Deployment and performance evaluation of mobile multicoupon solutions

M. Francisca Hinarejos¹ · Andreu-Pere Isern-Deyà¹ · Josep-Lluís Ferrer-Gomila¹ · Llorenç Huguet-Rotger¹

Published online: 24 February 2018
© The Author(s) 2018. This article is an open access publication

Abstract

Mobile commerce (m-commerce) represents an important area of business with a huge potential revenue for merchants and great opportunities for customers to achieve better offers. One of the topics within m-commerce that needs important improvements on usability and efficiency is electronic coupon systems. This is because solutions for electronic coupons are focused on their theoretical definition, overlooking both their performance evaluation and their viability analysis. In this paper, we provide an evaluation framework to test multicoupon solutions. Under this framework, we evaluate two particular multicoupon schemes, obtaining interesting performance results, such as the fact that cryptographic operations are not always the most costly processes, as usually claimed by authors of multicoupon schemes. We show that message encoding can help to improve the efficiency of multicoupon solutions, even up to 50% of the overhead over the network. Moreover, we solve other issues related to the viability of these schemes, for example, using a standardized way to distribute non-standard cryptographic key material. Therefore, the viability of a multicoupon solution should be validated considering all involved factors, such as internal processing, networking, message encoding and current technology support.

Keywords Mobile commerce · Electronic coupon · Mobile devices · Implementation · Performance evaluation · Group key certification

1 Introduction

Coupons are a very effective marketing instrument used by merchants, helping to increase customer loyalty and to attract new customers. Both parts, customers and merchants, obtain benefits. According to the 2014 NCH Coupon Facts Report [51], consumer packaged goods marketers issued more than 310 billion coupons, and consumers made savings of \$3.6 billion by redeeming a total of 2.75 billion coupons in 2014. Another study, from Juniper Research [42], indicates that electronic coupon redemptions will double over the next years, reaching \$81 billion in 2021. The increase follows

an estimated \$40 billion in 2016. Booklets of coupons, a set of coupons that can be handled as a single unit, represent a step further. The aim is to increase, even more, the loyalty of customers, who can obtain enhanced benefits.

Being so useful, it is logical to provide electronic versions of paper-based coupons or booklets of coupons, to be used in e-commerce scenarios: e-coupons and multicoupons. Additionally, with the impressive market of smart-phones (more than 1.3 billion [14]) and the increasing number of mobile-broadband subscriptions (growing up to 6.9 billion [14,35]), coupons for m-commerce should be considered. In fact, this topic has attracted a remarkable attention [6,9,15–19,25,33,39,47,53,66,68] in recent years.

The main concern of those authors is the security of the proposed schemes: unforgeability, double-redemption protection, unlinkability, unsplittability, etc. And usually, a security analysis (more or less informal) is provided. Indeed, it is essential to meet some security requirements, but such compliance does not guarantee the viability of those solutions. This is a necessary but not sufficient condition.

Assuming that security requirements are met, it is very important to carry out performance assessments. About 57%

✉ M. Francisca Hinarejos
xisca.hinarejos@uib.es

Andreu-Pere Isern-Deyà
andreupere.isern@uib.es

José-Lluís Ferrer-Gomila
jlferrer@uib.es

Llorenç Huguet-Rotger
l.huguet@uib.es

¹ University of the Balearic Islands, Palma de Mallorca, Spain

of online customers leave a site after waiting 3 s for a page to load, and a 80% of these people will not return [45,50]. Therefore, a secure scheme for e-coupons with a bad response time will not be useful (probably will not be used) in the market. However, performance has been overlooked by most authors and such was the case that only a few of them consider the cost of cryptographic operations, ignoring other time-consuming aspects.

Therefore, simulations and, even better, implementations should be carried out to validate mobile coupons proposals. However, proposals are only sketched in papers. Thus, it is very difficult for third parties to implement and evaluate them, due to the lack of a complete specification of their algorithms. Moreover, they do not include a "formal" syntax (XML, ASN.1) giving clear details about which data are exchanged among parties when they communicate with each other. Communications can also be a determining factor to make someone use a mobile coupon solution or not. So, it is important to assess whether well-known and widespread communication mechanisms (e.g., Wi-Fi, 3G, NFC) are really suitable.

Considering all aforementioned issues, it is necessary to define guidelines allowing to evaluate, in a unified way, multicoupon schemes for mobile scenarios.

Contributions. We define the first framework for the deployment and performance evaluation of multicoupon solutions. In order to verify the viability of this framework, we present a practical experience on deploying and evaluating two multicoupon solutions. This deployment will help us to provide solutions to some problems related to the implementation of this type of schemes, that developers typically find when they try to implement their theoretical proposals on a real-world scenario. In order to do it, we take into account a real scenario composed by current commercial devices (for clients and servers) and network communications.

Firstly, the proposed framework helps to define homogeneously the entities, processes and metrics required to evaluate a multicoupon solution. Therefore, the first step is to define which entities are involved in the scenario, and which functionalities should be performed by them. It is of great importance to use a unified notation that allows a clear and unequivocal identification of entities and processes.

Secondly, the selection of an appropriate deployment scenario to test performance features of multicoupon solutions is not always trivial. Hence, we provide a set of considerations to take into account at the moment of deploying a multicoupon solution, in a scenario with real customers devices and communications technologies. In order to do so, we analyze current technologies (their current status, deployment and market support), which covers both client- and server-side platforms, the message encoding syntax, and

the communication technologies. Upon this study, we give advice about which of them could be a good option to shape the testing scenario.

Thirdly, we analyze different message encoding syntaxes to prove how message coding can impact on the overall efficiency. In fact, it can be even more important than the impact produced by the use of complex cryptographic operations. Thus, its influence over the total time perceived by customers should be analyzed by all authors.

Fourthly, considering all the above factors, we define two sets of metrics: one for general evaluation of the solution (total response time of each process and network overhead), and another one for detecting specific processes or functions that can lessen the performance of the scheme (time on processing and encoding, and communications delay). In this way, we obtain a complete performance analysis considering all factors that influence on the total response time perceived by customers. The evaluation of these metrics on a proposal provides real data about the viability of the solution in a real deployment scenario.

Fifthly, to validate the framework, we select two published multicoupon solutions that we define, deploy and evaluate following the specifications of the proposed framework. By means of the results, we can assert that the evaluated multicoupon solutions are suitable to be used in a real deployment scenario, and ready to be used by customers.

Finally, we solve some issues on the practicability of the selected proposals at the time of deployment. For example, one of the proposals is based on the use of a group signature scheme to provide revocable anonymity to customers, but there is no specification to manage this type of cryptographic keys. So, we propose a method to fit group public keys into the X.509 public key infrastructure to distribute them in a standard way.

Although the deployment scenario and the performance evaluation are addressed for multicoupon solutions, our suggestions, considerations and the provided experience could be relevant for developers and protocol designers of other m-commerce solutions covering different topics and applications.

Organization. This paper is organized as follows. We review the existent electronic multicoupon proposals in Sect. 2. In Sect. 3, we describe a general framework for deploying and evaluating multicoupon solutions. Next, in Sects. 4 and 5 we define two existing multicoupon proposals under the specifications of the framework. We describe, in Sect. 6, the implementation environment we have selected to develop and deploy the selected proposals, while in Sect. 7 we evaluate their performance. Finally, in Sect. 8 we conclude the paper.

2 Evaluation of existent electronic multicoupon proposals

There are commercial and scientific proposals providing electronic multicoupon solutions. Regarding commercial solutions, companies try to offer simple products at the lowest possible cost to avoid developing new solutions, and trying to reuse their already developed technologies. This way, they release their products to the market in a short time, but sometimes, these solutions neither consider all required customer functionalities nor solve all associated problems. For example, an easy way for companies to offer electronic coupons is using a simple web page where customers can obtain them. However, customers must go to the physical merchant store (or to an authorized store) to obtain the coupon in paper support, and only afterward they can redeem it using this paper copy. Solutions become more costly due to the fact of introducing this additional manual management. Therefore, commercial solutions are focused on providing basic functionalities at a minimum development cost, without considering all possible problems or additional costs.

In regard to the existing scientific proposals [6,9,15,18,19,25,33,39,53,66,68], these try to offer the greatest number of functionalities without considering the cost of those solutions on real scenarios. Moreover, most of the proposals do not provide enough information to analyze their viability from two main points of view: security and efficiency.

Security is the most studied issue when electronic multicoupon solutions are proposed, but even so, not all solutions provide all the security requirements needed. Table 1 summarizes the security requirements given by the analyzed scientific solutions. In general, proposals meet the basic

security requirements (unforgeability, unlinkability, coupon reuse avoidance, and customer anonymity), but this is not always true [11,33]. In [33], customers' privacy could be lost, while the security in [11] has an important drawback because the content of coupons could be modified by a cheater, so the basic unforgeability requirement is not really well provided. Besides, a real scenario requires even a larger list of security requirements and functionalities. For example, all solutions provide anonymity, but the anonymity revocation is not considered and, in some cases, revoking the anonymity of malicious users could be required. Another functionality that could be used for evaluating merchants reputation is merchants' behavior observed by customers and multicoupon providers. It is a critical aspect that may entail the loss of potential customers.

In any case, the most omitted issue by authors is to prove that their proposals are in fact efficient in terms of response time, because the majority of them are only focused on providing a security analysis. However, even though a solution could achieve a high degree of security, it could be impractical. For example, authors in [15] do not mention any kind of performance consideration, but the cryptographic operations behind the proposed scheme are considered complex, so they could require a high computational cost. In [18,53], authors try to conduct an efficiency comparison between their schemes based on how the number of coupons entails an increase in the cost of the involved protocols. As a result, authors in [18] claim that the cost of issuing and redeeming coupons is linear regarding to the maximum number of coupons in a multicoupon; meanwhile, the authors in [53] claim (without any proof) that their scheme has a constant computational cost regardless of the number of involved

Table 1 Multicoupons solutions: a comparative analysis

	[33]	[18]	[15]	[53]	[6]	[66]	[39]	[32]
Scenario								
Coupons	mc	mc	mc	mc	mc	mc	mc	mc
Merchants	mm	sm	sm	sm	mm	sm	mm	mm
Security								
Unforgeability	✓	✓	✓	✓	✓	✓	✓	✓
Coupon reuse avoidance	✓	✓	✓	✓	✓	✓	✓	✓
Unlinkability	✗	✓	✓	✓	✓	✓	✓	✓
Customer anonymity	✗	✓	✓	✓	✓	✓	✓	✓
Anonymity revocation	✗	✗	✗	✗	✗	✗	✓	✓
Confidentiality	✓	✗	✗	✗	✗	✗	✓	✓
Merchant Disaffiliation	✗	✗	✗	✗	✗	✗	✗	✓
Performance								
Implementation	✗	✗	✗	✗	✓	✗	✗	✗
Analytical	✓	✓	✓	✓	✗	✓	✓	✓

mc, multicoupon ✓ Yes, ✗ No

sm, single-merchant; mm, multi-merchant

coupons. Authors in [66] provide two schemes that improve the efficiency during the issuance and the redemption stages of previous proposals [15,18,53], but they do not provide proofs to support this claim.

