eIDeCert: A User-Centric Solution for Mobile Identification

Michele Schiano di Zenise[1], Andrea Vitaletti[1], Lisha Chen-Wilson[2], Lester Gilbert[2], David Argles[2]
[1]*University of Rome "Sapienza", Italy*
[2]*University of Southampton, United Kingdom*

Abstract

The necessity to certify one's identity for different purposes and the evolution of mobile technologies have led to the generation of electronic devices such as smart cards, and electronic identities designed to meet daily needs. Nevertheless, these mechanisms have a problem: they don't allow the user to set the scope of the information presented. That problem introduces interesting security and privacy challenges and requires the development of a new tool that supports user-centrity for the information being handled. This article presents eIDeCert, a tool for the management of electronic identities (eIDs) in a mobile environment with a user-centric approach. Taking advantage of existing eCert technology we will be able to solve a real problem. On the other hand, the application takes us to the boundary of what the technology can cope with: we will assess how close we are to the boundary, and we will present an idea of what the next step should be to enable us to reach the goal.

1. Introduction

Recent improvements in technology have made possible the creation of new forms of support, usually electronic, for personal data management that is more efficient and reliable than those which existed previously. Electronic identity (eID) aims to replace paper-based documents, thereby providing improved portability and durability. For example, a spilt cup of coffee could invalidate our passport. In some contexts, the possibility exists for the user to decide what information should be made public, thus supporting a user-centric approach to sensitive information management. Unfortunately, current technologies don't support user-centricity effectively. Moreover, these types of support raise a wide variety of issues relating mainly to security. In this context, the provision of a specialized eCertificate which controls access to private information is required. To gain maximum advantage from the eID concept, we may use it on a mobile device. This idea adds other security issues to existing problems and at the time of writing this article, only a few limited solutions have been proposed.

Taking advantages from the eCert project [2], an innovative project developed by Learning Societies Laboratory in Southampton's School of Electronics and Computer Science and focused on the creation of an eCertificates, we have been able to guarantee prevention of forgery, provision of privacy and interoperability in the eID context developing a new specific tool: eIDeCert. The following pages present the life-cycle that has allowed us to reach the goal. Starting from the theoretical analysis, we will show how eIDeCert has been developed. Moreover, we will underline the limits and advantages that come from the use of this approach.

An exemplifying scenario. Consider the following situation: UserX goes out clubbing and has to certify his age to enter. Unfortunately his wallet is untidy and it contains a lot of smart cards, other IDs, but not that one that the UserX needs. Disconsolate, UserX comes back home. Observe that with paper ID, users are forced to disclose all the sensitive information on the ID, not only the age.

More than 95% of the citizens in developed countries owns a mobile phone [3]. As a consequence, eCert for eID managed in mobile devices proposes itself as the tool able to be always available and to provide a huge variety of ID in order to avoid the previous scenario. According to [4], "One way to protect sensitive information is for the card not to reveal it at all but just to verify certain assertions", or in other words a major goal for eID card is to support all those security features which increase the control of the card owner over which data are disclosed about them and to whom. This is exactly the main focus of our user-centric approach that we will discuss in details in the next sections.

2. Underlying technologies

The eID exploits and leverages some key technologies, such as:

- eCertificate as policy for key management;
- ePortfolio to understand and underpin the choice about eCert;
- eID to understand what it is and the related requirements;
- Mobile agents as means to reach a big part of the population and to provide portability, mobility, scalability, data management and identification and control.

In the following we briefly discuss the role of these technologies in our proposed solution.

2.1. eCertificate

The word eCertificate is often used as synonymous with the concept of Public Key Infrastructure (PKI). A PKI provides a secure infrastructure based on authentication and proof of content. Specifically in cryptography, PKI denotes the authentication managed by CA (certification authority): each user receives a public key bound with its unique identity through the registration and issuance process provided, usually, by the certification authority. The link between unique identity and the public key identifies unambiguously for each user a certainty in:

- quality of information sent and received;
- source and destination;
- time and timing characterizing information;
- privacy;
- legal trustworthiness;

Thus PKI ensures confidentiality, integrity and availability with consideration to possible changes of hardware, software, data, policy and people.

2.2. ePortfolio

The Southampton eCert project is focused on providing user-centric control of personal data within the context of ePortfolios (EPs). The term ePortfolio identifies a digitalized collection of artifacts that represent an institution, group or an individual. The EP possibilities go beyond the previous definition: it can be used as an administrative tool (to manage and organize work), to monitor access to private information and it can be used as a means for exchanging ideas and feedback. The structure of an ePortfolio (collection of files) isn’t directly linked to the eID structure, which is a collection of text-line information, but its idea is quite close to that of eID: both EP and eID have to find an electronic and secure way to certify, set and show the users' private information. This feature underpins the eCert protocol which is designed to manage ePortfolios, providing security in this field of e-Learning.

