Access Control and Trust in the Use of Widely Distributed Services

Jean Bacon, Ken Moody, and Walt Yao

University of Cambridge Computer Laboratory
New Museum Site, Pembroke Street
Cambridge CB2 3QG, United Kingdom
{jean.bacon, ken.moody, walt.yao}@cl.cam.ac.uk

Abstract. OASIS is a role-based access control architecture for achieving secure interoperation of independently managed services in an open, distributed environment. OASIS differs from other RBAC schemes in a number of ways: role management is decentralised, roles are parametrised, and privileges are not delegated. OASIS depends on an active middleware platform to notify services of any relevant changes in their environment.

Services define roles and establish formally specified policy for role activation and service use; users must present the required credentials and satisfy specified constraints in order to activate a role or invoke a service. The membership rule of a role indicates which of the role activation conditions must remain true while the role is active. A role is deactivated immediately if any of the conditions of the membership rule associated with its activation become false.

Instead of privilege delegation OASIS introduces the notion of appointment, whereby being active in certain roles carries the privilege of issuing appointment certificates to other users. Appointment certificates capture the notion of long lived credentials such as academic and professional qualification or membership of an organisation. The role activation conditions of a service may include appointment certificates, prerequisite roles and environmental constraints.

We define the model and architecture and discuss engineering details, including security issues. We illustrate how an OASIS session can span multiple domains, and discuss how it can be used in a global environment where roving principals, in possession of appointment certificates, encounter and wish to use services. We propose a minimal infrastructure to enable widely distributed, independently developed services to enter into agreements to respect each other's credentials.

We speculate on a further extension to mutually unknown, and therefore untrusted, parties. Each party will accumulate audit certificates which embody its interaction history and which may form the basis of a web of trust.

1 Introduction

OASIS is an access control system for open, interworking services in a distributed environment modelled as domains of services. Services may be developed inde-

R. Guerraoui (Ed.): Middleware 2001, LNCS 2218, pp. 295–310, 2001.

pendently but service level agreements allow their secure interoperation. OASIS is closely integrated with an active, event-based middleware infrastructure. In this way we can notify applications of any change in their environment, making it possible to ensure that security policy is satisfied at all times.

OASIS is role based but has important differences from other RBAC schemes [3,4,5,6,7,8,11,12,13,14,15,16]:

- Roles are service-specific; there is no notion of globally centralised administration of role naming and privilege management.
- Roles may be parametrised, as required by applications.
- Roles are activated within sessions. A session is started by activating an initial role such as *logged_in_user*. Most roles have activation conditions that require prerequisite roles and a session of active roles is built up.
- All privileges are associated with roles. We use appointment instead of privilege delegation; the activation conditions of roles which convey the privileges to be granted may require appointment certificates.
- We provide an *active security environment*. Constraints on the context can be checked during role activation; the role may be deactivated if particular conditions become false subsequently.

Services name their client roles and enforce policy for role activation and service invocation, expressed in terms of their own and other services' roles. Section 2 gives the details of role activation conditions. An encryption protected role membership certificate (RMC) is returned to the user on successful role activation and this may be used as proof of authorisation to use this and other services and as a credential for activating other roles, according to services' policy. Engineering details are in Sect. 4.

In contrast with some RBAC schemes we do not support the delegation of privileges. Instead we use *appointment*, in which one function of certain roles is to issue appointment certificates which may be used, together with any other credentials required by policy, to activate one or more roles. There is no reason why the holder of the appointer role should be entitled to the privileges conferred by the certificates; for example, a hospital administrator need not be medically qualified. Appointment is discussed in Sect. 2.

RBAC, in associating privileges with roles, provides a means of expressing access control which is scalable to large numbers of principals. The detailed management of large numbers of access control lists, as people change their employment or function, is avoided. However, pure RBAC associates privileges only with roles, whereas applications often require more fine-grained access control. Parametrised roles extend the functionality to meet this need. Role activation conditions can check parameter values and the relationship between parameters, so that policy can express exceptions to the default access controls. OASIS role membership certificates are always principal specific but may or may not be parametrised, depending on the requirements of the application. Some applications may require anonymous certificates. Section 5 discusses this issue in more detail.

Appointment meets the requirement for long-lived credentials; in contrast privileges are associated with roles, which are activated in the context of a session. This allows an active security environment to be maintained, in which any breach of role membership conditions can be notified. Sections 4 and 5 discuss the architecture and engineering of sessions.

