Available online at www.sciencedirect.com
ScienceDirect
Computers & Education 48 (2007) 1–16
COMPUTERS & EDUCATION
www.elsevier.com/locate/compedu

A PKI approach for deploying modern secure distributed e-learning and m-learning environments

Georgios Kambourakis [a,*], Denise-Penelope N. Kontoni [b], Angelos Rouskas [a], Stefanos Gritzalis [a]

[a] *Department of Information and Communication Systems Engineering, University of the Aegean, 83200 Samos, Greece*
[b] *Department of Civil Engineering, Technological Educational Institute of Patras, 1 M. Alexandrou St., Koukouli, Patras GR-26334, Greece*

Received 11 June 2004; accepted 13 October 2004

Abstract

While public key cryptography is continuously evolving and its installed base is growing significantly, recent research works examine its potential use in e-learning or m-learning environments. Public key infrastructure (PKI) and attribute certificates (ACs) can provide the appropriate framework to effectively support authentication and authorization services, offering mutual *trust* to both learners and service providers. Considering PKI requirements for online distance learning networks, this paper discusses the potential application of ACs in a proposed trust model. Typical e-learning trust interactions between e-learners and providers are presented, demonstrating that robust security mechanisms and effective trust control can be obtained and implemented. The application of ACs to support m-learning is also presented and evaluated through an experimental test-bed setup, using the general packet radio service network. The results showed that AC issuing is attainable in service times while simultaneously can deliver flexible and scalable solutions to both learners and e-learning providers.

Keywords: Architectures for educational technology system; Distance education and telelearning; E-learning; M-learning; Trust; Security; Public key infrastructure

* Corresponding author.
E-mail addresses: gkamb@aegean.gr (G. Kambourakis), kontoni@teipat.gr (Denise-Penelope N. Kontoni), arouskas@aegean.gr (A. Rouskas), sgritz@aegean.gr (S. Gritzalis).

0360-1315/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compedu.2004.10.017

1. Introduction

In recent years, distance learning systems (DLS) have become one of the most significant and promising platforms to fulfill the vision for wide-range, life-long training to a wide variety of audiences. In a distance learning scenario, learners are not required to attend classes on a regular basis. Nearly all contacts between them and the teaching organization are carried out by conventional or modern telecommunication infrastructure, even if some tutoring activities may take place in a face-to-face condition.

The term "m-learning" has lately emerged and is associated with the use of mobile technology in education. In this paper, this term is considered as *"The point at which mobile computing and e-learning intersect to produce an anytime, anywhere learning experience"* as quoted in (Harris, 2001).

Trust is an important factor in either traditional face-to-face education or in distance learning procedures and especially in interactive and distributed e-learning. Mutual trust between the learner and the e-learning provider is vital and has to be correctly established to provide the appropriate level of confidence and assurance to both sides. For instance, the learner needs to trust the provider and his procedures, restricting access only to that sensitive personal information authorized by the user. On the other hand, the e-learning provider has to deploy and support reliable authentication, authorization and accounting (AAA) mechanisms, which certify that the user accessing the provider's network is someone authorized for the particular service. The trust levels also substantially affect user's motivation or aspiration for learning. Students and teachers, who take part in an impersonal distributed e-learning or m-learning environment, have to enjoy respect, autonomy and reliance to become a trouble-free working party for their learning and teaching activities (Clark & Mayer, 2002; Horton, 2000).

Trust is a central research topic in information security research and it gains increasing attention over the years. At the same time, e-learning services are spreading fast and are gradually enjoying universal acceptance. Nevertheless, very few papers attempt to blend trust issues with e-learning or m-learning applications. The rapid increase of the number of users taking part in e-learning services, results in a many-to-many trust model. As a result, existing security schemes in current e-learning systems e.g. symmetric key techniques like passwords and pre-shared secrets/keys, are inadequate and there is an urgent demand to provide more flexible, configurable and scalable security mechanisms that can self-adjust as fast as e-learning or m-learning systems evolve.

A public key infrastructure (PKI) (Adams & Lloyd, 1999; Nash, Duane, Joseph, & Brink, 2001; PKIX WG, 2004) is an all-encompassing security infrastructure whose services are implemented and delivered using public-key concepts and techniques. Attribute certificates (ACs) (Farrell & Housley, 2002; Oppliger, 2002; Oppliger, Pernul, & Strauss, 2000), have been suggested by the Internet engineering task force (IETF) PKI Working Group as an alternative to and better than X.509 public key certificates (PKCs) (ITU-T, 1997), for carrying authorization information. Attribute authorities (AA) bind the characteristics of an entity (called attributes) to that entity by digitally signing the appropriate AC. Attributes can specify group membership, role, security clearance, or other authorization information associated with the AC holder. Therefore, ACs can be used for controlling access to system resources and employing role-based authorization and access controls policies accordingly (Oppliger et al., 2000). ACs are theoretically similar to privilege access certificates (PACs), as used in SESAME (Vandenwauver, Govaerts, & Vandew-

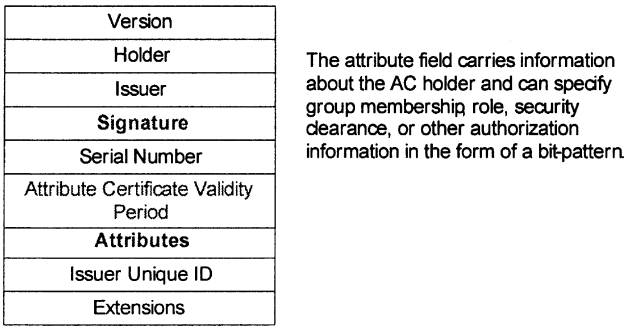

Fig. 1. Basic structure of an attribute certificate.

alle, 1997) and the Windows 2000 operating system. Two well known general purpose authorization systems that use ACs are PERMIS (Chadwick, 2002) and Akenti (Otenko & Chadwick, 2003; Thompson et al., 1999). AC based authorization is also an extension to the IETF transport layer security protocol (TLS). In the literature (Arseanult & Turner, 2002) CA entities are PKI elements, while AA entities are privilege management infrastructure (PMI) elements. However, for simplicity reasons, in this paper we use the term PKI for both CA and AA domains.

