# A User-Centric Anonymous Authorisation Framework in E-commerce Environment

Richard Au, Harikrishna Vasanta, Kim-Kwang Raymond Choo, Mark Looi
Information Security Research Centre
Queensland University of Technology, Brisbane, AUSTRALIA
{w.au, h.vasanta, k.choo, m.looi}@qut.edu.au

## ABSTRACT

A novel user-centric authorisation framework suitable for e-commerce in an open environment is proposed. The credential-based approach allows a user to gain access rights anonymously from various service providers who may not have pre-existing relationships. Trust establishment is achieved by making use of referrals from external third parties in the form of *Anonymous Attribute Certificates*. The concepts of *One-task Authorisation Key* and *Binding Signature* are proposed to facilitate pseudonymity in authorisation service. These mechanisms enhance user privacy and tackle the problem of scalability in identity-based access control systems.

## 1. INTRODUCTION

The Internet and World Wide Web propel the development of e-commerce and e-business. More and more conventional commercial activities are being reconstructed so that they can make use of the public Internet. Some examples are auctions, shopping, banking, paying bills and other personal business activities. The protection of user privacy is an important factor behind the popularity of e-commerce on the Internet [21]. Users will find it unacceptable that their daily online activities can be freely recorded, linked and traced back to their identities unconditionally [18].

The basic approach to enhance user privacy is to minimise the release of unique personal information, e.g. user identity, whenever possible. This means the security architecture should not solely depend on unique entity identifiers. Without explicit identifiers, it becomes far more difficult for outsiders to track individuals in such systems. Thus anonymity can effectively enhance user privacy and protect users from personal exposure to various security threats. By definition, anonymity is the state of being not identifiable within a set of subjects, the anonymity set [23]. It can be classified into two categories [9]:

- *Full Anonymity*: An anonymous record or transaction is one whose data cannot be associated with a particular individual, either from the data itself or by combining the transaction with other data [12][11], e.g. cash payment.
- *Pseudonymity*: A pseudonymous record or transaction is one that is identified by a pseudonym and the transaction cannot, in the normal course of events, be associated with a particular individual. If a specific piece of additional data is available, then the transaction data can be linked to that party.

Full anonymity and unlinkability may lead to increased misuse by anonymous users and present a security risk that is unacceptable in many applications [16]. Pseudonymity is therefore more suitable to the e-commerce environment. For a pseudonymity mechanism to be effective, there must be some legal and technical protections to assure that the revocation of pseudonymity can only be made by trusted third parties in a controlled manner.

Information about a user's identity can be revealed at two levels in the communication model:

- *At application level*: through the application data or content exchanged;
- *At network level*: via the network addresses of the connecting devices used.

Research on connection anonymity includes: Mix-net [10], Crowds [25] and Onion Routing [24]. In this research, focus is placed on anonymity at the application level only.

Traditionally, access control adopts the framework of subjects, objects and access rights. While authentication establishes the identities of the subjects (network users), authorisation provides users with certain rights to access objects (services and applications). User authentication provides the mechanism by which access control can be implemented on network data, as well as by which auditing and network monitoring are made easier. In certain environments, establishing a user identity automatically provides the user with a set of privileges. To determine the type of access appropriate for a user, the user's identity is compared to an access control list (ACL). If a user's identity appears on the list, the user is granted the access corresponding to that identity. This identity-based authorisation depends on reliable user authentication techniques. The explicit use of unique names or other permanent identifiers makes it difficult to implement anonymity services.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ICEC'04, Sixth International Conference on Electronic Commerce
Edited by: Marijn Janssen, Henk G. Sol, and René W. Wagenaar
Copyright 2004 ACM 1-58113-930-6/04/10…$5.00

In this paper, a novel authorisation framework is proposed that allows the user to generate a short-term cryptographic key and use it as his/her explicit identifier in an activity. The user collects a number of anonymous attribute certificates from different referee servers and submits them to the service provider which makes the authorisation decision based on these referral certificates. As user identity is not a mandatory attribute in these certificates, anonymity can be supported in the authorisation service. The paper is organised as follows. In Section 2, the authorisation framework is introduced with an architectural overview. In Section 3, the features and definition of anonymous attribute certificate are explained. In Section 4, the concept of binding signature is proposed. In Section 5, an overview of the authorisation protocols is described. In Section 6, the mechanism for revocation of anonymity is discussed. In section 7, the mechanism of chained referrals is developed. In section 8, a scenario is given to demonstrate the usefulness of the authorisation framework in e-commerce applications. The paper finishes with the conclusions and future work.

# 2. A NEW COLLABORATED AUTHORISATION FRAMEWORK

Authorisation decisions have so far remained in the hands of end systems, which centrally maintain and enforce access control information without direct involvement by other components in the distributed security infrastructure. Traditional access control systems employ basic mechanisms for identifying legitimate users before granting services by providers. Each user is given a new username and password to be used when accessing the application server. The system administrator needs to create accounts for new users in the registration process. This authorisation mechanism is simple and works well in relatively small and closed systems. However, there are several limitations when applying the mechanism in an open and distributed environment, such as business-to-customer (B2C) e-commerce on the global Internet:

- *Scalability*: The burden of managing such user accounts can be high for large populations of eligible users;
- *Reachability*: It is even more difficult for a service provider to collect and verify information from foreign users in other administrative domains;
- *Efficiency*: Web users often want to have instantaneous access to their targeted Web resources/services, thus a complex user registration process for creating new user records to the database is intolerable.

Suppose there are $m$ service providers and $n$ users in an enterprise. If the entities are independent in this many-to-many relationship, the total number of user accounts in the system will be $m$ times $n$. The information provided by a user is actually repeated in a number of different established accounts.

