Simplified WiFi hotspot payment with mobile phones using IMS and NFC

Abstract

In this paper a novel service is presented which allows a user to pay for a WiFi hotspot using his or her mobile phone. This reduces the need to enter sensitive information into a web page with unknown trustworthiness in a public space. It also is very convenient, especially when the mobile phone should be used for surfing. If the hotspot is in a cafe, then the time spent waiting in line for service could be used to perform the payment. To prove that it works a prototype is created.

Table of contents

1.Introduction

When a user takes his laptop to a commercial WiFi hotspot he or she has to deal with a variety of inconveniences. In most cases the user is welcomed by a webpage of the WiFi HotSpot Operator (WHO) where he is prompted to buy a credit (in some way) to use the hotspot. For this the user needs to give the operator some payment details even though the user does not necessarily trust the operator. Another problem might be that none of the required payment options can be fulfilled by the user. Also in case of attempting to authorize a device with limited screen space such as an iPhone or a PDA the procedure might become very complicated if not impossible.

The goal of this paper is to outline a system that enables the user to securely and conveniently pay for hotspot access and authenticate as many devices as the user wishes with as few steps as possible.

1.1.Requirements

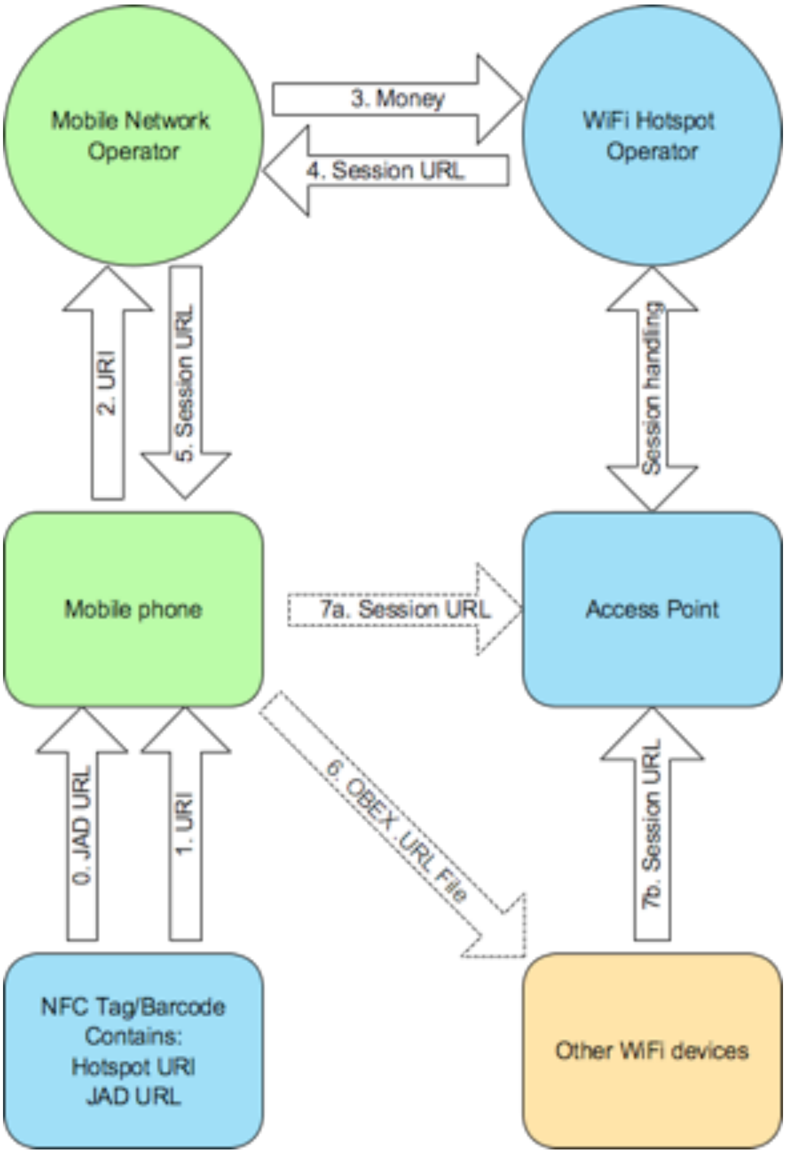
A trusted entity is required that can handle payment details. Many people have a mobile phone subscription via a contract with a mobile network operator (MNO) in order to use their phone. The MNO has some way to get paid by the customer (subscriber): be this via a prepaid contract, post paid contract, or any other kind of contract. Therefor the user already trusts the MNO enough to give them their money or give them a promise of payment. This trust relationship can be used by the subscriber in order to pay the WHO. The benefit for the customer is that the customer only needs to deal with one entity - specifically this is an entity that the subscriber already trusts and has a payment agreement with. The MNO often seeks to provide additional services to the customer and thus distinguish themselves from their competitors. Another benefit for the MNO is that some of those hotspots are already operated by the MNO, thus do not need agreements with foreign WHOs. However, some of these hotspots will not have agreements with this WHO, thus the MNO will need to establish agreements with them directly, via a broker, or via a clearance house.

