IDA-Pay: a secure and efficient micro-payment system based on Peer-to-Peer NFC technology for Android mobile devices

Luca Mainetti, Luigi Patrono, and Roberto Vergallo

Original scientific paper

*Abstract:***The evolution of modern mobile devices towards novel Radio Frequency (RF) capabilities, such as Near Field Communication, leads to a potential for delivering innovative mobile services, which is still partially unexplored. Mobile proximity payment systems are going to enhance the daily shopping experience, but the access to payment security resources of a mobile device (e.g. the "Secure Element") by third party applications is still blocked by smartphone and Operating System manufacturers. In this paper, the IDA-Pay system is presented, an innovative and secure NFC micro-payment system based on Peer-to-Peer NFC operating mode for Android mobile phones. It allows to deliver mobile-to-POS micro-payment services, bypassing the need for special hardware. A validation scenario and a system evaluation are also reported to demonstrate the system effectiveness and performance.**

***Index Terms:*NFC, Android, Micro-payments, Performance Evaluation.**

I. Introduction

Recently, mobile devices are playing a very important role in linking humans and networks. Currently, a mobile phone can be used to update Facebook status, check-in to physical places, share digital music, and more yet to come. In particular, mobile payments are receiving special attention from a number of business actors: bank institutions, telecom operators, shopkeepers, technology providers, and so on [1].

Mobile payments involve two different areas:

- the Mobile Remote Commerce, i.e. the buying and selling of goods and services through the use of the smartphone (or similar devices);
- the Mobile Proximity Payment, i.e. the using of the smartphone to enable proximity payments.

The adoption of Near Field Communication (NFC) technology in today's smartphones is essential for the delivery of mobile proximity payments. NFC follows exactly the Internet of Things (IoT) paradigm, as it is considered the double-click of the IoT[2]. Cisco has designed an infographic[3] that offers a simple example of how IoT will affect our everyday life. It states that by 2020, there will be 50 billion "things" connected to the Internet - everything from our body, car, alarm clock and even cows.

The special interest of Google's Android towards NFC is favoring the spread of smartphones with embedded NFC readers; such devices are becoming popular and they are going to play a fundamental role in people's life, as they allow to supply a wide range of ubiquitous applications such as: access control, consumer electronics, healthcare, information collection and exchange, loyalty and coupons, payments and ticketing.

Mobile phones create a lot of secure and convenient conditions for payment operations, e.g., battery, keyboard, screen, storage, and 3G network. In future, these smartphones will represent the personal electronic wallet (e-wallet) for most people replacing the current plastic credit cards.

However NFC payments are yet to attain to their market potential [4]. This is mainly because only Google's Android [5] has reached a significant share of the NFC mobile market. Moreover, only the Google Wallet mobile application supports NFC micro-payments in the US for Android mobile phones. Third party applications cannot take advantage of the security resources embedded in Android devices because Google Wallet is the unique application having privileged access to such resource, namely the "Secure Element" (SE). The SE is a special memory area where trusted applications can store and retrieve sensitive user information, such as the credit card number and the Card Verification Value (CVV) code. Furthermore, Google Wallet is limited to work with affiliated credit card issuers (e.g. MasterCard, Visa), so no chance is given to alternative payment ecosystems.

The research work summarized in this paper aims to develop an innovative mobile micro-payment system which can be easily used in alternative ecosystems to implement custom payment scenarios. Such system must ensure the same security level of traditional credit card payments, without the need of any hardware intervention (SIM or SD cards replacement) by smartphone's owners. That is, the user has only to download the application from the market and install it onto the device. Moreover, multiple payment networks (e.g. credit cards, money transfer, couponing) should be easily configurable. Finally, the interaction between the user and the system must be bi-directional, so the system can return rich collectible feedbacks to the user.

Manuscript received November 21, 2012; revised January 7, 2013.

The material in this paper was presented in part at the 20th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2012), Split, Croatia, Sept, 2012.

Authors are with the Department of Innovation Engineering at University of Salento, Lecce, Italy (email: {luca.mainetti, luigi.patrono, roberto.vergallo}@unisalento.it).

1845-6421/12/8265 © 2012 CCIS

To achieve the abovementioned goals, a security architecture based on both symmetric and asymmetric encryption was designed. In our system, the end-user can securely access innovative and unlimited mobile services by waving his/her smartphone proximate to an ad-hoc NFC Point-Of-Sale (POS). We called our system IDA-Pay, as it was designed and developed in the IDA Lab (IDentification Automation Laboratory) of University of Salento (Italy). The system was validated in our laboratory (in vitro) considering the raised requirements; an effective validation (in vivo) with real actors has not been performed yet as it represents a future work.

This paperextends ourprevious work [6]including a performance evaluation of the proposed secure NFC micro-payment system in order to demonstrate mainly itseffectiveness.

This paper is organized as follows. In section II, an overview about NFC technology and standards is given. Section III reports the state of the art related to security issues in NFC mobile payments. Presentation and discussion about the IDA-Pay secure architecture are outlined in section IV. Section V discusses about implementation issues and strategies, as well as presenting a validation scenario. A qualitative and quantitative system evaluation is reported in Section VI. Finally, section VII summarizes our key messages and sketches future research directions.

II. Overview on NFC technology

NFC is a short-range wireless technology derived from the Radio Frequency IDentification (RFID) [7][8] family which, even if is widely diffused and adopted in applications even quite far from the canonical ones related to logistics[9]-[14], it is not adequate for micro-payments. NFC has been standardized and promoted by Sony, Philips and Nokia, which founded the NFC Forum [15] in 2004. To date, the NFC Forum boasts more than 160 members.

NFC is based on inductive-coupling, where loosely coupled inductive circuits share power and data over a distance of a few centimeters (i.e. less than 5 cm). NFC devices inherit the basic technology of proximity such as RFID tags in High Frequency (HF) band (i.e., 13.56 MHz) and contactless smartcards, but they have a number of key new features.

The specification for NFC is given by ISO/IEC 18092 NFC IP-1 [16] and ISO/IEC 14443 [17] contactless smartcard standards. According to such standards, NFC offers three different operating modes, illustrated in Fig.1:

1. Reader/Writer mode. The NFC device is capable of reading NFC Forum-mandated tag types. The reader/writer mode on the RF interface is compliant to the ISO 14443 and FeliCa schemes.
2. Card Emulation mode. The device can emulate an existing contactless card without adaptors in the existing payment infrastructure. A card and a tag are technically the same; however, contactless cards used in e-ticketing and payment today include additional technology to store secure data.
3. Peer-to-Peer (P2P) mode. Two active NFC devices can exchange data, such as virtual business cards or digital photos. P2P mode is standardized on the ISO/IEC 18092 specification.