In the context of e-learning, ACs can effectively implement and support popular authorization mechanisms such as role-based access control (RBAC) (Ferraiolo, Cugini, & Kuhn, 1995). RBAC focus on the users and the jobs users perform. According to that, a collection of application specific operations (procedures) is called a *role*. Subjects, e.g. users or processes, derive their access rights from the role(s) they are performing. The basic structure of an AC is shown in Fig. 1.

Various research groups are exploring security issues in e-learning and m-learning environments. However, very few works propose full-fledged security frameworks. *SDLearn* (Furnell et al., 1998, 1999) and *iLearning* (Graf, 2001; Graf, Busch, & Wolthusen, 1999), developed within respective R&D projects did succeed to identify and meet certain security requirements, like user authentication, confidentiality and privacy, but partially left unaddressed the issue of generality, required in these systems. It is worth noting that both aforementioned architectures, try to deal with some security issues by applying public key methods. For instance, SDLearn framework uses a hybrid model of symmetric/public key encryption to provide confidentiality, while iLearning supports non-repudiation using digital signatures.

Moreover, public key services in both approaches are not incorporated in the core system architecture but rather are employed in a self-contained manner to deal with specific security issues. Unfortunately, this approach influences adversely the scalability and flexibility of the whole systems. Other works (Chadwick, Tassabehji, & Young, 2000; Diatchenko, Miloslavskaya, & Tolstoy, 2002) face the problem rather statically trying to apply mostly local or case-oriented solutions. For example, in (Chadwick et al., 2000) a managed PKI was used in order to secure electronic preparation of examination papers.

In this paper, we propose a practically secure general architecture – trust model, which is able to support e-learning and m-learning activities. Our scheme deploys public key technology and RBAC logic to secure communications and other procedures between all active components while at the same time offers maximum scalability, flexibility and reduced administration costs,

especially as the number of participants increases. Further on, the paper evaluates the feasibility of the discussed solutions in terms of service time response in the emerging mobile educational systems, using a test-bed with low-end mobile devices and a limited bandwidth general packet radio service (GPRS) network.

Although our work uses the proposed architecture mainly to provide authentication, authorization, non-repudiation of origin services and message confidentiality and integrity, other important key elements can be supported as well. For instance, tamperproof evaluation of tests, protection of courseware material, secure delivery of test material, etc., can be effectively maintained using the same infrastructure and model.

The rest of the paper is organized as follows: Section 2 depicts and discusses our proposed trust model. We put particular emphasis on how PKI is incorporated into the system as this is essential to support ACs technology. Section 3 evaluates our model providing some basic interaction scenarios between learners and providers. Section 4 concisely describes and measures the performance of a scenario, in terms of service time, when ACs are employed in the mobile environment. Finally, our conclusions are presented in Section 5.

2. Trust model specification and architecture

As our model is based on public key cryptography, each entity (user, network element or automaton) has a unified asymmetric key pair: a public key and a private key (Oppliger, 2002). In this key pair, the keys are sufficiently different, so that the knowledge about one key does not allow derivation or computation of the other. This means that the public key may be made publicly available, provided that the other key remains private. The certification authority (CA) (Adams & Lloyd, 1999; Nash et al., 2001) is the trusted authority, responsible for creating and providing the corresponding PKC, which binds the specific public key with the identity of the entity. The private key is used to form digital signatures and is a key known only to the entity.

There are two basic alternatives for generating a client key pair. Key generation may be *distributed*, that is built into the client product. In this case, the client generates the key pair, stores the private key locally in a secure manner, formats a certificate request to the CA and finally receives a PKC back from the CA. Alternatively, in *centralized key generation*, a system at the enterprise administration center generates the key pair, executes a transaction with the CA to have a certificate issued, and then delivers the key pair and the corresponding PKC to the client for importation into its local store. In this paper, we presume that key generation is distributed. Keys and certificates can be stored or transferred in hard disks, smart cards, diskettes, etc. For instance, eToken (www.ealaddin.com/etoken) enables the users of RSA's PKI systems to generate and store private keys and digital certificates inside the token when using a standard Web Browser, hence creating a secure environment and allowing full portability and maximum ease of use.

The number of keys needed for authentication and encryption in a symmetric key system, with n network entities communicating with each other, is $O(n^2)$. On the other hand, in a public cryptosystem, the corresponding need for keys is $O(n)$. Therefore, when n increases, the costs of key generation and distribution associated with the introduction of a new network entity, e.g. a new e-applications server, are quite different. In the symmetric model, we need to establish n new secret keys, while in the asymmetric case we only need 1 new key pair (private, public) for any new

network entity. Nevertheless, sometimes it is necessary to maintain dual key pairs for every user. For example, in cases when it is desirable to backup and recover the encryption key pair, whereas the signing key pair should not leave the user's possession (VeriSign, 2002). In any case, it is obvious that the public cryptosystem solution is far more scalable and effective, especially as the number of participants is increasing rapidly. In the rest of this paper we presume the use of single key pair PKJs.

Moreover, the support of flexible learning procedures in e-learning requires solutions that are not only suitable for such environments, but also interoperable with other platforms. This is necessary in order to trim down problems like redundancy, out-of-sync content versions and mismatch of security profiles when learners move from one platform to another. The successful use of public key cryptosystems in wired and lately in wireless environments has proved its usability, scalability and effectiveness, making it an ideal solution within a highly adaptive security framework.

2.1. Model components and interactions

There are four logical domains or subsystems in our model depicted in Fig. 2. Service, PKI and other agent's subsystems may belong to one corporate domain, or may be distributed in three or more interworking sub-networks. The four domains are:

1. The client side (*user agent*), which requests services bound by the appropriate ACs (credentials) that he holds. Authentication is handled by the public key certificate that he possesses.
2. The server side (*service agent*), which provides services requested by the client. These services, excluding authentication, can include multimedia content, file management, web content, discussion groups, course registration etc. A service can be obtained or authorized if the client possesses the valid AC.