Using the phone to implement the method of payment,requires that the phone needs to know the hotspot details the subscriber wishes to pay for. A problem here is that for example only 12 of 154 Sony Ericsson phones have WiFi[SEMC Phones] interfaces. So another technology needs to be used to collect the hotspot details in order that the subscriber can provide this information to their MNO. Without WiFi the user could use e a visual way of identifying the hotspot. Most hotspots have a sign that a hotspot is available. So the easiest way would be to require that the user enters the unique ID printed on the sign. However, that is not very convenient. There are at least 2 other simpler ways. One is to print a barcode on the sign and scan (or photograph) this; while the other method is to user near-field communication (NFC). Today 2D Barcodes are increasingly popular and are widely used for mobile tagging[WP:Mobile tagging]. They frequently contain a URL that can be visited with the mobile phone. However, for each kind of tag a software is required to decode them. NFC is a technology based on RFID and standardized by the NFC Forum. By using NFC the mobile phone can read the contents of RFID tags by approaching the tag with the phone. Using the NFC Forum's smart poster specification[NFC Specs] the user only needs to touch the sign and they will be prompted to either open the URL or for specially programmed records a custom application can be launched automatically.

Once the user has the ID of the hotspot, the next step is providing this information to their MNO in order to pay. However, the WHO needs to know which devices the user wants to use. One way is that the user tells the hotspot some unique device id that should be use with the hotspot, the other is that the user uses some token to tell the hotspot that the device is legitimate. The first option would require the user to enter some information and probably manually. This is inconvenient. The token on the other hand needs to be transfered from the mobile phone to the device(s). This could be done by the using bluetooth object exchange (OBEX) which is supported by 146 out of 154 Sony Ericsson phones.

To summarize, the best experience for the user comes with an NFC enabled phone and a device that support bluetooth OBEX. As the NFC is not very wide spread yet, an alternative is to use the camera for the barcode. Also the phone needs to be able to execute a program for selecting the WiFi plan and authenticate against the MNO.

2.System architecture



At first the user approaches the visual sign with his mobile phone. Now there are 2 cases. Either the user uses this service for the first or he already has the required application installed. As a first time user he retrieves the URL for the Java program from the sign by either typing it in manually, reading the NFC tag or reading the Barcode. Once the application is installed the unique ID in form of an URI for this particular hotspot is retrieved. This URI is send to the MNO via IMS by the application. The MNO then pays the WHO for the service and charges the users account. The WHO responds with at least one unique session URL which is forwarded to the users phone. If the phone supports WiFi the user can visit the session URL and start surfing immediately. In most cases however he probably wants to transfer the session URL from the mobile phone to the device which can be done from within the application by using bluetooth OBEX. With the visit of the session URL the device that invokes the URL is bound to the hotspot.

2.1.Step details

2.1.1.Installing the software

For the proposed system a software on the mobile phone is required. The reason for this is that an application provides a multitude of authentication possibilities, while a simple web page visit might not yield enough information. For example an application could send the International Mobile Subscriber Identity (IMSI) or Temporary Mobile Subscriber ID (TMSI). Even special functions on the SIM card could be used. This guarantees that the mobile phone indeed is a subscriber of this service.

