mCoupons – Der digitale Gutschein für mobile Geräte

Der fälschungssichere digitale Gutschein, gedacht für mobile Geräte wie Handys, PDAs und Smartphones, wurde im Zuge des Projektes SNAP (Secure NFC Applications) entwickelt. Dieses Gemeinschaftsprojekt bestehend aus den Projektpartnern IAIK (TU-Graz), EMT (TU-Graz) sowie den Studiengängen ETM und ISW der FH JOANNEUM GesmbH. wird von der FFG und der NXP (ehemals Philips) finanziert.

Zur Feststellung einer notwendigen Verschlüsselung der kommunizierten Daten mittels NFC (Near Field Communication) musste die Gefahr des Abhörens und möglicher Beeinflussung der NFC Elektronik eruiert werden. Auch wenn die Gefahr eines „Abhörens" durch die geringe Sendeleistung und dem dadurch geringen maximalem Abstand beider NFC Geräte bei nur ca. 1,5 cm liegt kann durch bestimmte Umgebungsbedingungen (z.B. Metallrohre in Wänden, Metallgeländer) das Elektromagnetfeld physikalisch verschleppt und theoretisch an entfernter Stelle abgehört werden. Diese theoretische Sicherheitslücke wurde mittels Messungen und Versuchen durch das EMT (Institute of Electrical Measurement and Measurement Signal Processing) ermittelt.
Weiters wurde durch das EMT die Möglichkeit von Side-Channel Attacken (Magnetfeldveränderungen, Spannungsveränderungen) des bestehenden AES Security Chips aufgezeigt.

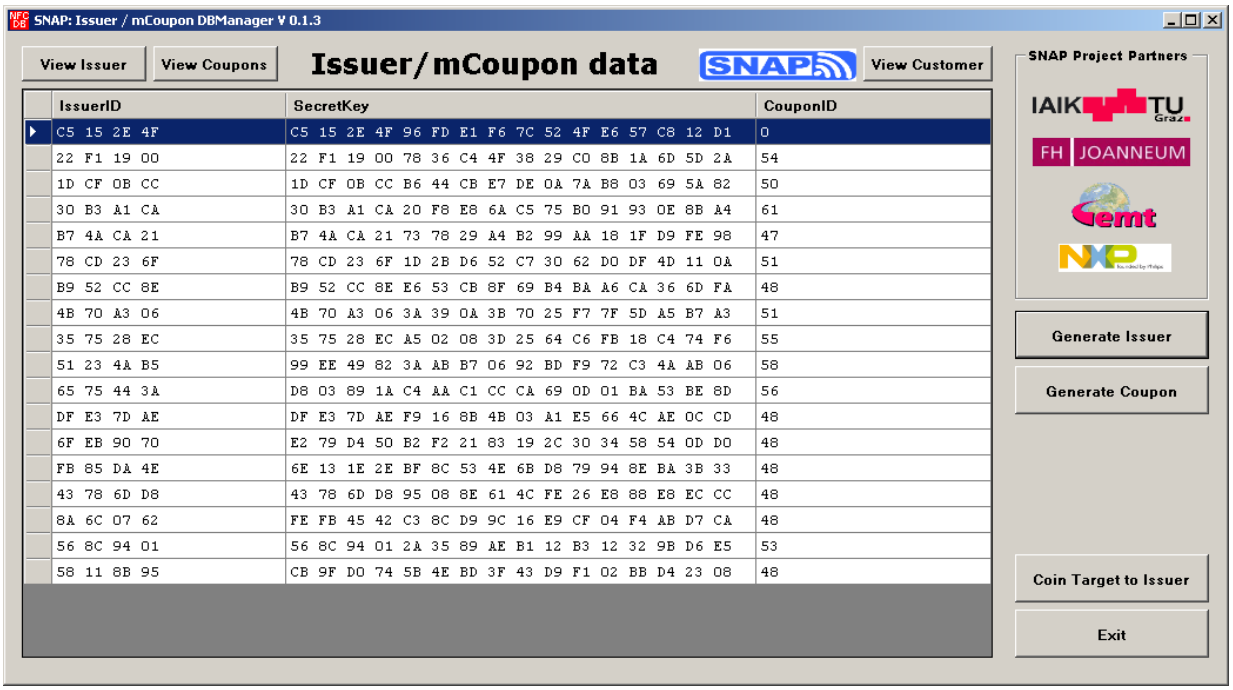
Aufgrund dieser Erkenntnis wurde vom Projektpartner IAIK (Institute for Applied Information Processing and Communication) der von der IAIK im Vorgängerprojekt ART entwickelte AES (Advanced Encryption Standard) Security Chip zur Verbesserung der Datensicherheit erweitert.
Um auch im Falle eines „Mithörens und Aufzeichnens" während des drahtlosen Datenaustausches oder durch das hacken (verändern) gespeicherter mCoupon Daten einen Missbrauch (z.B. unberechtigte Vervielfältigung und unberechtigtes einlösen) von mCoupons zu verhindern wurde das mCoupon Protokoll entwickelt. Dieses Protokoll kann in einfacher (simple) oder in der erweiterten (extended) Variante eingesetzt werden, wobei es bei der Verwendung des erweiterten Protokolltyps zusätzlich möglich ist, sogenannte digitale Zertifikate zu verwenden.