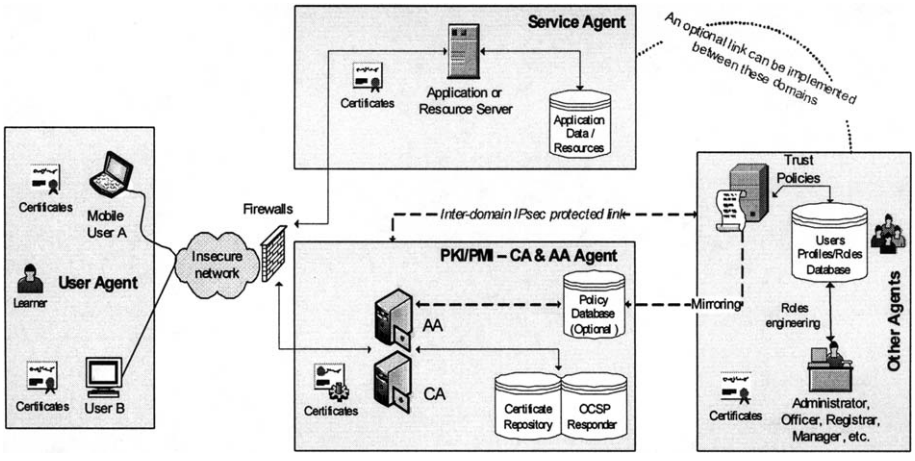

Fig. 2. Trust model for e-learning/m-learning architecture.

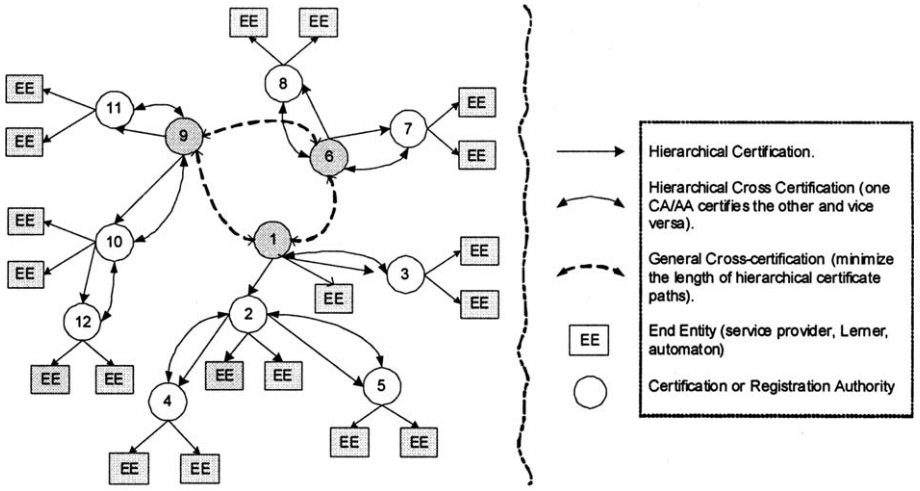

Fig. 3. Hybrid trust model.

3. The PKI/PMI subsystem (*CA & AA agent*) that issues and signs public key certificates and attribute certificates. In some implementations CA and AA functionality can be combined. For instance, a CA and an AA can be hosted by the same machine, operating simultaneously. However, ideally these entities should be completely separated, since the authority that issues PKCs is usually very different from the authority that issues ACs (PMI) (Oppliger, 2002). PKI can be deployed and maintained by the e-learning service provider or can be operated by a trusted third party (TTP), also known as certification service provider (CSP). For example, two or more collaborating providers can trust a common TTP. In general, as the number of TTPs and CAs increases, we can organize a hierarchical or hybrid trust model capable of controlling and managing an unlimited number of distributed collaborative providers and its inter-domain relationships (Fig. 3). For example, in case of two e-learning providers and it's corresponding certification authorities, CAa & CAb, CAa issues $\text{Cert}(\text{CAa})_{\text{CAa}}$[1] (the root certificate) and $\text{Cert}(\text{CAb})_{\text{CAa}}$ (the cross-reference certificate). Respectively, CAb issues $\text{Cert}(\text{CAb})_{\text{CAb}}$ and $\text{Cert}(\text{CAa})_{\text{CAb}}$. AA maintains a certificate database (DB) for accounting and supporting non-repudiation of origin services. Non-repudiation services provide unforgeable evidence that a specific action occurred. Although repudiation is a jurisprudential term and not a piece of technology, in cases of dispute, public key technology can provide all useful evidence. As a result, remote students are able to digitally sign their work, using their private key, and thus prove that they are the owners; the training institution can issue digitally signed receipts for the work submitted, and signed message authentication codes (MAC) to certify message content. On the other hand, symmetric encryption cannot guarantee non-repudiation. In this case, since both the originator and recipient share the symmetric encryption key, either party can generate the proof.
4. *Other agents*, which are responsible to define the *trust policy* and administer the system. More specifically, they are responsible for:

[1] $\text{Cert}(\text{X})_{\text{Y}}$ = Public key certificate of X with format X.509v3 (or subset) issued and signed by Y.

Table 1
Example of role assignment represented by bit patterns

Role	Admin	Instructor	**Ph.D.**	M.Sc.	Student	Registrar	Other1	**Other2**
Role ID	0000000	0000001	0000011	0000101	0000111	0001000	0001001	0001011
Learner1	0	0	**1**	0	0	0	0	**1**

Learner1 is active to both Ph.D. and Other2 roles.

