard and, thus, is able to communicate with existing RFID readers. In reader/writer mode, NFC devices can access contactless smartcards, RFID transponders and NFC tags. The NFC Forum specified four tag formats based on different existing RFID transponders. Many NFC devices even support additional non-standard tag formats like MIFARE Classic. NFC tags are basic data containers that offer read and write functions to store and retrieve data. In an NFC ecosystem these tags are used to store content like Internet addresses (URLs), telephone numbers, text messages (SMS) or electronic business cards. By simply touching a tag with an NFC device the information is transferred. The content is structured according to the NFC Data Exchange Format (NDEF, [3]). NDEF is a standardized format for storing formatted data on NFC tags and for transporting data across a peer-to-peer NFC link.

As of today many use cases based on NDEF exist. These use cases cover smart posters, the exchange of business cards and using NFC as an enabler for other, especially wireless, communication technologies. The basic principle of these use cases is "*it's all in a touch*" [4]. This means that simply touching an object with an NFC device immediately triggers an action. The term "smart poster" refers to posters, flyers and other advertising material equipped with NFC tags. For instance, these tags may convey an Internet address which provides further information about an advertised service, a telephone number for an advertised hotline or a ready-made SMS message for a ticket ordering service. Several applications are already in the field. They focus mainly on the smart poster use case and integrate NFC into existing web-based or SMS-based ticketing and information systems. Examples for such applications are

- the "ÖBB Handy-Ticket", a web-based train ticket in Vienna, Austria [5],
- the "Wiener Linien HANDY Fahrschein", an SMS-based e-ticket for the public transport system in Vienna, Austria [6],
- payment at Selecta vending machines [7],
- ticketing and current traffic information for the public transport system in Gothenburg, Sweden [8] and
- traffic information and guidance for the public transport system in London, UK [9].

Although, the number of available applications increases continuously, there is still a multitude of security problems. A serious risk is the manipulation of NFC tags. An attacker may replace (unprotected) tag content or even replace whole tags with modified tags. By, for instance, manipulating Internet addresses or telephone numbers in smart poster tags it is possible to redirect the user to a forged website for phishing user credentials or trick the user into sending an SMS message

[1] http://www.nfc-forum.org/

DOI 10.1109/NFC.2010.10

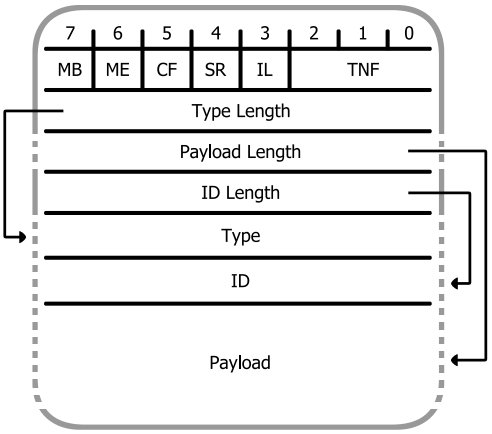

Fig. 1. An NDEF record consists of multiple header fields and a payload field. The header contains five flags – *Message Begin* (MB), *Message End* (ME), *Chunk Flag* (CF), *Short Record* (SR) and *ID Length Present* (IL), a type classification (*Type Name Format*, TNF), the length information for fields of variable length, a type identification (Type) and an optional record identifier (ID) [3].

Fig. 2. Multiple records form an NDEF message. The flags MB and ME are set for the first and the last record respectively.

to a costly premium rate service.

An important measure against manipulation of tag content is activating the permanent write protection of each distributed tag. Unfortunately, this only protects against modification of a certain tag. An attacker could still replace the whole tag or add additional tags to the infrastructure. One approach to diminishing the risk of such an attack is the use of digital signatures [10]. With a combination of digitally signed NDEF messages and a trustworthy certification infrastructure, the users (or their NFC equipment) have a means to distinguish genuine and forged tags.

This paper gives a short introduction to the NFC Data Exchange Format and its use cases. Furthermore, we outline various potential security threats concerning NDEF-based applications that have been identified in related publications. We describe the potential of digital signatures for NDEF messages. Finally, we discuss the advantages and disadvantages of various ways to digitally sign an NDEF message.