Collaborated or federated management of authorisation attributes can avoid the problem of updating multiple user accounts as users' information changes.

In the commercial world, a service provider usually makes an authorisation decision based on the acquired information about the user's attributes, such as club memberships, financial assets and various licenses. The user's identity is just one of the attributes, but it is not important at all if that user has no pre-existing relationship with the service provider. Very often, these supporting attributes for authorisation are contributed from some trusted third parties, such as banks, business partners or other well-known business entities. This process makes use of the trust relationships between different organisations and companies. Following this concept of trusteeship, a new dynamic authorisation framework is developed making use of external referee servers, which supply certified attributes about a user on-the-fly, to support the authorisation process at the stage of privilege allocation.

## 2.1 Authorisation without User Authentication

As a broader definition from ISO [28], authorisation is the act of determining whether an authenticated entity has the right to execute an action. Thus, authorisation can be separated into two stages [2]:

- *Stage 1- Privilege Allocation*: Granting rights and/or privileges (authorisation credentials) to a particular entity;
- *Stage 2- Access Control Enforcement*: Using these privileges in combination with access decision rules at the resource to determine if access should be granted to the entity.

It is argued that user authentication can be eliminated in the second stage of authorisation. In [3], an authorisation architecture was designed to allow users to access resources directly and securely with the pre-loaded authorisation credentials (one-shot authorisation token). The mechanism eliminates the repeat of user authentication at every access. In this paper, the research is continued with a focus on how to keep the user anonymous at the stage of privilege allocation in the authorisation service.

## 2.2 Architectural Overview

Referring to Figure 1, there are three main entities in the proposed user-centric authorisation framework.

### *User/Client*

The user makes use of his/her client application to initiate communication with the service provider requesting access to an application or resource. After receiving the requirements for granting access, the user requests appropriate external referral servers to issue some referral credentials.

### *External Referee Servers*

In the real world, a user has many business relationships with commercial or governmental entities. For example, a person has a credit account in a credit card company, a driving license from the transport department and a variety of memberships in different clubs. Different business relationships exist among these organisations and a trust infrastructure can be formed. These external business entities can act as referee servers and provide referrals to their clients upon requests. User identification/authentication may be required before issuing

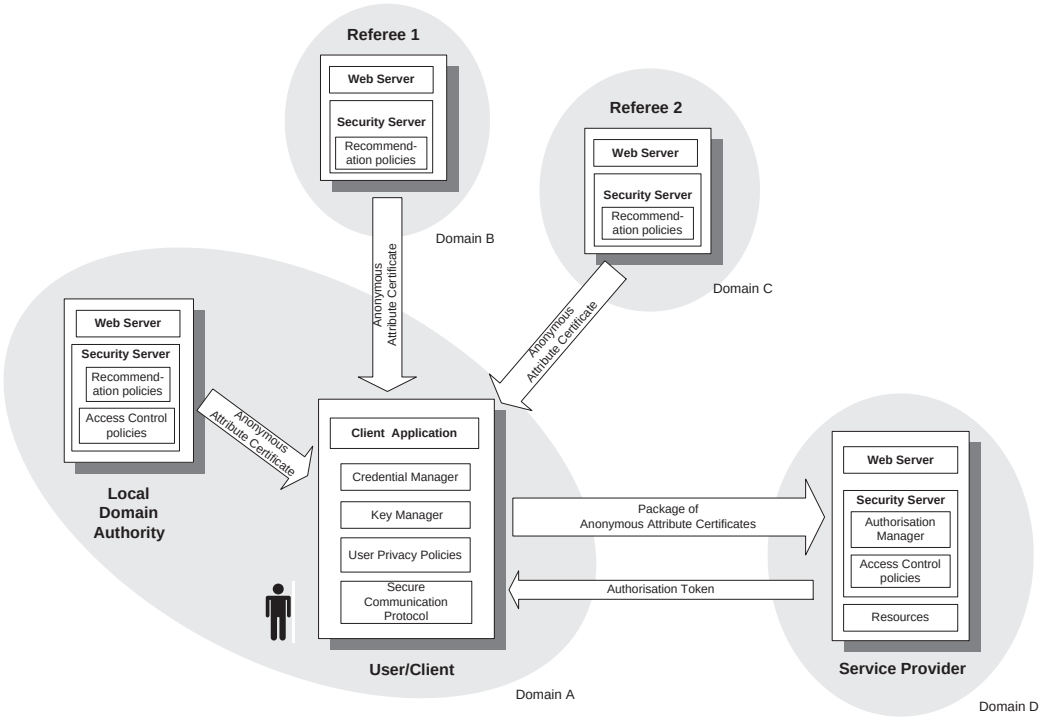


**Figure 1: User-Centric Anonymous Authorisation**

referrals (in the form of anonymous attribute certificates) to the user.

### *Service Provider/Application Server*

The service provider/application server makes the authorisation decision based on the assessment of the referral credentials submitted by the client. Clearly, it is assumed that the service provider has some trust relationships with different referee servers involved and accepts those certified attributes they supply. Realistic examples are the virtual enterprise environment and commercial federation.

## 3. ANONYMOUS ATTRIBUTE CERTIFICATES (AAC)

In the e-commerce environment, different merchants provide their resources/services on the Internet and grant access rights to users in both local and foreign administrative domains. These users range from corporate business partners with strong trust relationships to potential customers without any previous interactions. For B2C e-commerce and other applications on the Internet, we argue that access control systems should focus on authorisation rather than user authentication. Customers prefer to remain anonymous for privacy and security reasons. For service providers, the ultimate goal is to verify the access rights of the users rather than their identities. The mechanism of the *anonymous attribute certificate* is proposed to provide authorisation services without relying on user identification and authentication in such a diversified and distributed environment.

