mCoupons: An Application for Near Field Communication (NFC)

Sandra Dominikus and Manfred Aigner
Institute for Applied Information Processing and Communications
Graz University of Technology
Inffeldgasse 16a, 8010 Graz, Austria
firstname.lastname@iaik.tugraz.at

Abstract

Near Field Communication (NFC) is a short-range wireless communication standard. It is very comfortable for the user as communication starts without any further configuration steps when bringing two devices very close together. This property makes the standard well suitable for the use with mobile coupons (we call it mCoupons). Nowadays, coupons are a very common way for companies to promote their products or services. These coupons are available as paper-based coupons or electronic coupons (which for example are used on the web). We propose a new form of coupons, so called mCoupons, which can be downloaded from a poster or a newspaper equipped with a passive NFC device to a mobile device. With this mobile device the user can then cash in the mCoupon at the cashier. In most of the cases, the value of each coupon is relatively low but misuse in larger scale can result in a major damage for a company. Therefore, mCoupons shall be secured against attackers. In this paper, we propose protocols for a secure mCoupon system using NFC technology. This protocol can be a basis for future micro-payment or ticketing services.

1 Introduction

Near Field Communication (NFC) is a short range wireless communication standard defined in the ISO/IEC 18092 standard [7]. It is expected that in the future most of the mobile devices will be equipped with an NFC interface. NFC works at 13.56 MHz and can be used for communication between two active devices or between an active and a passive device. Active devices are powered by a battery, passive devices gain their energy from the electromagnetic field of the active device. The communication always happens between an NFC initiator and an NFC target. The active device can take over both roles, whereas a passive device is always the NFC target. The initiator sends requests to the target, the target answers these requests.

NFC works in a very intuitive way for the user: Two NFC devices start their communication by bringing them closely together (it is also called "touching" each other). The touching of the components is seen as a declaration of intention from the user. This easiness of use makes the technology preferable for the mCoupon system we propose. Coupons (paper-based as well as electronic ones) are a common and well-established way of marketing. Companies use this means to establish consumer relationships or to reward the consumer for looking at their advertisement or out of other reasons to improve their business.

mCoupons are coupons that can be collected and stored on a mobile device (like a mobile phone or PDA). We propose a system, where the client uses an NFC-enabled device. The user receives an mCoupon from a passive NFC target and cashes it in at a terminal of the merchant (also using the NFC interface). In this paper we will show the differences of mCoupons to existing paper-based coupons or systems of secure e-coupons for web stores. Further, we will describe the mCoupon concept as well as the protocol and the requirements for the interacting parties of the system.

2 Motivation for Using mCoupons

A significant difference of virtual or electronic coupon systems to paper-based coupons is the fact, that issue and pay-in are performed without direct human interaction. Copies of unprotected electronic coupons can be produced without any relevant cost for the attacker. A distributed attack can therefore easily be mounted in an automated way. Blundo et al. give in their paper [2] some examples, how multiple cash-in of electronic coupons generated respectable loss. A secure system of electronic coupons provides protection against attacks with a level of security that should make an attack more expensive than the value an attacker could gain from a successful attack.

Coupons are often given out to collect some information about customers. For example, the filled-out coupon con-

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE

tains the address of the client which can be used for subsequent advertisement. The possibility to convert the gathered information into usable data in electronic format without significant costs, is probably the major advantage of an electronic coupon system. Electronically collected coupons can be stored and processed in databases with basically no additional costs compared to paper based coupon systems. To allow further processing the client information from paper coupons needs to be typed in manually to convert them into electronic data. More accurate information can be gathered (e.g. exact collection time, or specific up-to-date information about the client) and the value of the coupon can be made dependent on this additional data (e.g. the earlier a coupon is paid in, the higher is its value). Calculation of the costs for collection of coupons and the handed out gifts can be done in a fully automatic way. Although this additional information is very important for coupon system providers, collecting and further management of the gathered data of clients is not the focus of this paper. The focus of this paper is to describe an mCoupon system which is secured against illicit use of clients or attackers.