II. NFC Data Exchange Format

The NFC Data Exchange Format (NDEF, [3]) defines the format and the rules for exchanging data structures through NFC. Application specific data structures along with type information are packet into NDEF records. The layout of an NDEF record is depicted in Fig. 1. Multiple records form an NDEF message as shown in Fig. 2.

An NDEF record consists of multiple header fields and a payload field. The header contains five flags – *Message Begin* (MB), *Message End* (ME), *Chunk Flag* (CF), *Short Record* (SR) and *ID Length Present* (IL) – a type classification (*Type Name Format*, TNF), the length information for fields of variable length, a type identification (Type) and an optional record identifier (ID).

MB and ME mark the first and the last record of an NDEF message respectively. The flag CF, if set to 1, specifies that the payload of this record is continued in the next record. SR defines the size of the Payload Length field: When SR is 0 the payload length is a 4-byte unsigned integer, otherwise it is a 1-byte unsigned integer. This flag is useful to reduce the memory consumption of short records. If the flag IL is set to 1, then the optional ID field and its corresponding length field are present.

The value of the TNF field determines the format of the type information:

- 0h: The record is empty. The fields Type, ID and Payload are not present and their length fields are set to zero.
- 1h: The Type field contains the relative URI (Uniform Resource Identifier) of an NFC Forum well-known type according to the NFC Record Type Definition (RTD, [11]).
- 2h: The Type field contains a MIME media type identifier (RFC 2046).
- 3h: The Type field contains an absolute URI (RFC 3986).
- 4h: The Type field contains the relative URI of an NFC Forum external type according to the RTD.
- 5h: The record contains data in an unknown format. No type information is present and the length of the Type field is zero.
- 6h: The record continues the payload of the preceding chunked record. No type information is present and the length of the Type field is zero.
- 7h: Reserved for future use.

The ID field may be used to specify a unique identifier (in the form of a URI) for each record. This identifier can be used to cross reference between records.

The Payload field carries the actual data. The data is formatted according to the type information in the Type field. If e.g. the Type field specifies the MIME type “text/x-vCard”, then the payload is an electronic business card using the vCard file format. If the type information contains the NFC Forum well-known type “urn:nfc:wkt:U”, then the payload is a URI according to the URI Record Type Definition [12].

A data packet can be divided into multiple record chunks. In this case the first record contains the type information and the optional record identifier. The remaining chunks do not carry this information, but instead have their TNF field set to “unchanged” (6h). Except for the last chunk, every record chunk has its *Chunk Flag* set.

A. Record Types

The NFC Forum has defined a set of well-known type specifications. They cover basic data types, like the Text Record Type [13] and the URI Record Type [12], as well as complex data structures for specific use cases, like the Smart

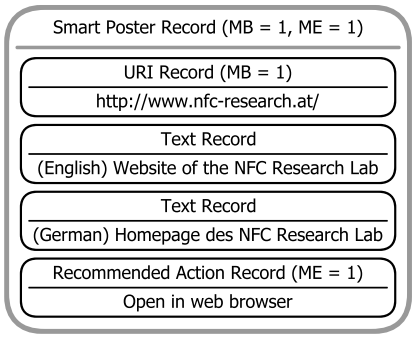

Fig. 3. A smart poster record is composed of a URI record and several informative records. Upon reading this exemplary record, an NFC-enabled mobile phone opens the URL "http://www.nfc-research.at/" in a web browser. Before actually starting the web browser the device may ask the user for permission to open the page. In this case the phone could choose to display one of the text records to describe the URL.

Poster Record Type [14], the Generic Control Record Type [15] or the Connection Handover Reference Application [16].

1) Text: Text records contain descriptive text along with language and encoding information. These texts are usually tagged to some other record (e.g. a URI) and typically describe that records content or function.

2) URI: A URI record conveys a URI reference. This could be, for example, an Internet address (Uniform Resource Locator, URL), an e-mail address, a telephone number or an SMS message.

3) Smart Poster: The Smart Poster Record Type extends the functionality of the URI Record Type. This is achieved by adding optional information like descriptive titles in one or more languages (based on the Text Record Type), an icon (based on an image or video MIME type) or an action that specifies what the receiving device should do with the URI. The payload of a smart poster record is an NDEF message which is composed of a URI record and all its attached informative records. Fig. 3 illustrates a smart poster record that points to a website.

The smart poster record is not limited to posters with active content. Instead, it can be used whenever a URI is used to trigger an action and when it should be combined with human readable information.