However this application needs to be installed on the mobile phone of the user. There are numerous possibilities all of which require the user to use some common sense to not install malware. For example when instructions for installing the application are printed on the sign the user needs to judge whether those information are authentic or they might have been altered. An attacker for example could print its own sign which contains a slightly different URL/Phone number which then installs the bad software on the users phone. When the user bought the phone at the MNO store it would be best if the software would come pre-installed. If the user has not installed the software it could be send to the mobile phone by providing a URL where the program could be downloaded. Another alternative is a bluetooth OBEX beacon which sends the application to every visible phone. This has the downside that users that are just passing by are also spammed, even if they are not interested in using the hotspot and it increases the costs of a sign significantly. So the cheapest and most convenient way is to provide it as a mobile tag barcode, a printed URL and as a NFC smart poster. Especially the NFC smart poster way is very convenient because it requires only a short touch with the phone instead of aiming the camera or typing the URL.

2.1.2.The URI of the hotspot

With the application in place it is easy to instruct the user to walk to the sign and retrieve the URI by either touching it (NFC) or taking a picture. The URI contains information that uniquely identify this hotspot. So at least it needs to contain the operator and an identification of the hotspot. For example: urn:who:homerun.telia.se:stockholm:kista_gallery_food_court.

For advanced security purposes the URI might contain a time variant part which could guarantee that the user is physically present at this sign. This however only works for the NFC tag because printed barcodes are not alterable. A small display might display the time variant number which the user would need to enter when he is using the barcode. However all this effort is questionable. It increases the costs of the sign significantly without gaining many benefits.

Some attack scenarios: One possible way to abuse the system would be to buy credits for the hotspot without using it. This would be beneficial for the WHO because he gets money without doing anything. This could be done by emulating the software on a computer. Another scenario would be a user buying credits using his MNO. For example by replaying a listened authentication.

Both scenarios are depending on the fact that it is possible to use the service from anything but a mobile phone without a valid SIM card. So it is crucial that the application makes sure that it is the correct SIM card and uses a challenge response system which can only work with the correct SIM card. If the attacker is in possession of the real SIM card or a copy the user is in big trouble anyway. With a valid SIM card it is in the MNOs interest to make sure that someone is paying the bill at the end of the day. So, yes the user can buy credits without being in place, however if such an abuse is detected the MNO can determine the cell IDs the user is currently in at the next time such a credit payment is due and verify that it is within a reasonable area around the hotspot. While fraud might be possible it is also easy to detect for the MNO.

2.1.3.Buying credits

The user sends the URI of the hotspot to the MNO. The MNO then looks whether there is some agreement with the WHO and if yes what credits are available. Those credit information are then transfered to the mobile phone and the application presents them. The user should be able to choose what kind of credit he wants (time or volume based credits) and for how many devices he wants to buy them. The selection is then transfered to the MNO who forwards it to the WHO. The WHO responds with a URL that contains a unique session ID.

2.1.4.Using the session

The session ID needs to remain secret as it can be used to authorize devices for using the credits. So it needs to be a HTTPS URL with an IP. This avoids that upon first invocation on the hotspots access point an attacker can sniff it and use the credit for its own purpose. When the session ID is invoked for the first time the access point registers the devices' MAC address with the session. If the number of allowed devices that were bought with the credit is exceeded the new device will be rejected. This protects the user from unauthorized usage of his credit in case a sniffing attack was successful. While the transfer of the session ID to the mobile phone can be seen as well protected a transfer from the phone to another device yields potential security problems that have to be considered. The bluetooth OBEX is fairly secure when it is initiated from the mobile phone. A summary of bluetooth security[Bluetooth security] shows that most attacks are based on pulling or attacking the authentication of bluetooth not on decrypting a self initiated transfer. It also has to be considered that the session ID is very likely to be used shortly after the transfer, which shortens the window of opportunity significantly.

3.Implementation

The mobile phone application is implemented in Java Mobile Edition (JME). It uses the Java Specification Request (JSR) 281[JSR 281] for communication to the IMS server and JSR 82[JSR 82] for the bluetooth OBEX. The hotspot sign is a programmed Mifare classic 1k card[Mifare Info] with an Mifare Application Directory (MAD)[Mifare MAD] indicating NFC records in sector 1-5. The first NFC Data Exchange Format (NDEF) record is of the well known type smart poster. This enables readers without the supporting application to get more information about usage of this access point. The second record is a custom record and contains the URN which is used by the application to identify the hotspot. On the hotspot operator side a web-service is responsible for creating the sessions.