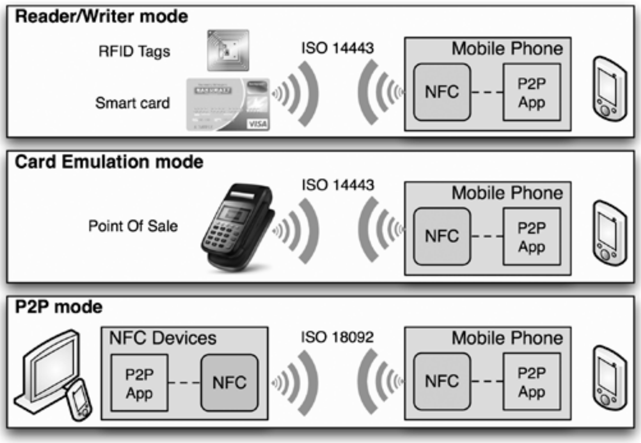


Fig. 1. NFC operating modes and standards

NFC always involves an initiator and a target; the initiator actively generates an RF field that can power a passive target (e.g. tags, stickers, cards) or an active target as well (i.e. a second NFC reader). Available bit rates for NFC are 106/216/414 Kbps.

Data stored into NFC passive targets follows the NFC Data Exchange Format (NDEF) Forum specification [18]. NDEF is a binary message format that can be used to encapsulate one or more application-defined payloads, which may be of a variety of types and sizes. The format and the rules for building standard record types in NDEF messages are included in NFC Record Type Definition (RTD) Technical Specifications [19]. Four specific RTDs are available: Text, URI, Smart Poster, and Generic Control.

The exchange of data between two active targets (P2P) takes place on top of the Logical Link Control Protocol (LLCP) [20]. LLCP is an OSI layer-2 compact protocol, based on the industry standard IEEE 802.2, designed to support either small applications or network protocols. Google released the Android NDEF Push Protocol (NPP) [21], a simple open protocol built on top of LLCP which is designed to push an NDEF message from a NPP client to a NPP server (one way). A device that supports NPP always run an NPP server, and may run a NPP client as well. This allows for bi-directional NDEF exchange between NPP devices.

III. Related works

Working in card emulation mode implies to access the SE. It plays a key role in the security management, as it hosts the firewalled applications and user credentials, and it controls security and cryptography using an onboard microprocessor and software.

There are three ways to implement the SE:

1. In the SIM. This has the advantage of portability, and is the preferred approach in GSM countries; the drawback is that the user must use a special purpose SIM.
2. Embedded SE component. This is a separate chipset in the handset. Its main advantage is that it is convenient for handset manufacturers to implement quickly. The drawback is that handset and OS manufacturers rule the access to the SE.
3. A removable SE component. This is a theoretical

approach to create a removable separate chipset in the handset; often it is implemented as a SD card. Again, the drawback is that the user must use special hardware.

The embedded SE solution is adopted in Android NFC mobile phones. Samsung's Nexus S, Galaxy Nexus, and Galaxy S III have this feature implemented in the PN65N NFC chipset, which includes the SE. However the Android NFC APIs do not provide SE software interfaces, except for the Google Wallet application, that can work in card emulation mode and rely on the existing payment infrastructure.

In literature, some attempts to address such limits exist. In [22], the first experiment of a NFC-based payment application fully supporting the EMV (Europay, MasterCard and VISA) international standard [23] is described. In this solution, the SE is embedded within a special SIM card. The same approach is used in [24], where Nokia NFC mobile phones use special SIM cards to guarantee privacy and security in car parking payment transactions in Italy. Other interesting SIM-based solutions are proposed in [25]-[28].

In [29], the pros and cons of combining NFC with the SIM card in the handsets are described. Although the technology itself is already working properly, the authors admit that the processes for integration of both into one mobile device still need time. At present, it is necessary for the users to replace their own SIM card with new ones. Moreover, SE itself is not completely threat-safe, as its centralized logic may be object of Denial of Service (DoS) and Relay attacks [30][31].

IV. System Architecture

In IDA-Pay, P2P mode is used to transfer payment information between the Android NFC smartphone and the POS. The forced choice to use P2P suffers the major drawback of not being compatible with the existing NFC POS infrastructure; nevertheless, EMV virtual POS services are offered almost by every bank. Moreover, the choice to use P2P mode instead of card emulation leads to several benefits, e.g. the user can receive customizable confirmations onto his/her smartphone, so s/he can keep track of the payments at any time (even when off-line).

In Fig. 2, the high level IDA-Pay system architecture is shown. In particular:

1. The client is an Android NFC smartphone with the IDA-Pay App installed on. In IDA-Pay client, there is no need for a SE, as the sensitive information for every configured credit card is stored in a secure file placed in the smartphone's memory, which is called the Credit Card File (CCF).
2. The IDA-Pay POS is a desktop client connected to the Internet. It has also an NFC interface and runs the IDA-Pay POS application to exchange payments data with the buyer.
3. The IDA-Pay Gateway (GW) is a Web server that forwards the payment request incoming from the IDA-Pay POS to the right Credit Card Network (CCN) endpoint (e.g. an EMV compliant virtual POS).

The security architecture designed for IDA-Pay follows in Dominikus's footsteps [32].

When a payment is requested, the user inserts a PIN and the CCF is passed to the IDA-Pay POS through a NFC P2P link.

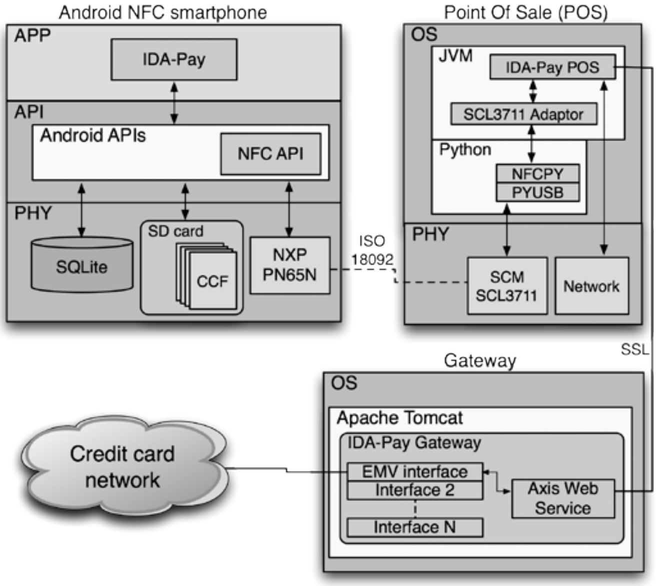


Fig. 2. The IDA-Pay architecture

The CCF encrypts a plain text credit card file M containing the device ID, the credit card number, the month/year of expiration and the CVV. The device ID is the phone's fingerprint; it is used to hard-link the CCF with the smartphone, so the CCF, if stolen, cannot be used by any different device. It is calculated by combining hardware information such as the smartphone's CPU serial number and the Bluetooth card MAC address.