Um die NFC Technologie für den Benutzer akzeptabel zu machen war es weiters Voraussetzung dieses NFC Feature immer (bei eingeschaltetem Gerät) in Betrieb zu halten (anders als bei z.B. Bluetooth oder WLAN welches der Benutzer immer manuell ein- und ausschalten muss). Um bei einem ständigem Betrieb des NFC Features eine energiesparende Erkennung (wichtig für mobile Geräte) „gleichgesinnter" NFC-Devices zu ermöglichen wurde vom Studiengang ETM (Elektronik und Technologiemanagement der FH JOANNEUM) eine spezielle, extrem energiesparende, Neighborhood – Detection entwickelt. Mittels dieser Detektierung werden im nahen Umfeld benachbarte NFC Geräte eruiert. Durch diese energiesparende Methode ist das explizite ein- und ausschalten der NFC Funktionalität nicht Notwendig (wie bei Bluetooth oder WLAN).

Zur Demonstration des Zusammenwirkens der NFC Hardware und der Sicherheitseinrichtungen sowie einer möglichen realen Nutzung der mCoupons wurde vom Studiengang ISW (Infrastrukturwirtschaft der FH-JOANNEUM) der Prototyp eines digitalen Gutschein Ausfolge- und Einlöse- Systems entwickelt.

Der von ISW (Infrastrukturwirtschaft der FH-JOANNEUM) entwickelte Prototyp besteht aus dem mCoupon Aussteller (Issuer) welcher das mCoupon per Near Field Kommunikation (NFC) dem Client (Initiator) nach dessen Wunsch (der Client ist die Kunden-Applikation welche den Empfang eines mCpoupons initiiert, daher auch Initiator genannt) übermittelt.
Diese Client- (Initiator) Applikation ist die eigentliche, für den Kunden relevante Applikation, welche je nach NFC fähigem mobilem Gerät entsprechend gestaltet werden muss. Durch Interaktion des Benutzers, z.B. betätigen eines „Thouch"-Buttons in der Nähe (aufliegend – 1,5 cm Entfernung) zu einem mCoupon Aussteller (Issuer) wird die Übermittlung des mCoupon angefordert und über das DEP (Data exchange protocol) der beiden NFC-Devices übertragen.

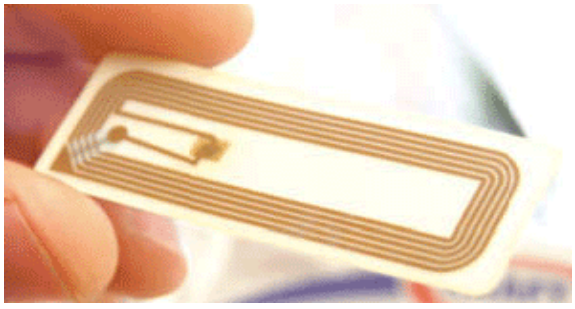
Um eine sichere Ausstellung und Einlösung eines mCoupon zu gewährleisten muss eine funktionierende mCoupon-Infrastruktur vorhanden sein. Dazu gehört das Anlegen der Aussteller-ID mit einem geheimen AES Schlüssel in eine (zentrale) Datenbank. Bei unserem Prototyp wird in eine (derzeit lokale Datenbank) gespeichert und das Verwalten der Issuer- und mCoupon Daten erfolgt über die DBManager-Applikation.