3.1.IMS

Ericsson provides an IMS environment which earlier was called IMSInnovation. This has been renamed to Mobile Java Communication Framework (MJCF)[Ericsson MJCF]. Unfortunately the person responsible for creating the IMS accounts was on vacation and without account it was not possible to use the IMS platform. So the phone is directly contacting the hotspot operator's web-service at the moment.

3.2.OBEX

For the transfer of the session URL the bluetooth Object Push Profile (OPP)[Bluetooth OPP] is used. At first a device discovery is launched. Currently the implementation does not show all devices but selects the MacBook. Then a service discovery is started for the short form UUID 1105 as specified in the Assigned Numbers - Service discovery Protocol[Bluetooth assigned numbers]. Then a file push is started. The type is "text/plain" and the file name "session.url". The file format is a very simple .url file.

```
[InternetShortcut]
URL=<sessionURL>
```

This file format is understood by most operating systems and usually assigned to open the browser. At least on the Mac the user can choose to automatically accept and open files from trusted (paired) devices. So the URL can automatically be opened after reception.

3.3.NFC

In the first sector of a NFC Mifare classic card is usually a MAD.

```
Page 0: 1: 00 01 03 e1 03 e1 03 e1 03 e1 03 e1 00 00 00 00 ................
Page 0: 2: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
Page 0: 3: a0 a1 a2 a3 a4 a5 78 77 88 c1 d3 f7 d3 f7 d3 f7 ......xw........
Page 1: 7: d3 f7 d3 f7 d3 f7 7f 07 88 40 d3 f7 d3 f7 d3 f7 .........@......
Page 1:11: d3 f7 d3 f7 d3 f7 7f 07 88 40 d3 f7 d3 f7 d3 f7 .........@......
Page 1:15: d3 f7 d3 f7 d3 f7 7f 07 88 40 d3 f7 d3 f7 d3 f7 .........@......
Page 1:19: d3 f7 d3 f7 d3 f7 7f 07 88 40 d3 f7 d3 f7 d3 f7 .........@......
Page 1:23: d3 f7 d3 f7 d3 f7 7f 07 88 40 d3 f7 d3 f7 d3 f7 .........@......
```

In Page 0: 0 is the ID of the tag and some manufacturer information stored. It is not writable and not relevant to this application. The first byte 00 is a CRC checksum which most of the times is ignored. Thus it is ignored here as well. The second byte 01 is an info byte. It points to the card publisher sector and is not used here. The 2 bytes e1 03 indicates that a sector is used by an application registered to the NFC Forum (by Philips SC, 1 sector width) as specified in the MAD list of registered applications[MAD List]. The position of the 2 bytes refers to the sector it is describing. An increase in offset of 2 bytes increases the sector by 1 starting with sector 1. So the highlighted bytes are indicating an NFC application in sector 1. The next e1 03 are for sector 2 and so on. A sector with 00 00 is considered empty. The A-key a0 a1 a2 a3 a4 a5 and access bits 78 77 88 are required as described in the MAD specification section 3.9. The B-Key d3 f7 d3 f7 d3 f7 is probably specified in the document Application Note MIFARE Standard as NFC Forum Tag 1.1 which is only available upon request. The general purpose byte c1 is a required indicator for the MAD (MAD available=1, Multi-application card=1, MAD version code 01) as specified in MAD specification section 2.4. Each sector that is indicated as NFC Application sector by the MAD requires A and B Key d3 f7 d3 f7 d3 f7 and the access bits 7f 07 88. The general purpose byte is set to 40. Those are probably also specified in the application note.