In practice, distributed systems contain many domains; for example the healthcare domain comprises subdomains of public and private hospitals, primary care practices, research institutes, clinics, etc. as well as national services such as electronic health record management. Access control policy may be dictated by national law and/or may come from organisational decisions. In [1] we present an early attempt at pseudo-natural language policy expression and its automatic translation into first-order predicate calculus. Such a mechanism is crucial for any large-scale deployment of policy; it is essential to maintain consistency as policies evolve. The formal expression of policy and its automatic deployment is an independent thread of research which is not the main focus here.

Widely distributed services may establish agreements on the use of one another's appointment certificates. Within a session at a user's place of work it is often necessary to make cross-domain invocations of remote services. For example, a doctor carrying out emergency treatment for a patient who is away from home may need access to the patient's health record. Another example is the negotiated use of remote digital libraries or databases. This is a natural part of the architecture and is described in Sect. 3.

Sections 5 and 6 discuss extended uses of the OASIS approach. Suppose someone temporarily leaves their home base to work as a known principal in a known domain. For example, a doctor employed in a hospital may need to work temporarily in a research institute, perhaps in another country. There may be a reciprocal agreement between the hospital and the research institute to accept electronic evidence of medical qualifications, subject of course to validation by the issuer.

A different scenario is that a member of some organisation may have the right to use the services of another, negotiated between the two organisations. An analogy is that the English and Scottish National Trusts may give privileges to each other's members. Here, any paid-up member of a local organisation may apply to use a known remote organisation.

The above scenarios assumed trusted services inhabiting mutually aware domains, such as the healthcare domains of different regions or nations. More generally, we may wish to set up an infrastructure for a world in which roving computational entities encounter previously unknown, and therefore untrusted, services. Both parties should be able to present checkable credentials to provide evidence of previous successful interactions. We discuss the difficulties of establishing such an infrastructure and look at some of the risks involved.

In summary: in Sect.s 2 and 3 we outline the OASIS model and architecture. In Sect. 4 we highlight some significant engineering issues, including the integration of OASIS access control with authentication and secure communication. In

Sect. 5 we discuss the use of OASIS for widely distributed but mutually aware domains. In Sect. 6 we speculate on how principals and services might have their interactions certified in order to establish a more general scenario that includes mutually untrusted parties.

2 The OASIS Access Control Model

[17] presents the OASIS model in detail including the formal semantics. Here we outline the basic model.

OASIS embodies an open, decentralised approach, appropriate to deployment in distributed systems. Roles are defined by services and services may interoperate, recognising one another's roles, according to service-level agreements. Central to the OASIS model is the idea of **credential-based role activation** at a service. The credentials that a user possesses, together with side conditions which depend on the state of the environment, will authorise him or her to activate roles.

Activation of any role in OASIS is explicitly controlled by a **role activation rule**. A role activation rule specifies, in Horn clause logic, the conditions that a user must meet in order to activate the role. The conditions may include **prerequisite roles, appointment credentials** and **environmental constraints**.

A **prerequisite role** as a condition for a target role means that a principal must already have activated the prerequisite role before it can activate the target role. Some services define roles which do not include prerequisite roles in their role activation rule. An example is a login service which defines a role *logged_in_user*. The activation conditions might include proof that the user is employed by or registered at the organisation to which the system belongs, and can supply authentication evidence, but a prerequisite role is not required. By activating such a role, called an initial role, a user may start an OASIS session; other roles' activation rules may include the prerequisite role *logged_in_user*. Figure 1 illustrates role dependency, with a role activation rule for service C that includes only prerequisite roles.

Appointment credentials were motivated in Sect. 1. Being active in certain roles gives the principal the right to issue appointment certificates to one or more other principals. Appointment certificates may be used, together with any other credentials required by policy, to activate one or more roles. They are certificates whose lifetime is independent of the duration of the session of activation of the appointer role. They may be long-lived, such as when they are used to certify academic or professional qualification, employment or membership of an organisation. They may be transient, for example when certifying that someone is authorised to stand in for a colleague who is called away while on duty. In some cases the appointment is made with the intention of allowing the appointee to activate a specific role. An example is that a screening nurse in an Accident and Emergency (A&E) Department may allocate a patient to a particular doctor. He/she issues an appointment certificate to the doctor who may then activate the role *treating_doctor* for that patient. In other cases the