IssuerID	SecretKey	CouponID
C5 15 2E 4F	C5 15 2E 4F 96 FD E1 F6 7C 52 4F E6 57 C8 12 D1	0
22 F1 19 00	22 F1 19 00 78 36 C4 4F 38 29 C0 8B 1A 6D 5D 2A	54
1D CF 0B CC	1D CF 0B CC B6 44 CB E7 DE 0A 7A B8 03 69 5A 82	50
30 B3 A1 CA	30 B3 A1 CA 20 F8 E8 6A C5 75 B0 91 93 0E 8B A4	61
B7 4A CA 21	B7 4A CA 21 73 78 29 A4 B2 99 AA 18 1F D9 FE 98	47
78 CD 23 6F	78 CD 23 6F 1D 2B D6 52 C7 30 62 D0 DF 4D 11 0A	51
B9 52 CC 8E	B9 52 CC 8E E6 53 CB 8F 69 B4 BA A6 CA 36 6D FA	48
4B 70 A3 06	4B 70 A3 06 3A 39 0A 3B 70 25 F7 7F 5D A5 B7 A3	51
35 75 28 EC	35 75 28 EC A5 02 08 3D 25 64 C6 FB 18 C4 74 F6	55
51 23 4A B5	99 EE 49 82 3A AB B7 06 92 BD F9 72 C3 4A AB 06	58
65 75 44 3A	D8 03 89 1A C4 AA C1 CC CA 69 0D 01 BA 53 BE 8D	56
DF E3 7D AE	DF E3 7D AE F9 16 8B 4B 03 A1 E5 66 4C AE 0C CD	48
6F EB 90 70	E2 79 D4 50 B2 F2 21 83 19 2C 30 34 58 54 0D D0	48
FB 85 DA 4E	6E 13 1E 2E BF 8C 53 4E 6B D8 79 94 8E BA 3B 33	48
43 78 6D D8	43 78 6D D8 95 08 8E 61 4C FE 26 E8 88 E8 EC CC	48
8A 6C 07 62	FE FB 45 42 C3 8C D9 9C 16 E9 CF 04 F4 AB D7 CA	48
56 8C 94 01	56 8C 94 01 2A 35 89 AE B1 12 B3 12 32 9B D6 E5	53
58 11 8B 95	CB 9F D0 74 5B 4E BD 3F 43 D9 F1 02 BB D4 23 08	48

DBManager-Applikation des SNAP Prototyps

In dieser Applikation werden die Issuer- (ID / Schlüssel) und die mCoupon-Daten erstellt oder für Demonstrationszwecke automatisch generiert. Die Issuers, in Zukunft im Prinzip das selbe wie die bekannten RFID-Smartlabels aber mit dem speziellen IAIK Chip werden danach mit diesen Daten beschrieben und können auf, bzw. in Zeitungen, auf Werbetafeln und Werbeflächen, auf Flugblätter, Auslagen von Läden, usw. geklebt werden. Diese Klebe-Labels werden im Zusammenhang



RFID Smart Label / passive target

mit der NFC-Technologie auch passive Targets genannt. Durch die mobile Kundenapplikation, wie unten beschrieben, werden diese Daten angefordert und danach als mCoupon über das NFC Air-Interface gesendet und gespeichert. Derzeit wird dieses passive Target durch eine weitere Applikation simuliert, dem sogenannten Tagsimulator.

Tagsimulator

Dieser Tagsimulator kommuniziert mittels eines von NXP zur Verfügung gestellten USB- oder RS232- PN53x NFC Board mit anderen NFC Geräten. Das Board wird über die Applikation in den Target-Mode versetzt und simuliert so ein zukünftiges passives Target. Die für einen gültigen Aussteller notwendigen Daten werden nach dem Anlegen in der Datenbank (DBManager) mittels eines zweiten PN53x Boards über das Air-Interface an den Simulator übertragen. Ein Prototyp des zukünftigen realen Issuers wird derzeit von der IAIK entwickelt.