Some approaches for secure electronic coupons (E-Coupons) have been published already for usage in web stores ([3], [8], and [11]) or peer to peer systems ([12]). In the published solutions, e-coupons are distributed by email or can be collected from web pages. They are paid-in directly at the online store of a merchant. mCoupons are issued by NFC-enabled issuers (e.g. in a newspaper or an advertisement poster) and are stored on a mobile device which can be carried to a cashier by the client. E-coupon systems require online access of the issuer, the client and the merchant. The system of mobile coupons (mCoupons) works without online access of the client and the issuer, but provides protection against illicit use.

3 The mCoupon Concept

mCoupons are electronic coupons and can be stored and cashed in using a mobile device. The coupon is issued by an *issuer*, equipped with an NFC interface. The issuer can be attached for example to a newspaper advertisement or a poster. The client carrying a mobile device (also equipped with an NFC interface) can now "touch" the issuer and a valid mCoupon is stored on the mobile device. Each client can possess more than one mobile device, but then he also possesses more than one virtual identity (one for each mobile device) in this system. The client can now take the mCoupon to a *cashier*, which is also an NFC-enabled device, and cash in the coupon. The cashier verifies the validity of the coupon and hands over the bonus product or service. The basic mCoupon mechanism is shown in figures 1 and 2.

Unprotected data on mCoupons can be easily copied or

Figure 1. Issuing an mCoupon

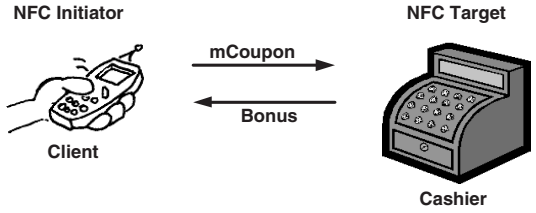

Figure 2. Cashing in an mCoupon

modified without significant costs. For this reason security measures should be taken. There are a couple of security issues that have to be addressed:

- **Multiple Cash-In:** An attacker shall not be able to use the same mCoupon multiple times.
- **Unauthorized Generation:** An attacker shall not be able to issue his own mCoupons.
- **Manipulation:** mCoupons shall not stay valid after a manipulation.
- **Unauthorized Copying:** An attacker shall not be able to produce a valid copy of an mCoupon and cash it in.

Some of these features are optional, e.g. multiple cash-in or copying of coupons is sometimes permitted. The features depend on the system requirements. In section 5 we present an mCoupon system which is secured by standard cryptographic means. The system implements countermeasures against the mentioned security threats where some of the features can be omitted. The next section deals with the basics of cryptographic authentication which is used as basic countermeasure in our system.

4 Authentication Principles

Authentication means to proof a certain identity to another party. Cryptographic authentication is mostly based on showing the knowledge of a secret key. Challenge-response authentication (also called strong authentication) is often used in practice, even in smart-card applications. In principle, a challenge is sent to the party, which wants to proof its identity. Then this party returns a response, which

can be verified, and the party is then authenticated. Authentication principles are explained in more detail e.g. in [9].

A broadly used standard, where challenge-response authentication techniques are described is the ISO/IEC 9798 standard [5], [6]. Basically, one-way authentication works like shown below:

$$\begin{aligned} A \rightarrow B &: ID \\ B \rightarrow A &: R_B \\ A \rightarrow B &: F_K(R_B) \end{aligned}$$

Party A wants to authenticate against party B and claims to have a certain identity by sending its ID. B sends a time-variant challenge (R_B) to A. A can authenticate itself by showing, that it knows the secret or private key K. A calculates a function F, the result depends on the secret (or private) key K and the random number R_B. This result is sent to B. B verifies the response and can in that way verify the identity of A. It is very important that the challenge is time-variant: The attacker can log the communication between A and B and if a challenge is used a second time, the attacker can answer this challenge without knowing K. Therefore variation of the challenge is vital for the security of the authentication protocol.