Only two of the analyzed proposals dealing with multicoupons provide a limited computing performance evaluation on a laptop [6,32]. In both works, authors evaluate the computational time required by involved protocols, but without considering any kind of network communication or message coding. All entities were executed on an isolated laptop, and all internal operations were not implemented (cryptographic operations were the focus of the performance). Even though we take a look at the single coupons scenario in the literature, we can find similar results about the performance evaluation conducted by the authors of these proposals. There are mobile coupon solutions that present a performance analysis using limited scenarios [16,22,46,67,68], sometimes using some sort of mobile devices [11]. But in the vast majority of cases, authors consider limited laboratory scenarios without taking into account any other relevant elements that can decrease the efficiency of those solutions. Thus, these restricted testing scenarios do not contribute to show real results. In addition, efficiency of multicoupon solutions is a critical concern, even more important than for single coupon solutions.

Summarizing, the performance of multicoupon solutions has been mainly evaluated in terms of the number of cryptographic operations carried out by the involved parties, instead of being based on real measures. Therefore, this type of analysis does not prove the viability of these proposals on real networks and devices. Moreover, it is unclear whether technology to support the required cryptographic background is really available (mainly software and standardized structures). This is a critical concern for the success of electronic multicoupon solutions (and electronic mobile commerce in general), that must be considered from the design to the development stages.

3 A framework for evaluating multicoupon solutions

An evaluation framework must help to define homogeneously the entities, processes and metrics required to evaluate a multicoupon solution. Therefore, the first step is to define which are the entities involved in the scenario and which functionalities should be performed by them.

3.1 Multicoupon scenario: design and definition

Reviewing the literature, we can verify that authors use different names to describe a same entity or a same process. Therefore, as a key part of the framework, we generalize the

multicoupon scenario to give a uniform view of entities, relationships, and functionalities associated with these entities.

We can differentiate four entities and seven main processes. The entities are:

- *Manufacturer (MA)* entity that wants to use multicoupons as a marketing instrument to attract new customers or to increase loyalty of previous customers.
- *Coupon provider (CP)* entity in charge of generating and/or distributing multicoupons to customers, on behalf of manufacturers.
- *Customer (C)* entity who wants to use multicoupons to obtain products or services, with some incentive (discount or gift).
- *Merchant (M)* entity that provides products or services and accepts coupons from a multicoupon (to be redeemed) from a customer.

Although we have defined four entities, multicoupon proposals could involve other entities, usually due to the particular cryptographic schemes used. For example, there are solutions where a public key infrastructure is used for entities' authentication. We do not consider these special entities in this general scenario, because they depend on the security features of each proposal, and are not really specific to the multicoupon scenario.

Regarding processes, i.e., the set of internal processing and communications required between each entity, we can find:

- *Setup/agreement* all entities in the multicoupon scenario should obtain all required data to operate in the scheme (e.g., obtaining secret parameters for security purposes), or interact with other entities aimed to reach an agreement on the terms of the involvement of each entity. For example, some functionalities are typically delegated from some entity to another one, but always under an agreement between them, e.g., the issuance of multicoupons could be performed by a coupon provider on behalf of a manufacturer if those entities agree.
- *Finish* the system must provide mechanisms in order all entities can safely leave the system (remove information from the database, revoke keys, cancel agreements, etc.).
- *Issuing* customers obtain multicoupons directly from the coupon provider or, alternatively, from the manufacturer or the merchant.
- *Redeem* customers redeem coupons with a merchant to obtain goods or services (with a discount or gift).
- *Validation* merchants could require to check the validity of a multicoupon, and thus they should contact with the manufacturer or the coupon provider, before accepting a multicoupon redeemed from a customer.

Fig. 1 Entities and processes involved in a general multicoupon scenario

- *Clearing* merchants send all the redeemed coupons to the manufacturer (or the authorized entity, e.g., a coupon provider) to obtain the amount of money agreed with him.
- *Transferring* a customer could transfer single coupons (from his multicoupon), or even a complete multicoupon, to another customer.

Figure 1 depicts the general scenario described above. However, a less number of entities could be required in particular cases. Sometimes, the manufacturer does not exist, and its functionalities are taken over by a merchant. For example, in a simple case where there are customers and a merchant, issuing and validation processes are performed by the merchant, so the clearing functionality does not appear, because it is not needed.

3.2 Multicoupon deployment: technical alternatives

Once the general scenario is defined (participants and processes are selected), the system must be deployed in the most realistic way to gather appropriate performance measures.

In this section, we give some important considerations that both researchers and developers must take into account at the time upon development, deployment and evaluation of multicoupon solutions. It is not just for multicoupon solutions (or other similar schemes), but also to put into production m-commerce solutions in general.

In order to build the production scenario, we have to select solutions to cover every part of the scenario: the server side (merchant, manufacturer and coupon provider), the customer application, the way how to exchange messages among entities, and how to encode these messages. In general, we have to analyze the available and current solutions based on the following main features:

- *performance*,
- *functionalities*,

- *cost of development and deployment*, and
- *security*.

Moreover, it is also important to study the scalability and the availability of standardized methods to achieve interoperable and scalable multicoupon solutions.

3.2.1 Server platform

In all schemes, there are entities running as servers (i.e. merchants, coupon provider, manufacturer), so system administrators must choose which server platform is the most suitable to run these server applications, according to the most important requirements in this case: scalability, functionalities and security. First, the server platform must provide a reliable and scalable service, so resources should be supplied as needed (e.g., computing power, storage and network bandwidth) without requiring to change the environment. In addition, the solution must provide means to execute complex exchange of messages among server entities and customers (not just request–response such as http-based applications), in order to implement all the protocol flows, no matter how complex they are. Moreover, it must allow the access to disk to store data and logs.

We have two main options to deploy servers: a fully self-managed server or relying on some sort of cloud solution. Next, we discuss the available options to provide the best choice according to the desired features for m-commerce solutions.

On the one hand, a deployment on a self-managed server gives the ability to control and configure all the system parameters, from the hardware infrastructure to the software stack. In terms of evaluating and testing the functionalities of the schemes as a proof-of-concept, a single machine where servers and clients run could be enough. In fact, this is the option chosen by the few proposals that obtain performance measures (the same machine to run server and client developments) [26,32]. However, it does not take into account some external effects that must be considered in a real deployment scenario, such as communication lines, network bottlenecks, bounded computing capabilities, limited availability of resources, security levels. Instead, as a major difference from previous works on implementation and performance evaluations, we reckon on a deployment and testing scenario where servers and clients run on different platforms, even in different locations, as in real scenarios.

On the other hand, cloud solutions are an interesting, affordable, distributed and production-ready solutions to allocate resources without worrying on the underlying physical infrastructure. Cloud customers are typically charged by a cloud provider only for the resources they have been used within a billing cycle. It means that most of the cloud solutions are scalable in a way that customers can add resources as

Table 2 Server platform solutions: a comparative analysis

	Own server	SaaS	PaaS	IaaS
Deploy client apps	✓	✗	✓	✓
Type of applications	Any	Provider-based	Web-based	Any
Resources scalability	Own	Provider	Provider	Provider
Performance tests	Local–global	✗	Global	Global
Security control	Own	Provider	Provider-partially	Client
Examples	–	Google drive [30]	Azure [64]	Amazon EC2 [24]

✓ Yes, ✗ No

they really need. Privacy is yet another critical concern with regard to cloud computing due to the fact that customers' data and business logic are spread among servers, which are owned and maintained by a cloud provider. Therefore, there are potential risks that either confidential data (e.g., financial data, health records) or personal information (e.g., personal profile) could be disclosed to public or business competitors [62,65]. Mainly, cloud solutions can be classified in three different types, based on the functionalities they offer, and the degree of control given to them: Software-as-a-Service (SaaS) [30,69], Platform-as-a-Service (PaaS) [29,64] and Infrastructure-as-a-Service (IaaS) [24,59]. Each of these three types of cloud solutions has key features that have been summarized in Table 2.

Although all server solutions present advantages and disadvantages (see Table 2), the reality is that the choice would be influenced by the type of service or solution to be deployed. So, the server platform should be carefully selected after a deep inspection. For example, the use of an owned server allows to control all the stack, giving this way the ability to apply a fine grained configuration. However, deployments on this kind of platforms imply an inherent complexity to control a large list of features, from the underlying infrastructure to the concrete software application, that usually results on larger costs. Thus, the IaaS cloud solution could be the best choice due to the fact that it joins together the delegation of the infrastructure management to the cloud provider (as opposed to an owned server), and the ability to install and configure the complete software stack (limited by cloud providers in the case of SaaS and PaaS solutions).

3.2.2 Developing client app

Similarly to the considerations about the server platform, some issues should be addressed to choose the way in which the client application must be developed. First, the solution must support not only applications based on web connections (with the request–response pattern) but also connections in which several messages should be exchanged within a single connection. Secondly, it should be able to deal with external software libraries (not only standard libraries) to provide

additional functionalities, such as data encoding or credentials management. Finally, minimum response time and high performance provisioning are both main concerns regarding to mobile applications that require to do complex operations on the client side. Of course, there are other aspects that affect the response time, such as computational power of mobile devices as well as the efficiency provided by external cryptographic libraries, but sometimes developers cannot control them.

There are many frameworks to develop applications for mobile devices. They can be classified in different ways, but for our purpose, we classify them into two main categories: platform dependent and platform independent. Table 3 gives a comparative analysis of these categories.

In order to develop applications based on the platform, we must use native frameworks provided by each platform vendor, such as Android (Google), iOS (Apple) or Windows Phone 10 (Microsoft). This means that the code reusability is low, so this makes supporting multiple platforms costly. However, this alternative has many benefits, since these kinds of frameworks have been designed to allow a rapid development (including simplicity), a direct access to platform APIs, and better performance.

In the second category (platform independent), we find many cross-platform solutions under the concept of *write once, run everywhere* [4,48] (e.g., PhoneGap, Titanium, Corona, JQuery Mobile, React Native). Unlike native development, these solutions are oriented to develop multi-platform mobile applications with a high degree of code reuse. Going deeply into the classification, we can differentiate two main types of platform-independent solutions: web-based frameworks and hybrid solutions.