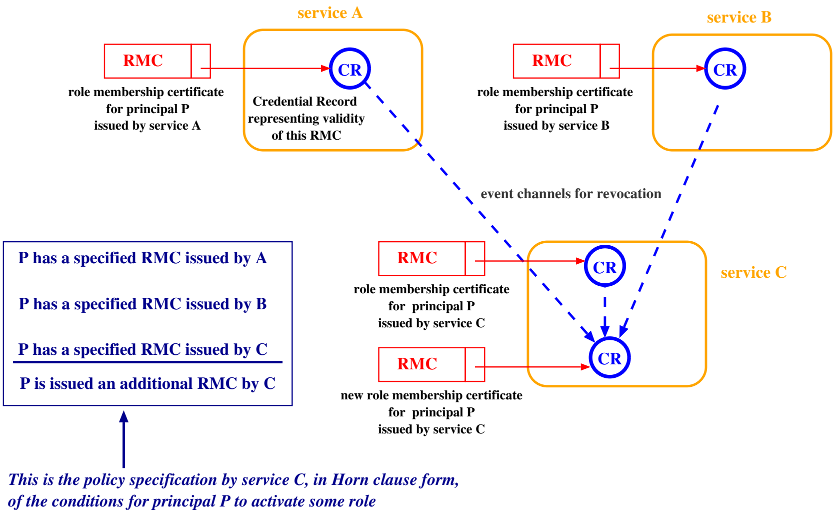
service A
RMC
role membership certificate
for principal P
issued by service A
CR
Credential Record
representing validity
of this RMC
service B
RMC
role membership certificate
for principal P
issued by service B
CR
event channels for revocation
P has a specified RMC issued by A
P has a specified RMC issued by B
P has a specified RMC issued by C
P is issued an additional RMC by C
RMC
role membership certificate
for principal P
issued by service C
CR
service C
RMC
new role membership certificate
for principal P
issued by service C
CR
This is the policy specification by service C, in Horn clause form,
of the conditions for principal P to activate some role

Fig. 1. Role dependency through prerequisite roles

appointer may have no reason to know which roles will include the appointment certificate in their activation rules. In general, there is no reason why the appointer should be entitled to the privileges conferred (through subsequent role activation) by the appointment certificates it issues. In a computerised system it is natural that an administrator should enter a role and issue academic, professional or membership appointment certificates; a hospital administrator need not be medically qualified. If an application requires delegation then it can be built using appointment. The role of the *delegator* must be granted the privilege of issuing appointment certificates, and a role must be established to hold the privileges to be assigned. Finally an activation rule must be defined to ensure that the appointment certificate is presented in an appropriate context.

OASIS roles are parametrised; without parameters RBAC cannot meet the requirements of some application domains. For example, the English “Patients’ Charter” allows patients to specify who may see which parts of their health records; the policy deployed at any service which provides health care should take account of such directions by the patient. Default policy is role based for scalability; for example “doctors may access the records of patients registered with them”. Policy as implemented must respect individual exceptions indicated by patients; for example “Fred Smith” (although a doctor) “may not access my health record”. Constraint checking during service use allows such exceptions to be enforced; it is vital that doctors who access patient records may be identified individually. For access control in a file service it is necessary to indicate individual owners of files as well as groups of users. In general, OASIS role parameters might be the identifier or location of the computer, the name of the activator of the role, some identifier of the activator, such as a public key or health service identifier, the patient the activator is treating, and so on.

As noted above, in addition to prerequisite roles and appointment credentials, role activation rules may include **environmental constraints**. These may be user-independent constraints or conditions on user-dependent parameters. Examples of user-independent constraints are the time of day and the location or name of a computer. An example of a condition on user-dependent parameters is that the user is a member of a group; this may be ascertained by database lookup at some service. Another is that parameters are related in a specified way; for example the doctor has the patient registered as under his/her care. Again, a check can be made against a database. A simple parameter check may ascertain that the user is a specified exception to a general category who may activate the role.

The membership rule for a role indicates those predicates for activating the role that must continue to be true for the role to remain active. In Sect. 4 we show how membership rules are monitored and enforced in our active environment.

As in traditional RBAC, roles convey privileges; specifically, the privilege of method invocation (including object access) at services. The conditions for service invocation are possession of role membership certificates of this and other services together with environmental constraints. Since roles are parametrised we can enforce policies on object access such as “doctors may access patients’ health records” together with exclusions on individual objects: “Joe Bloggs’ health record may not be accessed by Fred Smith”.

3 OASIS Architecture

Figure 2 shows the architecture of an OASIS service which defines roles. Services may also be OASIS-aware and specify roles of other services as credentials to authorise their use, without themselves defining roles, see [10].

A client activates a role by presenting credentials to a service that enable the service to *prove* that the client conforms to its policy for entry to that role (path 1). The service validates the credentials. This may involve checking back with certificate-issuing services and checking environmental conditions. If the checks succeed a *role membership certificate* (RMC) is issued (path 2). This may be presented with subsequent requests to use that service, together with any other credentials specified by the service’s policy (path 3). The service validates the credentials, checks any constraints and, if all is well, the invocation proceeds (path 4). The design of RMCs and how they are validated is discussed in Sect. 4.