Der Aussteller (Issuer) bestimmt nach der Initiierung des Clients den Protokoll-Typ (simple oder extended). Im Falle des einfachen Protokolls werden die mCoupon Daten und ein AES verschlüsseltes Datenpaket, bestehend aus den mCoupon Daten, als mCoupon an den Client gesendet und auf diesem gespeichert. Mittels dieser Kundenseitigen-(Client)-Applikation können die empfangenen mCoupons verwaltet (gelöscht, eingelöst, eingesehen) werden.

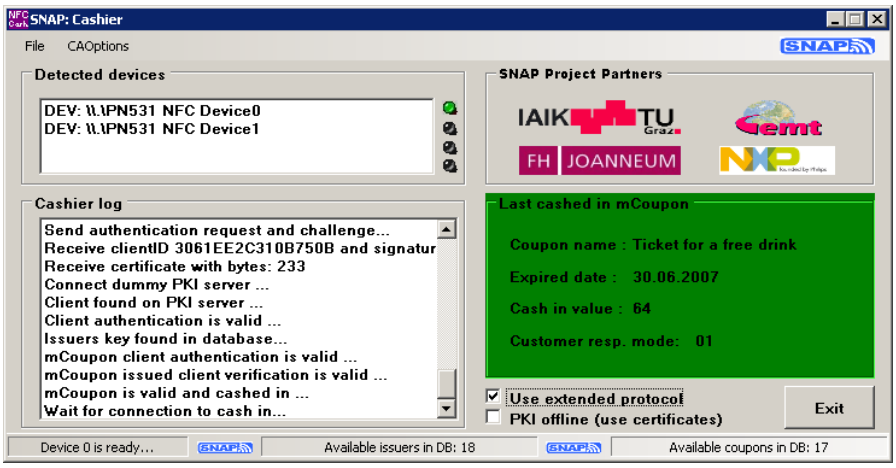
Client- (Initiator-) Applikation am PDA

Unter der Verwendung des einfachen Protokolls empfangene und gespeicherte mCoupons können bei Bedarf unter mobilen Geräten (soweit eine standardisierte Speicherung oder Konvertierungs-Schnittstellen zwischen den einzelnen Applikationen vorhanden sind) weitergegeben (z.B. über NFC-DEP, Bluetooth, WLAN, ActiveSync) werden.
Durch den verschlüsselten Zusatz des übertragenen mCoupons ist ein nachträgliches hacken (verändern) der mCoupon-Daten und eine gültige spätere Einlösung nicht möglich.

Wird durch den Aussteller (Issuer) das erweiterte Protokoll vom Client (Initiator) angefordert muss dieser (spätestens beim einlösen des mCoupons, bei der Ausstellung aber schon gültige RSA-Keys) eine registrierte (z.B. beim mCoupon Provider) Client-ID und den zugehörigen öffentlichen Schlüssel besitzen (genaueres dazu etwas später). Durch die Anforderung des Ausstellers an den Client muss dieser, die vom Aussteller gesendeten zufällig generierten Challenge, mit seinen RSA-Keys (privatem und öffentlichen Schlüssel) verschlüsselt (signiert) an den Aussteller zurücksenden. Der Aussteller integriert diese Client-spezifische Signatur in