Different functions F with appropriate cryptographic properties can be used for authentication. The most important property is, that only a person knowing the secret key can produce valid results. The results depend on the challenge and the secret as well. We can differentiate between symmetric and asymmetric methods. When using symmetric methods, both parties possess the same key, which is called *secret key*. As B knows the secret key and the challenge, he can easily verify A's response. The drawback of symmetric authentication methods is the complex key management. All communicating parties work with the same key. If the key of one party is compromised, the whole system becomes insecure and a new symmetric key has to be distributed. Key establishment and distribution is costly. Symmetric authentication methods are best suited for closed systems, where all devices are under the control of one central instance.

Asymmetric authentication methods are well suited for open systems. Each party possess a public and a private key. With the private key one party can sign or encrypt a message. With the public key, which is open to everybody, each other party can verify the encryption or the signature and can in that way verify the identity of the sender. Asymmetric methods are in general more time- and power-consuming than symmetric algorithms, but key management is easier. Some common methods for key management are described in the next subsection as they are recommended for the use with our protocol.

It is good practice to use standardized protocols and algorithms. The advantage is, that such protocols and algorithms are broadly used and tested in real-world applications. Existing security flaws are likely to be found and occur in a very early stage when the standardized algorithm is introduced. In standardized cryptographic algorithms which are already in use for a certain time, these flaws have already been removed. Therefore, standard protocols and algorithms guarantee a well known level of security. Furthermore, interoperability between different systems is more likely with protocols conforming an international standard. Our system works with standardized protocols and algorithms.

4.1 Basics of Public Key Infrastructures

For asymmetric authentication techniques we need public-key cryptography, i.e. each party has a key pair consisting of a private and a public key. With the private key, the party can sign challenges and with the public key each other party can verify the signature. The aspect we want to point out in this subsection is the access to the public keys. The public keys of the authenticating parties are published, but the party which wants to verify the signature has to know how to get the public key of the party which wants to authenticate itself. We want to discuss in short two mechanisms: certificates and a PKI server.

When using a PKI server, the public keys of all parties are stored in a central database. The verifier has to know how to contact that server and has to trust in the integrity of the data stored on it. The verifier can send a request to the server with the ID of the authenticating party, the server responds with the corresponding public key. Figure 3 shows an asymmetric authentication using a PKI server. The authenticating party A generates a public-private key pair. The public key (Pub_K) together with the ID of A is transferred to a PKI server. A sends its ID to party B. B gets the public key of A by sending the ID to the PKI server. The transmission of the public key shall be secured as well, e.g.

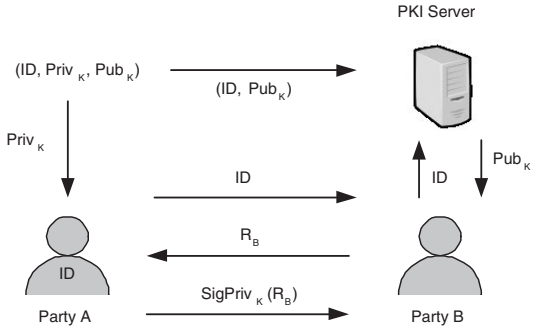

Figure 3. Assymmetric Authentication with a PKI Server

the server itself can sign the transmitted data. B sends a challenge to A, which is signed by A using the private key. B can now verify the signature by using the public key of A.

Figure 4 shows an asymmetric authentication using certificates. When using certificates, we need a certification