4) Generic Control: The Generic Control Record Type provides a common framework for describing pretty much any action. In comparison to the smart poster record, it is a more general way to define an action that should be executed on an NFC device. For instance, a generic control record may be used to trigger a function or update a property of an NFC device.

5) Connection Handover: The reference application "Connection Handover" provides a means of using NFC as an enabler for some other communication technology. The connection handover specification defines record types, message structures and a handshake protocol for establishing a link through virtually any alternative carrier.

III. Related Work

The growing number of use cases and actual applications leads to an increasing awareness of security risks. In [10], Madlmayr et al. highlight the security and privacy aspects. They point out that "*the inhibition threshold of touching a tag or a reader with the mobile phone is probably much lower than making an intended connection with a wire.*" Thus, the average user will not be able to distinguish forged tags from genuine tags. Consequently, NDEF-based applications are potentially vulnerable to various phishing attacks.

In [17], Mulliner reveals several flaws in the NDEF implementation and the web browser of the Nokia 6131 NFC[2] mobile phone. Moreover, he confirms several possible attacks on NDEF applications (with an emphasis on smart poster records) when tested on that mobile phone platform. Many of them follow a similar pattern: The phone usually displays the title followed by the URI reference (Internet address, telephone number ...). An attacker could use a specially crafted title record to show a falsified URI and push the real target URI off the screen. A user is likely to fall for that trick without even noticing the manipulated URI. Therefore, an attacker could take advantage of this approach by redirecting the users to phishing websites or by redirecting telephone calls or SMS messages to his own premium rate service [17]. The possible attacks are not limited to the smart poster record: The records of generic control or connection handover applications as well as any other type of record can be forged in a similar manner.

To prevent such attacks an NFC device has to verify the authenticity and the integrity of the received NDEF records. There are several approaches to reducing the risk of such attacks. In [18], Schoo and Paolucci suggest that spoofing of NFC tags can be prevented by registering all genuine tags in a database back-end and by using a certified application on the NFC device that compares the tags' data with the data stored in that back-end database. Another method to assure authenticity and integrity of NFC tags is digitally signing the NDEF records. In [19], Kilås evaluates several digital signature algorithms for NDEF messages and their feasibility and performance on mobile Java platforms. Digital signatures are also the solution that the NFC Forum chose for securing the NFC Data Exchange Format. Their Signature Record Type specification reached the state of a stable draft in April, 2009[3] but has not yet been released to the public[4].

IV. Digital Signatures

The digital signature of a data packet is calculated in two steps: First, a hash value is calculated for the data packet. The hash guarantees data integrity. Second, the hash value is encrypted with the signer's secret key. As only the signer has the secret key, this step assures the authenticity of the signature and the signed data.

[2]http://www.forum.nokia.com/devices/6131/
[3]http://nfc-forum.org/specs/spec_dashboard/, retrieved on Nov. 20th, 2009
[4]Nov. 20th, 2009

Digital signatures based on public-key cryptography in combination with a trustworthy certification infrastructure have several important properties [20]:

- *authentic:* The signing party can be determined unambiguously.
- *unforgeable:* Only the holder of a secret signing key can create an authentic signature.
- *non-reusable:* A signature is bound to the signed data and cannot be used for any other data. Thus, a digital signature assures the integrity of the signed data.

Therefore, signed NDEF data allows the receiving NFC device to determine if the data has a certain origin and if it is free of manipulations.

A. Averting Attacks

By the means of digital signatures, an NFC device has the possibility to identify the origin of the signed NDEF messages. Based on that information a decision can be made, whether NDEF records should be allowed to trigger certain events, like opening a specific website, calling a specific telephone number or initiating a specific alternative carrier. Yet, there are several types of attack that cannot be averted with digital signatures. Among them are the malicious modification of unlocked tags and the use of valid signed tags in other than the intended places.

V. Discussion: Signing NDEF Messages

There are many possible ways to sign NDEF messages. To stay compatible with the NDEF format and as there is already a Signature Record Type Definition on its way, we will focus on approaches where the signature is packed in its own record type and attached to the signed message. Our discussion focuses on which parts of an NDEF message should be covered by the digital signature and on how to handle the signatures. The actual implementation of a signature record is beyond the scope of this paper.