On the one hand, in web-based developments it is the browser who determines the set of functions available to the web application (e.g., JQuery Mobile, Wink). This means that only technologies provided by all browsers on all devices can be used by web applications. Obviously, these frameworks are HTTP-based, so this fact restricts the type of application to be developed. Moreover, the developed web application is actually a front-end showing data received from the server side.

Table 3 Frameworks for developing mobile apps: a comparative analysis

	Platform dependent	Platform independent	
	Native	Hybrid	Web-based
Access to libraries	Unlimited	Limited (API)	Limited (web browser)
Development cost	High	Low	Low
Client-side operations	Complex	Limited	Screen show based
Communication type	Not limited	Not limited	Request–response
Performance	High	Medium	Low
Security	High	Medium	Low

On the other hand, hybrid solutions are a set of programming languages and APIs provided by the framework (e.g., Titanium, Corona). At the end, there is a conversion step that creates platform-specific applications (native code for each vendor platform). However, the use of these kind of solutions often does not include all functionalities available in native platforms. In fact, cross-platform applications present worst performance than that obtained from native code applications [31].

3.2.3 Communication technologies

In this section, we review the most important available communication technologies, because clients and servers have to communicate through a network. In this case, the decision should consider which is the best option in terms of coverage, bandwidth, cost, user adoption and devices support. Among network access technologies available in current mobile devices, we can classify them according to their range coverage: 3G mobile network (very large coverage), Wi-Fi local area network (medium coverage) and NFC (Near Field Communication) (very short coverage).

The Wi-Fi local area network (IEEE 802.11 related standards) is a widely accepted technology to provide wireless connectivity to homes, offices, and hot-spots. It gives a throughput up to 600 Mbps [34] with a coverage area of about hundreds of meters. However, the throughput depends on many factors (e.g., the number of connected users, the distance to the access point). In any case, this throughput is referred to the internal network capacity instead to the link between the access point and the Internet. Moreover, Wi-Fi is becoming ubiquitous in urban areas, thanks to the growing popularity of community networks such as FON [27], and the rise on the number of cities providing free Wi-Fi access. This makes Wi-Fi communications a realistic scenario in urban areas.

Regarding to 3G, it is based on wireless voice telephony that has evolved to also allow data transmission, providing connectivity with large coverage. Latest deployed versions could provide rates up to 56 Mbps in the downlink, and 22 Mbps in the uplink. As well as Wi-Fi, the data rate is

shared among all users connected to a specific base station, and is affected by other factors, such as movement, weather, distance to base station. In fact, it is common to achieve data rates of about 2 Mbps for stationary users, 384 Kbps for slow-moving users, and 128 Kbps for users in fast-moving vehicles [55,58].

Finally, NFC differs from Wi-Fi and 3G in some key points: the range of coverage, the amount of bandwidth available, and the configuration at the beginning of the transmission [52]. NFC has a radio coverage of few centimeters, so it means that NFC can be only used by proximity communications in which devices are physically close each other. In addition, the bandwidth is very limited because it only allows data transfer rates of around few Kbps. As a difference, NFC does not need any kind of previous configuration since it is enough that devices being within the coverage area.

Both Wi-Fi and 3G are widespread wireless communications [28] that are present in all current mobile devices, and they are also technologies well known by users. In opposition, although NFC is not a new technology, it is not yet widely deployed as well as there are still few mobile devices and providers with NFC support.

3.2.4 Messages

Implementation of protocol messages requires to make a decision about the syntax. There are mainly two syntaxes to structure messages and data in general: XML (eXtended Markup Language) and ASN.1 (Abstract Syntax Notation Number One).

The XML language was developed in 1996 under the auspices of the World Wide Web Consortium (W3C) [13], and it was designed keeping in mind the following objectives: to simplify implementations, to be compatible with HTML, and to be human-legible. Its main drawback is the achieved performance due to its uncompressed textual syntax [49]. Although message length can be reduced using data compression techniques, it increases the internal processing cost. Over the last years, the XML syntax has been the preferred format to transfer business information over the Internet, receiving a lot of attention.

Before the success of XML, security protocols based on cryptography had been defined using the ASN.1 syntax. In broad outline, the ASN.1 [37,38] is a standard way to describe a message (or data structure) to be sent to, or received from, a network. ASN.1 is an ISO/ITU standard based on the OSI model, and it was first defined in 1984. ASN.1 is divided into two parts: on the one hand, the rules of syntax for describing the contents of a message in terms of structure and data type; and on the other hand, the way how to encode and decode each data type to be transmitted.

Some authors have emphasized on the performance efficiency of the ASN.1 syntax over XML [49,57], although their conclusions, from our point of view, rely on partial tests. In [49], authors perform tests on a desktop computer, considering only a single type of structure: attribute certificates. Meanwhile, in another work [57], tests are performed on an isolated powerful laptop (without external message transmission). Moreover, the impact on the user application performance on mobile devices (such as smart-phones, tablets) has not been studied yet. We think that, although interesting, their findings cannot be directly extrapolated to mobile scenarios. Since all tests are conducted locally at the same computer, data about cause–effect of user constrained devices and communications are not considered at all.

Therefore, the choice on the syntax to describe messages has clear implications on their size as well as on the time to encode and decode them. As a consequence, it affects the response time perceived on the client side, which is actually a critical issue, as already explained in the introduction section. In fact, it could lead up to the dissatisfaction on the customer side which would be a reason to even stop using the m-commerce service. Motivated by this issue, we define new performance metrics in Sect. 3.3. Using these metrics, we will analyze in Sect. 7 not only the impact of each communication technology on the mobile device side, but also the performance that entails encoding (at transmission time) and decoding data (at reception time) on the client side.

3.3 Multicoupon evaluation

Once defined, and implemented the scheme, it is important to define performance metrics with the following objectives:

- detecting the processes, or functions that can lessen the performance of the scheme,
- providing homogeneous parameters to compare the efficiency of different solutions.

As explained in Sect. 2, almost all proposals claim that their scheme is more efficient than the previous proposals. But, they were mainly evaluated in terms of the number of cryptographic operations, regardless of other effects (on the

performance), such as delays in communication or data coding.

Next, we define the response time and network overhead metrics, considering separately each process defined in Sect. 3.1:

- *Response time* The required time to execute each process must be evaluated. The time required to complete each process depends on many factors; some rely on the design phase (e.g., the choice of cryptographic tools) or in the deployment phase (e.g., selected cryptographic libraries or the message syntax), as already explained in this section. In order to evaluate this metric in different schemes, it is very important to consider some issues. As different multicoupon proposals can involve different entities and processes, the time of each process must be evaluated properly. For example, the redemption of a coupon by a customer (*redeem*) could require that the merchant must verify the validity of the coupon (*validation*). Thus, the total time perceived by the customer must reflect the execution of both processes.
- *Network overhead* The network resources used by each process are measured as the amount of data transferred by each message (message size). The choice on the syntax to describe messages has clear implications on the messages size (Sect. 3.2.4), and in turn it has an influence on several parameters, such as delay in communications. Moreover, the selected cryptographic tool (to provide different security requirements) may imply that participants must exchange a larger volume of information.

A total response time of each process can be an insufficient metric to evaluate a multicoupon scheme. If we want to improve the scheme, it is useful to identify which factors introduce a larger overhead to the different processes. Thus, we divide the total time used by each process into the following three metrics:

- *Time spent on processing data* This refers to the computational and memory access operations, not related to communications. There are several aspects that affect this processing time, such as computational power of mobile devices, efficiency provided by external cryptographic libraries, the used framework for developing.
- *Time spent on encoding data to be sent and decoding received data* One of the factors not considered in previous proposals is coding, and as explained in Sect. 3.2.4, this could have clear implications on performance.
- *Time delay due to communications channel* Although the best option in terms of coverage, cost, user adoption and devices support, must be considered (as explained in Sect. 3.2.3), the delay in sending and receiving data must be also analyzed. Sometimes, sending short messages

may be more feasible than sending fewer messages but longer.

3.4 How to apply the framework

Two main multicoupon scenarios exist in the literature: single-merchant and multi-merchant scenarios. Regarding the single-merchant scenario [15,18,19,25,53], merchants mostly are in charge of issuing multicoupons to customers and afterward verify these coupons. Hence, several processes are not required because are internal functions, i.e., merchants perform manufacturer and coupon provider functionalities, and they do not require to execute the validation process. However, customers have to obtain different multicoupons to operate with different merchants. Therefore, its main drawback is the flexibility.

In the multi-merchant scenario [6,32,33,39], a coupon provider must be included in order to issue multicoupons to customers on behalf of a manufacturer. Thus, customers can use these multicoupons at different merchants, and these merchants must have a previous agreement with the issuer before accepting multicoupons. This is the most complete scenario, and to verify the applicability of the presented framework, we select two existing multi-merchant proposals that we define, deploy and evaluate under the specifications of the framework.

The first proposal is the multicoupon scheme proposed in [32], because this is one of the latest published proposals of multicoupons. And the second one is the scheme proposed by Hsueh et al. [33], which is one of the most cited solutions for multicoupons. Moreover, both solutions are well defined by authors giving a detailed description about messages flow, the content of messages, and the validations to be performed by each participant. This is a very important issue in order to implement those solutions properly.

4 Definition of the Hsueh et al. [33] Proposal

In this section, we briefly describe the scheme proposed in [33], applying the framework defined in Sect. 3.

In Sect. 4.1, we take a glance about the relevant cryptographic tools considered by the solution. Next, in Sect. 4.2 we unify the definition of the proposal, notation and processes involved in the protocol, under the specifications of the framework.

4.1 Relevant cryptographic tools

In order to study the deployment viability of any multicoupon scheme (software availability and efficiency on constraint devices), it is important to know which cryptographic functions are required, as we explain in Sect. 6. Next, we present

the most relevant cryptographic tools used by [33] to provide authentication and confidentiality: asymmetric cryptography. This is a well-known cryptography, so we will not extend the explanation. We just give the notation and procedures necessary for their use.

Asymmetric Cryptography. The notation of the most relevant elements generated by a public key scheme is:

- pk_i The public key for user i .
- sk_i The private key for user i .
- $Cert_i$ The certificate for user i .

The above parameters may be used for the following procedures:

- *CertGen* An entity i generates a key pair (pk_i, sk_i) and requests the authentication of the keys to an authority. At the end of this procedure, a certificate is issued to the requesting entity ($Cert_i$). This certificate contains information about the identity of the requesting entity and her public key (pk_i).
- *Sign_i* (m) The signer i uses her secret key sk_i to compute a signature on a message (m). This procedure provides non-repudiation of the information.
- *Verify* A verifier uses the public key pk_i (included in $Cert_i$) to verify whether the provided signature is valid and made by the correct user (owner of the corresponding sk_i).
- $c = \mathbb{E}_{pk_i}[m]$ A sender encrypts a message (m) using the public key of the recipient giving as a result a ciphertext (c). This procedure provides data confidentiality; only the owner of the private key can read the message.
- $\mathbb{D}_{sk_i}[c]$ A recipient decrypts c using her private key.