2.3. eID

Just as the ePortfolio is proposed as a substitute of the paper-based portfolio, the eID guarantees the same functionalities as an ID card. Usually an eID is a plastic smart-card (like a bank card) that provides personal user identification through a microchip containing information. The electronic side of this support consists of authentication and requires a document to be signed with a digital signature. This paper presents a secure protocol for eID management which is completely electronic: it allows the user to set a scope for his information, provide security through the migration of the eCert protocol, and mobility through the use of mobile devices.

2.4. Mobile agents

We envision a mobile solution based on mobile agents (MAs) implemented on mobile phones. A MA is a software agent which can transport its state between different environments without loss of features. More specifically a mobile agent is autonomous, self-taught (with respect to its environment) and mobile.

Security problems for MA include the protection of the Agent Execution Environment (AEE) from malicious agents, of agents from a malicious AEE, of one agent from another, of an AEE from another AEE, of the communication to and from the AEE, and of the host from the AEE. In order to solve these problems we could use the principles shown in [1]:

- *for the most natural applications of MA, the participants cannot be assumed to trust one another,*
- *any agent-critical decisions should be made on neutral (trusted) hosts,*
- *unchanging components of the state should be sealed cryptographically.*

The eCert central system provides the same features. Thus the migrated eCert technology is sufficient to guarantee these principles in practice.

2.5. Existing systems

There are many examples of eID applications. They include the following:

Identification eID: is a government-issued document for online and offline identification. Usually this type of document allows digital signing and uses a chip which contains the same information legible on the card plus information for the identification like signature key and certificates. Examples are the biometrical passport [5] and the italian CIE [6] (electronic identity card): the first one is a combination of paper and electronic ID that provides the same features described above.

Access badge (private eID): is used for entry to reserved areas managed by automated access control. It can be equipped with various technologies, but usually uses barcodes or magnetic stripes to carry an identification number. An example of this type is the IDcard [7] for the students of Southampton University. This card provides authentication for all the universitary stuff, including the public transport.

Financial eID: can be defined as a "middle way" between the previous two. The mechanism of authentication is close to an access badge (authentication through an automated control with magnetic stripe) but the new eIDs belonging to this group has a chip that contains all the private information of the user. Moreover the process of authentication is made stronger by the use of a personal secure number (PIN). Bank cards [8] are typical examples of this category.

Each type of eID analyzed provides security, authentication, and has a track record in the security field. Moreover they use the same cryptography model that the developers would use for the project, so we have to think: what is the innovation inside the project? Why could it become a useful tool for the day life? The answers are: mobility and customizability. None of the previous eID examples allow the user to set a scope for his data and no one has an implementation usable by mobile. The project shown in this paper aims to reach both this goals and, in this sense, it's a very innovative project.

3. eCert protocol

The eCert project is a UK government-sponsored project [9]. At the heart of this project is an eCertification protocol which is being developed to address security issues which originally arose as a concern within the field of ePortfolios. The project aims to implement an electronic version of a Qualification Certificates System, which will overcome the authorization problems that we are facing in ePortfolios. This eCertificate system will be at least as valid as the paper-based certificates, and will be usable either as a standalone application or served within other applications, such as ePortfolios. eCert aims to be easy to use and suit all levels of students while including high security methods to prevent forgery. The system is designed with user centric approach, such that the students will have control over the usage of such eCertificates lifelong. With the provided verification service, the whole system is secured, not just the eCertificate.

The eCert policy is alredy integrated and operative the UK ePortfolio system called “eFolio”, and an Australian system, the Mahara. Both systems can now be fully utilized and it is possible to prove the usage of the system successfully as it can not only be used standalone but can also be plugged into other applications. The eCert system’s accessibility and scalability have also been improved after taking a considerable number of observations and recommendations from this subproject.

The concept of the eCert protocol needs to be evaluated to test its usage in a wider domain. For this reason, it seems a good idea to leverage the UK government user-centric structure of eCert for ePortofolios to achieve the creation of a new protocol with similar features but focused on eID in the mobile environment.

3.1. Overview

eCert divides the system stakeholders into three categories, together with the eCert server, form the four actors of the system; their relationships are defined and shown in figure 1:

- the issuer is the entity that manages student records, provides and issues the eCert, after the user identification;
- the owner is the entity that, if identification succeeds, receives eCert. Then he is able to set access control to his eCerts, and finally he can distribute them;
- the reviewer is the entity that views the information provided by the owner and he checks if they are valid.
- the eCert server: is the entity that provides services for issuer to issue, owner to manage, and reviewer to verify eCerts, it provides security control and support functions for the services.