Table 2
Example of a role–permissions table

Description	**Role ID**	Permissions									
		Per1	Per2	Per3	Per4	Per5	Per6	Per7	Per8	Per9	Per10
Admin	0000000	1	1	1	1	1	1	1	1	1	1
Instructor	0000001	0	0	1	1	1	1	1	0	0	1
Ph.D.	0000011	0	0	0	0	1	1	1	0	0	1
M.Sc.	0000101	0	0	0	0	0	1	1	0	0	1
Student	0000111	0	0	0	0	0	0	1	0	0	1
Registrar	0001000	0	0	0	0	0	0	0	1	1	1
Other1	0001001	0	0	0	0	0	0	0	0	1	1
Other2	0001011	0	0	0	0	0	0	0	0	0	1

(a) Creating roles (e.g. general manager, registrar's employee, tutor of program X, student of program Y in year Z, etc.). A role does not allow users to be directly associated with permissions, instead it specifies what operations are authorized on the applications data and the conditions of those operations. Roles are bound with each service or application on the server side. The exact types and number of roles depend on the complexity of the provider's system and its organizational structure.
(b) Assigning roles to each person or entity. Thereafter, an AC can bind the role with the person's identity. Two or more roles which correspond to different ACs (and different permissions) can be assigned to a person using this schema. Conflicts of interest may arise as a consequence of a learner gaining authorization for permissions associated with conflicting roles (Ferraiolo, Sandhu, Gavrila, Kuhn, & Chandramouli, 2001; Sandhu, Coyne, Feinstein, & Youman, 1996). For example, if one role restricts access to a specific course and another role approves it, the system must prohibit the same user from being assigned (or active in) both roles at the same time. Thus, it may be acceptable for a person to be a member of an examiner role and a student role in different courses, but unacceptable to take on both roles within the same course. For instance, if the system has eight roles and a user have been assigned two of them, the attributes field in the AC (Fig. 1) can be represented by a bit pattern as depicted in the third line of Table 1. It is implied that for each role exists a corresponding role–permissions table (Table 2).

2.2. Further architectural issues

As our model requires an AC for every sensitive transaction, and since AC could be issued *on-the-fly*, it is essential to have the AA server check against the roles/policy DB in order to make sure

that he is going to issue the correct AC. Taking this into account it is not possible to deliver all the ACs in batch. In case that this AA-to-DB communication is performed on-demand (on-line) – as shown in Fig. 2 – and having the possibility that the PKI/PMI can be a CSP, we can employ a security protocol like IP secure (IPsec) (Kent & Atkinson, 1998) to secure the link, thus constructing a virtual private network (VPN). For example, IPsec encapsulating security protocol (ESP) in tunnel mode can be applied to offer integrity, confidentiality (and protection against traffic analysis) services between these network entities. IPsec uses the Internet key exchange (IKE) protocol (Kaufman, 2004) for peer authentication. In this case, IKE can be configured to use public key based authentication with certificates. A different, but rather static option is to use pre-shared secrets.

Another approach is to incorporate a secondary roles/policy DB inside the PKI/PMI domain. This DB could be a mirror of the corresponding DB inside the other agent's domain. For example, it can be batch updated once, or several times during the day time. However, the requirement for secure communication channel between these domains still remains.

A final alternative, is to have AA issue only role ACs (role assignment ACs–RAAC). Then we could dispense those ACs altogether to the corresponding learners. Consequently, when a client needs to acquire a specific service, he provides the analogous RAAC to the service agent. Then the service agent, probably using an access control policy server, can make control decisions by querying the other agent's domain DB to see – based on the client's RAAC and the permissions assigned to that role – if the user is entitled to utilize the specific e-learning resource. This role-to-permissions or role specification query can be performed either on-line or in a local policy DB that is maintained by the service access control policy server. In all above scenarios, it is implied that the DB is adequately protected inside each domain and all sensitive data are stored in encrypted form.

Two network entities can effectively authenticate each other by exchanging their public key certificates. In the context of public key technology enabled protocols, like TLS or secure sockets layer protocol (TLS/SSL) (Dierks & Allen, 1999; Frier, Karlton, & Kocher, 1996) and IPsec, this yields robust authentication and end-to-end protected communication. Public key authentication is not limited only to the client/server model, but it can also be applied to peer-to-peer communication. For instance, two students of class X can be mutually authenticated and securely exchange data using e.g., their e-mail clients.

2.3. Additional security issues

It is worth noting that even when authentication and encryption issues are solved, problems like one student passing the exam for someone else still remain. The problem persists even if more expensive technologies like biometrics, smart-cards and monitoring or surveillance hardware are used. One can say that this issue is similar to e-voting security issues, which still remain to be solved. Even so, the majority of distance learning institutions organize examinations, which take place under controlled conditions.

ACs can be issued to the user either at the request of the provider (*push model*) or at the request of the user (*pull model*). In the first case, the provider requests from AA the ACs that correspond to specific roles or even temporary actions. AA validates the requests, creates the correct ACs, updates its DB and forwards them to the proper users. According to the pull model, a user can