### 3.1 Comparing X.509 and SPKI/SDSI Certificates

In recent years, much work has been done in the establishment of a global Public Key Infrastructure (PKI) which enables the use of public key cryptography for encryption and digital signatures. While PKIs can provide strong authentication services, e-commerce applications require additional use of authorisation services in order to allocate appropriate privileges to different groups of users in a flexible and efficient way. The concept of Privilege Management Infrastructure (PMI) has been introduced for this purpose. The Internet Engineering Task Force (IETF) has established two working groups to operate in these particular fields in the Internet community:

- *PKIX Working Group*: This group is building a PKI based on ITU-T (International Telecommunications Union) Recommendation X.509 [27]. For the purpose of authentication, public key (identity) certificates are used to maintain a strong binding between a globally unique name and a public key. Attribute certificates are designed to bind a user's name and one or more privilege attributes. They can serve a more general purpose and can be used for authorisation.
- *SPKI Working Group*: The aim of this group is to develop a simple and flexible distributed authentication and authorisation infrastructure based on digital certificates and local name spaces [15]. An egalitarian model of trust is used instead of the

global hierarchical infrastructure as in X.509. Two types of certificates are defined: name certificates and authorisation certificates. A name certificate, signed with the private key of the issuer, links a user local name to his public key. The concatenation of the public key of the certificate issuer with the local name represents a SPKI/SDSI unique global identifier. An authorisation certificate grants specific authorisation from the issuer to the subject (or principal) of the certificate. The subject can be a local name, a group or a public key.

To date, X.509 is more widely used than SDSI/SPKI but it has many disadvantages in large scale implementation. The root of the problems is the use of a global namespace, which inherits many management and security problems [14]. SDSI/SPKI follows a decentralised approach for authentication and authorisation and is more flexible for implementing security controls in large-scale distributed systems. However, the weakness of SDSI/SPKI is that it is more difficult to determine the certification path than the hierarchical infrastructure of certification authorities as in X.509. In conclusion, both X.509 and SDSI/SPKI have their advantages and disadvantages. This framework makes use of both approaches.

## 3.2 One-Task Authorisation Key (OTAK)

This paper proposes to use different cryptographic keys separately for authentication and authorisation purposes. When a common key is used by a user for both services, it may seem to be very convenient that the two services can be completed in just one validation operation. However, there are many drawbacks. The use of a common key increases the complexity of key management and weakens the robustness of the protocol. The authentication key usually binds with the user's identity. Since the identity information does not change very often, so the validity period of the authentication key can be rather long. However, the attributes and contents in authorisation may vary frequently, and even be different at every access. The requirements of authorisation keys are different in terms of validity and security.

In this authorisation framework, it is proposed that the user generates a private/public key pair solely for authorisation in each task. Due to the considerably large space used, the authorisation public key can be assumed to be globally unique and suitable for use as an explicit identifier of the client in that single task. Thus it is named the *One-Task Authorisation Key*. Its separation from the identity (authentication) key allows an easier implementation of multiple authorisations to a single user while reserving the property of unlinkability and possibly anonymity.

## 3.3 Binding privilege attributes to One-Task Authorisation Key

In this research, it is assumed that a Public Key Infrastructure (PKI) has been already in position on global scale, i.e. every user has an identity certificate (e.g. X.509 public key certificate or SPKI name certificate) for authentication purposes and that there are proper certification paths to verify the certificates.

Referring to authorisation certificates, following the ideas in SPKI/SDSI [15], it is argued that the user's privilege attributes should not be linked to his/her identity directly. Instead, this paper proposes to bind the privilege attributes to the user's one-task authorisation public key, which is generated by the user and acts as the explicit identifier in the certificate. With reference to the X.509 Attribute Certificate, this work has developed the so-called *Anonymous Attribute Certificate* (refer to Figure 2). Basically, it is similar to the X.509 Attribute Certificate except that the *Holder* field is replaced by a *One-Task Authorisation (Public) Key* field and a new field, named *Anonymity Revocable*, is added to indicate the capability of anonymity revocation upon request. The issuer signature is created by signing all the other fields with the issuer's private key.

The design has the following advantages [4]:

- *Anonymity Support and Enhanced Privacy*: One-task authorisation keys are used directly without reference to the names of the key owners. The users can remain nameless without taking any special measures. It becomes difficult to correlate different tasks/activities of a single user over time because the keys, which are the explicit identifiers in the activities, are randomly scattered. Using separate keys when communicating with different entities, or when performing different unrelated tasks, prevents the easy combination of gathered information for a single entity. In this way, the properties of anonymity and unlinkability are achieved.

- *Cost Effective and Convenience*: The one-task authorisation key has a short lifetime as compared to the authentication key; thus, the security requirements may be lower in many applications. Then the key length can be shorter, which results in faster computation. This feature becomes an advantage for mobile devices with limited computing resources. New authorisation keys are created by the user and the public part can be delivered directly to entities involved. When two entities create a trust relationship, they pass the key directly without relying on the support of trusted third parties. This eliminates the cost of running a Certificate Authority for centralised key management (e.g. key distribution and revocation) as in X.509. The direct authorisation of a key is convenient because the key will also sign the access request. The ownership of the key can be verified directly without a trusted third party. Since the key is used as the identifier, there is no need to use a global naming scheme.