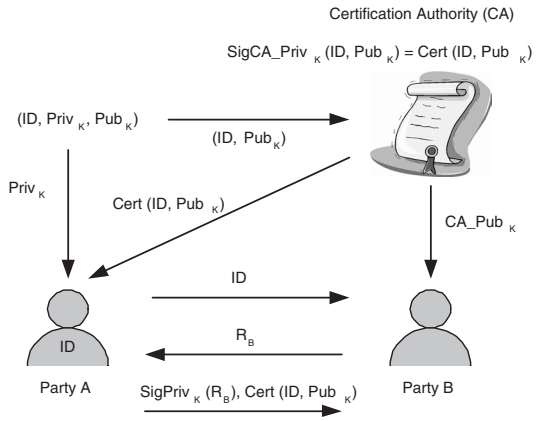

Figure 4. Assymmetric Authentication with Certificates

authority (CA). The verifier has to trust in the CA and get the public key of the CA (CA_Pub_K) in a secure way. The authenticating party can get a certificate from the CA by sending its ID and its public key (Pub_K) to it. The CA signs these data with its private key (CA_Priv_K). The result of this signature is the certificate ($Cert(ID, Pub_K)$). During authentication, a party not only sends a signature for the challenge R_B but also the certificate. The verifier checks the validity of the certificate with the public key of the CA (CA_Pub_K). He also checks if the ID in the certificate corresponds to the ID of the authenticating party. If all conditions hold, the verifier now knows the public key of the authenticating party and can verify the signature of the challenge. An offline system is possible with certificates, if no revocation information is needed. If a verifier wants to approve, that a certificate is not revoked by the CA, he also has to have online access to a database, where a list of all revoked certificates is available.

For more information about public key infrastructures please refer to [1], [5] or [9]. In the next section, we describe how authentication can be used to handle the security issues of mCoupons.

5 The mCoupon Protocol

We start this section with a discussion how cryptographic (and also non-cryptographic) methods are used as countermeasures against the four security issues mentioned in 3 and continue with explaining the "simple" mCoupon protocol, which does not provide security against unauthorized copying. The last subsection describes the "advanced" mCoupon protocol, which also covers the unauthorized-copying threat.

5.1 Multiple Cash-In

Multiple cash-in can easily be prevented by the cashier. The cashier is a device with enough computational power and storage to store the data from all mCoupons which are already cashed in. If more than one cashier are used, the data from the used mCoupons can be stored in a central database, where each cashier has access. If a client sends a request to cash in a new mCoupon, the data from this coupon are compared with the mCoupons in the memory. If the same mCoupon is found, the cashier rejects cash-in. In that way, the cashier can prohibit multiple cash-in of the mCoupons without making use of cryptographic methods.

5.2 Unauthorized Generation

The issuer has to prove, that it is authorized to generate mCoupons. We want to apply symmetric authentication to reach this goal. Each authorized issuer knows a secret key and is able to calculate a symmetric encryption algorithm, namely the Advanced Encryption Standard (AES) [10]. This algorithm is used because hardware implementations of this algorithm are already suitable for passive RFID environments (see [4]).

The client sends a challenge to the issuer, the issuer encrypts this challenge using its secret key. Unfortunately, the client is not able to verify the response as he does not know the secret key. The verification is done at cashing-in. Besides additional data, the mCoupon mainly consists of the challenge and the response. The client sends these data to the cashier. The cashier knows the secret key of the issuer and can verify, that the mCoupon was issued by an authorized issuer.

Any attacker can try to generate new valid mCoupons by producing challenge-response pairs of his own. As the attacker does not know a valid secret key, he will not succeed in producing a valid challenge-response pair. As mentioned before, the outcome of the cryptographic function used for authentication depends on the challenge as well as on the secret key. So, a random challenge-response pair generated without knowing a correct secret key will not be accepted by the cashier. In that way, unauthorized generation of mCoupons is prevented.

5.3 Unauthorized Manipulation

If an mCoupon is modified, the cashier shall notice this manipulation and rejects cash-in. The mCoupons consist

of a non-encrypted (the challenge), and an encrypted part (the response) depending on the challenge and the secret key. In order to produce a manipulated challenge-response pair which is still valid, an attacker will have to manipulate both, the challenge and the encrypted part. He can easily alter the challenge, but as the response also depends on the secret key, he will not be able to produce a valid response to that challenge. In that way, manipulation of mCoupons is prevented.

5.4 Unauthorized Copying

Neither an attacker nor the client itself shall be able to copy or delegate the mCoupon to another mobile device. In order to reach this goal we have to apply client authentication in the mCoupon system. The key that is used for authentication shall be bound to the client's mobile device. Before the issuer hands the valid mCoupon over to the client, the client has to authenticate itself against the issuer. We suggest to use an asymmetric authentication method because key distribution and scalability is provided easily. Furthermore, the system we propose is also able to work offline. Asymmetric operations in hardware are costly in terms of time, area and energy consumption. As the issuer is a passive NFC device we have to keep the circuit small and asymmetric operations are out of scope. Therefore, it is not able to verify the received client authentication data.