For the sign a Mifare classic card is used. The first NDEF message is a smart poster, the second one a custom message. They are embraced by a Type Length Value (TLV) block.

```
Page 1: 4: 03 C6 91 02 6D 53 70 91 01 52 55 03 77 68 6F 6D ....mSp..RU.whom
Page 1: 5: 2E 70 72 69 63 65 72 61 74 2E 63 6F 6D 2F 6E 66 .pricerat.com.nf
Page 1: 6: 63 49 6E 66 6F 3F 75 72 6E 3D 68 6F 6D 65 72 75 cInfo.urn.homeru
Page 2: 8: 6E 2E 74 65 6C 69 61 2E 73 65 3A 73 74 6F 63 6B n.telia.se.stock
Page 2: 9: 68 6F 6C 6D 3A 6B 69 73 74 61 5F 67 61 6C 6C 65 holm.kista.galle
Page 2:10: 72 79 5F 66 6F 6F 64 5F 63 6F 75 72 74 11 03 01 ry.food.court...
Page 3:12: 61 63 74 00 51 01 0C 54 02 65 6E 4D 6F 72 65 20 act.Q..T.enMore.
Page 3:13: 69 6E 66 6F 54 15 3C 77 68 6F 6D 2E 70 72 69 63 infoT..whom.pric
Page 3:14: 65 72 61 74 2E 63 6F 6D 3A 68 73 6C D1 01 38 55 erat.com.hsl..8U
Page 4:16: 13 77 68 6F 3A 68 6F 6D 65 72 75 6E 2E 74 65 6C .who.homerun.tel
Page 4:17: 69 61 2E 73 65 3A 73 74 6F 63 6B 68 6F 6C 6D 3A ia.se.stockholm.
Page 4:18: 6B 69 73 74 61 5F 67 61 6C 6C 65 72 79 5F 66 6F kista.gallery.fo
Page 5:20: 6F 64 5F 63 6F 75 72 74 FE
```

This reads as following:

0x00: 03 c6 TLV block as specified in NFCForum Type 1 Tag section 2.4

03 = NDEF Message TLV - Contains the NDEF message

c6 = Length of block

0x02: 91 02 6D 53 70 NDEF Record as specified in NFCForum TS-NDEF section 3.2

91 = NDEF header Type name format (TNF) = 0x01 (NFC Forum well-known type as specified in NFCForum Record Type Description (RTD)) Short record (SR)=1, Message begin (MB)=1

02 = Length of record name

6D = Length of Smart Poster data

53 70 = ASCII “Sp” record type is of urn:nfc:wkt:Sp as specified in NFCForum SmartPoster RTD

0x07: 91 01 52 55 03 XX NDEF Record as specified in NFCForum TS-NDEF section 3.2

91 = NDEF header TNF = 0x01, MB = 1, SR = 1

01 = Length of record name

52 = Length of URI payload (82 bytes)

55 = ASCII “U” record typs is of urn:nfc:wkt:U as specified in NFCForum RTD URI

03 = “http://” as specified in NFCForum TS RTD URI section 3.2.2

XX = ASCII URL for information about this hotspot and software download.

0x5D: 11 03 01 61 63 74 00 NDEF Record as specified in NFCForum TS-NDEF section 3.2

11 = NDEF header TNF = 0x01, SR=1

03 = Length of record name

01 = Length of the “act” record name

61 63 74 = ASCII “act” record typs is a local type.

00 = Open browser as specified in NFCForum SmartPoster RTD section 3.3.3

0x64: 51 01 0C 54 02 65 6E XX NDEF Record as specified in NFCForum TS-NDEF section 3.2

51 = NDEF header TNF = 0x01, Message End (ME) = 1, SR = 1

01 = Length of record name

0C = Length of Text payload (12 bytes)

54 = ASCII “T” record typs is of urn:nfc:wkt:T as specified in NFCForum RTD Text

02 = Status byte: UTF-8, two-byte language code

65 6E = UTF-8 “en” ISO two-character language code: English

XX = UTF-8 Text description

0x74: 54 15 3C XX NDEF Record as specified in NFCForum TS-NDEF section 3.2

54 = NDEF header Type name format (TNF) = 0x04 (NFC Forum external type as specified in NFCForum Record Type Description (RTD)) SR=1, ME=1

15 = Length of record name

3C = Length of custom message (highlighted in light-purple)

XX = ASCII URN for external types as specified in NFCForum TS RTD section 2.2 translates to “urn:nfc:ext:whom.pricerat.com:hsl”

0x8C: D1 01 38 55 13 XX NDEF Record as specified in NFCForum TS-NDEF section 3.2

D1 = NDEF header TNF = 0x01, MB = 1, ME=1, SR = 1

01 = Length of record name

38 = Length of URI payload (82 bytes)