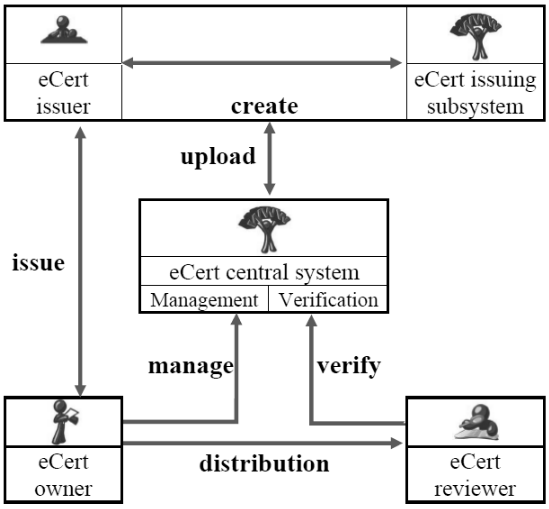


Figure 1. eCert actors and relations

The actors interact each other through the processes shown in figure. The eCert issuer creates an eCert through the issuing subsystem, and issues it to the owner. The owner can set access control to the eCert before distribute it out to the reviewer. Finally the reviewer communicates with the eCert server through the verify relation which is the act of checking the validity of information. The issuer updates the verification related information at anytime during the process. It contains a request requesting the checking process and acknowledgement (confirming the document’s validity and showing the eCert).

eCert aims to secure the whole e-certificate system, not just the eCertificate itself. We now show how to reach this goal.

3.2. Structure and functionalities

The eCert structure provides central services, but requires user-orientated storage. It is composed of an issuing subsystem and an online central system. The issuing subsystem is for installed within registered education institutions only; and the online central system is in turn composed of an management subsystem and an verification subsystem. The management subsystem is for eCert owners to set access control values to their eCerts, and the verification subsystem is for reviewers to verify the received eCerts. This solution is able to guarantee a unique standard for all issuing institution; issuer side: signs the documents with his private key, encrypts the whole document with system private key, issue the documents to the owners' corresponding institution accounts, and update the verification related information to the central system (e.g. revokes a document that it was withdrawn); owner side: sets the access info's through an access token contained into the XML metadata and sends the eDocument to reviewer; reviewer side: uses the online service and the access token for verifying the genuine of the eDocument. Figure2 shows the structure.

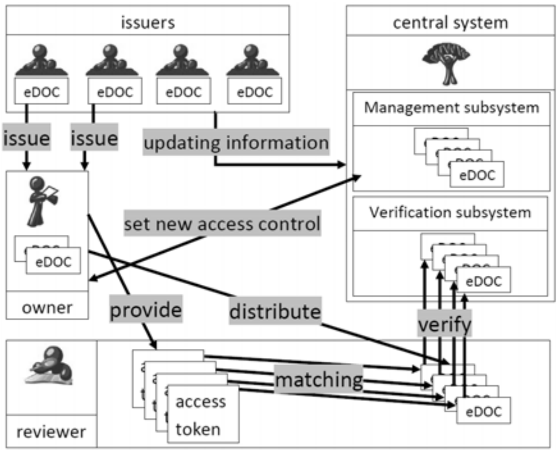


Figure 2. eCert system

The eCert policy also provides the following functionalities that aim to solve the problems related to the chosen architecture:

- each owner has one and only one system ID that acquires at the moment of first registration into the system and it has a lifetime-duration. It must be attached in each owner's information;
- the issuers have to be certified to avoid that fake issuers are present into the system. Moreover all the member that belong to an issuer have to be certified also;
- owner: system ID relation is 1:1;
- system ID: eCert relation is 1:many;
- the central system has to maintain a revocation list for all issued eCerts that may be uploaded by the institutions.

3.3. eCert system design summary

The eCert project addressed the eCertificate problem that exists within the traditional digital signing method when it is applied to non static content eDocuments, such as eCertificates; it defined the eCertificate file structure, so that it contain not only the qualification award information, but also the transcript information and any supported evidence files, which can be in any format; it has proposed a new digital signing method to cooperate with the designed structure and to meet the eDocuments' ownership right.

The new signing method not only bound the related files together, but also allow the eCertificate owners to set access control value of who can see what and for how long to the signed eDocument, while remaining the integrity of the signature, without the need of resigning by the initial issuing body; an additional encrypt key will be added after the signing to ensure that only the receiver with the corresponding decrypt key can access the file; it has also proposed a newly designed centralized verification service for such digitally signed and access controlled distributed eCertificates.