In order to ensure confidentiality, only the IDA-Pay GW should be able to understand the CCF; so it is obtained by encrypting the M file using a Public Key Infrastructure (PKI). When a new credit card is configured in the IDA-Pay App, the M file is encrypted using the IDA-Pay public key (e_{Pub}) and the CCF is created. The used encryption algorithm is the RSA with a key length of 2048 bits.

$$M_{Pub} = e_{Pub}(M) \tag{1}$$

The encrypted message resides on the smartphone's memory, so it may be subject of spoofing attacks. The user authentication is necessary when the access to M_{Pub} is requested. For this reason we used an additional level of encryption based on the symmetric key algorithm AES, where the 128 bits key k_{PIN} is computed on the basis of a 5 digits seed (PIN) inserted by the user at the moment of card configuration. So:

$$CCF = k_{PIN}\left(e_{Pub}(M)\right) \tag{2}$$

Such process is modeled in the card configuration interaction scenario (Fig. 3). In Fig. 4, the payment interaction scenario is shown. When a new payment is requested, the user selects a configured credit card on its Android smartphone and the IDA-Pay App prompts the user to re-type the PIN. The k_{PIN}symmetric key is re-computed, so it can be used to try to decrypt the CCF. If the PIN is right, M_{Pub} is obtained and it is ready to be transferred to the requesting POS through the NFC

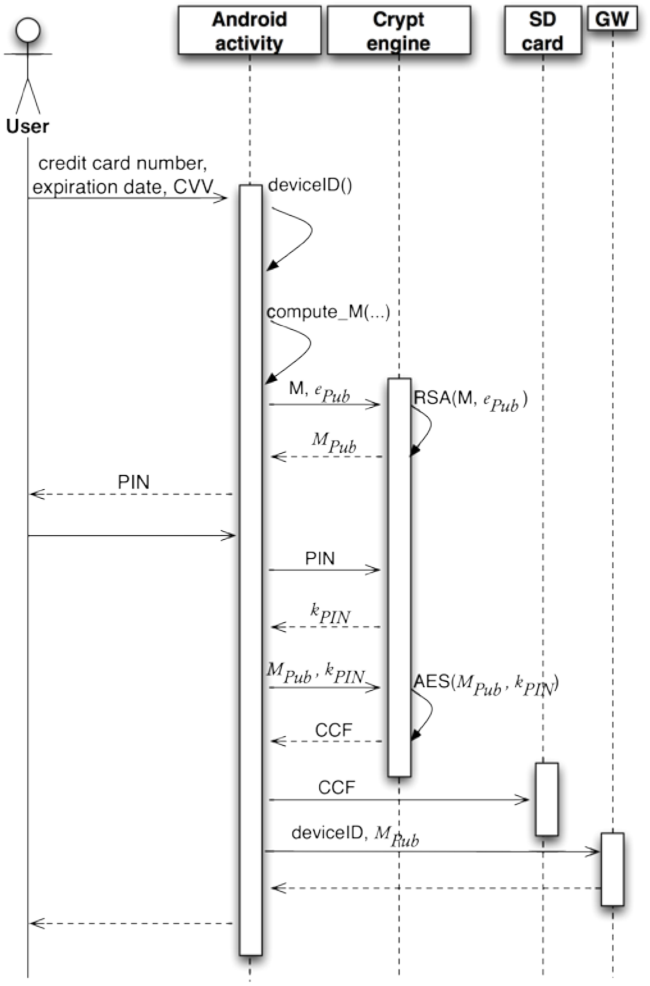


Fig. 3. The card configuration interaction scenario

P2P link, along with the recomputed device ID. The IDA-Pay POS works as a relayer, as it forwards the CCF, the device ID and the payment amount to the IDA-Pay GW in the POST body of a SOAP message. The SOAP request is transferred by using a Secure Socket Layer (SSL) tunnel. After the device ID has been validated, the gateway can extract the M plain file using the IDA-Pay private key e_{Priv} and it forwards the payment request to the right credit network (e.g. a virtual EMV POS).

V. Implementation and Validation

The IDA-Pay App prototype was implemented and validated by using a Samsung Nexus S mounting Android 4.1.2 (i.e., Jelly Bean version). We chose this smartphone model because the Nexus S was the first Android device to support NFC in both hardware and software. In our opinion, it represents the best trade-off between performances and cost; the double-level encryption process is almost instantaneous on the Nexus S' ARM Cortex A8 CPU (see section VI). The package *javax.crypto* from Android SDK was used in order to produce the CCF.

The IDA-Pay POS prototype uses a SCM Microsystems SCL3711 USB Dongle NFC reader, as it is the best-supported reader by NFCPY software libraries [33]. We used such libraries because they were the most advanced NFC P2P desktop libraries at the time of implementation (Jan. 2012).

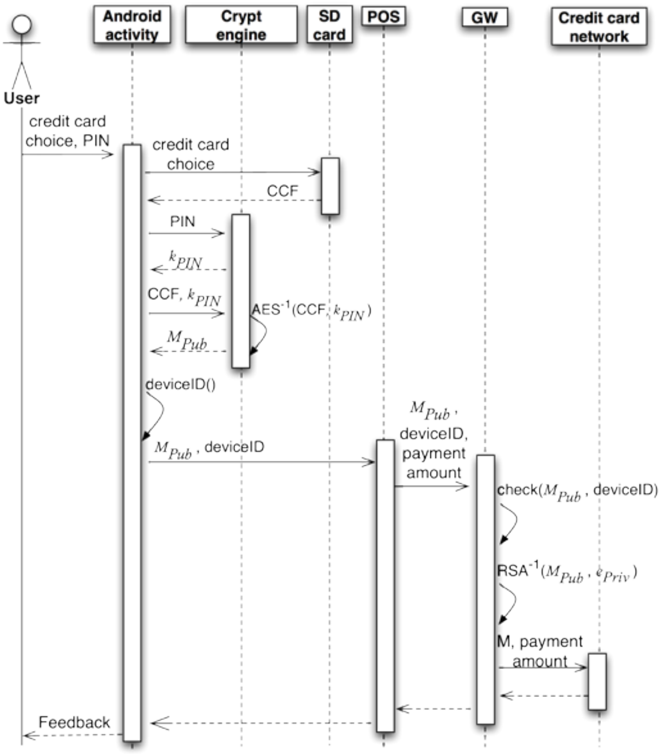


Fig. 4. The payment interaction scenario

NFCPY libraries are written in Python and they implement the Google's NPP specifications.