4.2 Scheme sketch: unifying actors and processes

Under the framework notation, in the [33] proposal we identify a manufacturer, a coupon provider, merchants, and customers. Moreover, [33] requires the participation of a Certification Authority to provide key certification, as explained in Sect. 4.1.

Regarding processes, the main difference from the general scenario is that the $\mathcal{MA}$ is involved in both the issue and the transferring process. In this scheme, $\mathcal{MA}$ issues multicoupons and delegates their management to $\mathcal{CP}$.

Next, we explain the main tasks that should be performed in each identified process: Setup, Issuing, Redeem, **Validation**, Transferring, and Clearing.

1. Setup. Although authors do not specify a setup process, all entities $\mathcal{C}$, $\mathcal{M}$, $\mathcal{CP}$ and $\mathcal{MA}$ should execute this process to receive a public key certificate ($Cert_X$, where $X = \mathcal{C}, \mathcal{M}, \mathcal{CP}, \mathcal{MA}$) from a Certification Authority (CA), attesting that a public key (pk_i) belongs to a specific entity.

Fig. 2 [33] Setup process flow: customer certificate request

We have included in Fig. 2, as an example, the steps that should be conducted by $\mathcal{C}$. As we can see, $\mathcal{C}$ sends a certificate request, and at the end of this process, she obtains the corresponding certificate.

2. Issuing. $\mathcal{C}$ is allowed to engage with $\mathcal{CP}$ to issue a signed multicoupon, $\mathbb{MC}$. Three entities are involved in

this process (Fig. 3): $\mathcal{C}$, $\mathcal{CP}$, and $\mathcal{MA}$. Below, we review the structure of a $\mathbb{MC}$. It consists of the following two elements:

- $\mathbb{MC}_{Sign}$ It is a signed structure containing: root indexes (X_0, Y_0) , an expiration date (EXD) , a mobile customer identification (MUC) , and a serial number (SN) .

$$\mathbb{MC}_{Sign} = \text{Sign}_{\mathcal{MA}}(X_0, Y_0, EXD, MUC, SN)$$

- $\mathbb{MC}_x$ It is the data structure that defines all coupons within a multicoupon. The solution generates iteratively (by means of a hash chain procedure, $H(X_{i+1}) = X_i$) p indexes from an initial secret index X_p up to the last index, called *root* (X_0) .

$$\mathbb{MC}_x = X_{p-1}, X_{p-2}, \dots, X_0$$

$\mathcal{C}$ starts the process sending a signed request to $\mathcal{CP}$. She provides the unique code of the $\mathcal{C}$'s mobile device (MUC), along with data about the coupon (the authors do not specify this information). Upon receiving data, $\mathcal{CP}$ encrypts the data received from $\mathcal{C}$ plus a serial number (to identify the multicoupon to be generated) and sends it to $\mathcal{MA}$. $\mathcal{MA}$ is

Fig. 3 [33] Issuing process flow

Fig. 4 [33] Redeem process flow

the entity in charge of generating the random values used as seeds to calculate the indexes of the multicoupon. He generates MC_{Sign} and encrypts the seeds. These encrypted seeds are forwarded to C by CP , and MC_{Sign} is stored by CP .

In this process, CP acts as an intermediary (she only generates the serial number). And curiously, MC_{Sign} is never known by C , but she can compute the indexes belonging to MC , i.e., MC_x .

3. Redeem. The Redeem process includes the functionalities of the Validation process and works as follows (Fig. 4). C prepares an array of data filled by the first unused coupon index (X_i), together with the serial number of the multicoupon, and an identifier for the current transaction (R_{id}), and sends it to M . M forwards this information (plus his identifier, ID_M) encrypted with the public key of CP , to CP . CP uses SN to obtain the corresponding MC_{Sign} , and updates the database of redeemed coupons. Then, CP sends MC_{Sign} to M who checks if X_i provided by C belongs to MC_{Sign} . If so, he sends a proof of redemption to C and the process ends.

4. Clearing. M can request MA a money transfer in exchange of a received coupon from a customer (Fig. 5). To request a clearing, M has to provide MA with the serial number of the multicoupon, the index identifying the specific coupon in the multicoupon, and the proof sent to C at the end of the Redeem process. If all validations hold, MA

Fig. 5 [33] Clearing process flow

authorizes a deposit to the account of M based on a previous agreement between both parties.

5. Transferring. The owner of a multicoupon (C_1) can transfer a shareable multicoupon to another customer (C_2) (Fig. 6). First, C_1 sends the secret index Y_q , SN and CP 's public key to C_2 . Next, C_2 encrypts her MUC and sends it to C_1 , who finally returns this information signed to C_2 as a proof of the transferring.

5 Definition of the Hinarejos et al. [32] proposal

In this section, we give an overview of the scheme proposed in [32], applying the framework in a similar way as Sect. 4

Fig. 6 [33] Transferring process flow

does. In Sect. 5.1, we give a short review about the relevant cryptographic tools considered by the solution. Next, in Sect. 5.2 we unify the definition of the proposal, notation and processes involved in the protocol, under the specifications of the framework.

5.1 Relevant cryptographic tools

Next, we present the most relevant cryptographic tools used by [32] to provide anonymity and untraceability: group signatures and partially blind signatures.

GROUP SIGNATURE A group signature is a public key signature scheme that involves a group of signers, each of them holding a membership certificate, composed by a group public key (common to all group members), and a secret and individual signing private key [7]. Any member of this group can sign messages on behalf of the group, while the identity of the signer is hidden within the group. The resulting signature is publicly verifiable by anyone regardless of whether the verifier belongs to a group or not. However, only a third party (called group manager) can trace the signature in case of dispute or abuse, revealing this way the identity of its originator.

There are several group signature proposals in the literature, but the authors of [32] use the pairing-based Boneh–Boyen–Shacham (BBS) group signature [10]. Next, we provide the notation of the most relevant elements generated by the BBS group signature scheme:

- pk^G The group public key
- sk_i^G The secret signing key for user i
- σ The group signature on a message m

According to [7], a group signature is defined by the following procedures:

- **KeyGen^G** The group manager computes a group public key (pk^G) for a group of n members, and n secret signing

keys ($sk_i^G, \forall i \in [1, n]$) to be assigned to each member within the group.

- **Sign^G** The signer uses her own secret signing key sk_C^G to compute a signature (σ) on a message (m).
- **Verify^G** A verifier uses the group public key (pk^G) to verify whether the provided group signature (σ) is valid and made by a user who belongs to the group.
- **Open^G** The group manager traces a signature revealing the real signer's identity.

Every relevant element is composed by different values obtained by means of complex operations that rely on mathematical groups, which are out of the scope of this paper. We address the reader to [10] for a detailed description of the group signature scheme [40].

PARTIALLY BLIND SIGNATURE The notion of partially blind signatures was introduced in [1] to address the main shortcoming of blind signatures: the signer has no control over the message to be signed except for those bound by the public key. It means that in blind signatures, the signer can only sign data provided by the requester, but the latter cannot add any additional information to the signature. So, a partially blind signature allows the signer to explicitly include some information (usually called common information) into the blind signature, under some previous agreement with the requester. Therefore, this concept is a generalization of blind signatures in such a way they are a special case of partially blind signatures, where the common information is a null string [54].

According to [2], a partially blind signature is defined by the following procedures:

- **Init^{PBS}** Both signer and requester obtain a key pair (pk_i, sk_i) from a secure public key cryptosystem (such as RSA).
- **Request^{PBS}** The requester asks for a partially blind signature to the signer, sending the message to sign (m) in a way the signer cannot read it (by blinding data), together with the previously agreed common information (Γ).
- **Sign^{PBS}** Upon receipt of data from the requester, the signer uses his secret key to compute a signature on this data (σ') and sends it to the requester.
- **Extract^{PBS}** The requester unblinds the received signature and obtains the final expression of the partially blind signature (σ).
- **Verify^{PBS}** A verifier can check the partially blind signature (σ) to determine whether it has to be *accepted* or *rejected*.

Similar to group signatures, there are several partially blind signature schemes suitable to be used in mobile multicoupon scenarios. Authors selected a simple and secure scheme based on the RSA factorization problem [20].

5.2 Scheme sketch: unifying actors and processes

Under the framework notation, in the [32] proposal we identify a coupon provider, merchants, and customers. Besides entities defined in the framework, [32] considers a new entity, different from previous proposals on multicoupons, called group manager ($\mathcal{G}$). This party comes due to the group signature scheme (see Sect. 5.1), and it is the entity in charge of providing anonymity to customers, as well as it is the only party who can revoke the anonymity of customers when it is required (i.e., when a customer misbehaves).

Regarding processes, as the manufacturer does not appear, the coupon provider should perform her functions. Thus, the clearing process is carried out between a merchant and the coupon provider. Moreover, the setup, registration and affiliation processes defined in [32] correspond to the setup process of the framework.

Next, we explain the main tasks that should be performed in each identified process: Setup, Issuing, Redeem, Validation, and Clearing.

1. Setup Both $\mathcal{G}$ and $\mathcal{CP}$ must execute a setup protocol in order to receive requests from the other parties. $\mathcal{G}$ has to execute the $KeyGen^G$ group signature procedure to create a public group key (pk^G), and the related set of secret signing keys for signers ($sk_i^G, \forall i \in [1, n]$). Moreover, each customer who wants to use multicoupons has to register her real identity at $\mathcal{G}$, in order to receive a group key pair (pk^G, sk_C^G) used to sign, and to prove the fact that she actually belongs to the claimed group. During the customer registration (Fig. 7), $\mathcal{G}$ links the real identity of $\mathcal{C}$ to the corresponding signing key in order to provide anonymity revocation in case of misbehavior.

Besides, $\mathcal{CP}$ and each $\mathcal{C}$ have to call to the $Init^{PBS}$ procedure from the partially blind signature scheme to deploy their own RSA key pair. Meanwhile, each $\mathcal{M}$ who wants to accept multicoupons issued by $\mathcal{CP}$ must affiliate to $\mathcal{CP}$.