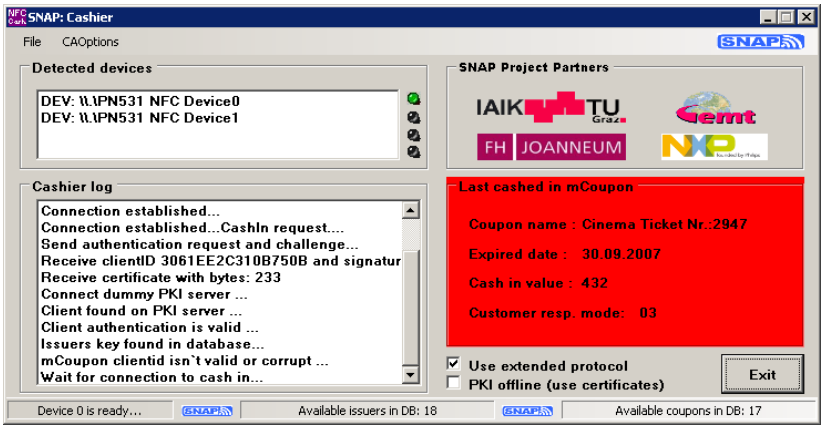
die zu sendenden mCoupon Daten welche an den Client übermittelt und dort gespeichert werden.
Unter der Verwendung des erweiterten Protokolls empfangene und gespeicherte mCoupons können unter Umständen zwar weitergegeben werden, da jedes empfangene mCoupon aber Client-spezifische, signierte Daten enthält können diese nur vom jeweiligen rechtmäßigen Empfänger (Client) auch wieder erfolgreich eingelöst werden.
Wie bei der Verwendung des einfachen Protokolls ist durch den verschlüsselten Zusatz des übertragenen mCoupons ein nachträgliches hacken der mCoupon-Daten und eine gültige spätere Einlösung nicht möglich.

Die vom Client gesammelten und gespeicherten mCoupons können nun an den dafür vorgesehenen „Kassen", den sogenannten Cashiers eingelöst werden. Für die künftige reale Anwendung könnte man sich eine erforderliche Infrastruktur ähnlich dem Bankomatkassen-System vorstellen. Zu unserem Proto-Typ gesellt sich eine weitere Applikation, der so genannte Cashier hinzu.



Cashier erfolgreiches einlösen

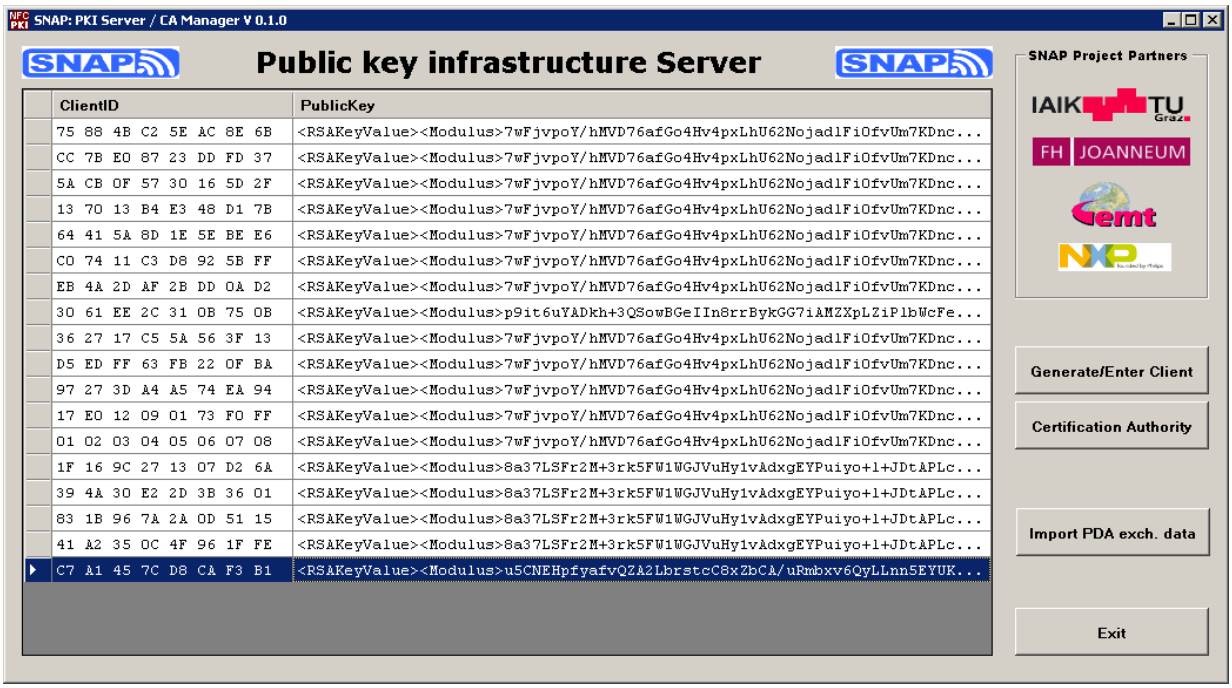
Dieser Cashier ist mit einem oder mehreren PN53x Boards ausgestattet welche, ähnlich dem Tagsimulator, in den Target Mode versetzt werden und auf einen Cash-In Request der Kundenapplikation wartet. Erfolgt dieser Request im Lesebereich (aufliegend – 1,5 cm) eines Cashier NFC Devices, z.B. durch klicken des CashIn-Buttons (nach der Auswahl des gewünschten mCoupon in der mCoupon-Liste) wird, im Falle des einfachen Protokoll-Typs, das selektierte mCoupon an den Cashier gesendet und der AES (Advanced Encryption Standard) verschlüsselte Zusatz des mCoupons entschlüsselt und verglichen.