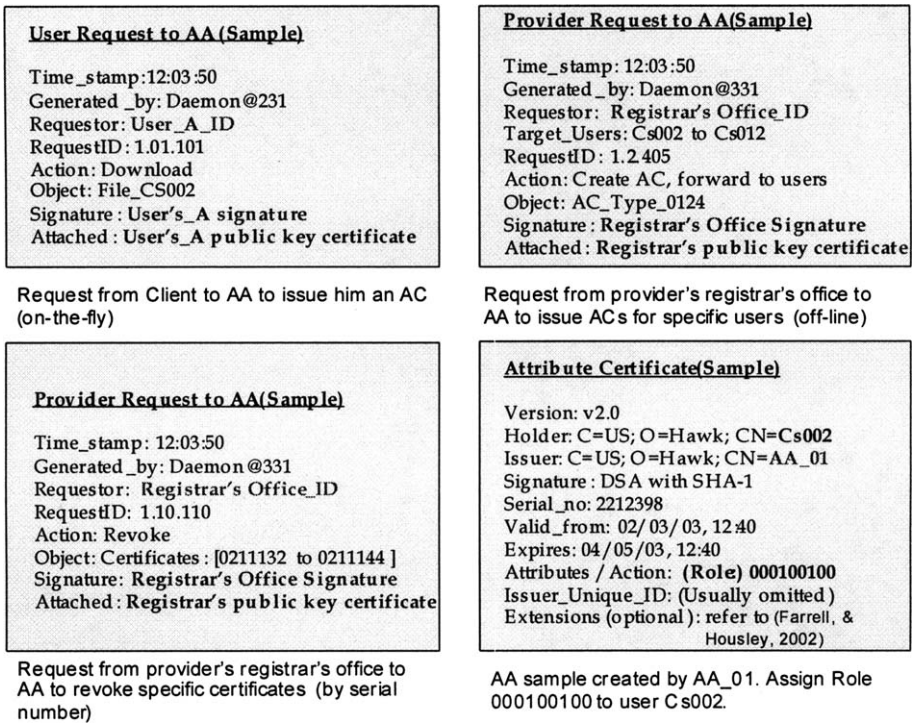

Fig. 4. Attribute certificate and requests (samples).

request an AC *on-the-fly* from the serving AA. The issued AC has to be in accordance with the roles previously assigned to that user, in order to avoid conflicts. Both types of requests are signed with the issuer's private key (Request||Digital_Signature) [2] and can be transmitted in clear-text, as they are actually useless to anyone who intercepts it.

In Fig. 4, we present three types of such potential authenticated requests and an AC sample. However, an attacker is still able to exploit a denial of service (DoS) attack, e.g. by intercepting and tampering the requests or the issued ACs. Allied to DoS attacks are distributed DoS attacks (DDoS) which deploy multiple machines to prevent legitimate users of a service from using that service (Mirkovic, Martin, & Reiher, 2002). The service is denied by sending a stream of packets to a victim (e.g. to the AA server or to the application server) that either consumes some key resource, thus rendering it unavailable to legitimate clients, or provides the attacker with unlimited access to the victim machine so he can inflict arbitrary damage (e.g. delete course material). Firewall and intrusion detection systems can be applied accordingly to ensure the availability of learning services and materials. Other sort of attacks or threats, expressly for ACs, can be found in (Farrell & Housley, 2002).

[2] $\text{Hash}_{(16\_bytes)}$ = MD5(Request) and Digital_Signature = $(\text{Hash}_{16\_bytes})_{\text{Issuer's\_Private\_Key}}$.

The syntax of the commands from the client/registrar to the AA is contained in the authenticated requests depicted in Fig. 4. In our scenarios we used three such fields named "RequestID", "Action" and "Object". However, these fields are shown for demonstrative reasons, as the request structure depends on the requirements embedded in each deployment. Thus, the applications could be adjusted to handle any specific type of request. For instance, the "RequestID" field in the form of a bit-pattern could designate a unique combination role–permission (key) in the roles/policy DB. The developer can tune this field as appropriate, making as many classifications he wishes to. Additionally, the "Action" field can specify concrete actions like "Revoke", "Download", "Read", "Update" etc., in relation to the "RequestID" field. Finally, the "Object" field may define the resource that the "Action" field is applied to. For instance, "Download learning material for the course CS001".

Finally, we note that ACs can have an extended life, e.g. four months (one semester), or can be temporary (transaction oriented; take part in a different virtual class meeting). One of the advantages of these temporary certificates, having a short life, is that they do not usually need to be revoked and will therefore need not be included in any certificate revocation list (CRL) (Iliadis et al., 2003). If they are issued in respect of a pre-paid service, they certainly not require any revocation at all. The need for certificates revocation arises in cases in which private keys are lost or compromised, the rights of access are changed, or it is desired to change keys as a precaution against cryptanalysis.

3. E-learning scenario

In a typical e-learning or m-learning client/server scenario, the user is connected to the provider's network and browses the service categories that define what services are currently available and which role has access to these services. Note, that we omit authentication procedures from our architecture for brevity. At some point the user makes a selection, requesting for example to download a file that refers to course "CS203". A server agent (thread) is subsequently generated to dispatch the request.

The server agent interactively asks the user to provide the AC corresponding to the requested service. Upon reception, the server agent has to validate the AC. First the AC's signature authenticity and origin is verified. It is implied that the certificate must be signed by an AA that the server agent trusts, while the public key certificates of all trusted CA/AAs can be kept in the server's cache memory. Next, the certificate's time expiration field is checked and finally, if appropriate, the server confirms that the AC is not included in the last retrieved CRL. Another option to test against revocation is the on-line certificates status protocol (OCSP) responder (Iliadis et al., 2003; Myers et al., 1999). Revocation for specific certificates can be requested by the registrar's office for example. If the certificate is valid and in accordance to the requested service, the server agent provides the service, otherwise it can offer the following options to the user:

(1) Allow him change his request.
(2) Allow the provider adjust his role and provide him the appropriate AC at some time later (push model).
(3) Allow him request the requisite AC from an AA *on-the-fly* (pull model).

If the user selects the last option, the user agent constructs on behalf of the user the appropriate request. After that, the user agent signs it with the user's private key and forwards it along with his public key certificate to the appropriate AA. AA shall validate the request (signature, expiration time, etc.), using the user's public key found in the user's public key certificate. Next, AA checks the user credentials by querying the provider's user's policy DB (according to the provider's specific policies) or – in case of mirroring – it's local roles/policy DB (see Section 2.2). This last step guarantees that the AC will conform to the roles that have been assigned to the user. In case the learner has been assigned two or more roles, the main point of this process is to reduce the role set, so that the resulting group does not have any mutually exclusive permission. If everything is acceptable, AA shall issue and forward the corresponding AC back to the user.

For a given transaction, another option is to pass the corresponding link (to the AC), instead of the actual AC, to the e-learning application server. However, the certificate forwarding is safer to be done by the user. If the client delivers a certificate URL, rather than the certificate itself to an application server, he implicitly requests from the server to do the work (retrieve the certificate from the AA's certificate repository). The danger is obvious: a DoS attack is possible when a malicious client deliberately passes an invalid certificate URL. For example, an attacker can flood the application or resource server with a large number of invalid certificate URLs, forcing the server to consume resources (retrieving ACs that never issued) and gradually become unavailable to the legitimate users.