55 = ASCII “U” record typs is of urn:nfc:wkt:U as specified in NFCForum RTD URI

13 = “urn:” as specified in NFCForum TS RTD URI section 3.2.2

XX = ASCII URN for hotspot identification

0xCA: FE TLV block as specified in NFCForum Type 1 Tag section 2.4

FE = Terminator TLV - Marks end of Messages

3.4.Web-service

The web-service is implemented in JAX-WS which is shipped with JDK 6[JAX-WS]. The code is very lightweight and easy to understand.

```
@WebService
public class WirelessHotspotOperator {
       @WebMethod
       public URL[] buyCredit(
              @WebParam(name = "transactionReference") String transactionReference,
              @WebParam(name = "planID") String planID
              ) throws IllegalArgumentException {
          return generateURLs(); //Actually there is some more logic in here
       }
       @WebMethod
       public PlanInfo[] getPlans() {
              return PLANS;
       }
       public static void main(String[] args) {
              Endpoint.publish(
                     "http://whom.pricerat.com:8180/WirelessHotspotOperator",
                     new WirelessHotspotOperator());
       }
}
```

The MNO at first calls the getPlans() method to retrieve all possible data plans and forward it to the mobile phone. There the user chooses one and sends the chosen plan to the MNO. The MNO initiates a money transaction which is referenced by the transactionReference variable. If everything is ok, session URLs are returned. The number depends on the plan that was chosen. Each device gets its own sessionURL to avoid abuse.

The downside of using this simple JDK 6 method is that https is not available which is a no-go for a production system.

4.Conclusion

The prototype works very well and provides the designed benefits. When the application is installed the process of touching the sign, buying the credit and sending the session URL to the device takes less than 15 seconds. Only 4 clicks for confirming the plan are required. An ideal scenario would be a user waiting in line to buy a coffee while performing these actions. This way he can immediately start surfing and does not need to enter any sensitive information in public.

However the implemented prototype is not complete and lacks features such as storing the session URLs, choosing the device to send the data to or using the barcode. Those are mostly UI issues which need not much time to implement. In general the UI would require some serious polishing.

In order to further increase the security a dedicated application for notebooks could be considered. For example the session URL could be pre-bound to a random MAC address. Once the notebook receives this MAC address it could change its wireless LAN MAC address to this random MAC. This would have the benefit of a slightly increased security and of an increased privacy for the user. The WHO would only know which MNO the user is using without seeing any further details from the payment.

5.Appendix: References

SEMC Phones Phone Gallery

http://developer.sonyericsson.com/device/searchDevice.do?restart=true

WP:Mobile tagging Mobile tagging - Wikipedia, the free encyclopedia

http://en.wikipedia.org/wiki/Mobile_tagging

NFC Specs NFC Forum : Specifications

http://www.nfc-forum.org/specs/

Bluetooth security Bluetooth Security & Hacks. Andreas Becker. August 16, 2007

JSR 281 The Java Community Process(SM) Program - JSRs: Java Specification Requests - detail JSR# 281

http://jcp.org/en/jsr/detail?id=281

JSR 82 The Java Community Process(SM) Program - JSRs: Java Specification Requests - detail JSR# 82

http://jcp.org/en/jsr/detail?id=82

Mifare Info MIFARE Classic from NXP Semiconductors

http://www.nxp.com/#/pip/pip=%5Bpfp=41863%5D%7Cpp=%5Bt=pfp,i=41863%5D

Mifare MADAN MAD Mifare application directory

http://www.nxp.com/acrobat/other/identification/M001830.pdf

Ericsson MJCF Mobile Java Communication Framework | Ericsson Labs Developer

http://developer.labs.ericsson.net/apis/mjcf

Bluetooth OPP Bluetooth.com | OPP

http://www.bluetooth.com/Bluetooth/Technology/Works/OPP.htm

Bluetooth assigned numbers Bluetooth assigned numbers

http://violator.kiev.ua/docs/Bluetooth_assigned_numbers_SDP.pdf

MAD List MIFARE Application Directory (MAD); list of registered applications

http://www.nxp.com/acrobat/other/identification/MAD_overview.pdf

JAX-WS Introducing JAX-WS 2.0 With the Java SE 6 Platform, Part 1

http://java.sun.com/developer/technicalArticles/J2SE/jax_ws_2/