Instead of verifying the authentication of the client, the issuer postpones the verification. The exchanged authentication data is integrated into the mCoupon data and is encrypted by the issuer. In that way, an attacker is not able to modify the authentication data without destroying the validity of the mCoupon. At cashing-in the client has to authenticate himself once more. He has to use the same private key as for authentication at the issuer's site. The cashier has all the necessary resources to verify the authentication response. It can also extract the authentication data exchanged at issuing the mCoupon. Now, the cashier also verifies this authentication data and checks if the same private key is used for both authentication processes. This condition holds, if the data exchanged in both authentication procedures are valid for the same public key and if this public key is linked to the client's ID. If this is the case, the cashier can be sure, that the mCoupon was issued for exactly the same client who wants to cash the coupon in.

An attacker can copy the mCoupon from a client, but he does not know the client's private key for authentication. Therefore, the attacker is not able to authenticate himself against the cashier. If he uses another valid private key for authentication, the authentication data stored in the mCoupon are not valid, because they were generated with another private key. They are only valid for the client's authentication key (= private key). The same is true if a client wants to delegate his mCoupon, e.g. copy it to another device. The authentication key is bound to the device and therefore the cashier notices, that the authentication key at issuing and at cashing-in differ and rejects the coupon.

Using client authentication, we establish a virtual geographic link between the client and the issuer. The client can prove in that way that he physically "touched" the issuer at a moment in the past. This property cannot only be used for mCoupons, but also for all other applications, where geographic links are of importance (e.g. paper-chase games, orienteering).

In the next subsections we describe two mCoupon protocols with different security features.

5.5 The "Simple" mCoupon Protocol

At startup of the system, some initialization steps have to be taken.

- The passive NFC target attached to the poster or the advertisement (= issuer) has to be initialized, i.e. a secret key (K_I) has to be assigned to the target. The secret keys of all targets have to be stored in a database. All cashiers must have access to that database in order to be able to verify the mCoupons. The access control to that database has to be defined, because only authorized cashier are allowed to know the secret keys. Here, also authentication can be used.
- The client has to install a software handling the mCoupons on his mobile device.

The client "touches" the issuer with his mobile device. The client generates a challenge (R_M) and sends it to the issuer. The issuer attaches some informative data (*Offer*), e.g. about the type, issuing time and validity range of the coupon, and encrypts the challenge and additional data using the secret key K_I. Then, it sends a valid mCoupon to the client's mobile device. The mCoupon consist of the issuer's ID (ID), the challenge, the additional informative data, and the encryption result (EK_I = response). Figure 5 shows the issuing of an mCoupon in the simple protocol.

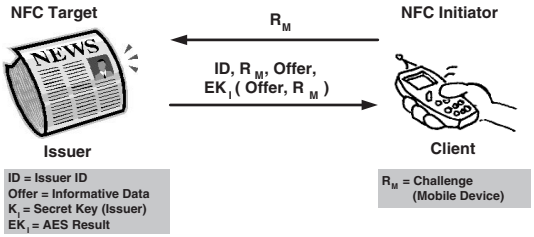

Figure 5. Issuing an mCoupon

For cashing-in, the client "touches" a cashier with his mobile device. The mobile device sends the mCoupon to

the cashier over the NFC interface. As the ID of the issuer is part of the mCoupon, the cashier can now search for the appropriate secret key of the issuer (K_I) in the key database. The mCoupon also contains the challenge from the mobile device (R_M) and the additional informative data (*Offer*). With these inputs, the cashier can verify the encryption result from the issuer (EK_I). The cash-in process for the simple protocol is shown in figure 6.