Fig. 7 [32] Setup process flow: customer registration

2. Issuing Once $\mathcal{C}$ is registered with $\mathcal{G}$, she is allowed to engage with $\mathcal{CP}$ to issue a signed multicoupon, MC . Before we explain how the Issue protocol works, we review the structure of MC (Fig. 8). It is composed by the two following elements:

- MC_ω It is the data structure that defines all coupons within a multicoupon. Therefore, given a number of m coupons, the solution generates iteratively (by means of a hash chain procedure) $2m + 1$ hash identifiers from an initial random and secret *multicoupon seed* (ω_{seed}) up to the last hash identifier, called *multicoupon identifier* (ω_0). Then, each coupon (c_i) is defined as a pair of consecutive hash identifiers, where the left identifier is called *payment information* ($c_i^{pay} = \omega_{2i-1}$) and the right one is called *proof information* ($c_i^{proof} = \omega_{2i}$), for all $0 < i \leq m$ (i denotes the i -th coupon within the set of coupons). $\mathcal{C}$ has to keep MC_ω in secret, with the exception of ω_0 , which is part of the public information MC_{PBS} .
- MC_{PBS} It is the partially blind signature over the multicoupon identifier (ω_0). The final MC_{PBS} also conveys some verification data (*verif_data*), commonly agreed beforehand (Γ) between $\mathcal{C}$ and $\mathcal{CP}$, which defines the multicoupon features. These features can be different depending on the concrete application or service, but it would be common to consider parameters such as the number of coupons within the multicoupon, the value or discount achieved by each coupon, time marks to fix their validity.

Once $\mathcal{C}$ has generated MC_ω , she starts the issuing process (Fig. 9) by blinding the multicoupon identifier (ω_0) from $\mathcal{CP}$ (using $Request^{PBS}$). Upon receiving data, $\mathcal{CP}$ verifies the common information (Γ) and signs the received blinded data together with Γ by using his private key (sk_{CP}). During the last step (by means of $Extract^{PBS}$), $\mathcal{C}$ completes the process without further involvement of $\mathcal{CP}$. As a result, $\mathcal{C}$ unblinds

Fig. 8 [32] MC structure composed by MC_ω and MC_{PBS}

Fig. 9 [32] Issuing process flow

the received signature and obtains the final MC_{PBS} . This way, CP does not know all the final values of MC_{PBS} , so CP cannot trace the multicoupon. The Issuing process is completely anonymous because neither CP knows any data about the identity of C, nor the resulting MC contains information related to her identity. Thus, C can operate from this moment with her issued MC in an anonymous way.

3. Redeem Now, C is ready to redeem (Fig. 10) coupons from her multicoupon to obtain a service from any M affiliated to CP. The Redeem process is defined by four steps, in which C can redeem either single or multiple coupons using a single protocol call. This is an important enhancement from the point of view of computing and networking efficiency, not considered by other authors.

The Redeem process works as follows. C prepares an array of data ($data_1$) filled by the first unused *payment information* (c_i^{pay}), together with the target M's identifier (id_M), and an identifier for the current transaction (id_r). Then, C signs this data with her secret signing key ($Sign_C^G(data_1)$), and sends it to M, who applies some verifications, such as group signature and partially blind signature validations, he checks whether the provided coupon belongs to MC, and whether it was not previously redeemed, etc. If all verifications are correct, M acknowledges it by means of a signature on the provided service (s) and $data_1$ (he communicates that he is committed to the transaction). Then, both C and M are engaged in a similar exchange as the previous one, by sending the corresponding *proof information* (c_i^{proof}). If all verifications hold, M updates his stored data and acknowledges C in a similar way as before. The reuse avoidance can be performed checking whether any single coupon is either in the local database, or in the CP's global database (executing online the Clearing process).

4. Clearing (online or offline). Similar to [33], the Validation process is included in another process, in

Fig. 10 [32] Redeem process flow

this case, in the Clearing process. M can request CP a money transfer to his account balance in exchange of a list of received coupons from customers (Fig. 11). M is allowed to claim coupons, either every time a set of them is received from C (on-line Clearing), or when he has a list of received coupons from customers (offline Clearing).

To claim coupons, M has to provide CP with information received from customers during the Redeem process, i.e., $data_1$ during the first step, and $data_2$ during the third step of the Clearing process. Upon validation, CP acknowledges this data telling M whether these coupons had been used before or not (reuse detection). If all verifications hold, CP authorizes a deposit to the account of M with the right amount of money, according to either the number and value of provided coupons, or based on a previous agreement between both parties.

6 Multicoupon solutions implementation

In this section, we describe the implementation environment selected to develop, deploy and thereafter evaluate the perfor-

Fig. 11 [32] Clearing process flow

mance of the two multicoupon solutions presented in Sects. 4 and 5. Figure 12 maps the processes defined in Sects. 4.2 and 5.2 to a general deploying scenario. At the beginning, we present the concrete technologies selected from the alternatives discussed in the framework (see Sect. 3.2). Secondly, we list the main libraries used along the code development. It is worth to mention that the development has been conducted with the Java language. Finally, we provide some insights about how a multicoupon customer app has been designed and developed together with the experience provided by the app's user interface.

Fig. 12 Devices and communications planning

6.1 Prototype environment

6.1.1 Server platform

Regarding to the server platform, we select two different options from those described in Sect. 3.2.1: an own fully managed server and two Amazon EC2 instances. Figure 12 shows all communications among parties for both selected multicoupon solutions, and we also differentiate between communications proposed by [32] (solid lines) and those defined by [33] (dotted lines). For example, the Issuing process involves $\mathcal{C}$ and $\mathcal{CP}$ for both solutions, but this process also covers the participation of $\mathcal{MA}$ in [33]. In the same way, the Redeem process involves both $\mathcal{C}$ and $\mathcal{M}$ for both solutions and also $\mathcal{CP}$ in [33]. Finally, the Clearing process is the only one in which two servers are involved, where communications are performed between $\mathcal{M}$ and $\mathcal{CP}$ in [32] and between $\mathcal{M}$ and $\mathcal{MA}$ in [33].

6.1.2 Client application

Concerning to the platform to develop the client application, we have studied each considered mobile platform from Sect. 3.2.2, and we have decided to develop the application's core on the native Android Platform, instead of other systems, due to the following key reasons:

- Android provides a well-structured API to developers to build applications using the widespread and supported Java language.
- It offers a wide market of commercial handset devices to test developments.
- It is supported from a huge community of developers with a large number of resources, examples and so on.
- Android is open source, and so, anyone can contribute with enhancements.
- Android is supplied free of charge.

- A native framework performs better than cross-platform frameworks.
- Cryptographic operations must be implemented, and these operations can only be implemented with native code.

In addition, developing on the Android Platform covers a large percentage of mobile devices, and a large market share. In fact, according to data provided by Google, Android is the leading mobile OS, and the platform with better growth rates, reaching recently up to 1.4 billion active users worldwide [63]. It means that more than 1.5 million of new Android devices are activated every day.

Even though the application's core is implemented using Java, we decide to implement the User Interface (UI) with PhoneGap [56], as an example of a cross-platform development framework that falls into the category of hybrid solutions (see Sect. 3.2.2). It contributes to decrease the time invested on application development, because PhoneGap allows developers to build mobile applications using web technologies, such as HTML5, Javascript and CSS.

Notice that the same interface could be used by all solutions, because all processes are transparent to customers. That is, customers only need to obtain coupons and redeem them regardless of the underlying solution. Thus, the used technologies for the UI are valid for [33] and [32].

6.1.3 Communication technology

As described in Sect. 3.2.3, the next choice is the communication technology. Although NFC seems a promising radio technology suitable to be applied to proximity transactions, we discard it, not only because it is not yet a mainstream technology, but also due to the lack of full support of the Android API. In fact, the Android API does not allow accessing all the NFC features, such as full peer-to-peer NFC communication to implement complex message exchanges. Instead, the API only exposes the Android Beam feature. It consists on a very limited peer-to-peer NFC communication, where an application can only send some data (text, contact card, image, photo, etc.) to another Android device with NFC. However, it does not allow to receive messages different than confirmation messages, reporting whether transmissions were finished properly or not. Summing up, we select the Wi-Fi and 3G wireless technologies to provide a consistent comparative analysis, because these are the technologies available in all current mobile devices.

6.1.4 Messages

The way to encode messages is an important choice as regards customer's response time and the amount of data transferred to the network, as stated in Sect. 3.2.4. So, we evaluate both

XML and ASN.1 to provide an extended analysis. This way, we get performance measures to study the effect of message syntax and encoding on the whole performance of [33] and [32].

6.2 Key distribution

The two multicoupon schemes, that we are analyzing, use public keys that should be distributed securely. X.509 digital certificates [21,36] are a very extended instrument to certify data (such as public keys, user privileges), in which an entity called CA (Certification Authority) is in charge of managing X.509 Public Key Certificates (PKCs).

The security of [33] is based on Public Key Infrastructure (PKI), that relies on the use of X.509 PKCs. Thus, PKCs can be used without changes.

On the other hand, [32] uses a group signature scheme to provide revocable anonymity of customers (as described in Sect. 5.1). So, it considers a group manager entity as the party who is in charge of generating and distributing group public keys (along with the corresponding set of user private keys), in a secure way. However, group signature schemes do not define how group public keys have to be distributed. It is an important drawback when a solution based on group signatures has to be deployed on a real and productive scenario. In the following, we propose a method to provide certified credentials when group signatures are involved.

The main difference, in reference to X.509, is that in a group signature model, a set of users have the same group public key, while they have different associated private keys. But, in X.509 keys come in pairs (one public key corresponds uniquely to a single private key) in which the private key is bound to the identity of a single entity. It is completely different from group signatures. Therefore, we have to provide a solution to fit group signatures into the X.509 public key infrastructure.

We address the three most important requirements that the solution for group public keys management must accomplish. Firstly, the certificate should not contain any information to identify or link the signer. Secondly, data included in the certificate must hold the group identifier and the corresponding group public key. Lastly, the format of the certificate specification must be supported by the most widespread user applications, e.g., Transport Layer Security (TLS) protocol.

```

X.509 Certificate Profile
-----
BSCertificate ::= SEQUENCE {
    -----
    subject          Name,
    -----
    subjectPublicKeyInfo SubjectPublicKeyInfo,
    extensions       [3] EXPLICIT Extensions OPTIONAL
}

```

If we take a look at the X.509 v3 certificate format, reproduced in the listing X.509 Certificate Profile, it can be observed that the definition of the fields does not restrict its use for other data structures.

```

Public Key Info
-----
subjectPublicKeyInfo ::= SEQUENCE {
    algorithm            AlgorithmIdentifier,
    subjectPublicKey     BIT STRING
}

```

The SubjectPublicKeyInfo (listing Public Key Info) field is used to carry the public key and to identify the algorithm used (e.g., RSA, DSA) [21]. However, this field can contain any ASN.1 type encoded as a string of bits. For the particular case of the BBS group signature, listing ASN.1 Group Public Key represents the ASN.1 syntax to define the BBS group public key (see Sect. 5.1).

```

ASN.1 Group Public Key
-----
SGSPublicKey ::= SEQUENCE {
    g1      OCTET STRING,
    g2      OCTET STRING,
    h       OCTET STRING,
    u       OCTET STRING,
    v       OCTET STRING,
    omega   OCTET STRING
}

```

The most important point about group public keys is that they can be represented as a complex and structured ASN.1 type composed by ASN.1 simple types. Thus, the output of the SGSPublicKey structure can be converted into a bit string accepted as a value of the field subjectPublicKey in the X.509 standard. The Subject field defines who is the owner of the public key (who holds the public key). In our case, it is the group identifier instead of a specific user identity. This way, the Subject field identifies the concrete group instead of linking the signer.