The system provides security control for verification against eCertificate expire time, access period, ownership, signing key status, qualification award status, owner controlled section display.

The whole design cooperate together to enable the issued eCertificates can be securely distributed and verified independently from the issuing body and satisfy the ownership right, without requiring storage in the verification system, these also provide huge advantages of lifetime validation and avoiding many of the database attacks.

3.4. Advantages and disadvantages

eCert provides a unique, secure and trusted system for the management of data in a web environment with a secure user-centric approach. This user-centric focus is key to this project and a further advantage would be to consider this protocol as a possible candidate for a future national level standard. On the other hand, the protocol is thought to manage ePortfolio in the web environment: a reverse engineering process to adapt the system is needed. Even if the ideas of EP and eID are quite close, their structure is different: eCert is not able to manage eID. Furthermore, the issue process is different in terms of the data transmitted and the execution environment. These issues will now be analyzed.

4. Design

The idea of an eID managed in mobile environments constitutes a powerful and innovative tool to manage personal identity in an easy and comfortable way, but it offers many security challenges as that to guarantee security in the whole the process. In order to do this it is clear that:

- reviewer has to be able to verify the identity and the validity of the document shown;
- the owner has to be able to show only the information that he wants to show (in this case date of birth) and he has to be able to manage the access to this information;
- the issuer has to be trusted to manage a huge quantity of information and to sign its document with the eCert policy.

Apart from the security and adaptation issues, we have to consider a technical problem related to the mobile environment: to find a good and quick way to pass the eID to the reviewer. Two possible solutions were considered, NFC [10] and qr code [11]. Although the two means are equally efficient, the increasing popularity of qr codes and the availability of a qr reader within mobile devices support the choice of the qr code over NFC.

Figure 3 gives the use case diagram for eID eCert:

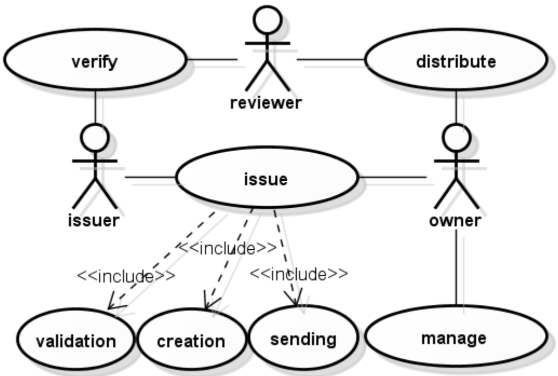


Figure 3. Use case diagram

4.1. eID vs eCert

The review of the underlying technologies identified that the idea of EP is quite close to eID in that both need to find an electronic and secure way to certify and show the users' information. Although eCert provides secure eCertificates in the web environment, studies on eCert and in the literature point out that it can't be directly applied to eID for two reasons; differences in the file structure and the existence of a personal account. eCert is a stand-alone system, but it has been created to manage EPs and this feature is evident in the structure of the managed files.

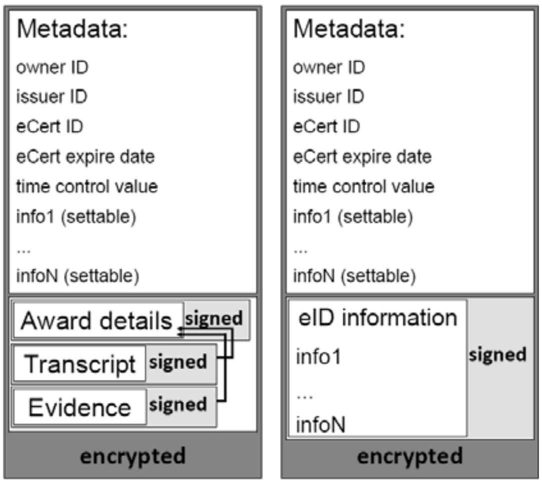


Figure 4. (a) eCert file; (b) eID-eCert file

The eCert files are a collection of files, individually signed, and encrypted together with metadata that makes it possible to set a file's scope. Instead the eID files are a collection of line-text information gathered in a single signed file and encrypted together with metadata that makes it possible to set the scope of this information. This problem is related to the code of the issue process: a recoding is sufficient to make the system able to recognize the eID file. Thus, the recoding aims to make able the system to use the structure in figure 4.(b) instead the structure in figure 4.(a). Note that with the notation "settable" we set the visibility for that element of the file: other files in the case of eCert, and text-line information in the case of eID.