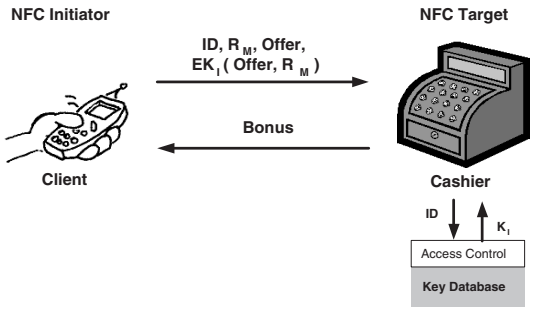

Figure 6. Cashing In an mCoupon

As only authorized issuers and cashiers know the secret key (K_I), unauthorized generation and manipulation of mCoupons is prevented. It is up to the cashier to decide if multiple cash-in should be prevented. The cashier can store the mCoupons in a (central) database and check if further coupons are identical. Copying of the coupons is not prevented by this protocol. Everyone can transfer the mCoupon to another device and cash this coupon in. The cashier will not make any difference about the clients. In some applications, this can be a feature, but sometimes it is better to prevent copying. Following, we describe an advanced protocol which also provides copy protection.

5.6 The "Advanced" mCoupon Protocol

The advanced mCoupon protocol implements client authentication in addition to the features of the simple protocol described above. For client authentication, asymmetric authentication methods are used. The client, or better the client's mobile device, has to generate a key pair consisting of a private and a public key. The initialization of the advanced protocol requires the following steps:

- The passive NFC target has to be initialized, i.e. a secret key (K_I) has to be assigned to the target.
- The client has to install a software handling the mCoupons on his mobile device. During this initialization step a public-private key pair is generated and an ID is assigned to the mobile device. The private key is stored on the mobile device. The public key shall now be linked to the ID of the mobile device. For user convenience, this shall be a fully automated process without any user interaction.

  The first alternative is that the public key and the corresponding ID are transferred to a PKI (= public key infrastructure) server. The access to the PKI server can be public, i.e. also all cashiers have access to it. The server has to make sure, that its database can only be modified by authorized parties and all links (ID to public key) are valid. For setting up an offline system, the public key can be signed by a trusted CA and the certificate, containing the public key and the ID of the mobile device, is stored on the mobile device.
- The cashier has to take care of the access to the clients' public keys. In case of a PKI server, the cashier has to trust this server and has to have access to it. In case of public-key certificates, the cashier has to make sure that it is able to verify the signature from the CA, i.e. it has to install and to trust in the public key of the CA.

Public-key crytpography and infrastructures are already used in established systems and applications. We suggest to use standard solutions for key management in asymmetric systems in the mCoupon system as well.

The protocol works as follows: The client "touches" the issuer and sends a request to get a valid mCoupon containing the challenge R_M for the issuer. In the advanced protocol, the issuer wants the client to authenticate himself, so the issuer also sends a challenge (R_I) to the client's mobile device. The mobile device signs this challenge by using its private key (PrK_M). It sends the signature and its ID (ID_M) to the issuer. The issuer is not able to verify the signature, but uses the authentication data as input for the AES encryption. It attaches additional informative data (*Offer*) and the authentication data to the challenge and encrypts this input using the AES. Like in the simple protocol, the valid mCoupon consists of the issuer's ID, the client's challenge and the encryption result. Figure 7 shows the issuing process for the advanced protocol.

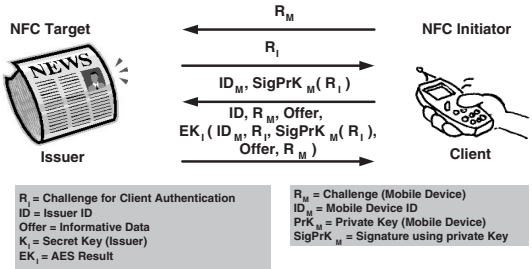

Figure 7. Issuing an mCoupon

The client takes the mCoupon to the cashier. The cashier wants the client to authenticate himself and sends a chal-

lenge R_C. The client's mobile device signs the challenge with its private key (PrK_M) and sends the result ($SigPrK_M(R_C)$) together with its ID (ID_M) back to the cashier. The mobile device can also send a certificate containing its public key ($Cert(PuK_M)$). The cashier needs the public key from the mobile device in order to verify its signature. If the client has sent a certificate, the cashier has to make sure, that the signature of the CA is valid and the CA is trusted. If this is the case, the cashier can verify the client's signature and get the client's public key. If the client did not send a certificate, the cashier has to contact the PKI server and sends the client's ID to the server. The server responds with the corresponding public key. If the cashier is not able to verify the client's signature, he rejects the mCoupon.