Cashier – erfolgloses einlösen

Um den erforderlichen geheimen Schlüssel zu erhalten verbindet sich der Cashier mit der Issuer Datenbank und bekommt über die Issuer-ID den zugehörigen geheimen Schlüssel als Grundlage für eine erfolgreiche Entschlüsselung mitgeteilt. Ist die Issuer-ID nicht vorhanden, der Schlüssel und/oder das entschlüsselte Resultat falsch, oder das Gültigkeitsdatum des mCoupon abgelaufen kann das mCoupon nicht eingelöst werden.

Für die Verwendung des erweiterten Protokolls werden zusätzlich zu den mCoupon Aussteller-Daten wie ID und geheimer AES-Schlüssel noch Client spezifische Sicherheitsdaten benötigt. Um mCoupons mittels erweitertem Protokolls erfolgreich zu empfangen und danach einzulösen benötig jede Kundenapplikation eine eigene Client-ID und ein eigenes RSA-Schlüsselpaar (private and public key) welche derzeit beim erstmaligen Start der Applikation automatisch generiert werden. Die Client-ID und der öffentliche Teil des RSA-Schlüssels müssen nun, ähnlich der Aussteller-ID und des AES Schlüssels, in der Datenbank eines PKI (Public Key Infrastructure) – Servers registriert (gespeichert) werden.



PKI-Server und CA Applikation

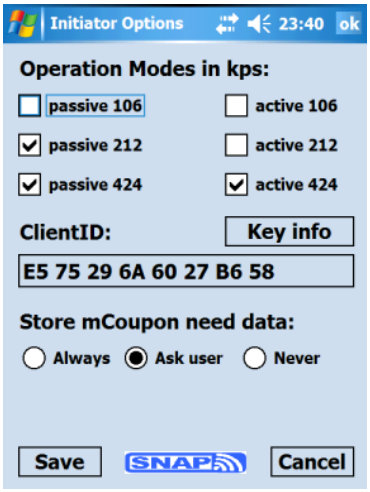
Um dies zu realisieren wurde die PKI-Server Applikation entwickelt welche die Client-IDs und die zugehörigen öffentlichen Schlüssel des RSA-Schlüsselpaares der einzelnen Clients verwaltet. Zur Generierung und Verwendung von digitalen Zertifikaten, wurde in dieser Applikation zusätzlich die Funktionalität einer autorisierten Zertifizierungsstelle (CA - Certification Authority) implementiert. Diese CA kann, wiederum über ein eigenes RSA-Schlüsselpaar für die jeweiligen Clients ein digital signiertes, Client spezifisches Zertifikat ausstellen.

Die vom Client generierte ID und der öffentliche Teil des Schlüssel müssen entweder händisch in die PKI-Server Datenbank eingetragen (z.B. Cut & Paste) oder importiert werden. Die Client Applikation erstellt aus der ID und dem öffentlichen Schlüssel auch ein Exchange Datenfile das in die PKI-Server Applikation importieren werden kann.

Nachfolgend eine möglichst informative, aber allgemein gehaltene Erläuterung der Funktionalität des Verfahrens zum erweiterten Protokoll:
Zur erfolgreichen Ausstellung und Einlösung eines mCoupons mittels des erweiterten Protokolls wird vom Client ein Ausstellungs-Request an den Aussteller gesendet. Daraufhin fordert der Aussteller (Issuer) die ID des Clients und eine Signatur mittels eines Authentication Request mit einem zufällig generierten Challenge vom Client an.