To correctly process the certificate, it must include information about the type of public key that it conveys. This information can be included in the field called algorithm from the structure SubjectPublicKeyInfo. However, there is not a standardized value to identify the group public key, although X.509 allows to define new identifiers as needed. To maintain the interoperability, we propose to use the extensions field to specify that this certificate is in fact a group key certificate. This new extension must be defined as critical, i.e., this extension must be understood by any entity in charge of validating the certificate.

Moreover, the use of X.509 allows taking advantage of the already available tools, as well as the infrastructure to manage X.509 certificates. That is, from the use of secure containers to store certificates and public keys (e.g., PKCS12, PKCS8), to the PKC management messages (e.g., CMP) [3,43].

Therefore, using the X.509 PKC we accomplish the three requirements for using group public keys: unlinkability to the private key holder, feasibility of group public key representation, and the use of standardized mechanisms to manage these keys.

6.3 Secure storage of credentials

Server entities, as much as the customer application, must store credentials in a secure way. We use the features provided by the KeyStore Java class, which exposes methods to store credentials in a secure repository protected by password. This way, only authorized users and applications can access and extract information from the key store after they type the right password.

However, the way Android exposes key stores to users and applications is different from the standard KeyStore class. Android provides the *system key store*, which is something like a vault where only authorized users and applications can access stored private data. Unfortunately, for versions before to 4.0 (Ice Cream Sandwich), the *system key store* is only available for VPN and Wi-Fi authentication procedures. Thus, applications are not granted the access to it. Instead, Android 4.0 and up bring several enhancements that allow applications to access the *system key store* to keep personal credentials, such as private keys and digital certificates, by means of a new API named KeyChain [8]. However, it is only available from 4.0 Android version. To preserve compatibility with older devices, we use the standard method based on the KeyStore class. So, the application saves his own key store on the smart-phone's storage.

6.4 Libraries

In addition to the standard Java libraries, a number of third-party libraries are used. Next, we outline the most important ones:

- *Java Bouncy Castle* The Java Bouncy Castle library [12] is a very complete Java library that implements cryptographic algorithms and serves as a provider for the JCE (Java Cryptography Extension) framework. The Bouncy Castle library presents a lightweight and comprehensible API suitable for its use in any Java environment. Moreover, it is continuously supported and improved by a large group of developers. The Bouncy Castle library is used along the developed code as a common framework, methodology and provider to use some cryptographic primitives, in addition to build ASN.1 encoding and decoding procedures where they are needed.
- *Spongy Castle* Due to class name conflicts, Android does not allow applications to include the official release of Bouncy Castle. It happens in older Android devices due

to the differences between the internal version of Bouncy Castle (Android includes an old and trimmed down version of Bouncy Castle), and the last release of the library bundled by the application. However, there is a project called Spongy Castle [61] distributing a renamed version of the library to work around this issue. It renames all the original packages from `org.bouncycastle.*` to the new one space on `org.spongycastle.*`. As a result, all of the features of the last version of Bouncy Castle can be included in any Android version.

- **Simple XML Framework for Java** The serialization of Java objects into a representation using XML can be a tedious work without using any specialized library. The first considered option was the JAXB (Java Architecture for XML Binding) [41], as it is the Java reference implementation for serializing objects to XML. However, it cannot be used on the Android platform due to the fact that Android does not allow calling to classes belonging the package `javax.xml.*`. As an alternative, we found the Simple XML Framework for Java [60] which is a lightweight but powerful library that provides a high-performance XML serialization and configuration framework for Java. Simple XML Framework works without further configuration or problems on Android, and it does not depend on any other underlying library.
- **Group signature implementation** We developed a custom library [40] to deal with computation related to the BBS group signature scheme required in [32]. The library relies heavily on the jpBC library [23], which is a

specialized Java library to work with pairing-based computations.

6.5 Customer application and user experience

Customer's app needs to be as simple as possible, keeping in mind that maybe users do not have previous skills in similar electronic commerce solutions. Thus, our implementation tries to minimize the customers effort at the time of using the multicoupon scheme.

As discussed in Sect. 6.1.2, a same application's UI (User Interface) could be used by both proposals ([33] and [32]), because customers only need to obtain multicoupons and redeem them regardless of the underlying proposal.

The general application's UI is organized into six screens in order to manage all processes related to a multicoupon scheme. In addition, we design some views to manage multicoupons, such as windows to show the remaining coupons or to configure parameters from digital certificates. Figure 13 shows three views captured from the customer application adapted to the [32] proposal.

The following list summarizes the functionalities of each application view:

- **Register View** This view allows the customer to request the needed credentials. It shows customer terms and conditions of the service, and it allows to start the registration process. At the end of the process, this view stores in a

Fig. 13 Screenshots from the developed UI of the Android customer application adapted to [32]. **a** The *IssueNewMC* view, **b** the *ManageMC* view, **c** the *Settings* view

secure way (see Sect. 6.3) the credentials received from a group manager ([32]), or a certification authority ([33]).

- **Issue New MC View** If the customer already owns her credentials, she can execute the Issuing process (see Fig. 13a). Depending on each particular solution, the consumer could choose some parameters, for example, the number of coupons included in a multicoupon. After that, the Issuing process starts, and without any more interaction from the customer, the multicoupon is received and stored.
- **Redeem View** This view allows the customer (in case she already has previously obtained a multicoupon) looking for merchants where multicoupons can be accepted. When a customer is interested in something offered by a particular merchant, she starts the Redeem process only clicking a button. In case a customer has more than a single multicoupon stored, the application requests the customer to choose which one has to be redeemed. The application returns the result of the redeem execution.
- **Manage MC View** This view provides the customer with an interface to manage her multicoupons (see Fig. 13b). Several data can be shown about each multicoupon: how many coupons are remaining, the expiration date, the purchase history, etc.
- **Settings View** It allows the customer to configure some application settings, such as install new personal certificates (see Fig. 13c).

- **Log View** The application stores all actions performed by the customer. It can be configured to show either informative messages (such as process beginning and finishing events) or more complex information (such as full message contents).

7 Performance evaluation of the multicoupon solutions

In this section, we present the performance measures obtained from the realistic implementation of the multicoupon schemes explained in Sects. 4 and 5. We describe the considered test scenario in Sect. 7.1, and then we conduct an extensive performance analysis in Sects. 7.2 and 7.3, evaluating the metrics of performance defined in Sect. 3.3.

7.1 Test scenario

In line with the considerations we stated in Sect. 3.2 and the reasoned choices we made in Sect. 6, Fig. 14 depicts the testing scenario we use to obtain performance measures. Moreover, Table 4 describes the main technical features of all selected testing services and devices, together with a short name to identify them easily throughout the analysis, as well as which entity is running in each server.

Regarding the server platform, as already stated in Sect. 6.1, we have picked out the Elastic Compute Cloud solution (EC2 for short) from Amazon. We have deployed and configured two cloud instances (that we call CS-1 and CS-2) running a custom Linux OS installation preconfigured by Amazon. Among the available options, we have selected the free tier micro instance (t1.micro) [5], because it is free of charge and provides a suitable level of resources to obtain consistent performance measures. Both EC2 instances are in different AWS availability zones (AWS-AZ), i.e., they are not physically close, because CS-1 has been deployed in Virginia AWS-AZ (EEUU), while CS-2 instance has been allocated in Frankfurt AWS-AZ (Europe). This way, our testing scenario models a complex production environment where machines may be allocated far away from each other. Besides, we also consider the use of a virtual machine (called

Fig. 14 Considered performance testing scenario for multicoupon solutions

Table 4 Considered testing devices

Device	Entity	Name	CPU	RAM	OS
Virtualbox machine	$\mathcal{M}$	PS (single-core 2.8 GHz)	1 virtual core	1 GiB (Last stable)	Debian Linux
EC2 t1.micro	$\mathcal{M}A$	CS-1 (≈ 1.0 – 1.2 GHz)	2 EC2 CU ^a	600 MiB (customized)	AWS Linux
EC2 t1.micro	$\mathcal{CP}$	CS-2 (≈ 1.0 – 1.2 GHz)	2 EC2 CU ^a	600 MiB (customized)	AWS Linux
HTC desire	$\mathcal{C}$	Desire (single-core 1 GHz)	Qualcomm snapdragon	512 MiB (API level 8)	Android 2.2
Google Galaxy Nexus	$\mathcal{C}$	Nexus (dual-core 1.2 GHz)	Texas instruments OMAP	1 GiB (API level 17)	Android 4.2.2

^aOne EC2 CU (compute unit) provides the equivalent CPU capacity of a 1.0–1.2 GHz 2007 Xeon processor [5]

Table 5 Network features of each communication path (estimated average)

Communication path		Transmission rate (est. avg.)		Latency (est. avg.)
Origin	Target	Downstream (Mbps)	Upstream (Mbps)	Round-trip (ms)
CS-1	PS	> 25	> 25	< 100
Customer app (ADSL)	CS-1/CS-2	< 3	< 0.3	> 200
Customer app (3G)	CS-1/CS-2	2–8	2–4	> 200

PS) running a Debian Linux OS allocated in a physical server in our corporate network.

Concerning the client platform, the developed Android client application is tested using two different devices equipped with different versions of the Android OS. They are, from top-class to bottom-class devices, Google Galaxy Nexus and HTC Desire. Each of them has different computing power, so testing the scheme with different devices with heterogeneous levels of performance will be very useful to prove whether the multicoupon schemes are in fact suitable to be widely used by different customer devices.

Table 5 summarizes features (average transmission rate and latency) from network connections among servers and customers. As stated before (see Sect. 6.1), we selected two different wireless network technologies: a 802.11 g local wireless network attached to Internet through a commercial broadband, and a 3G mobile network.