However, NFCPY are still a work in progress and the latest release (1.0 at the time of implementation) was still buggy. In particular, we had to solve a problem occurring when switching NPP client and server roles between the smartphone and the SCL3711. Such switching was needed for receiving feedbacks from the POS, as at the first phase of the payment the communication flows from smartphone to SCL3711, while at the second phase it flows from SCL3711 to smartphone. The SCL3711 could not switch from server to client role because it remained pending, waiting for other NDEF messages incoming from the smartphone. Our fix let the connections be closed when data exchange between the two NFC devices is terminated.

In the IDA-Pay POS implementation, NFCPY are wrapped in a Java standalone program, in order to follow the Android philosophy. However, the Python runtime environment must be installed on the POS machine.

The IDA-Pay GW exposes a set of Web services through the Tomcat servlet container. In order to support multiple CCNs, the Abstract Factory design pattern [34] was used, thus it is simple to implement multiple concrete adaptors for different CCNs, with no effort in re-engineering the code.

In Fig. 5, the validation scenario is reported. Here we assume that one (at least) credit card has been configured by the buyer into his own smartphone. In particular:

1. The cashier inserts and confirms the amount into the IDA-Pay POS application (Fig. 6.a) and tells the buyer to touch his NFC Android smartphone to the POS NFC reader.
2. The buyer starts the IDA-Pay App and taps a credit card from the configured credit card list (Fig. 6.b). The

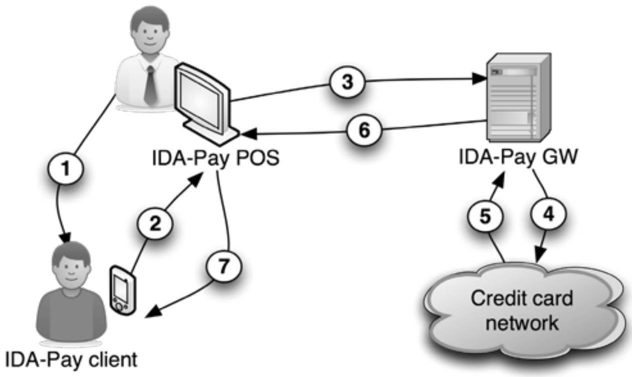


Fig. 5. Validation scenario

App prompts the buyer to insert the correspondent PIN, and then asks the buyer to touch the smartphone to the POS NFC reader (Fig. 6.c).

3. Payment details are transferred to the IDA-Pay GW.
4. The IDA-Pay GW instantiates a payment transaction with the right CCN.
5. The GW receives the result from the CCN.
6. The result is formatted and returned to the IDA-Pay POS, where it is shown to the cashier.
7. A plain text response is created and sent back to the buyer's smartphone through the NFC P2P link. A message dialog notifies the buyer that the payment has been sent successfully and that he can move the smartphone away from the POS NFC reader. The payment response is stored into the payments' archive (Fig. 6.d).

From step 3 to step 6 (included), it is not mandatory for the buyer to keep its smartphone in touch with the POS NFC reader. It is possible because after the information is transferred from the smartphone to the POS (step 2), the P2P role of the smartphone switches from 'initiator' to 'target'; in this state, the IDA-Pay app remains pending (Fig. 6.c), waiting for data incoming from the NFC interface. During this period of time, the buyer can remove the smartphone from the POS.However, it still remains mandatory for the buyer to keep the IDA-Pay app active in the foreground of the smartphone during this phase (application must not be closed).Step 7 will be accomplished as soon as the NFC smartphone will be in range.The buyer can close the application when step 7 is finished.

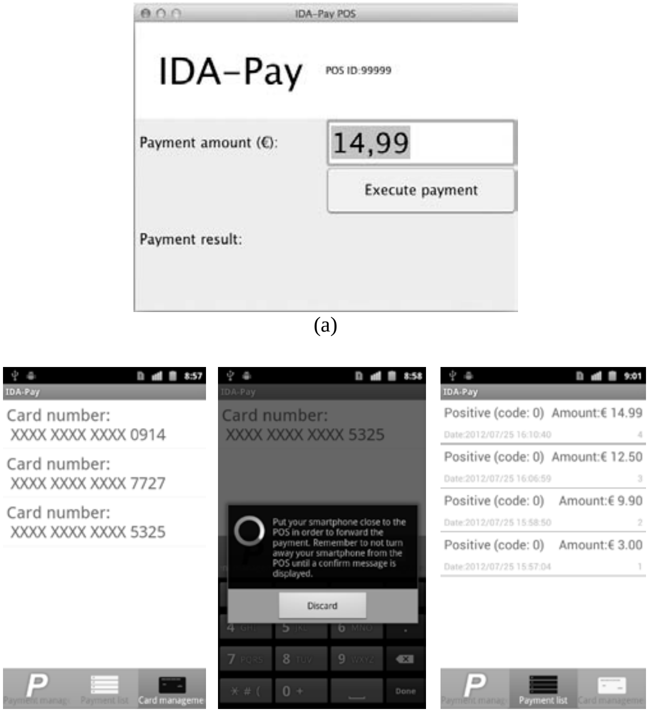
During the full payment process, it is not necessary for the buyer to be connected to the Internet. The Internet connection is only needed in the card configuration phase (see Fig. 3)

VI. Evaluation

In this section, a rigorous system evaluation is presented,both quantitative and qualitative.

A. Setup and Methods

In our experiment, the IDA-Pay app is running on a Samsung I9023 Nexus S with Android 4.1.2 (Jelly Bean) and a 1 GHz ARM Cortex A8 Hummingbird CPU. Anyway, further Android devices (even not NFC-powered) have been used (see Table I) to demonstrate the IDA-Pay crypt engine performances. Both the IDA-Pay POS and GW are running on the same 64-bit Ubuntu 12.04 LTS with an Intel Core2 Duo CPU T9600 @ 2.80GHz, 512MB RAM, OpenJRE Java 6, and Apache Tomcat 6, virtualized with VMware Fusion on a late 2008 Apple MacBook Pro.