An OASIS session may include remote, cross-domain service invocations based on prior service-level agreements. Figure 3 gives an example, taken from a national electronic health record (EHR) service. We assume a national EHR domain and many local healthcare domains such as primary care groups and hospitals. A hospital domain is likely to include many computerised services such as those which record pharmacy and X-ray usage. Figure 3 shows one of

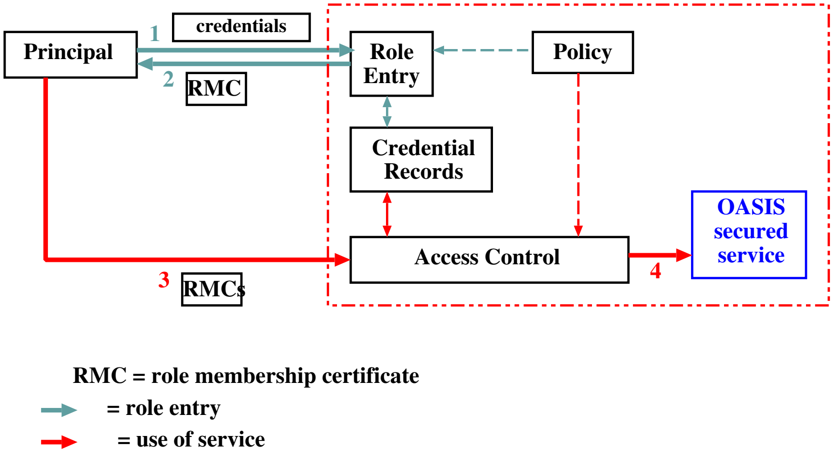

Fig. 2. A service secured by OASIS access control

these services; that which delivers EHR components as requested by authorised principals.

The scenario is that a doctor has successfully activated the parametrised role *treating_doctor (doctor_id, patient_id)* within the hospital domain where he/she is employed. The doctor needs to refer to certain records of treatment within the patient's EHR and invokes the local EHR service to make the request, with the RMC *treating_doctor* as credential. The local EHR service is OASIS aware, validates the credential by callback to the hospital administration, and the invocation proceeds. The local EHR service must now invoke the national EHR service to locate and acquire the EHR components. The national EHR service has a rolename which principals from authorised healthcare domains can activate, say *hospital(hospital_id)*. In the figure the role is active and the invocation **request-EHR** is made (path 1). Service level agreements between the national service and individual health care domains would establish a protocol to validate local RMCs so that the identity of the original requester can be recorded for audit. The detailed design of the national EHR service would specify the credentials and parameters required for the invocation. For example, the hospital certificate is the required credential and the *treating_doctor (doctor_id, patient_id)* certificate would form part of the audit record. The doctor and patient ids are shown embedded in the *treating_doctor* certificate but they could be included, in addition, as arguments of the calls.

The national EHR service validates the hospital certificate, notes that the requester is in the role *treating_doctor* and uses the parameter *patient_id* to find information on where components of the EHR are stored. Before returning the data to the hospital it is essential to check that the role is sufficient to permit access to the fields requested and that this particular doctor has not been excluded from access by the patient. If all is well the data is returned

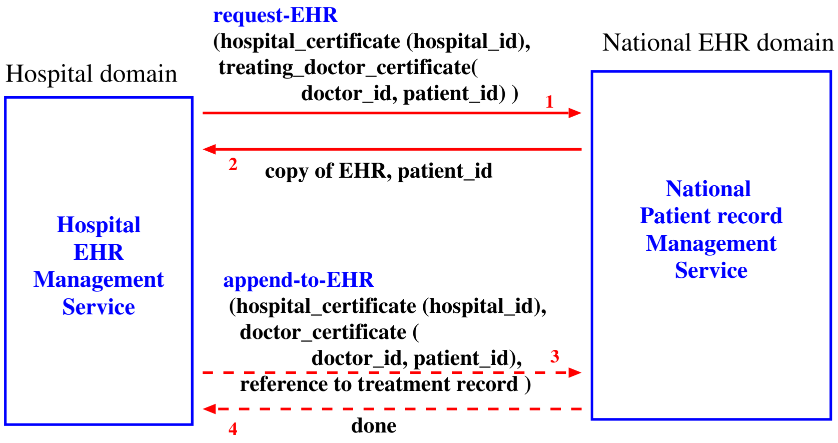

Fig. 3. An OASIS session with cross-domain calls

(path 2). The doctor carries out an item of treatment. Paths 3 and 4 show the authorised and audited inclusion of the record of treatment in the patient's EHR. The architecture must guarantee that these steps complete successfully, but the response time is not critical. This is a simplification of the way in which EHRs are likely to be managed.

The above examples illustrate how OASIS access control operates within and across domains. We now expand on some engineering issues.