As AC based authorization is also an extension to the IETF TLS protocol (Dierks & Allen, 1999), ACs can be used in end-to-end TLS protected sessions. Recent e-Europe Benchmarking Reports (http://www.elearningeuropa.info) find that the number of SSL equipped servers has increased considerably in Europe. Luxemburg for example, leads the European ranking with some 150 SSL servers per million inhabitants, while the USA ranks more than 300.

4. Testing ACs performance in a mobile scenario

From a technology perspective, handheld devices, such as personal digital assistants (PDAs) are more affordable today than before. From a pedagogical perspective, mobile learning supports a new dimension in the educational process (Chen, Kao, Sheu, & Chiang, 2002). Now, with mobile technologies, the range of education can be further extended by wireless connections to places where wired networks are not available or feasible. Mobile technologies fulfill the general requirements to support contextual lifelong learning by being highly portable, individual, unobtrusive and adaptable to the context of learning and the learner's evolving skills and knowledge (Insite, Ericsson, Telenor Mobil, & It Fornebu Knowation, 2001; Sharples, 2000).

In any case, handheld devices have their pros and cons (Ericsson, 2004; Luchini, Curtis, Quintana, & Soloway, 2002; Smith, Mohan, & Li, 1999) but without dispute, they are going to be particularly useful for people who have traditionally pursued distance education methods. Other important usability limitations are inherent in PDAs, like the confined screen size and the inevitable restrictions in the design of educational software for these devices. A number of good works that have been already conducted on usability topics which can be accessed in (Luchini, Quintana, & Soloway, 2004). Moreover, if used sensibly, adaptive mobile learning technologies have the

potential to revolutionize distance education by bringing the concept of *anytime and anywhere* to reality.

The contribution of this section is to provide evidence that delivering ACs using mobile devices and networks is attainable in terms of service times. It is generally known that mobile devices (PDAs, Pocket PCs, etc.) have relatively limited resources and computational power for processor-demanding public key operations when compared to desktop machines. Additionally, mobile networks currently provide limited bandwidth compared to wired ones. Under these conditions, it is necessary to test the feasibility of the proposed model in current and future mobile networks. In this work we used as a case study the delivery of ACs over the GPRS network.

4.1. Test bed setup

We constructed an experimental network architecture, which is illustrated in Fig. 5. The presumed mobile device is a low-end IBM ThinkPad 380 laptop computer that uses Windows 95B operating system. Contemporary wireless devices are featuring advanced architectures with Strong-Arm processors up to 400 MHz, memory capacities of 64 MB RAM and 48 MB ROM, support for various applications and strong operating systems (Kambourakis, Rouskas, & Gritzalis, 2004). The client (user agent) uses a Siemens ME45 mobile phone, in order to connect to the Internet over GPRS. The GPRS coding scheme (Korhonen, Aalto, Gurtov, & Laamanen, 2001) was CS1 (9.05 Kb/s) and the time slots for GPRS were varying from 3 to 4, thus having wireless network speeds in the range from 27 to 36 Kb/s. Network speeds for third generation mobile networks (3G) will be 144 up to 348 Kb/s for wide and up to 2 Mb/s for low coverage and mobility, which will substantially reduce transfer times. The same applies for IEEE 802.11× (wireless LANs), which currently enjoy network speed up to 54 Mb/s.

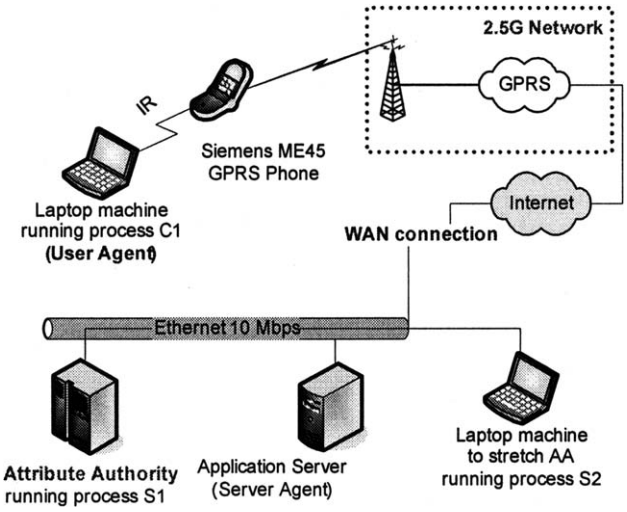

Fig. 5. Experimental hardware architecture for attribute certificates issuing.

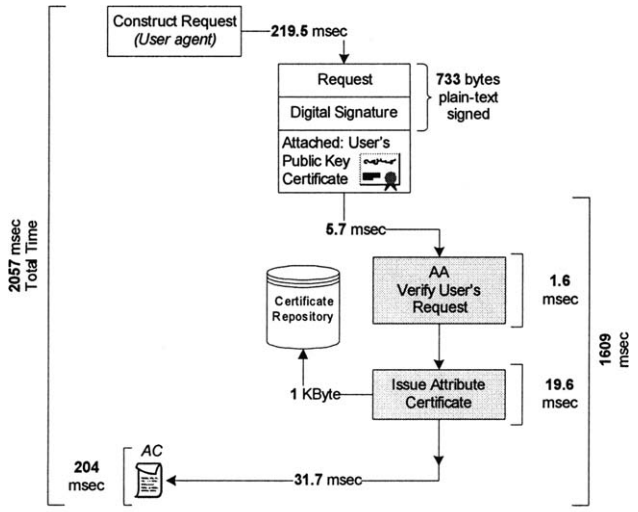

Fig. 6. Issuing attribute certificates: procedure and average service times.