Payment amount (€): 14,99

Payment result:

(a)

Card number: XXXX XXXX XXXX 0914

Card number: XXXX XXXX XXXX 7727

Card number: XXXX XXXX XXXX 5325

Card number: XXXX XXXX XXXX 5325

Put your smartphone close to the POS in order to forward the payment. Remember to not turn away your smartphone from the POS until a confirm message is displayed.

Positive (code: 0) Amount:€ 14.99 Date:2012/07/25 16:10:40 4

Positive (code: 0) Amount:€ 12.50 Date:2012/07/25 16:06:59 3

Positive (code: 0) Amount:€ 9.90 Date:2012/07/25 15:58:50 2

Positive (code: 0) Amount:€ 3.00 Date:2012/07/25 15:57:04 1

(b) (c) (d)

Fig. 6. Some screenshots of the IDA-Pay prototype: (a) the cashier inserts the amount into the IDA-Pay POS application, (b) the buyer starts the IDA-Pay App and taps a credit card from the configured credit card list, (c) the app asks the buyer to touch the smartphone to the POS NFC reader, (d) the payment response is stored into the payments' archive.

In order to simulate the network latency during the payment experience, an artificial cumulative delay T_D was introduced in the system. We define the Round Trip Time (RTT) as the length of time it takes for a payment instance to be sent plus the length of time it takes for the feedback to be received on to the smartphone. Formally, the RTT is defined as follows:

$$\begin{aligned} RTT &= T_{SEND} + T_{ELAB} + T_{RETURN} = \\ &(T_{NFC_{send}} + T_{SSL_{send}} + T_{CCN_{send}}) + \\ &(T_{POS} + T_{GW} + T_{PAY}) + \\ &(T_{CCN_{return}} + T_{SSL_{return}} + T_{NFC_{return}}) \end{aligned} \tag{3}$$

where:

- T_{SEND} is the amount of time needed to forward the payment instance from the smartphone NFC interface to the CCN. It is the sum of:
    - $T_{NFCsend}$, the time needed to move the information over the NFC P2P link, from the smartphone to the POS;
    - $T_{SSLsend}$, the time needed to move the information over the SSL link, from the POS to the IDA-Pay GW;
    - $T_{CCNsend}$, the time needed to send the information to the CCN.

- T_{ELAB} is the amount of time needed to process the information at different stages. It is the sum of:
  - T_{POS}, the time to process the information at the POS application;
  - T_{GW}, the time to process the information at the IDA-Pay GW;
  - T_{PAY}, the time needed to perform the transaction by the CCN.
- T_{RETURN} isthe amount of time needed to forward the payment result from the CCN to the smartphone NFC interface. It is the sum of:
  - $T_{CCNreturn}$, the time needed to forward the payment result from the CCN to the IDA-Pay GW;
  - $T_{SSLreturn}$, the time needed by the payment result to go back to the POS through the SSL link;
  - $T_{NFCreturn}$, the time needed by the payment result to go back to the smartphone through the NFC P2P link.

The RTT does not include the time needed by the buyer for inserting the PIN and putting the smartphone close to the POS.

In our experiments, the values T_{SSL} and T_{CCN} (both send and return) are not realistic because of our hardware setup (everything is on a PC). So we introduce an artificial cumulative delay T_D in our formula, obtaining a new RTT definition:

$$\overline{RTT} = RTT + T_D \quad (4)$$

Such delay time should be distributed among the different sub-systems; however, for evaluation purposes, a `Thread.sleep(T_D)` has been inserted in the POS application.

The range in which the delay T_D falls was estimated in 3-15 seconds; this range was obtained by a bank services company we have consulted, which observed the time experienced by a set of credit card customers in real usage scenarios. We have considered the full range 0-15 seconds in order to evaluate the system in clean setup conditions. We have discretized the range in 3 seconds intervals and we have evaluated the payment time at every interval.

B. *Qualitative evaluation*

In this paragraph, we evaluate the IDA-Pay system from a qualitative perspective. In Table II, a summarized comparison with similar payment technologies is reported. We have also considered remote mobile payment technologies (i.e. QR codes and SMS, anyway our considerations can be extended to any remote mobile payment technology such as eBay and PayPal mobile applications). The table shows that, at the time of writing (November 2012), a “perfect” mobile payment technology (i.e. without drawbacks) does not exist. Anyway, NFC P2P technology got the potential to be the most suitable payment technology for the future. Nevertheless, effort in standardization and diffusion must be spent.

In Table III, the risk matrix related to the IDA-Pay micro-payment system is reported. We used this tool to perform an ‘a posteriori’ evaluation about the harm events that can occur to the IDA-Pay system and the related amount of harm.

We assign both the probability of the event (A=Sure, B=Probable, C=Improbable, D=Impossible) and the effect (A=Disastrous, B=Denial of Service, C=Instability, D=Null). In particular, we focus our attention over the risks related with the client (IDA-Pay App) and POS sub-systems, which can be easily subject of theft, unauthorized access and corruption. The designed security system ensures strong threat immunity even in the extreme cases of smartphone stealing and malicious access.

C. *Quantitative evaluation*

In order to test and assess the performance of our micro-payment system, a campaign of experimentation for measuring the average time it takes a payment to be accomplished was held.

The number of repetitions (payments) carried out in order to estimate the mean value of T_D is able to guarantee a confidence interval equal to 95% and a maximum relative error less than 5%. As reported in Table IV, the number of repetitions is quite variable for each value of T_D; moreover, the variance is quite variable as well. Such variability is due to the partial immaturity of the NFCPY libraries, which let the feedback return time be subject to random delay. Anyway, the average payment time grows linearly when the delay T_D increases (Fig. 7); moreover, the percentile 80% is only slightly larger than the average for each T_D, which means that for the 80% of the times the payment time is close to the average.In clean setup conditions, the average payment time is about 5 seconds.

We have used further five different Android devices in order to demonstrate the efficiency of the IDA-Pay security system. The aim of our test is not to select the best Android device on which run the IDA-Pay App, but to demonstrate that *any* Android smartphone (even the old ones) can perform the double level encryption and decryption in very short time (few milliseconds). We obtained an average value over 100 repetitions of the encode algorithm time on each considered smartphone.