4 OASIS Engineering

There need not be a centrally dictated design of role membership certificates, although there is likely to be a unified design within each domain. An Oasis-aware service will validate a certificate presented as an argument via callback to the issuer. The service may cache the certificate and the result of validation in order to reduce the communication overhead of repeated callback. This requires an event channel so that the issuer can notify the service should the certificate be invalidated for any reason, see [2].

Figure 4 presents a possible RMC design. RMCs are encryption-protected to guard against tampering and are principal-specific to guard against theft, as shown in the figure. The *principal_id* is discussed further in Sect. 4.1. The role-name and any parameters may be recorded in the RMC and protected. The owner's personal identification may or may not be one of the parameters, depending on application requirements.

The issuer keeps information on the RMC, including its current validity, in a credential record (CR). The credential record reference (CRR) in the RMC allows the issuer and the CR to be located. Details of how the CR might be

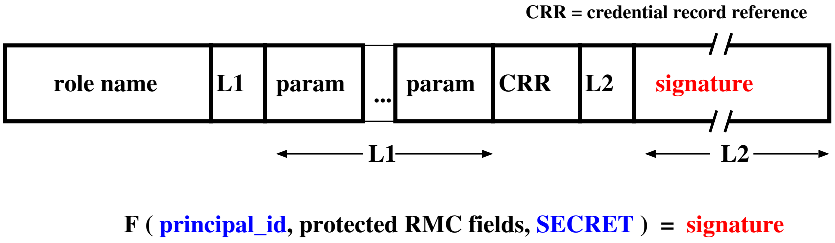
CRR = credential record reference
role name
L1
param
...
param
CRR
L2
signature
L1
L2
F (principal_id, protected RMC fields, SECRET) = signature

Fig. 4. A possible role membership certificate (RMC) design

designed were given in [9]. [10] discussed engineering issues for OASIS implementations in administrative domains that span many individual services. It is likely that certificates will not be issued and validated by each individual service as is possible in the architecture. Rather, a domain will contain one highly available service to carry out the functions of certificate issuing and validation. The paper outlined the design of such a service, including replication for availability together with consistency management.

It is important to note that OASIS is integrated with an event infrastructure [2]; this allows services protected by OASIS to communicate asynchronously, so that one service can be notified of a change of state at another without any requirement for periodic polling.

An OASIS session typically starts from the activation of an initial role, such as authenticated, *logged_in_user*. A user may activate further roles by submitting the credentials required to satisfy an activation rule for each. Active roles therefore form trees of role dependencies rooted on initial roles. If a single initial role is deactivated, for example the user logs out, all the active roles dependent on it collapse and that session terminates.

The event-based middleware infrastructure is used to monitor the membership conditions of active roles. Should any membership condition for a role become false the role is deactivated and the dependent subtree is collapsed. Figure 5 shows event channels that capture role dependencies in an OASIS session.

4.1 Engineering Authentication and Secure Communication

Here we discuss how OASIS might be integrated with standard authentication and encrypted communication. Within a firewall-protected domain OASIS might be used without encrypted communication and yet still provide acceptable access control. Colleagues are unlikely to wire-tap, although looking at a colleague's screen may well reveal confidential data. Appointment certificates might be copied from file spaces if care is not taken to protect them, but a thief should not be able to exploit them if the activation rules are well designed. Although call and return parameters are potentially visible "on the wire" the persistent data is protected from uncontrolled update.

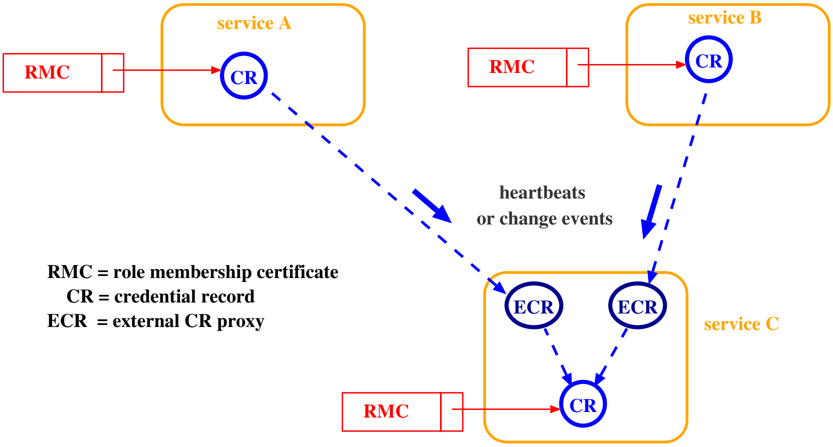

Fig. 5. Active security via an event infrastructure

If any visibility of data and certificates "on the wire" is unacceptable to an application, which must be assumed to be the case with cross-domain interworking, then encrypted communication must be used. Sensitive data might be encrypted selectively within a trusted domain. Data sent to a service can be encrypted with the service's public key and the public key of the caller can be included for encrypting the reply.