- *Higher Security*: While names of users are not explicitly advertised, attackers must systematically collect intelligence data about the system and analyse it in order to identify individual entities and their activities. As the explicit identifier, the authorisation key is different for each task, it reduces the risk of certain security threats, e.g. eavesdropping and replay. Even if an attacker manages to compromise a key used in one authorisation, he can only disclose information for one activity. Since other activities are independent, the scope of damage to the system may be confined and reduced.

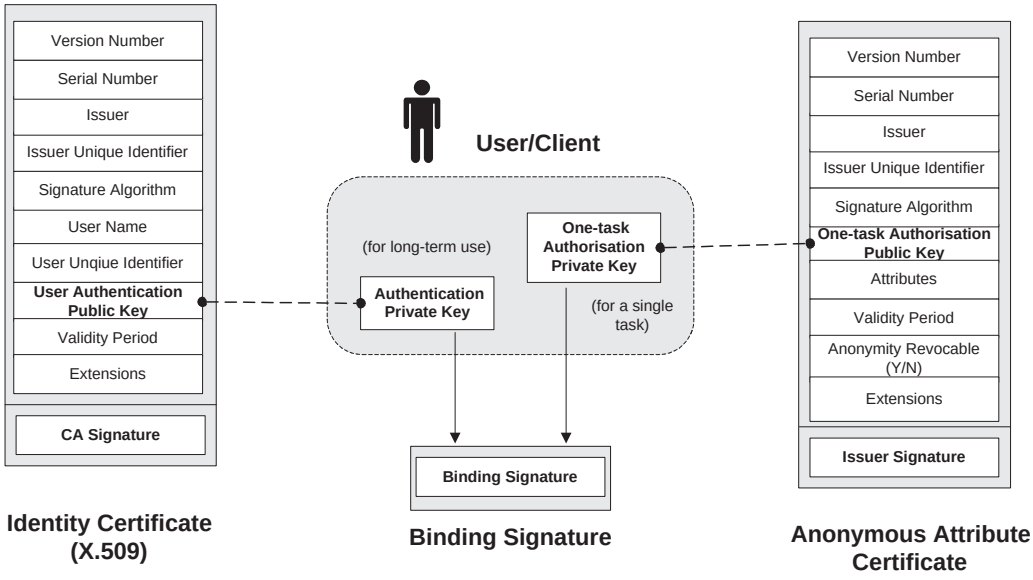


**Figure 2: Certificates for Authentication and Authorisation**

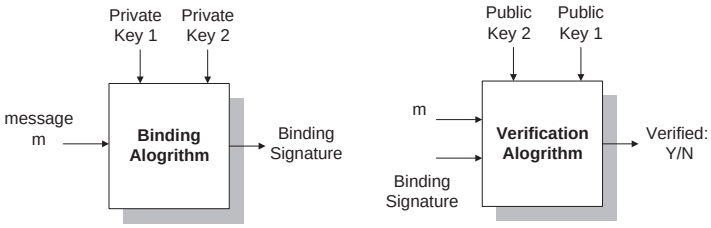


**Figure 3: Binding Signature Algorithm**

# 4. BINDING SIGNATURE

In the X.509 recommendations, a user has an identity certificate that binds his/her identity to his/her authentication public key. The certificate should have relatively long lifetime because the identity does not change very often. The user also has a number of relatively short-lived attribute certificates for authorisation purposes. These attribute certificates can be bound to his/her identity certificate using the common *Holder* field [17].

In our proposed authorisation framework, the two types of certificates are not linked to each other directly as in X.509 or SPKI. Instead, the user's anonymous attribute certificates are linked indirectly to his/her identity certificate through a *Binding Signature* as shown in Figure 2. Only the owner of the two private keys (i.e. authentication and authorisation private keys) corresponding to the public keys in the two certificates, is able to create the binding signature. Thus a binding signature can be used to reveal or prove the identity of the owner of the anonymous attribute certificate, when necessary.

## 4.1 Binding Signature Algorithm

Figure 3 illustrates how a binding signature is created and verified. In the creation process, a binding signature is the unique product of the operation of a cryptographic binding algorithm on a short message, $m$, using two private keys, $K_1^{-1}, K_2^{-1}$. In the verification process, the verification algorithm operates on the binding signature using the two corresponding public keys in correct order, the output is the binary verification result, either positive or negative. A positive result shows that the binding signature is created using the two private keys.

### A Demonstrative Example

For demonstrative purposes in this paper, we can define the binding and verification algorithms using the standard digital signature and encryption schemes as shown below (with reference to the notations in Table 1).

### *Creation of Binding Signature*

Firstly, the user prepares an input message $m$ which is composed of the two public keys, $K_U$ and $OTAK_U$, and a timestamp $T$. Then he/she signs the message with his/her authentication private keys, $K_U^{-1}$. The digital signature is concatenated with the same timestamp $T$ and the result is encrypted with the authorisation private key, $OTAK_U^{-1}$ to produce the binding signature.

$$BINDSIGN_{K_U^{-1}, OTAK_U^{-1}}(m) = ENC_{OTAK_U^{-1}}(SIGN_{K_U^{-1}}(m))$$
where $m = (K_U, OTAK_U, T)$.