TABLE I
MAIN FEATURES OF THE CONSIDERED ANDROID DEVICES

	Samsung Nexus S	Galaxy Nexus	Acer Liquid mini e310	Samsung Galaxy S Plus	HTC nexus one	Samsung Galaxy S3
CPU	1.0GHz ARM Cortex A8 Hummingbird	1.2GHz ARM Cortex A9 dual core	600MHz Qualcomm MSM7227	1.4GHz single core ARM Cortex A8	1.0GHz Qualcomm Snapdragon 8250	1.4GHz quad-core Exynos 4
RAM	512MB	1GB	512MB	512MB	512MB	1GB
Android version	4.1.2	4.1.2	2.3.5	2.3.5	2.3.7	4.1.2

TABLE II
PROXIMITY AND REMOTE MOBILE PAYMENT TECHNOLOGY COMPARISON

	NFC P2P (e.g. IDA-Pay)	**NFC card emulation (e.g. Google Wallet)**	**QR code**	**Contactless smartcard (e.g. MasterCard PayPass)**	**Contact smartcard**	**SMS**
Type of mobile payment	Proximity	Proximity	Remote	Proximity	Proximity	Remote
Level of adoption	Proprietary services	US only	Proprietary services	Worldwide, increasing adoption	Worldwide, full adoption	Proprietary services
Compatibility with existing POS infrastructure	NO	YES	NO	YES, but only with contactless POS	YES	N.A.
Can return rich feedbacks	YES	NO	YES	NO	NO	YES
Is freely implementable	YES	NO on Android, YES on other platforms	YES	N.A. (issued by banks and credit institutions)	N.A. (issued by banks and credit institutions)	YES
Security management	In the system	Secure Element	In the system	Secure Element	Secure Element	In the system

TABLE III
EFFECT/PROBABILITY RISK MATRIX
EFFECT: A=SURE, B=PROBABLE, C=IMPROBABLE, D=IMPOSSIBLE
PROBABILITY: A=DISASTROUS, B=DENIAL OF SERVICE, C=INSTABILITY, D=NULL

Effect/Probability	**Client**	**POS**	**Gateway server**	**NFC network interface**	**Credit card data**
Unauthorized access	D/B	C/C	A/C	C/C	D/C
Denial of Service	C/C	C/C	B/B	B/D	B/C
Theft	D/B	C/C	B/D	D/C	D/C
Corruption	C/B	C/B	B/B	B/D	A/C
Virus	C/B	C/B	C/C	B/D	D/D
Physical damage	D/B	D/C	B/D	B/D	A/C

TABLE IV
PAYMENT SYSTEM PERFORMANCE

T_D[s]	Number of reps	$\overline{RTT}(T_D)$						
		Average [s]	Median [s]	Min [s]	Max [s]	Percentile 80%	Variance [s^2]	St. deviation [s]
0	41	4.719	4.514	2.390	9.186	5.651	2,950.427	1.717
3	31	7.887	7.462	6.231	9.988	8.965	1,089.762	1.043
6	18	12.205	11.681	11.269	14.993	12.978	1,243.378	1.115
9	12	14.994	14.510	14.391	17.360	14.881	1,147.403	1.071
12	15	18.423	17.937	17.142	22.316	19.257	2,209.406	1.486
15	9	21.666	22.202	20.232	23.684	22.794	1,867.185	1.366

Fig. 8 shows the encode time histogram; the best performance (2ms) is obtained by the Samsung Galaxy SIII, equipped with a 1.4GHz quad-core CPU. The worst performance (11ms) is obtained by the Acer Liquid Mini E310, equipped with a 600MHz single core CPU.

We have monitored the CPU load during the 100 repetitions as well. We have used a third party Android app, NetMeter[35], in order to read the CPU peak usage after the execution of the 100 repetitions. Fig. 9 reports the results obtained by using the preselected smartphones. Beware that in Fig. 8 we have reported an average time, while in Fig. 9 we show a cumulative load of all the 100 repetitions, i.e. the CPU load may be significantly lower in real usage scenarios. As it is shown in Fig. 9, the CPU load is never higher than 40-50% on every smartphone. In Fig. 10, a NetMeter screenshot showingthe full CPU load graph for the Samsung Nexus S is depicted. The graph at the bottom is the one showing the CPU load. The black vertical arrow points at the peak related to the test execution time. The two small peaks shown on the left side ofthe peak in Fig. 10 are caused by the opening of the IDA-Pay test activity on to the smartphone.

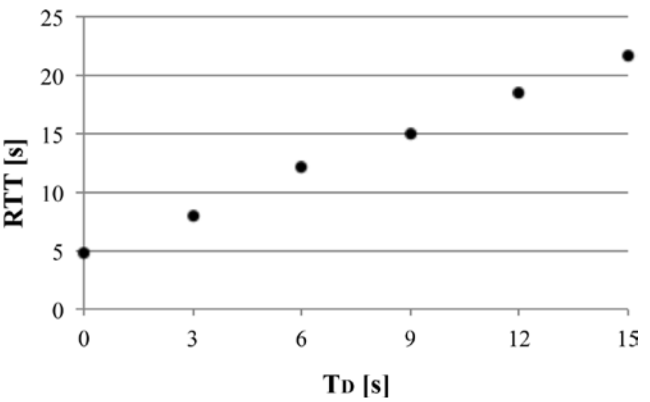


Fig. 7. Payment times grow linearly when the network latency increases

VII. CONCLUSION

In this paper, we have proposed and discussed the IDA-Pay, an innovative NFC-based mobile micro-payment system architecture and prototype for Android smartphones. In literature, several NFC-based micro-payment systems have already been presented; in order to avoid the security limitations imposed by smartphone and OS manufacturers, such works force the user to install additional hardware (special SIM or SD cards). Our system is SE-agnostic as it adopts NFC P2P mode and can be easily used and customized