First, suppose that certificates are sent and stored in clear in a local domain. Security properties and related issues are as follows:

– **Protection from tampering.** The fields of role membership and appointment certificates are protected by their signatures.
– **Protection from forgery.** A secret held by the issuing service is an argument to the encryption function of role membership and appointment certificates. A correct signature cannot be generated except by knowledge of the secret.
– **Protection of RMCs from theft.** OASIS RMCs are principal-specific. Although not visible as a parameter field in the RMC, a principal_id is an argument to the encryption function that generates the signature, see Fig. 4. A stolen RMC therefore cannot successfully be used by an adversary unless the principal_id can be guessed and forged. The choice of a suitable id is discussed below. In addition, an RMC can contain principal-related information as parameters in protected fields; this can help to prevent thieves from abusing appointment certificates.
– **Choice of a principal_id for RMCs.** Since OASIS is session-based a session-specific principal_id could be used, which would provide better security than a persistent principal_id. An unforgeable location-dependent principal_id would be ideal but that may not be easy to create. The principal, of course, would need to run the whole session from that location. The host_id,

IP address and process_id of the caller are perhaps too easy to forge. An alternative is that a key-pair can be created by the principal and the public key sent to the service to be bound into the certificate. The service can establish at any time that the caller holds the corresponding private key by running a challenge-response protocol. This is discussed further below.
- **Protection of appointment certificates from theft.** Unlike RMCs, appointment certificates have a lifetime which is independent of a session. They cannot be made principal-specific by using a session-dependent principal identifier. Like RMCs, the detailed design is issuer-dependent and the issuer validates them on demand. They can be made principal-specific by including a persistent principal_id as an argument to the encryption function, such as a long-lived public key of the principal. A long-lived appointment certificate is more vulnerable to attack than an RMC and it is likely that appointment certificates would be re-issued, encrypted with a new server secret, from time to time.

Many RBAC schemes do not have the OASIS distinction between potentially long-lived appointment certificates and session-based RMCs. An implementation of long-lived role membership would carry the same vulnerability to attack as OASIS appointment certificates. Session-based role activation is more secure, provided that we have strong authentication when an initial role is activated. Authentication is discussed further below.

Authentication

OASIS assumes that principals have been authenticated when appointment certificates and RMCs are granted to them. When a certificate is subsequently presented as a credential the assumption is that the principal presenting it is the principal to whom it was issued.

Authentication of a principal at the start of an OASIS session, when a long-lived appointment credential may be used to activate an initial role, is therefore an important issue. Subsequently, the RMC for the initial role is effectively an authentication token and all subsequent role activations and service invocations are provably from the same principal, as discussed above. Authentication might be through password-protected login in a working environment. In the future, biometrics will be used increasingly and might be included in an appointment certificate.

Integration with PKC

OASIS can be integrated with public/private key cryptography for authentication as well as secure comunication. A public key of the activator of an initial role could be used as the session key described above, bound into the signature of every subsequent RMC and sent to the service under the service's public key. The service can check that the activator has the corresponding private key by using a challenge-response protocol, such as ISO/9798. The issuing service produces a random challenge, encrypted with the public key presented by the activator, and a nonce. The client must respond with the challenge in plaintext encrypted with the nonce. Upon receiving this, the service can conclude that the activator has access to the private key corresponding to the public key presented, and can safely bind it into the signature. A similar challenge can be made at any time; in the extreme, every time a certificate is presented. In practice the challenge might be made at random during a session, and at selected times such as before sensitive data is sent.

5 OASIS for Multiple, Mutually-Aware Domains

Section 4 covered the case where a session within someone's normal workplace domain includes the invocation of services in other domains. We assumed prior agreement between domains/services on the use of each other's RMCs as credentials for role entry and/or service invocation. Now suppose someone wishes to work away from their home base temporarily, in a known domain. For example, a doctor employed in a hospital may need to work for a short time in a research institute, without changing employment.

The hospital and research institute trust each other; they are subdomains within a national healthcare domain. They agree that the home domain's administrative service will issue an appointment certificate to the doctor. This will serve as a credential for entering the role *visiting_doctor* in the research institute; a role which has more privileges than a minimal visitor's role such as *guest*.