The second difference is related to the nature of the information: the original protocol allows the issuing institution to have an account for each eCertificate owner in order to manage his information. In the eID, this step is unnecessary because it is reasonable to think that personal information like DOB (date of birth) is valid for a person's life-time. This issue mans we need to find a way to supply the lack of an account and thus, secure the eID-eCert on mobile devices. We therefore need a new encryption method to straighten the issue process between issuer and owner without using an intermediary account. In this sense, AES-128 [12] seems the best possible choice.

4.2. Using eCert's solutions in eID

In order to guarantee security, eCert employs additional features beyond the basic architecture with a central service and no central storage. Some of these must be maintained for the eID system.

About the assertion techniques:

- an XML signature can be used to encapsulate the documents in order to have an issuer's secure signature;

- XML metadata: the ownership information of an eCert can be stored in XML metadata and meanwhile the employee can create an eCert through the XML signature method;
- revocation lists: a check of information status is required. In fact if the key has been compromised or the information has been withdrawn, the system has to deny the access;
- auto request: the Certificate validation processes has to start automatically before the verification results are ready;
- timestamp: can also be added the XML Signatures to improve the security.

Concerning the privacy techniques, we may use the eCert signature method [2].

Regarding the lifetime validation techniques, the eCert central system provides a verification service for eCertificates issued throughout the UK. It would be ideal for solving the lifetime validation issue. Adopting this solution, we need to take care to consider the mass of stored data: in fact it implies that all the issued information should be stored in the central system.

5. Development

In the following, we are using the name eID to indicate the data to protect and eID-eCert for the protected data (eID file). Note that the issuer and the central system have the same cryptographic method, so all encrypted data sent by the owner is accessible by the central system.

5.1. Issue process

The issue process for eID exploits the same validation and creation processes as eCert for eCertificate. After the user (owner) request, the issuer gathers the owner’s information, sets an initial (default) access control and applies his signature (with his private key). This component is the eID. Afterwards, the issuer uses the system AES-128 key to encrypt the eID and obtain eID-eCert. The use of a single system key is possible considering the features of AES and the security of the system (we will develop it on a server): indeed, if the system is compromised, to steal a key or more than one presents the same level of complexity, because the operation of scan a list of keys is not really expensive in terms of computational complexity. Finally, the issuer updates the central system revocation list.

5.2. Manage process

The owner logs in to access the management system sending his eID-eCert. The central system looks for the validation of the signature, so, first of all it decrypts the file and then checks the signature. Moreover it is able to verify ownership, revocation and modification of eID-eCert. If the validation succeeds, the owner is able to interact with the central system in order to set a scope for the information. Again, the use of a single system key allows us to skip the original step of generation of a new key pairs (PKI) for the management. In fact, the AES key is hidden in the central system and we can use this feature to recognize if a user has the authorization to view the information or not. That will be clear in the implementation paragraph. When the owner has set a new visibility for the information, .the central system generates a qr_code that contains the eID with the new scope. This image is then sent to the owner that can distribute the eID toward the reviewer (human or not).

5.3. Distribute and verify process

The owner shows the qr code to the reviewer. The latter one uploads the qr code (through scanning) to central system that decrypts the eID-eCert and verifies revocation, modification and validation. If the checking is positive, the central system enables the reviewer to view the eID (with the user’s visibility setup), otherwise it sends an error message. We call the act of giving the information to the reviewer "sending", but actually the owner shows, not send his data: in fact the reviewer, through the use of a barcode reader, scans the barcodes and, gets back the contained information on his phone.

6. Implementation

eIDeCert has been implemented on the Android [13] platform. This platform is open source and in future its market could be a good testbed for this work. The core of the application, constituted by the eCert methods, is written in pure Java and linked to the Android interface with the use of PHP [14]. The latter one is used to build a web service that acts as central system. To store the eID information, we have used mySQL [15] and for the classes design UML [16]. The methodology used for the development of the protocol is Agile [17]. It has been very useful to understand the real user requirement and to improve it through a constant flow of feedback given by the developers of original protocol and by experts in security field. The first prototype of the application eIDeCert [18] turns out to be very powerful and applicable in a wide variety of scenarios (e.g. to prove own personal age, but also as substitute for digital access cards for reserved areas).

6.1. Issue process

At time of the writing, the application doesn’t have a real process of issuing. Mobile technologies are reaching new high levels of security, but it is reasonable to think that we can't allow the transmission of the original initial eID-eCert by means of mobile phone. If an attack succeeds in this step, the whole process will be compromised. So, how does the issuing happen? The development skips this step, because in order to prove the concept, we can assume without lack of generality that the owner has his file ready for the management step. The idea for the future is to use a mechanism equivalent to internet banking:

- the owner needs to fill in a request form giving personal information and unmistakably certifying his identity;
- the entity that manages the central DB, verifies the accuracy of the information and creates the eID-eCert;
- the eID-eCert is sent by mail to the owner;
- the owner receives the file together with other security information (PIN or other) that could be used to obtain a new eID-eCert in case of a problem with the first one (robbery, damage, etc).