### *Verification of Binding Signature*

In the verification process, the binding signature $BINDSIGN_{K_U^{-1}, OTAK_U^{-1}}(m)$ is firstly decrypted using the authorisation public key $OTAK_U$ to get the digital signature $SIGN_{K_U^{-1}}(K_U, OTAK_U, T)$. According to the standard signature scheme used, the digital signature is verified using the authentication public key $K_U$.

$$VERIFY_{K_U}(DEC_{OTAK_U}(BINDSIGN_{K_U^{-1}, OTAK_U^{-1}}(m))) \stackrel{?}{=} true$$

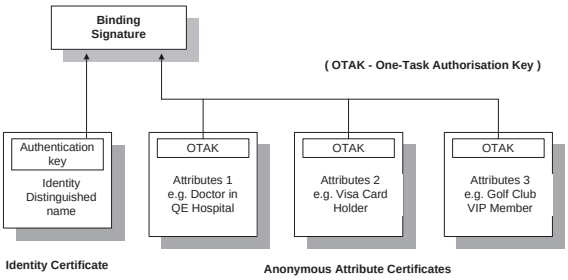


**Figure 4: Linking Multiple Certificates**

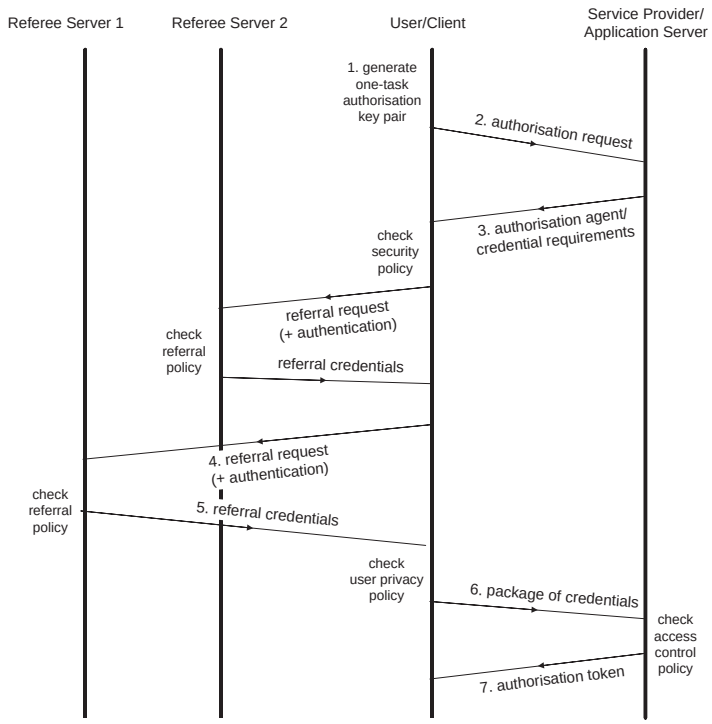


**Figure 5: Credential-based Authorisation Protocol**

If the verification is successful, it shows that the binding signature binds the authorisation public key $OTAK_U$ and the authentication public key $K_U$. Thus the binding signature becomes an indirect link between the identity certificate and the anonymous attribute certificates (see Figure 4). Note that the security of the binding signature relies on the security of the underlying encryption and signature scheme.

# 5. OVERVIEW OF THE ANONYMOUS AUTHORISATION PROTOCOL

In this section, we present an overview of the authorisation protocol, which is used to establish secure communication between the various entities as shown in Figure 5. The notation used throughout the remaining of this paper is introduced in Table 1.

| *Abbreviation* | *Description* |
|---|---|
| U | User/client |
| SPS | Service provider's server |
| ERS | External Referee Server |
| APS | Service Server |
| $Q_{ACC}$ | Access request |
| $Q_{REF}$ | Referral request |
| $AA$ | Authorisation agent |
| $SN$ | Serial Register Number |
| $K_A$ | Authentication public key of A |
| $K_A^{-1}$ | Authentication private key of A |
| $CERT(A)$ | Identity certificate of A |
| $PII(A)$ | Personal Identification Information |
| $OTAK_A$ | One-task authorisation public key of A |
| $OTAK_A^{-1}$ | One-task authorisation private key of A |
| $ENC_{K_A}(m)$ | Encryption of message m with public key of A |
| $DEC_{K_A^{-1}}(m)$ | Decryption of message m with private key of A |
| $SIGN_{K_A^{-1}}(m)$ | Signature (hashing & encrypting) of message m using private key of signer A |
| $VERIFY_{K_A}(S)$ | Verification of signature $S$ using public key of signer A |
| $BINDSIGN (K_{1(A)}^{-1}, K_{2(A)}^{-1})(m)$ | Binding Signature on message m using both the private keys of A, $K_{1(A)}^{-1}$ and $K_{2(A)}^{-1}$ |
| $AAC_n$ | $nth$ Anonymous Attribute Certificate |
| $AT_n$ | $nth$ Authorisation Token |
| $ENV$ | Environment parameters |

**Table 1: Table of Notations**

## 5.1 Credential-based Authorisation Protocol

**1. Generation of Authorisation Key Pair on Client**

The protocol begins by having the client U randomly generating a unique pair of public/private keys ($OTAK_U$, $OTAK_U^{-1}$) for every new task. The one-task authorisation public key $OTAK_U$ is used as his explicit identifier in the task. The one-task authorisation private key $OTAK_U^{-1}$ is stored on the client's secure repository and is used for generating signatures. Since the unique key pair is designed for one time use, this prevents replay attack on the protocol.