A. Backwards Compatibility

Backwards compatibility is an important requirement for the digital signature. There are two categories of compatibility:

1) Devices that do not support signatures: By using the approach of a dedicated signature record, devices that do not support signatures will simply ignore the unknown signature record. Thus, compatibility is not an issue.

2) Unsigned NDEF messages: Backwards compatibility to unsigned NDEF tags is a difficult topic. On the one hand, many current applications rely on unsigned tags. Therefore, an NFC device that blocks or ignores unsigned NDEF messages would render these applications unusable. On the other hand, an NFC device should distinguish between signed and unsigned data and use different levels of trust for each of these cases. If an NFC device would handle both cases, signed and unsigned, in exactly the same way, then the signature would be useless.

B. Authenticity vs. Authorization

When working with signed data, one has to distinguish between authenticity and authorization. When a signature is authentic, the signature's origin can be identified. This fact alone does not mean that the signing party is also eligible to sign that kind of record. Thus, each signing party needs to be certified for issuing a particular kind of records. For instance a certificate bound to *nfc-research.at* may only qualify for signing URI records that point to Internet addresses in the domain *nfc-research.at* and may not provide authorization for URLs in other domains.

C. Signing Individual Records

The digital signature could be attached to a single record, a group of records or the whole NDEF message. One NDEF message might be shared by more than one issuing party. Hence, signing the whole NDEF message with a single signature may not always be a desirable solution. The other extreme would be to sign each and every record individually. As tag memory is usually a very limited resource, this is not a reasonable solution either. Consequently, the best approach would be to group the records and sign each group individually. Probably the least memory consuming – and, therefore, most efficient – method is to sign a consecutive sequence of records and then immediately append the signature record.

D. Mixing Signed and Unsigned NDEF Records

Allowing only certain records to be signed and allowing multiple signing parties in one NDEF message also has a big drawback:

- What if a smart poster contains a signed title but an unsigned URI?
- What if a smart poster contains a signed URI but an unsigned title?
- What if a smart poster's title and URI are signed by different parties?
- ...

These questions are tightly linked to the issue of authorization. Especially the latter case may be abused to replace a smart poster's URI with a malicious URI. When the attacker has a valid certificate for the malicious URI he may even sign the forged part of the smart poster's NDEF message. Such cases and their possible exploitation for attacks must be considered thoroughly when implementing digital signatures for the NDEF format.

E. Message Begin Flag and Message End Flag

When a digital signature is applied to an NDEF message, one could either sign the whole records or only certain fields of the records. Signing each record as a whole leads to problems like the following:

When a signature is appended to a group of NDEF records, none of the signed records can have the ME flag set. As a result, including the ME flag results in an invalid signature when signing the last record of an NDEF message.

In general, it should be possible to move a group of signed NDEF records to any position within an NDEF message. Hence, including the MB flag or the ME flag into the signature is not useful.

F. Payload Field and Type Field

The central element to be protected by the digital signature is the records' payload. As the type identification determines the interpretation of the payload, the integrity of the Type field has to be guaranteed as well.

G. ID Field

NDEF records may be linked to other NDEF records through their ID reference. When the ID field of a referenced record is manipulated, any such links will be broken. An attacker could use this method to bypass a record in the signed NDEF message and to redirect the link to a new record (either unsigned or signed by the attacker).

H. Short Record Flag

The SR flag controls the size of the Payload Length field. When SR is set, the size is reduced from four bytes to one byte. When the signature includes neither this flag nor the Payload Length field, then repacking of NDEF records from one format to the other format would be possible. On the one hand, this could be used to reduce the size of an NDEF message without invalidating its signature. On the other hand, an attacker could use this feature to modify the fields that follow the Payload Length. If the length fields are not part of the signature, then there is no advantage for the attacker in manipulating the size of the Payload Length field, as its value could be modified anyways. Otherwise, three signed bytes could be moved from the Payload Length into the following fields (or the other way round) without changing the signature. Nevertheless, when this happens to change the size of the ID or the payload, signed parts of one record need to be included into an adjacent record. Thus, such an attack is not easily achievable.

I. ID Length Present Flag

The IL flag controls the presence of the ID Length field and, consequently, the ID field. When the signature includes neither this flag nor the length fields, then an attacker could add an ID field and use it to hide a suffix of the type identifier or a prefix of the payload without invalidating the signature. Similarly, an existing ID field could be integrated into the Type or the Payload field. If the length fields are part of the signature, then the lengths of Type, ID and Payload cannot be arbitrarily chosen. Therefore, as with the SR flag, such an attack is not easily achievable in that case.