In the home domain appointment certificates *employed_as_doctor (hospital_id)* are issued only to members of staff who can prove that they are academically and professionally qualified in medicine. Their credentials are checked before the *employed_as_doctor* appointment certificate is issued. The doctor can enter the role *visiting_doctor* in the research institute through an activation rule which recognises the home domain appointment certificate as a precondition that proves that the doctor is medically qualified; this activation rule is part of the policy established by the service level agreement between the hospital and the research institute. The research institute would check the validity of the appointment certificate during role activation by callback to the hospital.

The reciprocal side of the agreement would allow medical research workers to work temporarily in the hospital. The appointment certificate *research_medic (research_institute_id)* might similarly serve as a credential for activating medical roles at the hospital, subsuming *employed_as_doctor (hospital_id)* etc.

The fields of appointment certificates (and RMCs) are readable, although protected from tampering and theft as described above. Parameters recorded there may be read and any environmental conditions checked.

Group Membership Negotiations

A different scenario is that a member of some organisation may have rights to use the services of another, negotiated between the two services. An analogy is that an art lover can become a friend of the Tate Gallery, and receive the newsletter of Tate London, Tate St Ives or Tate Liverpool. A friend registered at one gallery will also receive the privileges of a friend at any other. Here, any paid-up member of a local organisation may apply to use a known remote organisation. The identity of the principal is not needed if proof of membership is securely provable. An appointment certificate (the electronic equivalent of a membership card) might indicate the organisation and the period of membership and might or might not include personal details.

Anonymity

Suppose that privacy legislation has been passed whereby someone who has paid for medical insurance may take certain genetic tests anonymously. The insurance company's membership database contains the members' data; the genetic clinic has no access to this. The insurance company must not know the results of the genetic test, or even that it has taken place. The clinic, for accounting purposes, must ensure that the test is authorised under the scheme.

A member of the scheme is issued a computer-readable membership card containing an appointment certificate and the expiry date. The member activates the role *paid_up_patient* at the clinic and proceeds to take the genetic test. The activation rule for the role comprises the appointment certificate and an environmental constraint requiring that the date of the (start of the) test is before the expiry date of the insurance scheme membership. The appointment certificate is validated at the issuing service (a trusted third party) before role activation can proceed.

6 Untrusted Environments and Principals

The above scenarios assumed trusted services inhabiting mutually aware domains, such as the healthcare domains of different regions or nations. Credentials issued by one service are accepted by another, and the role activation rule enforces the policy established when an agreement was set up between the services. This mechanism is only possible when the issuer of a credential is already known to the service receiving it.

More generally, we may wish to set up a minimal infrastructure, sufficient for a world in which roving computational entities encounter previously unknown, and therefore untrusted, services. Both parties should be able to present checkable credentials which provide evidence of previous successful interactions. This is analogous to the check on a person's credit record that is made before major purchases are authorised.

A certified record of an interaction between a principal and a service could contribute to the evidence of the trustworthiness of both parties. Such certificates might be exchanged and validated before a principal uses a previously unknown service. Each party may then take a calculated risk on whether to proceed: the service risks the client exploiting its resources in unintended ways, or failing to pay an agreed charge; the client risks breach of confidentiality, and poor or partial fulfilment. A formal approach might be for the parties to negotiate a contract before the service is undertaken, and together sign a certificate recording the outcome.

If a certificate issuing and validation (CIV) service already exists in a domain its function might be extended to generate such a certificate. After an interaction subject to contract the CIV service creates an *audit_certificate* which it issues to both parties and validates on request. An *audit_certificate* could be engineered, as described in Sect. 4, to contain sufficient information for the issuing service to be located.

There are of course snags associated with this proposal. It is made on the assumption that CIV services work only on behalf of trustworthy services, and that they are themselves trustworthy. In practice, a client and service might collude to build up a false history of trustworthiness. Similarly, a rogue domain might provide valueless audit certificates, or repudiate those issued to clients who had acted in good faith. The domain of the auditing service for a certificate is a factor that must be taken into account when assessing the risk.

These are deep waters. OASIS RBAC has been designed for deployment in a distributed environment, and allows mutually trusting services to interact securely. Ubiquitous computing, electronic business and mobile agents together promise great benefits, if only a means can be found to bound the risks inherent in computational interaction between unknowns. What is needed is an approach which will allow a trust infrastructure to evolve despite Byzantine behaviour by a minority of the principals.

7 Conclusion

We have brought together the various aspects of OASIS, outlining its model and architecture and discussing engineering issues, with a view to extending its use. We have outlined the security properties of OASIS certificates and have shown how OASIS can be integrated with a standard authentication and encryption infrastructure.