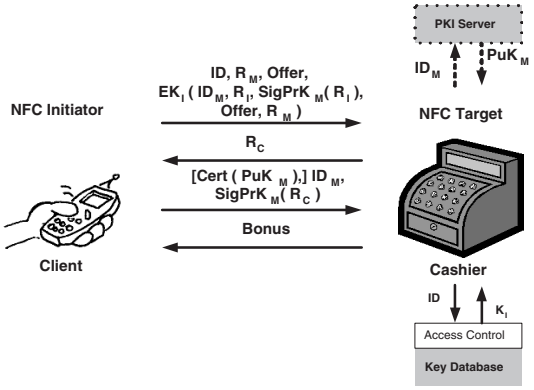

Figure 8. Cashing in an mCoupon

With the issuer ID it can get the secret key K_I. It decrypts the encrypted part of the mCoupon and gets access to the client-authentication data exchanged at issuing the coupon. The cashier compares the IDs (ID_M) and uses the public key from the present client to verify the authentication data. If the IDs are equal and the signature $SigPrK_M(R_I)$ can be verified with the present public key and the issuer's challenge contained in the mCoupon(R_I), the cashier can be sure, that the mCoupon was issued for the present client. Then the cashier verifies if the challenges (R_M) are the same in- and outside the encrypted part. If all these checks are successful, the cashier hands over the promised bonus.

Like the simple protocol, the advanced protocol offers countermeasures against unauthorized manipulation and generation of mCoupons. Furthermore, copy protection is guaranteed because the client has to authenticate itself at issuing as well as at cashing-in. Only if the same public key can be used to verify the authentication and the client's ID is linked to that public key, the mCoupon is valid. As the public key is bound to the client's mobile device by means of a PKI or certificates, a successful verification of both authentications implies that the same mobile device was used for issuing and cashing-in.

In the next section we try to identify the technical prerequisites for designing an mCoupon system in practice.

6 System Requirements

Various parties which work in different environments participate in the mCoupons protocol. Each of them is involved in different processes and has different requirements. In this section we want to identify the processes and requirements for each of the parties.

Activator. The activator is the party which takes care of the initialization of the issuer and the cashier. In general, it should be a device with enough computational power. The timing constraints for initialization are also not very limited. Required initialization processes are:

- **Initialization of the Issuer:** The activator generates secret keys for the issuers.
- **Management of the Key Database:** The key database contains the secret keys of all issuers. When initializing a new issuer, the activator stores its secret key in the database. The activator has to distribute the access key(s) to the database to all cashiers. Furthermore, the activator is responsible for revocation and re-distribution of the keys.
- **Initialization of the Cashier:** The cashier is initialized with the access key to the database. For the advanced protocol: If a PKI server is used to manage the public keys of the clients, the activator has to enable the cashier's access to this server. If certificates are used, the activator has to provide the cashier with the appropriate certificate of the trusted CA, such that the cashier is able to verify the client's certificate offline.
- **Management of the mCoupon Database:** In order to prevent multiple cash-in of mCoupons, there should be a central database where all used mCoupons are stored. The activator shall enable the access to this database for all new cashier devices.

Issuer. In general, issuer circuits are very small and attached to advertisement posters, newspapers or magazines. The issuer is a passive NFC device, which means that the energy for operation comes from the field of active NFC devices near the issuer. Thus, the available energy for the AES computation is extremely limited. The functionality required from the issuer can nowadays be provided even by a contactless smart-card chip or also by passive RFID or NFC devices equipped with an appropriate crypto hardware. Using contact-less smart-card chips as issuers

would, due to their high costs, prevent applications with a high number of issuers (e.g. issuers placed in magazines). For high-volume applications with lower costs per issuer, passive RFID tags with AES hardware can be used [4].

Client. The mobile device of the client can be a mobile phone or a PDA. It has to be able to perform asymmetric authentication. The private key has to be stored in a secure memory. The required calculations should be possible for a current standard mobile phone. The mobile device has an NFC interface and a software which supports the requesting, handling and cashing-in of an mCoupon. Nowadays, a standard mobile phone or PDA has enough computational power and memory to participate in the mCoupons protocol. It is expected that in the future NFC interfaces will be broadly used in mobile devices.