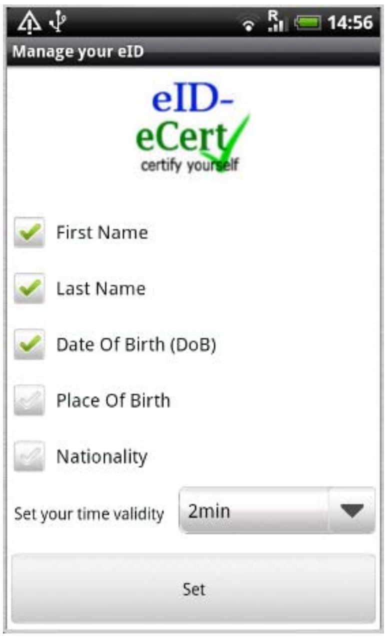
Note that we have also developed the code to generate the file (for the tests), so, in the future, it will be very easy to develop the issuing process.

6.2. Manage process

Selecting from the main menu the button “manage your eID” (figure 5), we can perform the actions described previously. In detail, this step is composed of three sub-actions:

- send eID-eCert: the user inserts the path and the name of the encrypted (and signed) eID and sends it to the server for validation;
- set the visibility for the information: if the validation succeeds, the server enables the interface to show a new layout (figure 5) where the user can select the information to visualize, otherwise an error message is given. The layout is intuitive through the use of check boxes. Moreover there is the possibility to set a timer for the validity (visualization time for the reviewer) for the chosen information;
- qr code generation: after sending the scope and the time validity values to the central system, the server generates and sends back a qr code that contains a link to the information on the server. The qr code is shown in a new layout.

Note, the final step in this process is significant in that it goes against the underlying philosophy of the eCert protocol that underpins eIDeCert, but it is necessary for now in order to prove the principle, due to the limitations of qr codes.



Manage your eID

☑ First Name

☑ Last Name

☑ Date Of Birth (DoB)

☐ Place Of Birth

☐ Nationality

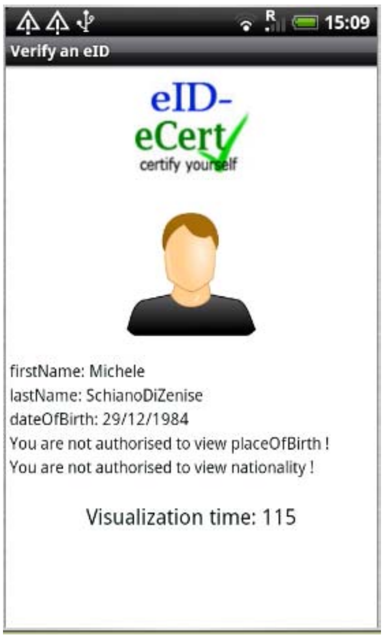
Set your time validity: 2min

Figure 5. eIDeCert: Manage your eID

Each part of this step is designed to be easy for the user. We have considered the possibility that the user may have different level of skills. Thus, the interaction needs to be simple, intuitive and clear. The interface guarantees these features. Concerning the qr code management, we have provided the possibility to save the image on the phone: this feature allows the user to show the eID even without internet connection and several time later.

6.3. Distribute and verify process

To distribute the eID, the owner shows the relative qr code image to the reviewer. The scanning happens by means of ZXing qr scanner [19] developed by Google. We have used this project in our one because it allows us to generate and scan the qr codes with the same precision and the same code. After the scanning the application acquires the link inside the qr code and requests the validation for the information related to it. If the validation succeeds (the process is the same described in the management process), the central system sends back an xml file with the accessible information or an error message. This XML is used by the interface to generate a layout for the visualization like that shown in figure 6. The data to visualize and the counter are based on the scope set in the figure 5. The photograph doesn’t belong to the scope because it is obligatory: that intuition allows us to avoid security issues related to the use of the phone. In other words, the photo avoids misunderstandings related to ownership of the eID.



firstName: Michele
lastName: SchianoDiZenise
dateOfBirth: 29/12/1984
You are not authorised to view placeOfBirth !
You are not authorised to view nationality !

Visualization time: 115

Figure 6. eIDeCert: Verify an eID

The counter represents the length in time that the reviewer has for visualizing the information. At the end of it, the data are deleted forever. Finally we have set another mechanism to prevent identity theft: when a layout of verification is built, the xml is deleted. Thus, the reviewer is not able to use the phone buttons home, back and menu to retrieve and save the xml. The interface is very easy and user-centricity persists: the information and the counter were set by the owner and the reviewer can't modify them.