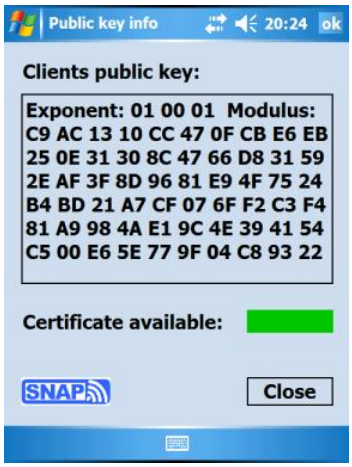
Der Client sendet seine Client-ID und die, anhand des Client RSA-Schlüsselpaares signierten Challenge an den Aussteller zurück. Die Client-ID und das signierte Datenpaket werden in den mCoupon-Daten integriert und (zusätzlich zum AES verschlüsseltem Datenpaket des Ausstellers) als mCoupon an den Client übermittelt und im Empfängergerät gespeichert.

Beim Einlösen eines mCoupons des erweiterten Protokoll-Typs wird nach dem Cash-In Request (siehe oben) vom Cashier ebenfalls ein Authentication Request und eine zufällig generierter Challenge an den Client (Initiator) gesendet.
Der Client sendet nun seine Client-ID und die, anhand des Client RSA-Schlüsselpaares signierten Challenge an den Cashier zurück. Der Cashier stellt danach eine Verbindung zum PKI-Server her und fordert anhand der empfangenen Client ID den zugehörigen öffentlichen Schlüssel aus der Datenbank an. Wird die Client ID in der Datenbank gefunden kann die vom Cashier gesendete Challenge und das vom Client signierte Ergebnis miteinander verglichen werden. Stimmen original Challenge und verifiziertes (mittels öffentlichen Schlüssels) Resultat miteinander überein ist die Authentifizierung des Clients erfolgreich abgeschlossen.
Wird die Client ID in der Datenbank nicht gefunden oder der Vergleich anhand des gefundenen öffentlichen Schlüssels korreliert mit der vom Client gesendeten Signatur nicht ist die Authentifizierung nicht erfolgreich und der Vorgang wird abgebrochen.



Client Option Daten (ID)

Bei erfolgreicher Authentifizierung wird vom Cashier ein Auhtentication Ready-Request zur Aufforderung das mCoupon zu übermitteln an den Client gesendet. Nach dem Empfang des mCoupons wird, wie beim einfachen Protokoll-Typ, über die Aussteller-ID der geheime AES Schlüssel aus der Datenbank angefordert und der verschlüsselte Zusatz der mCoupon Daten entschlüsselt und mit den eigentlichen mCoupon Daten verglichen.
Ist die Issuer-ID nicht vorhanden, der Schlüssel und/oder das entschlüsselte Resultat falsch, oder das Gültigkeitsdatum des mCoupon ist abgelaufen kann das mCoupon nicht eingelöst werden.



Schlüssel und Zertifikat

Unter der Verwendung des erweiterten Protokolls ist es auch möglich digitale Zertifikate auszustellen um ein erfolgreiches einlösen der mCoupons auch dann zu gewährleisten wenn eine permanente Onlineverbindung zum entsprechenden PKI-Server aus Gründen der Infrastruktur oder durch den temporären Ausfall der Onlineverbindung nicht möglich ist.

Wie oben beschrieben ist die PKI-Server Applikation in der Lage, ein aus den Client relevanten Sicherheitsdaten (ID und public key) über ein eigenes, für alle Zertifikate dieser Instanz gültiges, RSA-Schlüsselpaar signiertes Datenpaket (Certificate) auszustellen. Dieses Datenpaket beinhaltet unter anderem die Client-ID und den

öffentlichen Teil des Client spezifischen RSA-Schlüsselpaares. Dieses Zertifikat wird mittels dem CA (Certification Authority) eigenen RSA-Schlüsselpaares digital signiert und, für unseren Prototypen, abgespeichert und kann anschließend durch die Client Applikation importiert werden.