Cashier. The cashier verifies the validity of cashed-in mCoupons and hands over the promised bonus in case of success. The cashier is an active device with high computational power and energy resources. It must be able to verify digital signatures and to perform an AES calculation. Furthermore, it has access to the issuers' secret keys and the clients' public keys as well as to the mCoupon database, where all used mCoupons are stored. The functionality of the cashier can be provided by a standard PC solution with online access.

7 Conclusion

In this paper, we proposed an mCoupon system based on NFC technology. As NFC technology is very comfortable for the user, it is well suited for the use with mCoupons. mCoupons are mobile coupons, which can be stored on a mobile device. Although each mCoupon represents in general only a minor value, a misuse in large scale can result in a major damage for companies. Therefore, mCoupons shall be secured against multiple cash-in, unauthorized generation and manipulation, and copying. We have addressed these security threats and proposed two protocols for secure mCoupons with different security features. We derived the requirements for all components and showed that it is possible to build the proposed system with currently available technology. In contrast to existing systems of electronic coupons, mCoupons do not require online access of issuer and mobile client during pick-up and cash-in of coupons.

Currently, a prototype system of mCoupons is implemented using NFC-development kits and semi-passive RFID tags. Future extensions of the system will take NFC issuers with online access into account, which will extend the possible application range of mCoupons to electronic ticketing and micro-payment applications.

Acknowledgements. The development and prototype implementation of the proposed system is performed within the project SNAP, funded by FIT-IT from the Austrian bm:vit. All research and development is performed by partners of the initiative PROACT (Programme for Advanced Contactless Technology) at Graz University of Technology.

References

[1] C. Adams and S. Lloyd. *Understanding Public-Key Infrastructure*. New Riders Publishing, 1999.

[2] C. Blundo, S. Cimato, and A. D. Bonis. Secure E-Coupons. *Electronic Commerce Research*, 5(1):117–139, January 2005.

[3] C. Blundo, S. Cimato, and A. DeBonis. A Lightweight Protocol for the Generation and Distribution of Secure e-Coupons. In *Proceedings International WWW Conference 2002*. ACM, May 2002.

[4] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong Authentication for RFID Systems using the AES Algorithm. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems – CHES 2004, 6th International Workshop, Cambridge, MA, USA, August 11-13, 2004, Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 357–370. Springer, August 2004.

[5] International Organisation for Standardization (ISO). Information Technology - Security Techniques - Entity authentication mechanisms - Part 3: Entity authentication using a public key algorithm, 1993.

[6] International Organisation for Standardization (ISO). ISO/IEC 9798-2: Information technology – Security techniques – Entity authentication – Mechanisms using symmetric encipherment algorithms, 1999.

[7] International Organisation for Standardization (ISO). ISO/IEC 18092: Information technology - Telecommunications and information exchange between systems - Near Field Communication - Interface and Protocol, April 2004.

[8] M. Jakobsson, P. D. MacKenzie, and J. P. Stern. Secure and Lightweight Advertising on the Web. *Computer Networks (Amsterdam, Netherlands)*, 31(11–16):1101–1109, 1999.

[9] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. Series on Discrete Mathematics and its Applications. CRC Press, 1997. ISBN 0-8493-8523-7, Available online at http://www.cacr.math.uwaterloo.ca/hac/.

[10] National Institute of Standards and Technology (NIST). FIPS-197: Advanced Encryption Standard, November 2001. Available online at http://www.itl.nist.gov/fipspubs/.

[11] V. Patil and R. Shyamasundar. e-coupons: An Efficient, Secure and Delegable Micro-Payment System. In *Information Systems Frontiers*, volume 7, pages 371–389. Springer, December 2005.

[12] T. Shojima, Y. Takada, N. Komoda, H. Oiso, and A. Hiramatsu. An Incentive Attached Peer to Peer Electronical Coupon System. In M. Hamza, editor, *Proceedings Communications, Internet, and Information Technology - CIIT 2003*, November 2003.