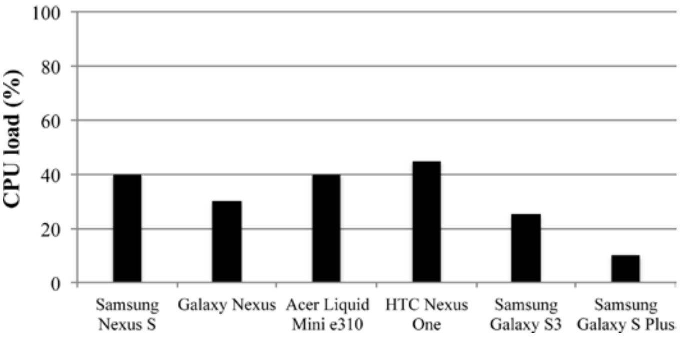


Fig. 8. Average encode time over 100 repetitions

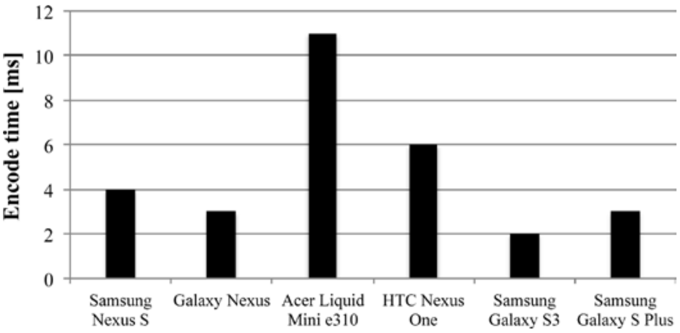
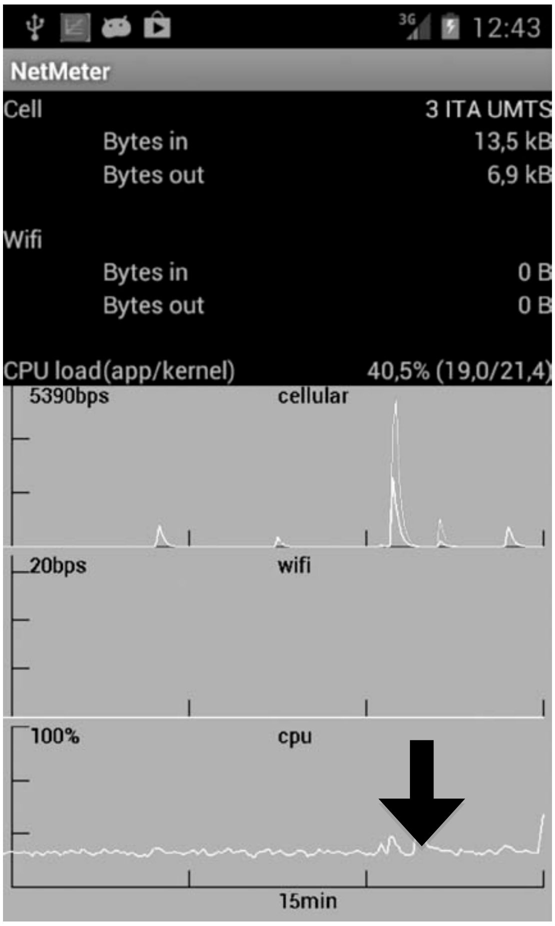


Fig. 9. Average CPU load over 100 repetitions



3G 12:43

NetMeter

Cell 3 ITA UMTS

Bytes in 13,5 kB

Bytes out 6,9 kB

Wifi

Bytes in 0 B

Bytes out 0 B

CPU load(app/kernel) 40,5% (19,0/21,4)

Fig. 10. CPU load graph for the Samsung Nexus S, the black arrow points the IDA-Pay CPU peak

in alternative micro-payments ecosystems. Nevertheless, thesame security level of traditional credit card payments is ensured, and the compatibility with popular credit card networks is preserved. Moreover, in IDA-Pay the data flow between the smartphone and the system is bi-directional, so the user can read and collect textual or binary feedbacks.

Nevertheless, only NFC-enabled smartphones can be used in IDA-Pay. Alternative solutions based on NFC stickers or QR codes could be considered in order to retrofit the system.

This paper extends our previous work, by adding a system evaluation. The experimental results obtained in this phase show a minimum payment time of around 5 seconds in ideal conditions; this number grows linearly as the network latency increases. The encryption algorithm, used in the validation phase to ensure security in the system, is very efficient, as it takes only 12 ms on a middle generation Android smartphone. The CPU load is never higher that 50% when 100 repetitions of the same algorithm are performed.

We spent our effort in the system validation and evaluation through a set of predefined test beds in our laboratory (in vitro). The next step is to validate our system in a pilot project with real users using the system in real scenarios (in vivo); moreover, biometric techniques to identify the cardholder will be also taken into account in order to avoid the need of a PIN, which can be easily forgotten, lost or stolen.

ACKNOWLEDGMENT

We want to thank Eng. Luca Marsano from SEI Consulting s.r.l. (Italy) for his valuable support in the evaluation phase of this work.

REFERENCES

[1] Polytechnic of Milan, School of Management, NFC & Mobile Payment Observatory:*Mobile Payment: traaspettative e realtà*, Feb. 2011.

[2] NXP: 3rd Workshop on RFID Systems and Technologies, http://www.rfidsystech.eu/20070612_1A_1235_Graeber_NFCIsTheDoubleClickInTheInternetOfThings.pdf , 2007-12-06

[3] Cisco:*The Internet of Things [INFOGRAPHIC]*, http://blogs.cisco.com/news/the-internet-of-things-infographic/.

[4] Gartner:*Hype Cycle for Emerging Technologies 2011*, http://www.gartner.com/DisplayDocument?doc_cd=215650.

[5] Android, http://www.android.com/.

[6] L. Mainetti, L. Patrono, and R. Vergallo:*IDA-Pay: an innovative micro-payment system based on NFC technology for Android mobile devices*, 20th International Conference on Software, telecommunications and Computer Networks, SoftCOM 2012, Split, Croazia, pp.1-6, 11-13 Sept. 2012.

[7] L. Catarinucci, R.Colella, M.De Blasi, L.Patrono, and L.Tarricone:*Enhanced UHF RFID Tags for Drug Tracing*, Journal of Medical Systems, Ed. Springer, vol 36, no 6, pp. 3451-3462, 2012.

[8] L. Catarinucci, R. Colella, M. De Blasi, L. Patrono, and L. Tarricone:*Experimental Performance Evaluation of Passive UHF RFID Tags in Electromagnetically Critical Supply Chains*, Journal of Communications Software and Systems, Vol. 7, No. 2, pp. 59-70, 2011.

[9] L. Catarinucci, R. Colella, M. De Blasi, L. Patrono, and L. Tarricone:*Improving Item-Level Tracing Systems Through ad Hoc UHF RFID Tags*, in Proc. of IEEE Radio and Wireless Symposium, RWW 2010, 2010.

[10] M. Maffia, L. Mainetti, L. Patrono, and E. Urso:*Evaluation of potential effects of RFID-based item-level tracing systems on the integrity of biological pharmaceutical products*, International Journal of RF Technologies: Research and Applications, vol. 3, Issue 2, pp. 101-118, 2012.

[11] G. Calcagnini, F. Censi, M. Maffia, L. Mainetti, E. Mattei, L. Patrono, and E. Urso:*Evaluation of Thermal and Non-thermal Effects of UHF*

RFID Exposure on Biological Drugs, IEEE Transactions on Information Technology in Biomedicine, Volume: 16, Issue: 6, pp. 1051-1057, 2012.