**2. Making Access Request to Service Provider**

The client U encrypts the access request $Q_{ACC}$ and his one-task authentication public key $OTAK_U$ with the public key of the application server APS with whom he desires to communicate. The resulting ciphertext is signed by U with his one-task authorisation private key $OTAK_U^{-1}$ to generate a (digital) signature. U then sends the ciphertext together with the signature to APS. Note that we assume the use of a secure channel in this protocol flow to prevent a man-in-the-middle attack, since otherwise, an adversary is able to hijack the entire message and claimed the one-task authentication public key $OTAK_U$ to be his own.

$$\mathsf{U} \longrightarrow \mathsf{APS}: \quad ENC_{K_{APS}}(Q_{ACC}, OTAK_U), \; SIGN_{OTAK_U^{-1}}(ENC_{K_{APS}}(Q_{ACC}, OTAK_U))$$

**3. Download of Authorisation Agent from Service Provider**

Upon receiving the message from U, the application server APS assigns an authorisation agent $AA$ and a unique serial register number $SN$, encrypts and signs with his private signing key $K_{APS}^{-1}$, and sends them to U. Note that we also assume the use of a secure channel in this protocol flow to prevent a man-in-the-middle attack.

$\mathsf{APS} \longrightarrow \mathsf{U}: \quad ENC_{OTAK_U}(AA, SN), SIGN_{K_{APS}^{-1}}(ENC_{OTAK_U}(AA, SN))$

The authorisation agent $AA$ is a trusted representative of the service provider APS. $AA$ carries customised authorisation policies and requirements according to each individual access request. Depending on the architectural design, the authorisation agent $AA$ can be either

- *static* - a simple list of requirements for granting requested privileges; or
- *dynamic* - an executable program that assists the client to acquire referral credentials (e.g. showing address of referee servers and managing the credentials received at a later stage).

## 4. Sending Requests for Referrals

Upon receiving the message from APS, the client U can execute the authorisation agent on the client platform. Depending on the requirements of the service provider, U may need to request for referral credentials from one or more external referee servers. Individual referee servers may or may not require to authenticate the requester (i.e. U) prior to issuing any referral credentials. The user authentication can be conducted using either an identity certificate (e.g. X.509) or other alternative means (e.g. username/password). We also consider anonymous chained referral, where the referee server examines the submitted attribute certificate(s) without the need for user authentication. The three possible scenarios are explained in greater details as follows.

*Scenario 1: User Authentication with Identity Certificate*

In this scenario, we consider user authentication with identity certificate. In Section 4, we propose a technique to generate a binding signature, where the user U has to encrypt some message (i.e. one-task authorisation public key $OTAK_U$ and the referral request $Q_{REF}$) with his one-task authorisation private key $OTAK_U^{-1}$ and his long-term authorisation private key $K_U^{-1}$. Consequently, this binds $OTAK_U^{-1}$ with $K_U^{-1}$ and provides the referee server a mapping of the user's real identity and his corresponding one-task authorisation public key as shown in Figure 3. Note that the binding signature on the referral request generated using U's one-task authorisation private key $OTAK_U^{-1}$ and long-term authorisation private key $K_U^{-1}$ provides non-repudiation.

$\mathsf{U} \longrightarrow \mathsf{ERS}: \quad ENC_{K_{ERS}}(Q_{REF}, OTAK_U, CERT(U)), BINDSIGN_{K_U^{-1}, OTAK_U^{-1}}(Q_{REF}, OTAK_U)$

*Scenario 2: User Authentication without Identity Certificate*

In this scenario, we consider user authentication without the use of identity certificate. Instead, alternative methods of user authentication, such as username/password, biometrics may be used by individual referee servers. The requesting user U will then need to provide some forms of personal identification/authentication information required, $PII_U$. U will then sign $PII_U$ and the referral request $Q_{REF}$ with his one-task authorisation private key $OTAK_U^{-1}$, and then send the generated signature together with the referral request $Q_{REF}$, $PII_U$, and his one-task authorisation public key $OTAK_U$ in encrypted form to ERS.

$\mathsf{U} \longrightarrow \mathsf{ERS}: \quad ENC_{K_{ERS}}(Q_{REF}, PII_U, OTAK_U), SIGN_{OTAK_U^{-1}}(Q_{REF}, PII_U)$

*Scenario 3: Anonymous Referral*

In situation where no user authentication is required prior to issuing the referral credentials, the referee server ERS may make use of other forms of information about the user, such as the IP address of the user's mobile device, his location in the mobile network, in order to establish a sufficient level of trust to grant an anonymous attribute certificate to the user. The user just need to sign the referral request together with his one-task authorisation key and them send them along to ERS.

$\mathsf{U} \longrightarrow \mathsf{ERS}: \quad ENC_{K_{ERS}}(Q_{REF}, OTAK_U), SIGN_{OTAK_U^{-1}}(Q_{REF})$

In the scenario of chained referral, where the user submits some third party referral credentials to the designated referee server in the referral request. The referee server assesses the user based on the attributes in the anonymous attribute certificates submitted. We shall discuss anonymous chained referral in greater detail in Section 7.

## 5. Sending Referral Credentials to Client

Upon successful verification of the received signature, an anonymous attribute certificate $AAC$ will be generated by the referee server ERS and sent to U.

$\mathsf{ERS} \longrightarrow \mathsf{U}: \quad ENC_{OTAK_U}(AAC)$

## 6. Submission of referral credentials to Service Provider

User U can decrypt the received message from the referee server ERS using his/her authorisation private key $OTAK_U^{-1}$ to obtain the anonymous attribute certificate. Then U can verify the signature of the issuer in the certificate to determine if it is generated by ERS. If the verification is successful, U can proceed to check if $AAC$ fulfills the required privacy and security policies.

Once U acquires sufficient anonymous attribute certificates from different referee servers to satisfy the authorisation requirements of the application server, these referral credentials together with U's serial register number will be encrypted with the public key of the service provider APS, which will then be signed with the one-task authentication private key of U and send to the service server APS. Non-repudiation is provided by the signature generated.