OASIS has important differences from other RBAC schemes which reflect our application-driven, engineering approach. Decentralised role management is essential for widely distributed systems with independently managed components. Parametrised roles are essential for many applications. It is often necessary to express exceptions from generic role-based access and the relationships between parameters may be an essential part of an access control policy. A session based approach provides stronger security than is possible with long-lived roles; a session key can be bound with role membership certificates. Further, a session context allows role activation rules to include environmental constraints. We also monitor role membership conditions throughout a session by implementing OASIS above event-based middleware. This allows us to deactivate roles immediately as dictated by policy. All of these factors contribute to an active security environment.

Appointment subsumes the delegation of privilege and captures the requirement for long-lived credentials. The distinction between session-based role activation and potentially long-lived appointment credentials adds to the security of the system.

We have shown how OASIS is used within and across domains and how principals might be supported in moving to known but remote domains. Service level agreements, with check-back to the issuing service for validation, are the supporting mechanisms. We have shown that anonymous service invocation is possible. Finally we have proposed that audit certificates might be issued as a record of the invocation of a service by a principal. Such certificates provide a distributed record of the histories of services and principals and might form the basis for interaction between mutually unknown parties.

Acknowledgements. We acknowledge the support of the Engineering and Physical Sciences Research Council (EPSRC) under the grant *OASIS Access Control: Implementation and Evaluation*. Members of the Opera research group in the Computer Laboratory made helpful comments on earlier drafts of this paper. We acknowledge many stimulating discussions with our colleagues on the "Secure" proposal, submitted to the EU's Global Computing Initiative.

References

1. J. Bacon, M. Lloyd, and K. Moody. Translating role-based access control policy within context. In *Proceedings of Policy 2001, Policies for Distributed Systems and Networks*, volume 1995 of Lecture Notes in Computer Science, pages 107–119. Springer-Verlag, 2001
2. J. Bacon, K. Moody, J. Bates, R. Hayton, C. Ma, A. McNeil, O. Seidel, and M. Spiteri. Generic support for distributed applications. *IEEE Computer*, pages 68–76, March 2000.
3. E. Barka and R. Sandhu. A role-based delegation model and some extensions. In *23rd National Information Systems Security Conference*, Baltimore, MD, October 2000.
4. E. Barka and R. Sandhu. Framework for role-based delegation models. In *16th Annual Computer Security Applications Conference*, New Orleans, Louisiana, December 2000.
5. M. J. Covington, M. J. Moyer, and M. Ahamad. Generalized role-based access control for securing future applications. In *23rd National Information Systems Security Conference*, Baltimore, MD, October 2000.
6. D. F. Ferraiolo, J. F. Barkley, and D. R. Kuhn. A role-based access control model and reference implementation within a corporate intranet. *ACM Transactions on Information and System Security*, 2(1):34–64, Feb 1999.
7. L. Giuri and P. Iglio. Role templates for content-based access control. In *Second ACM Workshop on Role-Based Access Control*, pages 153–159, Fairfax, Virginia, November 1997.
8. C. Goh and A. Baldwin. Towards a more complete model of role. In *Third ACM Workshop on Role-Based Access Control*, pages 55–61, Fairfax, Virginia, October 1998.

9. R. Hayton, J. Bacon, and K. Moody. OASIS: Access Control in an Open, Distributed Environment. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 3–14, Oakland, CA, May 1998. IEEE.
10. J. Hine, W. Yao, J. Bacon, and K. Moody. An architecture for distributed OASIS services. In *Middleware 2000*, volume 1795 of Lecture Notes in Computer Science, pages 104–120, 2000.
11. J. D. Moffett and E. C. Lupu. The uses of role hierarchies in access control. In *Fourth ACM Workshop on Role-Based Access Control*, pages 153–160, Fairfax, Virginia, October 1999.
12. M. Nyanchama and S. Osborn. Access rights administration in role-based security systems. In J. Biskup, M. Morgernstern, and C. Landwehr, editors, *Database Security VIII: Status and Prospects*, 1995.
13. M. Nyanchama and S. Osborn. The role graph model and conflict of interest. *ACM Transactions on Information and System Security*, 2(1):3–33, Feb 1999.
14. R. Sandhu. Role activation hierarchies. In *Third ACM Workshop on Role-Based Access Control*, pages 33–40, Fairfax, Virginia, October 1998.
15. R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-Based Access Control Models. *Computer*, 29(2):38–47, Feb. 1996.
16. R. T. Simon and M. E. Zurko. Separation of duty in role-based environments. In *10th IEEE Computer Security Foundations Workshop*, pages 183–194, Rockport, Massachusetts, June 1997.
17. W. Yao, K. Moody, J. Bacon. A Model of OASIS Role-Based Access Control and its Support for Active Security. In *Proceedings, Sixth ACM Symposium on Access Control Models and Technologies (SACMAT)*, pages 171–181, Chantilly, VA, May 2001