[12] A. L. Guido, L. Mainetti, and L. Patrono:*Evaluating potential benefits on the use of RFID, EPCglobal, and ebXML in the pharmaceutical supply chain*, International Journal of Healthcare TechnologyandManamegent, InderScience, vol. 13, Issue 4, pp. 198-222, 2012.

[13] R. Acierno, M. Maffia, L. Mainetti, L. Patrono, E. Urso:*RFID-basedtracingsystems for drugs: Technologicalaspects and potentialexposurerisks*, in Proc. of 2011 IEEE Radio and Wireless Week, RWW 2011 - 2011 IEEE Topical Conference on Biomedical Wireless Technologies, Networks, and Sensing Systems, BioWireleSS 2011.

[14] U.Barchetti, A. Bucciero, M. De Blasi, L. Mainetti, L. Patrono:*Implementation and testing of an EPCglobal-awarediscovery service for item-leveltraceability*, in Proc. of 2009 International Conference on Ultra ModernTelecommunications and Workshops.

[15] NFC Forum, http://www.nfc-forum.org/home/.

[16] International Standard ISO/IEC 18092, Information technology - Telecommunications and information exchange between systems - Near Field Communication - Interface and Protocol (NFCIP-1), 2010-04-20, ISO/IEC 2010, Switzerland.

[17] International Standard ISO/IEC 14443-1-2-3-4, Identification cards - Contactless integrated circuit cards - Proximity cards, 2008-07-15, ISO/IEC 2008, Switzerland.

[18] NFC Data Exchange Format (NDEF) Technical Specification (NDEF 1.0), 2006-07-24, NFC Forum, Inc., Wakefield (MA), USA.

[19] NFC Record Type Definition (RTD) Technical Specification (RTD 1.0), 2006-07-24, NFC Forum, Inc., Wakefield (MA), USA.

[20] NFC Logical Link Control Protocol (LLCP) Technical Specification (LLCP 1.1), 2011-06-20, NFC Forum, Inc., Wakefield (MA), USA.

[21] Android NDEF Push Protocol Specification, version 1, 2011-02-22.

[22] M. Pasquet, J. Reynaud, and C. Rosenberger:*Secure payment with NFC mobile phone in the SmartTouch project*,in Proc. of International Symposium on Collaborative Technologies and Systems, 2008 (CTS 2008), 19-23 May 2008, doi: 10.1109/CTS.2008.4543921.

[23] EMVCo., http://www.emvco.com/ .

[24] G. Benelli, and A. Pozzebon:*An automated payment system for car parks based on Near Field Communication technology*, in Proc. of International Conference for Internet Technology and Secured Transactions (ICITST), 2010, 8-11 Nov. 2010.

[25] X. Yu-ning: *Research on NFC and SIMpass Based Application*, International Conference on Management and Service Science, 2009. MASS '09., 20-22 Sept. 2009, doi: 10.1109/ICMSS.2009.5303274.

[26] H. Zhao, and S. Muftic:*The concept of Secure Mobile Wallet*, in Proc. of World Congress on Internet Security (WorldCIS), 2011, 21-23 Feb. 2011.

[27] E.-J.Steffens, A. Nennker, Z.Ren, M. Yin, and L. Schneider:*The SIM-based mobile wallet*, in Proc. of 13th International Conference on Intelligence in Next Generation Networks, 2009. ICIN 2009,, 26-29 Oct. 2009, doi: 10.1109/ICIN.2009.5357095.

[28] W.-D. Chen, K.E.Mayes, Y.-H. Lien, and J.-H. Chiu:*NFC mobile payment with Citizen Digital Certificate*, in Proc. of the 2nd International Conference on Next Generation Information Technology (ICNIT), 2011, 21-23 June 2011.

[29] G. Madlmayr, O. Dillinger, J. Langer, C. Schaffer, C. Kantner, and J. Scharinger:*The benefit of using SIM application toolkit in the context of near field communication applications*, in Proc. of International Conference on the Management of Mobile Business, 2007. ICMB 2007, July 2007 doi: 10.1109/ICMB.2007.62.

[30] M. Roland, J. Langer, and J. Scharinger:*Practical Attack Scenarios on Secure Element-Enabled Mobile Devices*, in Proc. of 4th International Workshop on Near Field Communication (NFC), 2012, 13 March 2012, doi: 10.1109/NFC.2012.10.

[31] A. J.Jara, A. F. Alcolea, M. A.Zamora, and A. F. G. Skarmeta:*Evaluation of the security capabilities on NFC-powered devices*, in Proc. of European Workshop on Smart Objects: Systems, Technologies and Applications (RFID Sys Tech), 2010, 15-16 June 2010.

[32] S. Dominikus, and M. Aigner:*mCoupons: An Application for Near Field Communication (NFC)*, in Proc. of 21st International Conference on Advanced Information Networking and Applications Workshops, 2007, AINAW '07, vol.2, 21-23 May 2007, doi: 10.1109/AINAW.2007.230.

[33] Python module for near field communication, https://launchpad.net/nfcpy. of Reusable

[34] Gamma, E. et al, 2007. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional, Redwood City, USA.

[35] NetMeter,https://play.google.com/store/apps/details?id=com.google.android.netmeter&hl=en

Luca Mainetti is an associate professor of software engineering and computer graphics at the University of Salento. His research interests include web design methodologies, notations and tools, services oriented architectures and IoT applications, and collaborative computer graphics. He is a scientific coordinator of the GSA Lab - Graphics and Software Architectures Lab and IDA Lab -IDentification Automation Lab at the Department of Innovation Engineering, University of Salento. He is the Rector's delegate at the ICT.

Luigi Patrono received his MS in Computer Engineering from University of Lecce, Lecce, Italy, in 1999 and PhD in Innovative Materials and Technologies for Satellite Networks from ISUFI-University of Lecce, Lecce, Italy, in 2003. He is an Assistant Professor of Network Design at the University of Salento, Lecce, Italy. His research interests include RFID, EPCglobal, Internet of Things, Wireless Sensor Networks, and design and performance evaluation of protocols. He is Organizer Chair of the international Symposium on RFID Technologies and Internet of Things within the IEEE SoftCOM conference. He is author of about 60 scientific papers published on international journals and conferences and four chapters of books with international diffusion.

Roberto Vergallo graduated cum laude in Computer Engineering at University of Salento (Italy) in October 2010. Currently he is a PhD student in Computer Science at the same university. His doctoral research aims to build a Classroom 3.0, in which not only humans but also machines can detect the context and interact with the environment. He also took part in several research projects regarding the development of middleware tools in order to support B2B exchange, supply chain management and healthcare interoperability.