$\mathsf{U} \longrightarrow \mathsf{APS}: \quad ENC_{K_{APS}}(OTAK_U, AAC_1, AAC_2, \ldots, AAC_n, SN), SIGN_{OTAK_U^{-1}}(ENC_{K_{APS}}(OTAK_U, AAC_1, AAC_2, \ldots, AAC_n, SN))$

## 7. Sending Authorisation Decision to Client

$VERIFY_{OTAK_U}(SIGN_{OTAK_U^{-1}}(ENC_{K_{APS}}(AAC_1, AAC_2, \ldots, AAC_n, SN)) \stackrel{?}{=} true$

The service provider (i.e., application server) APS can verify the received signature to determine if it is signed by

U (using his one-task authorisation key) and also that the anonymous attribute certificates in the verified signature are issued by the respective referee servers (i.e., assuming the PKI is in position, APS can obtain the public keys or identity certificate of the respective ERS, which can be used to verify the validity of the anonymous attribute certificates received). Upon successful completion of the verification process and the fulfillment of all requirements, APS will be able to make the decision whether to grant the requested access privileges to U. If APS decides to grant the requested access privileges to U, APS will send the authorisation token $AT_n$ to U as shown below.

$$\mathsf{APS} \longrightarrow \mathsf{U}: \quad ENC_{OTAK_U}(SN, AT_1), \; SIGN_{K_{APS}^{-1}}(ENC_{OTAK_U}(SN, AT_1))$$

There are several possible mechanisms for access control enforcement. In our proposed scheme, the application server creates an appropriate authorisation token specifying the privileges granted and then sends it to the client U. The mechanism of one-shot authorisation token due to Au *et al.* [3] can be deployed in our proposed scheme, where the one-task authorisation public key can be used as the unique identifier in the authorisation token. This implies anonymity for the entire authorisation process.

**8. Access of Service on Application Server**

$$\mathsf{U} \longrightarrow \mathsf{APS}: \quad ENC_{K_{APS}}(OTAK_U, AT_1, ENV), \; SIGN_{OTAK_U^{-1}}(AT_1))$$

For access control enforcement, U can submit the authorisation token $AT_n$ to the service provider in order to gain direct access. After verifying the privileges specified in the authorisation token, the application server can randomly generates a session key using some secure key generation algorithm and assigns this session key to U in order to establish a secure communication. From perspective of the application server APS, the access process remains the same for both local and external users from other domains. The process keeps track of U by his/her one-task authorisation public key, $OTAK_U$, bound in the authorisation token and not by U's identity.

## 5.2 Security of the Protocol

The primitives used in the credential-based authorisation protocol are the notions of a secure encryption scheme [5, 13, 19] and a secure signature scheme [6, 7, 8, 20]. Both notions are now relatively standard. For the security of the underlying encryption scheme, we consider the standard definitions of *indistinguishability of encryptions* (IND) due to Goldwasser and Micali [19] and *chosen-plaintext attack* (CPA). For the security of the underlying signature scheme, we consider the standard definition of *adaptive chosen-message attack* (ACMA) due to Goldwasser, Micali, and Rivest [20]. The credential-based authorisation protocol is secure if both the underlying signature scheme and encryption scheme are secure against ACMA and IND-CPA respectively.

# 6. REVOCATION OF ANONYMITY

Revocation is a mechanism for controlled anonymity, which reconciles groups with conflicts of interest: users who demand privacy, and law enforcement agencies. Anonymity revocation should be provided with the approval of a trustee - a trusted third party who should not be involved in the anonymity service. The identity can be revealed only in some well-defined circumstances. A controlled anonymity system provides a backdoor through which an identity can be traced. In order to revoke the anonymity, proper transition logs are required in both the referee servers and application server. In an authorisation process, the service provider holds the mapping of the user's authorisation public key and the privileges granted. It also has the list of referee servers which issue the anonymous attribute certificates. In practice, probably one or more referee servers require the revelation of the user's identity in the user authentication process when preparing credentials. In that case, the referee servers are able to map the authorisation public key to the user identity. If anonymous attribute certificates are created with anonymity being revocable, the *Anonymity Revocable* field in those certificates are filled with "Y". To guarantee the capability of anonymity revocation, the service provider can request the user to acquire at least one anonymous attribute certificate with anonymity revocable from the referees.

With the collusion of both the application server and referee server, the user identity in a particular task can be revealed. As a requirement in the authorisation framework, the referee servers are trusted not to disclose user's personally identifiable information and the mapping of his/her identity to the authorisation key improperly. In practice, some implementation of privacy management schemes, similar to Platform for Enterprise Privacy Practices (E-P3P) [22, 1] and Platform for Privacy Preferences (P3P) [26], can be developed for the protection of users against misuse or unauthorised disclosure of personal data including identities. Governmental legislation can be used to regulate and enforce the practice. In the commercial sector, organisations taking up the role of referee will not bleach the regulations easily because of the concern of business reputation and integrity. The user also has the liberty to use those referee servers that he/she trusts.

# 7. ANONYMOUS CHAINED REFERRALS

In this section, we propose an alternative referral request mechanism, *chained referral* that is based on some third-party anonymous attribute certificates submitted by the user. Anonymous chained referral is a suitable candidate for situations where anonymity of requesting user is of concern.