J. Length Fields

When the length fields are not included into the signature, then the size of the Type, ID and Payload field may be changed without requiring an update of the signature. As with the IL flag, this could be exploited to move bytes between the field boundaries. E.g. parts of the ID field or even the payload could be appended to the type identification or the other way round. The signed parts of subsequent records could even be completely included into the preceding record's Payload field.

When the length fields are signed, then it becomes more difficult for an adversary to change the fields' sizes. An attacker could only adjust the lengths in combination with the SR or the IL flag. But even then the values of the length fields cannot be arbitrarily chosen.

K. Chunked Records

The Chunk Flag allows the payload of one record to be split across multiple smaller record chunks. When only the fields Type, ID and Payload are signed, then a signed record can be divided into chunks or merged from multiple chunks without invalidating the signature. This feature is useful to join chunks to one record in order to reduce the overhead of multiple chunk headers. Yet, this feature is also prone to attacks:

Even when every other field and flag, except for the CF is protected by the signature, an attacker could clear a set CF flag to cut the remaining chunks off the record. That way parts of the chunked record's payload can be chopped off. However, the remaining chunks will trigger parser errors as their Type Name Format field states that they continue a previous record. Only if the TNF field is also not included into the signature, an attacker could change that field's value to "unknown" and, thereby, make the parser ignore the trimmed chunks. In a similar fashion a subsequent record could be appended as record chunk to the payload of its preceding record.

L. Type Name Format

When the TNF field is excluded from the signature, the meaning of the type identification could be changed without actually modifying the Type field. For instance, the well-known type "urn:nfc:wkt:U" can be changed to the external type "urn:nfc:ext:U" (although this identifier violates the RTD specification as it does not include a domain name.)

In combination with other unprotected fields even further manipulation is possible without voiding the signature. Particularly in combination with the length fields an attacker could change the type of a record to "unknown" and integrate the unused type field into the payload (or the ID respectively).

M. Limitations of Java's Contactless Communication API

Many mobile NFC devices, especially mobile phones, provide a Java platform. Java's Contactless Communication API (JSR 257) already includes a package for parsing NDEF messages. As soon as the method for digitally signing NDEF records is standardized, this method needs to be available to Java applications. Hence, signature creation and verification has to be either integrated directly into that NDEF parser API or must be built on top of it. Extending the API defined by JSR 257 involves lengthy administrative steps and it will take additional time until implementations are integrated into the firmware of the NFC devices. Therefore, building a signature library on top of the existing API would allow for the library to be available to the application programmers much faster. Unfortunately, the NDEF parser already puts a certain level

TABLE I
RECORD FIELDS WEIGHTED BY THE BENEFITS OF NOT SIGNING A FIELD AND THE DRAWBACKS THROUGH THE POSSIBLE ATTACK SCENARIOS.

Field name	Signature useful[a]	Possible on top of JSR 257[a]
Message Begin	−−	−−
Message End	−−	−−
Chunk Flag	+	−−
Short Record Flag	+	−−
ID Length Present Flag	+	−−
Type Name Format	+	−−
Type Length	+	++
Payload Length	+	−−
ID Length	+	++
Type	++	++
ID	++	++
Payload	++	++

[a] Possible weights are ++, +, − and −− (with ++ being a definitive *yes* and −− a definitive *no*.)

of abstraction on the NDEF records. For example, a chunked record is automatically combined into a single record without the intermediate record headers.

For a signature library on top of the NDEF parser API this abstraction renders the inclusion of header fields into the signature virtually impossible. Merely header fields that only exist in the first chunk of a chunked record, like Type Length and ID Length, can be included. Consequently, when JSR 257's NDEF parser needs to be used, only the fields Type, ID, Payload, Type Length and ID Length can be protected with a signature.

VI. CONCLUSION

While some fields have to be included into the signature in order to guarantee a minimum level of integrity and authenticity, the inclusion of some fields has advantages as well as disadvantages (Table I). Yet, some other fields should never be signed. A minimum of integrity and authenticity is achieved by signing the Type, ID and Payload fields. The MB and ME, however, should never be signed to allow moving blocks of signed records within an NDEF message.

Excluding the remaining fields from the signature has several benefits: Most of these fields cannot easily be handled through Java's Contactless Communication API. Moreover, records could be repacked to accommodate the signed message to memory requirements.