6.4. Trade off with the technologies

Qr codes are easy to use and and fast in operation. This is a big advantages for the eID-eCert user scenario, but it is also a problem. This technology allows us to store inside an image just a limited number of characters or bytes as shown in table 1:

Table 1. Qr code's bound

Numeric	7,089 characters
Alphanumeric	4,296 characters
Binary	2,953 bytes
Kanji	1,817 characters

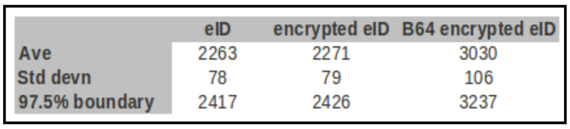
Unfortunately, after a series of attempts we have realized that the information contained in an eID-eCert are too much to respect the limits of the qr codes. In the next years, with the evolution of these codes, we could solve the problem easily. In order to prove the concept, the qr code generated by eIDeCert contains a link to a file stored on the server temporarily: that solution is against the eCert policy (no central system storage), but as said before we are counting to solve that conflict as soon as the qr technologies allow us.

7. Experimental Results

The Mobile eID system has been implemented on the Android platform using a couple of HTC Desire (hones and Android 2.2, although any smartphone running Android 2.2 should do). The system is easy to use and gives the participants confidence in the reliability of the data they can show and see.

It has been necessary to compromise the design as originally envisaged in order to ensure that system will work reliably in practice. However, current technology is not far from being able to deliver what is required in full.

A set of twenty representative eIDs have been produced in each of three versions; a raw eID, the same eID but encrypted, and the eID encrypted and encoded in B64 format ready for use in a qr code. Table 2 gives the results.



	eID	encrypted eID	B64 encrypted eID
Ave	2263	2271	3030
Std devn	78	79	106
97.5% boundary	2417	2426	3237

Table 2. eID code sizes

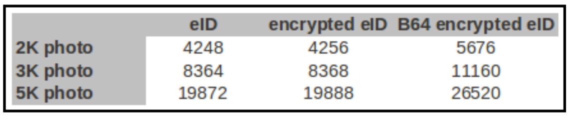
Using a standard passport-style format, but minus the picture, we can expect 97.5% of all likely required entries to produce a raw eID of 2417 bytes or less. Interestingly, the encrypted version of these eIDs is virtually identical, except that they have to round up to the nearest multiple of 8. In other words, encrypting the eID incurs no additional space demands, providing a full character set is available.

However, if we wish to use text characters only, rather than a full character set, we will need to encode using something like B64 encoding, and this increases the required 97.5% upper limit to 3237 bytes.

This means that in theory, we should have space to spare. However, the mobile phone technology available to us becomes unreliable over about 1.5 to 2K for alphanumeric codes, so in practice, we can't use native eIDs - although we are clearly close to being able to do so. Although we have had to compromise the design to make the eID project work in practice, next generation of phones will probably get us there.

Including an ID picture in the eID makes a significant difference. International passport regulations tend to specify a size for the user's photo, and although they allow for digital photos to be used, they don't specify a minimum quality in terms of pixels or colour-depth, for example. The file size depends heavily on the file format, but a little investigation indicates that something like the jpeg format yields good quality with small file sizes – perhaps this is not surprising in view of its origin. Table 3 gives the results of an investigation of eID file sizes if a photo is included.

Table 3. eID code sizes by photo size



	eID	encrypted eID	B64 encrypted eID
2K photo	4248	4256	5676
3K photo	8364	8368	11160
5K photo	19872	19888	26520

It is of interest to note that the relationship between photo size and final size of the eID would appear to follow a square law. For our purposes, this is not of major importance; what does matter is the quality of the photo. Without published criteria, it is necessary to assess this by eye; but there appears to be a marked difference between the 2K and the 3K versions. The 2K version is very small, whilst the 3K gives a usable photo in terms of recognition.

From the table, we can see that qr code technology is still quite a way off being able to cope with a full eID with photo. It will be interesting to see what the new generation of NFC-enabled smart phones will allow - this could be a perfect answer for the immediate future.

In the meantime, the mobile eID project demonstrates just how usable and friendly the concept is, and how it can give confidence to both user and reviewer in eCert terms.