Thus, the considered testing scenario reproduces the conditions we usually find on a real production scenario. On the one hand, mobile customers are very often connected to wireless networks in which transmission rates depend on several factors: movement of customers, peak-hours with high demand from users, distance to base stations, etc. On the other hand, servers usually have a lot of available resources, such as computing power and communication lines.

We execute performance tests for Issuing, Redeem, and Clearing, and each test is performed 20 times to obtain average measures.

7.2 Performance overview: response time and network overhead

Taking into account the testing scenario described in Sect. 7.1, we present performance results focused on the response time perceived by the customer, and the network resources consumed by each process (measured as the amount of data transferred by each message).

For the study about the message syntax implications (ASN.1 vs. XML), we analyze two main factors not previously considered by existent multicoupon solutions: first, the real message size transmitted on the network, and secondly, the time required to encode and decode messages (see

Fig. 15 Total message length for each process in [32] **Fig. 16** Total message length for each process in [33]

Sect. 7.3). Figures 15 and 16 show the results about the length of messages generated by each process considering the influence of the message syntax. If we analyze the most used processes (Redeem and Clearing), the network overhead can be reduced about a 50% using ASN.1 instead of XML to encode messages in [32]. Clearly, the use of the ASN.1 syntax reduces nicely the network overhead, arriving up to 62% for the Issuing process in [32]. In [33], the use of ASN.1 reduces the overhead about a 13% for all processes. Although this improvement does not seem very significant in [33], it has clear implications on the general efficiency, as discussed below.

Regarding the total response time for each protocol, Fig. 17 shows the results obtained on the customer application installed on Desire and considering the Wi-Fi connection. The most costly protocol in [32] is the Redeem due to the fact that it comprises the execution of cryptographic operations, such as group signatures (during the first and third steps), as described in Sect. 5.2. Moreover, the response time is very different whether it is considered XML or ASN.1 for-

Fig. 17 Total response time for each process in [32] and [33] (HTC Desire with Wi-Fi)

ments to encode messages. In fact, for each process in both multicoupon schemes, ASN.1 contributes to improve the performance perceived by customers (up to 40% for [32] and 37% for [33]). It is due to shorter messages (Figs. 15 and 16) and its encoding efficiency.

Therefore, both multicoupon solutions provide a response time under the maximum waiting time tolerable by customers. In fact, this tolerable waiting time could be higher because it depends, among other things, on the utility perceived by customers. These results prove that the use of online validations, where mobile customers are involved, is viable because it does not imply a large response time.

7.3 Analyzing influence factors on performance

In Sect. 7.2, we studied the total response time of each process. But, if we want to improve any scheme, it is useful to identify which factors introduce a larger overhead to the processes. Let us analyze in more detail the results obtained in Sect. 7.2, splitting the total time used by each process into the three metrics defined in Sect. 3.3:

- The time spent on processing data. Here, the values depend on the processing power of each device, mainly CPU and memory.
- The time spent on encoding data to be sent and decoding received data. We measure this time considering either XML or ASN.1 encoding formats.
- The time delay due to communications channel. We evaluate the effect of the network access technology used by the customer: Wi-Fi versus 3G.

First, let us focus on the processing and the encoding/decoding times for each process considering the customer device. Figures 18, 19, 20 and 21 show the time spent by the customer application to execute each process. Remember that Issuing and Redeem are the processes where customers are involved.

Aside the differences due to computing power of each mobile device, we detect some interesting results. In both

Fig. 18 Issuing process: processing and encoding/decoding in function of smart-phone in [32]

Fig. 19 Issuing process: processing and encoding/decoding in function of smart-phone in [33]

Fig. 20 Redeem process: processing and encoding/decoding in function of smart-phone in [32]

Fig. 21 Redeem process: processing and encoding/decoding in function of smart-phone in [33]

multicoupon proposals, it is notable the gap between encoding/decoding depending on which message syntax is considered. In fact, for the better case (considering the most powerful mobile device), encoding with ASN.1 is about 17

Fig. 22 Networking response time for each protocol in [32] and [33]

times faster than encoding with XML for [32] (Fig. 18), and 34 times for [33] (Fig. 19). The same trend is observed in all processes. Therefore, when XML is used, the encoding/decoding activities are a main concern, and this fact is further reinforced as the computing power of those handheld devices gets worse (Figs. 18 and 19).

Another interesting result is the relation between the time on processing data and the encoding formats. Apparently, the processing time must be independent of the message encoding. However, we observe that, in [33], the processing time is reduced when ASN.1 is used (arriving up to 66% for the Redeem process). If we analyze the processes flow of Issuing and Redeem, we notice that some encoded messages are encrypted and then forwarded to the manufacturer or the coupon provider. Those encoded messages (in XML are longer than in ASN.1) are encrypted, increasing the computational burden on the scheme.

Next, we study the influence of the network access technology. To perform this test, the customer application runs on Desire and the selected message syntax is ASN.1, because it is the better option in terms of efficiency, as we have proved above.

Figure 22 shows that executing processes using the Wi-Fi network is a better option than using 3G. It is due to the fact that Wi-Fi is attached to a wired Internet connection, which is often more stable, provides a better round-trip time, and offers better throughput than 3G. In addition, network activities consume a large amount of the total time required by each process. Comparing the results depicted in Figs. 17 and 22, we can observe that network activities can arrive up to 84% of the total time required for executing the Issuing process (66% for the Redeem process) in [32]. Similarly, more than a 60% of the total time in [33] is due to network activities. Therefore, besides the encoding/decoding format, the network throughput (and of course, stability) is a big concern.

8 Conclusions

The vast majority of papers that present solutions for electronic coupons are focused on the theoretical findings about their proposals, usually overlooking the performance eval-

uation and the viability study of the solution. Although a solution can provide a suitable security level, it could be unfeasible for a real scenario, for example due to lack of libraries for the required cryptographic tools, or their use can be burdensome for current mobile devices. There are authors in the literature that provide some performance tests of their proposals, but these tests present only partial results. Often these tests are executed on powerful devices (desktop computers or laptops) considering a local scenario in which all entities rely on the same machine, leaving some factors out, such as networking, message encoding. However, in this paper we prove that these external factors can become the most relevant ones reducing the perceived efficiency.

In this way, we provide solutions to some problems related to the implementation that developers typically find when they try to map theoretical findings to the real world. One of these main issues, not always trivial, is the selection of an appropriate deployment scenario to test performance features of electronic coupons solutions. The use of local testbeds can provide a false value of efficiency of the solutions. This is because latency of networks is not taken into account at all, so response times obtained from local testbeds cannot be considered as a reference value for the response time perceived by their potential clients. Regarding message encoding, we have shown how choosing an appropriate encoding can reduce two main issues regarding the efficiency of any solution: internal processing and network overhead. In fact, these issues entail an important benefit for mobile devices, that is energy saving, especially at the transmission time. The network access technology used by customers must be also considered at the performance evaluation stage, because this technology affects both the transmission time and the energy saving. Another issue is how to distribute non-standard credentials among untrusted entities. In this paper, we describe how standardized structures such as X.509 certificates can be used to achieve this goal, allowing to manage (issue, store and distribute) those credentials in an easy way, because all infrastructure and software about X.509 certificates is well known by, and available for, developers.

Thus, we provide an evaluation framework build upon a set of considerations to take into account when an electronic multicoupon solution has to be analyzed in terms of efficiency in a scenario with real customers and communication technologies. Under this framework, we analyze the real efficiency of two particular multicoupon schemes. By means of the obtained results, we can assert that these schemes are really efficient as well as viable solutions ready to be used by electronic coupons customers. On the one hand, cryptographic libraries for managing public keys and group keys are available to be used even on mobile devices, and their distribution can be achieved by means of X.509 certificates in an easy way. On the other hand, the use of ASN.1 for message encoding reduces the response time perceived by

customers regardless of the access technology used by them; it is between 10 and 46 times faster than encoding with XML, depending on the executed process. Besides, in the analyzed solutions, network activities can arrive up to 84% of the total time required for executing a process. It is a high value that must be considered for estimating the response time perceived by clients. Moreover, we have obtained interesting results about the cost of cryptographic operations with respect to other processes. As a result, we prove that the statement that cryptographic operations of a security protocol are usually considered as the most costly procedures could be false.

Therefore, the viability of an electronic coupon solution should be validated considering all factors, not just those related to internal processing, usually measured as the cost of cryptographic operations, but also taking into account other internal (e.g., message encoding) and external (e.g., networking) factors.