Nevertheless, not signing these fields opens up several vulnerabilities to attack scenarios. Some of these scenarios allow single records in the signed message to be hidden from the parser or references through the ID field to be broken intentionally without voiding the signature. These records could subsequently be replaced by new records that are either unsigned or signed by the attacker. This kind of attacks can be prevented by either signing the vulnerable fields or by putting adequate rules of authorization in place that prevent mixing signed records, unsigned records and records that are signed by multiple parties within one context.

ACKNOWLEDGMENT

This work is part of the project "4EMOBILITY" within the EU programme "Regionale Wettbewerbsfähigkeit OÖ 2007–2013 (Regio 13)" funded by the European regional development fund (ERDF) and the Province of Upper Austria (Land Oberösterreich).

REFERENCES

[1] *Near Field Communication Interface and Protocol (NFCIP-1)*, Ecma International Std. ECMA-340, Rev. 2, Dec. 2004.
[2] *Near Field Communication Interface and Protocol -2 (NFCIP-2)*, Ecma International Std. ECMA-352, Rev. 1, Dec. 2003.
[3] *NFC Data Exchange Format (NDEF)*, NFC Forum Technical Specification, Rev. 1.0, Jul. 2006.
[4] E. Chen, "NFC: Short ragne, long potential," News Article, Aug. 2007. [Online]. Available: http://www.assaabloyfuturelab.com/FutureLab/Templates/Page2Cols____1905.aspx
[5] "ÖBB Handy-Ticket," retrieved on Nov. 18th, 2009. [Online]. Available: http://www.nfc.at/cms/front_content.php?idart=113
[6] "Wiener Linien HANDY Fahrschein," retrieved on Nov. 18th, 2009. [Online]. Available: http://www.nfc.at/cms/front_content.php?idart=114
[7] "Zahlen am Selecta Automaten," retrieved on Nov. 18th, 2009. [Online]. Available: http://www.nfc.at/cms/front_content.php?idart=37
[8] "TeliaSonera and Västtrafik tests new mobile technology in Gothenburg," Press Release, TeliaSonera Sverige AB, Aug. 2007. [Online]. Available: http://www.teliasonera.com/press/pressreleases/item.page?prs.itemId=304418
[9] "Smart posters show passengers the way," Press Release, Transport for London, Aug. 2007. [Online]. Available: http://www.tfl.gov.uk/corporate/media/newscentre/archive/5832.aspx
[10] G. Madlmayr, J. Langer, C. Kantner, and J. Scharinger, "NFC Devices: Security and Privacy," in *Proceedings of the Third International Conference on Availability, Reliability and Security (ARES '08)*, Barcelona, Spain, Mar. 2008, pp. 642–647.
[11] *NFC Record Type Definition (RTD)*, NFC Forum Technical Specification, Rev. 1.0, Jul. 2006.
[12] *URI Record Type Definition*, NFC Forum Technical Specification, Rev. 1.0, Jul. 2006.
[13] *Text Record Type Definition*, NFC Forum Technical Specification, Rev. 1.0, Jul. 2006.
[14] *Smart Poster Record Type Definition*, NFC Forum Technical Specification, Rev. 1.0, Jul. 2006.
[15] *Generic Control Record Type Definition*, NFC Forum Technical Specification, Rev. 1.0, Mar. 2008.
[16] *Connection Handover*, NFC Forum Technical Specification, Rev. 1.1, Nov. 2008.
[17] C. Mulliner, "Vulnerability Analysis and Attacks on NFC-enabled Mobile Phones," in *Proceedings of the International Conference on Availability, Reliability and Security (ARES '09)*, Fukuoka, Japan, Mar. 2009, pp. 695–700.
[18] P. Schoo and M. Paolucci, "Do You Talk to Each Poster? Security and Privacy for Interactions with Web Service by Means of Contact Free Tag Readings," in *Proceedings of the First International Workshop on Near Field Communication (NFC '09)*, Hagenberg, Austria, Feb. 2009, pp. 81–86.
[19] M. Kilås, "Digital Signatures on NFC Tags," Master's thesis, Royal Institute of Technology (KTH), School of Information and Communication Technology, Stockholm, Sweden, Mar. 2009.
[20] B. Schneier, *Angewandte Kryptographie*. Addison-Wesley, 1996.