The IBM 380 incorporates a 150 MHz Pentium CPU and has 16 MB of RAM available. At the other end, the AA machine has a Pentium III 733 MHz processor with 256 MB RAM, running the Windows 2000 professional SP2 operating system. AA process S1 is multi-threaded. When it receives a message, it dispatches a thread (CA & AA agent) to process and respond to the request. AA has also a WAN connection available. The multi-threaded process S2 that loads/stretches the AA with virtual requests is running on another laptop machine that incorporates a Celeron 1.2 GHz processor with 256 MB RAM and is wired to the local network with a speed up to 10 Mbps.

The applications were developed in Java 2 and employed the well-known open-source Apache-style license OpenSSL toolkit (http://www.openssl.org) in version 0.9.6g to make them public key enabled (Viega, Messier, & Chandra, 2002). The whole procedure, depicted in Fig. 6, is in accordance with the pull model described earlier in Sections 2.3 and 3.

4.2. Measurements results

We experimented with various values for the arrival rate (following the negative exponential distribution) of ACs requests, which determines the virtual load offered to the AA. The total client's request size is about 733 bytes. Measurements were gathered from a set of 1000 transactions between the AA server and the client. Our experiments were conducted in different days and hours during a week period and 50% of the measurements were logged during peak hours. The average values of the time durations measured are presented in Table 3 and the probability density function of client's total time is shown in Fig. 7. Maximum and minimum service time duration was 4.18 and 1.18 s, respectively. We notice that the average total time of the transaction to complete is about 2.1 s, with a standard deviation of 0.35, which is generally acceptable by a user who demands "a fast and secure service".

Table 3
Measurements results

Average time in milliseconds								
Client					Attribute authority			
Time for the user agent to create request	Time to send request	Time to send request and receive response	Time to check response	Total time	Time to verify client's request	Time to create AC	Time to send AC back to client	Total time
219.5	5.7	1609	204	2072	1.6	19.6	31.7	149.3

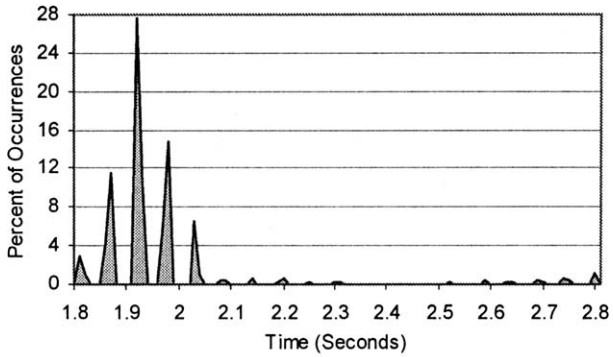

Fig. 7. Client's service total time PDF.

5. Conclusions

It is true that technical issues are not the only obstacle to deploy accredited and fully functional DLS systems. However, as users and providers rush to adopt e-learning and m-learning schemes, they also become aware of the need for security features and protection of their privacy. The constantly increasing population of e-learners demanding analogous services, expects from operators to provide features that will protect their data while in transit, safeguard their billing and customer information, provide reliable AAA mechanisms and offer availability and service quality. Thus, more flexible, dynamic and scalable mechanisms are necessary in order to support on-demand anytime/anywhere services and solutions in a many-to-many trust model integrated with the unsecured Internet environment.

PKI is a reality in wired networks and is about to be incorporated into mobile networks in the near future. In this paper, we discussed and investigated how public key certificates and attribute certificates, organized under a PKI, can provide strong mutual authentication and fine-grained trust control of common e-learning or m-learning services respectively. Other critical concepts like non-repudiation and message confidentiality and integrity were furnished as well. We experimented with on-the-fly attribute certificate generation, testing the performance of a prototype implementation. Results showed that ACs issuing is attainable in terms of service time, while simultaneously can deliver flexible and scalable solutions to both future wired and mobile operators and users.

Acknowledgements

We thank the reviewers for their valuable comments that helped us improve the quality and presentation of our work.

References

Adams, C., & Lloyd, S. (1999). *Understanding public-key infrastructure, concepts, standards and deployment considerations*. New Riders: Indianapolis, IN.

Arseanult, A., & Turner, S. (2002). Internet X.509 Public key infrastructure: Roadmap, PKIX Working Group. IETF Internet Draft. <draft-ietf-pkix-roadmap-09.txt>, July 2002.

Chadwick, D. (2002). The PERMIS X.509 based privilege management infrastructure. IETF Internet Draft. <draft-irtf-aaaarch-permis-00.txt>, April 2002.

Chadwick, D., Tassabehji, R., & Young, A. (2000). Experiences of using public key infrastructure for the preparation of examination papers. *Computers and Education, 35*(1), 1–20, Elsevier Science.

Chen, Y. S., Kao, T. C., Sheu, J. P., & Chiang, C. Y. (2002). A mobile scaffolding-aid-based bird-watching learning system. In *Proceedings of IEEE workshop on wireless and mobile technologies in education, Los Alamitos, USA* (pp. 15–22).

Clark, R. C., & Mayer, R. (2002). *E-Learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*. Jossey-Bass/Pfeiffer.

Diatchenko, J., Miloslavskaya, N., & Tolstoy, A. (2002). Designing secure distance learning system. In *Proceedings of The VII international conference on engineering and technology education, Santos, Brazil* (pp. 17–20).

Dierks, T., & Allen, C. (1999). *The TLS protocol version 1.0*. IETF RFC 2246, January 1999.

Ericsson, A. S. (2004). *Mobile learning: the next generation of learning project*. Available from http://learning.ericsson.net/mlearning2/.

Farrell, S., & Housley, R. (2002). *An Internet attribute certificate profile for authorization*. IETF RFC 3281.

Ferraiolo, D. F., Cugini, J. A., & Kuhn, R. D. (1995). *Role-based access control (RBAC): features and motivations*. Available from http://hissa.ncsl.nist.gov/rbac/newpaper/rbac.html.

Ferraiolo, D. F., Sandhu, R., Gavrila, E., Kuhn, D. R., & Chandramouli, R. (2001). Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security, 4*(3), 224–274.