We may give a specification for our requirements for eID. Firstly, any technology we use needs to offer a maximum capacity of 2426 bytes if the data can be transferred in encrypted form, but 3237 if it needs to be transferred in B64 format. The standards say that qr codes can cope with up to (about 4K), but in practice, the current qr technology on phones seems to become unreliable in the region of 1.5 to 2K, so we're close to, but over, the boundary for reliable text-only transfers of the full eID. Some codes of over 2K work reliably every time on the phone, but change a letter, or add a full stop, and suddenly it will never work – it becomes very fragile. With a very small photo (2K) included, even the B64 encrypted version is quite close to the theoretical limit of the qr code. The quite reasonable quality medium-sized photo (3K) is not so very large when coded up, but now is significantly over even the current theoretical limit. Yet we are tantalizingly close to being able to implement a user-side storage version of the eID on current technology. Furthermore, this year we may expect to see NFC phones appearing, which would be an alternative way to solve the size problem.

8. Conclusions and future work

Initial results indicate a real possibility for using eCert to manage eIDs in the mobile environment supporting user-centric management of sensitive information.

We can distinguish between two types of possible improvement: the first one is related to the improvement of the code and the second one related to improvement of the protocol methodology. In relation to the methodological issues, it will be advisable to find a way to reduce the use of the central system (by reducing message exchanges and access into the DB) and, most importantly, to make an identification picture for each eID available, in order to improve security and privacy. Concerning this category, a revocation list should be used that allows the system to recognize when an eID is not still valid. Finally we are also studying the possibility to reduce the use of the keys in the system using a single AES key instead of a multiple approach.

We plan to evaluate the usability of our system by using real testbeds. A possible testbed is related to access to alcoholic drinks: we should be able to certify age so that the salesman can be sure of that. Another way is to release a beta version on the Android market and wait for feedback in order to improve the protocol. Finally we plan to submit the protocol to a group of security experts following the methodology of think aloud [20] and to make improvements based on any suggestions offered.

Finally, in the early months of 2011, eIDeCert has been presented to the World Congress on Internet Security and to JISC 2011. The application has got genuine interest from the audiences and in general, the idea to have a flexible, always available and user-centric eID has made the community enthusiastic and very favorable.

9. References

[1] Schiano di Zenise, M., Vitaletti, A. and Argles, D. “A User-Centric Approach to eCertificate for Electronic Identities (eIDs) Management in Mobile Environment”, *The World Congress on Internet Security (WorldCIS-2011)*, 21-23 February, 2011, London, UK, pp.212-217

[2] Chen-Wilson, L., Gravell, A. and Argles, D., “Giving You back Control of Your Data: Digital Signing Practical Issues and the eCert Solution”. *The World Congress on Internet Security (WorldCIS-2011)*, 21-23 February, 2011, London, UK, pp.107-113

[3]http://www.itu.int/ITU-D/ict/statistics/ict/index.html, (Access date 22 mar 2011).

[4] Naumann, Giles Hogben, “Privacy features of European eID card specifications”, *Network Security, Volume 2008*, Issue 8, August 2008, Pages 9-13, ISSN 1353-4858, DOI: 10.1016/S1353-4858(08)70097-7

[5]http://www.direct.gov.uk/en/TravelAndTransport/Passports/Applicationinformation/DG_174159/, (Access date 22 mar 2011).

[6]http://www.servizidemografici.interno.it/sitoCNSD/pagina.do?metodo=homePage&servizio=navigazione, (Access date 22 mar 2011).

[7]http://www.soton.ac.uk/sais/idstudio/idstudio.html, (Access date 22 mar 2011).

[8]http://www.unicreditbanca.it/it/privati/conti/genius/one/?idc=14626, (Access date 22 mar 2011).

[9]http://www.jisc.ac.uk/whatwedo/programmes/aim/ecert.aspx, (Access date 22 mar 2011).

[10]http://www.nfc-forum.org/specs/spec_list/#refapps, accessed 22mar2011.
[11]http://www.denso-wave.com/qrcode/index-e.html, accessed 22mar2011.

[12] NIST, Announcing the Advanced Encryption Standard (AES), *Federal Information Processing Standards Publication 197*, 2001.

[13]http://developer.android.com/index.html, accessed 22mar2011.

[14] http://php.net/docs.php, (Access date 22 mar 2011).

[15] http://www.mysql.com/, (Access date 22 mar 2011).

[16]http://en.wikipedia.org/wiki/Unified_Modeling_Language, accessed 22mar2011.

[17] http://agilemanifesto.org/, (Access date 22 mar 2011).

[18] http://sourceforge.net/projects/cert/, (Access date 22 mar 2011).

[19] http://code.google.com/p/zxing/, (Access date 22 mar 2011).

[20] Maarten W. van Someren, Yvonne F. Barnard, Jacobijn A.C. Sandberg, "*The think aloud method - A practical guide to modelling cognitive processes*", Academic Press, London, 1994.