Der Einlösevorgang unter Verwendung von Zertifikaten erfolgt ähnlich wie ohne vorhandene Zertifikate. Verfügt der Client über ein Zertifikat wird dieses nach dem Authentication Request des Cashiers zusätzlich zur Client ID und dem signierten Challenge des Cashiers mitgesendet. Der Cashier, welcher in diesem Fall über den öffentlichen Schlüssel der CA verfügen muss, verifiziert die Gültigkeit des empfangenen Zertifikates über den öffentlichen Schlüssel der CA. Ist diese Verifizierung erfolgreich wird die aus dem Zertifikat erhaltene Client ID mit der vom Client empfangenen ID verglichen. Sind die beiden IDs identisch wird über den, im Zertifikat enthaltenen öffentlichen Schlüssel, die vom Client empfangene Signierung die Challenge des Cashier verifiziert. Ist auch diese Verifizierung gültig wurde der Client erfolgreich authentifiziert. Anschließend erfolgt die Überprüfung der mCoupon Daten wie oben beschrieben.

Durch die Verwendung von Zertifikaten ist eine Onlineverbindung zu einem PKI Server nicht notwendig, der jeweilige Cashier benötigt zur Verifizierung der Zertifikate nur den öffentlichen Teil des RSA Schlüssels der jeweiligen CA.

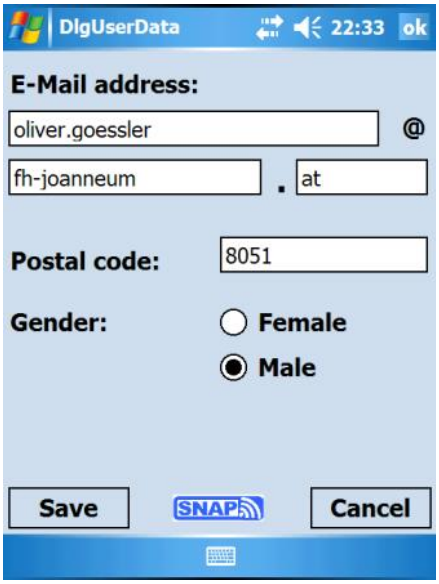
Das ganze klingt nun vielleicht etwas kompliziert, aber für den zukünftigen Einsatz dieser digitalen Gutscheine wird sich der potenzielle Benutzer über interne Vorgänge sowie Schlüsseln hier und Schlüsseln da keine Gedanken machen müssen. Nach dem Aufbau einer entsprechenden Infrastruktur wird sich der Benutzer nur die, für sein mobiles Gerät geeignete, Client (Initiator) Applikation des Anbieters downloaden und installieren müssen. Key und ID Generierung, Registrierung bei einer PKI, Zustellung und Integration/Installation eines Zertifikates werden dann automatisch beim Download oder der Installation der Applikation erfolgen.

mCoupon Inhalt - Info des Clients

Zur Demonstration einiger zukünftiger Varianten der mCoupons wurde in diesem Prototyp die Möglichkeiten eingebaut einen Informations- oder Werbetext mitzusenden, und diesen auf Wunsch bei Empfang durch den Client automatisch anzuzeigen, sowie akustische Signale zu lassen.

Werbetext nach mCoupon Empfang

Weiters ist es möglich für spätere Kundenanalysen oder Statistiken Kundenspezifische Daten mitzusenden, in diesem Prototyp wurde z.B. die E-Mail Adresse, die Postleitzahl und das Geschlecht realisiert.
Wird von einem empfangenen mCoupon einer dieser drei Userdaten für eine erfolgreiche Einlösung benötigt werden die entsprechenden Benutzerdaten aus der Clientkonfiguration eruiert und mit dem entsprechenden mCoupon abgespeichert.
Beim Einlösen werden die Benutzerdaten, soweit vom jeweiligen mCoupon gefordert, in einer Kundendatenbank gespeichert und können danach wenn gewünscht ausgewertet werden.



Userdaten Option des Clients

Autor: Oliver Gößler, Infrastrukturwirtschaft an der FH JOANNEUM GesmbH.

.