Acknowledgements This work is partially financed by the European Social Fund and the Spanish Government under the Projects TIN2014-54945-R and TIN2015-70054-REDC.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Abe, M., Fujisaki, E.: How to date blind signatures. *Asiacrypt'96. Lecture Notes in Computer Science*, vol. 1666, pp. 244–251. Springer, Berlin (1996)
2. Abe, M., Okamoto, T.: Provably secure partially blind signatures. *CRYPTO 2000. Lecture Notes in Computer Science*, vol. 1880, pp. 271–286. Springer, Berlin (2000)
3. Adams, C., Farrell, S., Kaese, T., Mononen, T.: Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP). RFC 4210 (Proposed Standard) (2005). <http://www.ietf.org/rfc/rfc4210.txt>
4. Allen, S., Graupera, V., Lundrigan, L.: Pro Smartphone Cross-Platform Development: iPhone, Blackberry. Windows Mobile and Android Development and Distribution. Apress, Berkely, CA (2010)
5. Amazon Web Services: Amazon Elastic Compute Cloud: Instance Types and Compute Resources Measurement. <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html#MeasuringComputeResources>. Accessed 23 Feb 2018
6. Armknecht, F., Escalante, B.A.N., Löhr, H., Manulis, M., Sadeghi, A.R.: Secure multi-coupons for federated environments: privacy-preserving and customer-friendly. In: Proceedings of the 4th International Conference on Information Security Practice and Experience, *Lecture Notes in Computer Science*, vol. 4991, pp. 29–44. Springer, Berlin (2008)
7. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of group signatures: formal definitions, simplified requirements, and a construction based on general assumptions. *Advances in Cryptology: EUROCRYPT 2003. Lecture Notes in Computer Science*, vol. 2656, pp. 644–644. Springer, Berlin (2003)
8. Blog, A.D.: Unifying key store access in Ice Cream Sandwich. <http://android-developers.blogspot.com.es/2012/03/unifying-key-store-access-in-ics.html>. Accessed 23 Feb 2018
9. Blundo, C., Cimato, S., De Bonis, A.: Secure e-coupons. *Electron. Commer. Res.* **5**, 117–139 (2005)
10. Boneh, D., Boyen, X.: Short signatures without random oracles. *Advances in Cryptology: EUROCRYPT 2004. Lecture Notes in Computer Science*, vol. 3027, pp. 56–73. Springer, Berlin (2004)
11. Borrego-Jaraba, F., Castro, P., Gonzalo, C., Luque, I., Gomez-Nieto, M.A.: A ubiquitous nfc solution for the development of tailored marketing strategies based on discount vouchers and loyalty cards. *Sensors* **13**(5), 6334–6354 (2013)
12. Bouncy Castle Library: <http://www.bouncycastle.org/java.html>. Accessed 23 Feb 2018
13. Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E., Yergeau, F.: Extensible markup language (xml) 1.0. (5th edn). In: World Wide Web Consortium, Recommendation REC-xml-20081126 (2008)
14. BuddeComm: Global mobile communications: statistics, trends and regional insights. Report <https://www.budde.com.au/Research/Global-Mobile-Communications-Statistics-Trendsand-Regional-Insights.html>. Accessed 23 Feb 2018
15. Canard, S., Gouget, A., Hufschmitt, E.: A handy multi-coupon system. *Applied Cryptography and Network Security. Lecture Notes in Computer Science*, vol. 3989, pp. 66–81. Springer, Berlin (2006)
16. Chang, C.C., Lin, I.C., Chi, Y.L.: Secure electronic coupons. In: *Information Security (AsiaJCIS)*, 2015 10th Asia Joint Conference on, pp. 104–109 (2015)
17. Chang, C.C., Sun, C.Y.: A secure and efficient authentication scheme for e-coupon systems. *Wirel. Pers. Commun.* **77**(4), 2981–2996 (2014)
18. Chen, L., Enzmann, M., Sadeghi, A.R., Schneider, M., Steiner, M.: A privacy-protecting coupon system. *Financial Cryptography and Data Security. Lecture Notes in Computer Science*, vol. 3570, pp. 578–578. Springer, Berlin (2005)
19. Chen, L., Escalante, B.A.N., Löhr, H., Manulis, M., Sadeghi, A.R.: A privacy-protecting multi-coupon scheme with stronger protection against splitting. In: *Proceedings of the 11th International Conference on Financial Cryptography and 1st International Conference on Usable Security, Lecture Notes in Computer Science*, vol. 4886, pp. 29–44. Springer, Berlin (2007)
20. Chien, H.Y., Jan, J.K., Tseng, Y.M.: RSA-based partially blind signature with low computation. pp. 385–389 (2001)
21. Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., Polk, W.: RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Technical Reports (2008). <http://tools.ietf.org/html/rfc5280>
22. Dai, X., Grundy, J.: NetPay: an off-line, decentralized micro-payment system for thin-client applications. *Electron. Commer. Res. Appl.* **6**(1), 91–101 (2007)
23. De Caro, A., Iovino, V.: jPBC: Java pairing based cryptography. In: *Proceedings of the 16th IEEE Symposium on Computers and Communications, ISCC 2011*, pp. 850–855. IEEE, Kerkira, June 28–July 1 (2011)
24. Elastic Cloud Computing (EC2): <http://aws.amazon.com/ec2/>. Accessed 23 Feb 2018
25. Escalante, B.A.N., Löhr, H., Sadeghi, A.R.: A non-sequential unsplitable privacy-protecting multi-coupon scheme. *GI Jahrestagung* **2**, 184–188 (2007)
26. Escalante, B.A.N.: Privacy-protecting multi-coupon schemes with stronger protection against splitting. In: *Master's Thesis. Department of Computer Science, Saarland University* (2008)
27. FON Project: Your global WiFi network. <http://corp.fon.com/>. Accessed 23 Feb 2018

28. Gass, R., Diot, C.: An experimental performance comparison of 3G and Wi-Fi. In: Krishnamurthy, A., Plattner, B. (eds.) *Passive and Active Measurement. Lecture Notes in Computer Science*, vol. 6032, pp. 71–80. Springer, Berlin (2010)
29. Google App Engine (GAE). <https://cloud.google.com/products/app-engine/>. Accessed 23 Feb 2018
30. Google Drive. <https://drive.google.com>. Accessed 23 Feb 2018
31. Han, Y., Choi, Y., Hong, J.K.: Experience on the development of a comsoc application for smart phones. *IEEE Commun. Mag.* **50**(4), 106–112 (2012)
32. Hinarejos, M.F., Isern-Deyà, A.P., Ferrer-Gomila, J.L., Payeras-Capellà, M.: Mc-2d: an efficient and scalable multicoupon scheme. *Comput. J.* **58**, 758–778 (2015)
33. Hsueh, S.C., Chen, J.M.: Sharing secure m-coupons for peer-generated targeting via eWOM communications. *Electron. Commer. Res. Appl.* **9**, 283–293 (2010)
34. IEEE 802.11™-2012: Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications
35. International Telecommunication Union: ICT facts and figures. In: *The World in 2015*. <http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf>. Accessed 23 Feb 2018
36. International Telecommunication Union: Information technology "Open Systems Interconnection" The Directory: Public-key and attribute certificate frameworks. Technical Corrigendum 2. Series X: Data Networks, Open System Communications and Security Directory (2008). ITU-T Recommendation X.509
37. International Telecommunication Union: Information Technology—Abstract Syntax Notation One (ASN.1): Specification of Basic Notation. ITU-T Recommendation X.680 (2002)
38. International Telecommunication Union: Information Technology—ASN.1 Encoding Rules—Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER), and Distinguished Encoding Rules (DER). ITU-T Recommendation X.690 (2002)
39. Isern-Deyà, A.P., Hinarejos, M.F., Ferrer-Gomila, J.L., Payeras-Capellà, M.: A secure multi-coupon solution for multi-merchant scenarios. In: *IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 655–663 (2011)
40. Isern-Deyà, A.P., Huguet-Rotger, L., Payeras-Capellà, M.M., Mut-Puigserver, M.: On the practicability of using group signatures on mobile devices: implementation and performance analysis on the android platform. *Int. J. Inf. Secur.* **14**(4), 335–345 (2014)
41. Java Architecture for XML Binding (JAXB). <http://jaxb.java.net>. Accessed 23 Feb 2018
42. Juniper Research: Mobile and Online Coupons: Loyalty and Beacon Engagement 2016–2021. <https://www.juniperresearch.com>. Accessed 23 Feb 2018
43. Kaese, T., Peylo, M.: Internet X.509 public key infrastructure—HTTP Transfer for the Certificate Management Protocol (CMP). RFC 6712 (Proposed Standard) (2012). <http://www.ietf.org/rfc/rfc6712.txt>
44. Larry Page: Google's CEO: Android activations (2013, March). Google+ profile. <http://googleblog.blogspot.com.es/2013/03/update-from-ceo.html>. Accessed 23 Feb 2018
45. Lee, Y., Chen, A.N., Ilie, V.: Can online wait be managed? The effect of filler interfaces and presentation modes on perceived waiting time online. *MIS Q.* **36**(2), 365–394 (2012)
46. Li, W., Wen, Q., Su, Q., Jin, Z.: An efficient and secure mobile payment protocol for restricted connectivity scenarios in vehicular ad hoc network. *Comput. Commun.* **35**(2), 188–195 (2012)
47. Liu, W., Mu, Y., Yang, G.: An efficient privacy-preserving e-coupon system. In: *Inscrypt, Lecture Notes in Computer Science*, vol. 8957, pp. 3–15. Springer (2014)
48. Mobile Frameworks Comparison Chart Project. <http://www.markus-falk.com/mobile-frameworks-comparison-chart/>. Accessed 23 Feb 2018
49. Mundy, D., Chadwick, D.W.: An XML alternative for performance and security: ASN.1. *It Prof.* **6**(1), 30–36 (2004)
50. Nah, F.: A study on tolerable waiting time: how long are web users willing to wait? *Behav. Inf. Technol.* **23**(3), 153–163 (2004)
51. NCH Marketing Services: 2014 NCH Coupon Facts Report. <http://www.nchmarketing.com>. Accessed 23 Feb 2018
52. NFC Forum. <http://nfc-forum.org/our-work/specifications-and-application-documents/>. Accessed 23 Feb 2018
53. Nguyen, L.: Privacy-protecting coupon system revisited. *Financial Cryptography and Data Security. Lecture Notes in Computer Science*, vol. 4107, pp. 266–280. Springer, Berlin (2006)
54. Okamoto, T.: Efficient blind and partially blind signatures without random oracles. TCC 2006. *Lecture Notes in Computer Science*, vol. 3876, pp. 80–99. Springer, Berlin (2006)
55. Pei Zheng et al: *Wireless Networking Complete—Chapter 3: An Overview of Wireless Systems* (2010)
56. PhoneGap: Framework for building cross-platform mobile apps. <http://phonegap.com/>. Accessed 23 Feb 2018
57. Ruiz-Martínez, A., Canovas, O., Gomez-Skarmeta, A.F.: Design and implementation of a generic per-fee-link framework. *Internet Res.* **19**(3), 293–312 (2009)
58. Shrimali, M.: Future trends in mobile commerce: service offerings, technological advances and security challenges. *J. Environ. Sci. Comput. Sci. Eng. Technol.* **1**(2), 101–117 (2012)
59. Simple Storage Service (S3). <http://aws.amazon.com/s3/>. Accessed 23 Feb 2018
60. Simple XML Framework. <http://simple.sourceforge.net/>. Accessed 23 Feb 2018
61. Spongy Castle Library. <http://rtyley.github.io/spongycastle/>. Accessed 23 Feb 2018
62. Subashini, S., Kavitha, V.: A survey on security issues in service delivery models of cloud computing. *J. Netw. Comput. Appl.* **34**(1), 1–11 (2011)
63. Website: Techcrunch. Android Now Has 1.4 Billion 30-Day Active Users Globally (2015, September). <https://techcrunch.com/2015/09/29/android-now-has-1-4bn-30-day-active-devices-globally/>. Accessed 23 Feb 2018
64. Windows Azure. <http://www.windowsazure.com/>. Accessed 23 Feb 2018
65. Xiao, Z., Xiao, Y.: Security and privacy in cloud computing. *IEEE Commun. Surv. Tutor.* **15**(2), 843–859 (2013)
66. Xin, L., Xu, Q.L.: Practical compact multi-coupon systems. *IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS)* **3**, 211–216 (2009)
67. Yang, J.H., Chang, C.C.: A low computational-cost electronic payment scheme for mobile commerce with large-scale mobile users. *Wirel. Pers. Commun.* **63**(1), 83–99 (2012)
68. Zhang, B., Teng, J., Bai, X., Yang, Z., Xuan, D.: P3-coupon: a probabilistic system for prompt and privacy-preserving electronic coupon distribution. In: *PerCom*, pp. 93–101. IEEE (2011)
69. Zoho: <http://www.zoho.com/>. Accessed 23 Feb 2018