Frier, A., Karlton, P., & Kocher, P. (1996). *The SSL 3.0 protocol version 3.0*. Available from http://home.netscape.com/eng/ssl3/draft302.txt.

Furnell, S. M., Onions, P. D., Bleimann, U., Gojny, U., Knahl, M., & Roder, H. F., et al. (1998). A security framework for on-line distance learning and training. *Internet Research, 8*(3), 236–242.

Furnell, S. M., Bleimann, U., Girsang, J., Roder, H., Sanders, P., & Stengel, I. (1999). Security considerations in online distance learning. In *Proceedings of Euromedia, Munich, Germany* (pp. 131–135).

Graf, F. (2001). Secure iLearning. In *Proceedings of communications and multimedia security issues of the new century* (pp. 267–281). Massachusetts, USA: Kluwer Academic Publishers.

Graf, F., Busch, C., & Wolthusen, S. (1999). Courseware needs security. In *Proceedings of ICCE 1999 international conference on computers in education, Chiba, Japan*.

Harris, P. (2001). *Goin' mobile, ASTD's online magazine all about e-learning*. Available from http://www.leamingcircuits.org/2001/jul2001/harris.html.

Horton, W. (2000). *Designing web-based training: How to teach anyone anything anywhere anytime* (1st ed.). New York: John Wiley & Sons.

Iliadis, J., Gritzalis, S., Spinellis, D., de Cock, D., Preneel, B., & Gritzalis, D. (2003). Towards a framework for evaluating certificate status information mechanisms. *Computer Communications, 26*(16), 1839–1850, Elsevier Science.

Insite, A. S., Ericsson, A. S., Telenor Mobil, A. S., & It Fornebu Knowation, A. S. (2001). *M-learning: experiences from the use of WAP as a supplement in learning, pilot project*. Available from http://www.insiteint.com/download/M-learning_wap.pdf.

ITU-T Recommendation X.509. (1997). *Information technology-open systems interconnection-the directory: authentication framework* (equivalent to ISO/IEC 9594-8, 1997).

Kambourakis, G., Rouskas, A., & Gritzalis, S. (2004). Experimental analysis of an SSL-based AKA mechanism in 3G-and-beyond wireless networks. In *Wireless personal communications (WPC). Special issue on security for next generation communications*. Kluwer (prepublication date: 4/2004).

Kaufman, C. (2004). *Internet key exchange (IKEv2) protocol*. <draft-ietf-ipsec-ikev2-13.txt>, March 2004.

Kent, S., & Atkinson, R. (1998). *Security architecture for the Internet protocol*. RFC 2401, November 1998.

Korhonen, J., Aalto, O., Gurtov, A., & Laamanen, H. (2001). Measured performance of GSM HSCSD and GPRS. In *Proceedings of the IEEE international conference on communications (ICC'01), Helsinki, Finland*.

Luchini, K., Curtis, M., Quintana, C., & Soloway, E. (2002). Supporting learning in context: extending learner-centered design to the development of handheld educational software. In *Proceedings of IEEE international workshop on wireless and mobile technologies in education, Los Alamitos, USA* (pp. 107–111).

Luchini, K., Quintana, C., & Soloway, E. (2004). Design guidelines for learner-centered handheld tools. In *Proceedings of ACM conference on human factors in computing systems* (pp. 135–142). Vienna, Austria: ACM Press.

Mirkovic, J., Martin, J., & Reiher, A. (2002). A taxonomy of DDoS attacks and DDoS defense mechanisms. UCLA CSD Technical Report No. 020018. Electronically available from http://www.lasr.cs.ucla.edu/ddos/ucla_tech_report_020018.pdf.

Myers, M. et al. (1999). *X.509 Internet public key infrastructure online certificate status protocol–OCSP*. IETF RFC 2560.

Nash, A., Duane, W., Joseph, C., & Brink, D. (2001). *PKI implementing and managing E-security*. Berkeley: RSA Press.

Oppliger, R. (2002). *Internet and intranet security* (2nd ed.). Norwood, MA: Artech House.

Oppliger, R., Pernul, G., & Strauss, C. (2000). Using attribute certificates to implement role based authorization and access control models. In *Proceedings of the 4 Fachtagung Sicherheit in information systemen (SIS2000), Zurich, Switzerland* (pp. 169–184).

Otenko, S., & Chadwick, D. (2003). A comparison of the Akenti and PERMIS authorization infrastructures, version 2.1, July 2003. Electronically available from http://esc.isi.salford.ac.uk/download/AkentiPERMISDeskcomparison2-1.pdf.

PKIX Working Group, 2004. Public-key infrastructure (X.509) (pkix), Last Modified: 2004-09-07. Electronically available from http://www.ietf.org/html.charters/pkix-charter.html.

Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control models. *IEEE Computer, 29*(2) <http://csrc.nist.gov/rbac/sandhu96.pdf>, IEEE Press.

Sharples, M. (2000). The design of personal mobile technologies for lifelong learning. *Computers and Education, 34*, 177–193.

Smith, J., Mohan, R., & Li, C. (1999). Scalable multimedia delivery for pervasive computing, *ACM Multimedia*. Available from www.ee.columbia.edu/~jrsmith/htm/pubs/acmmm99.pdf.

Thompson, M., Johnston, W., Mudumbai, S., Hoo, G., Jackson, K., & Essiari, A. (1999). Certificate-based access control for widely distributed resources. In *Proceedings of the 8th USENIX security symposium, Washington, DC*.

Vandenwauver, M., Govaerts, R., & Vandewalle, J. (1997). How role based access control is implemented in SESAME. In *Proceedings of the 6th workshops on enabling technologies: infrastructure for collaborative enterprises* (pp. 293–298). IEEE Computer Society Press.

VeriSign. (2002). Enterprise key management, white paper 005308, September 2002.

Viega, J., Messier, M., & Chandra, P. (2002). *Network security with OpenSSL*. CA: O'Reilly & Associates.