In responding to an referral request, the referee servers may use available information in the anonymous attribute certificates submitted by the requesting user without the need for user authentication. The requesting user U can reuse the original one-task authorisation public key $OTAK_U$ which is binding with a number of anonymous attribute certificates $AAC_1, \ldots, AAC_n$ as shown in Figure 4. U submits the referral request $Q_{REF}$ together with the anonymous attribute certificates $AAC_1, \ldots, AAC_n$ and his signature $SIGN_{OTAK_U^{-1}}(Q_{REF})$ to the designated referral server ERS requesting a new anonymous attribute certificate.

$$\mathsf{U} \longrightarrow \mathsf{ERS}: \quad ENC_{K_{ERS}}(Q_{REF}, OTAK_U, AAC_1, \ldots, AAC_n), \; SIGN_{OTAK_U^{-1}}(Q_{REF})$$

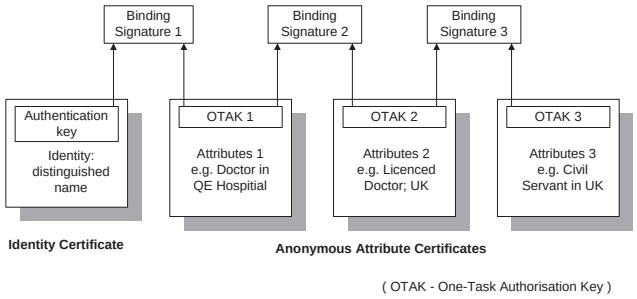


**Figure 6: Chained Certificates**

Alternatively, U can generate a new one-task authorisation key pair $(OTAK_{U_{New}}, OTAK^{-1}_{U_{New}})$ and use this newly generated one-task authorisation public key $OTAK_{U_{New}}$ as the unique identifier for the new anonymous attribute certificate. The newly generated and the original one-task authorisation private keys (i.e. $OTAK^{-1}_{U_{New}}, OTAK^{-1}_{U}$ respectively) are then bound together by a binding signature as shown in Figure 6.

$$\mathsf{U} \longrightarrow \mathsf{ERS}: \quad ENC_{K_{ERS}}(Q_{REF}, OTAK_U, OTAK_{U_{New}}, AAC_1, .., AAC_n), \; BINDSIGN_{OTAK^{-1}_U, OTAK^{-1}_{U_{New}}}(Q_{REF}, OTAK_{U_{New}})$$

In many cases, the degree of anonymity can be increased using anonymous chained referrals. As an example, suppose John has generated two one-task authorisation keys, $OTAK_1$ and $OTAK_2$. He uses $OTAK_1$ as the identifier requesting the university registrar to issue an anonymous attribute certificate A with the attribute *Student of London University*. Obviously he needs to disclose his identity and details in this application. Then, without revealing his identity to the International Student Authority, John submits certificate A and key $OTAK_2$ requesting another anonymous attribute certificate B with the attribute *University Student in UK*. Certificate B used with key $OTAK_2$ as his identifier provides a higher degree of anonymity because of the larger anonymity set.

## 8. A DEMONSTRATIVE SCENARIO

Peter intends to purchase a house and needs to acquire a mortgage for it. He tries to apply for loan pre-approvals from a few financial institutions. He can then choose the one with the best terms for completing the settlement process with the seller. As is common with any mortgage requirement, the user needs to prove his repayment capacity for the loan requested. The financial institutions usually require the user's employment details, banking history, credit rating and proof of other incomes and assets. At this preliminary stage of pre-approval application, Peter does not want to reveal his identity to the financial institutions.

Peter is currently employed full-time as a lecturer at Queensland University of Technology (QUT), banks with MiniBank, and has a superannuation account with one of the superannuation service providers from QUT, UniSuper. UniSuper and ISRC bank are corporate companies. Peter also has a credit account in a credit card company, CreditUnion.

Peter now applies for a loan pre-approval from ISRC bank. He wants to be anonymous and does not want to reveal all of his financial details, especially his superannuation account information in UniSuper, to ISRC bank before the loan has been approved.

Peter generates two private/public key pairs, $\{OTAK_1{}^{-1}, OTAK_1\}$ and $\{OTAK_2{}^{-1}, OTAK_2\}$, for authentication. Using $\text{OTAK}_1$ as his identifier, he requests MiniBank and CreditUnion to provide the anonymous attribute certificates certifying his bank accounts and credit limit. He then uses $OTAK_2$ to request UniSuper to issue a anonymous attribute certificate certifying his superannuation account. Next, Peter uses $OTAK_1$, the binding signature of the two keys ($OTAK_1$ and $OTAK_2$) and the anonymous credential provided by UniSuper to request the employer QUT to provide a chained referral ascertaining that he has an existing superannuation deposit in UniSuper. Thus, QUT provides two anonymous credential to Peter. One is the anonymous attribute certificate certifies that he is employed at QUT with a certain salary. The other one asserts the superannuation he holds in UniSuper.

Using $OTAK_1$ as his identifier, Peter submits these anonymous attribute certificates to ISRC bank. ISRC bank can verify that:

- Peter has financial power for repaying the debt using deposits in MiniBank;
- He has a good credit limit in CreditUnion for emergency use;
- He is an employee of QUT with a stable income and a superannuation account in UniSuper.

Based on the information provided, ISRC bank can decide whether to grant the loan pre-approval or not. Note that even if ISRC bank and UniSuper collude, they cannot reveal the user identity because different identifiers (one-task authorisation keys) are used for the two companies. Only QUT has information about the binding signature of the two keys.

If the requirements for the loan are satisfied, ISRC bank will issue a loan pre-approval in the form of an authorisation token to Peter. If Peter later decides to accept the loan, he may submit the pre-approval and will need to reveal his identity in further procedures.

## 9. CONCLUSIONS AND FUTURE WORK

Using the user-centric approach, this paper has developed a new authorisation framework for open distributed systems. The one-task authorisation key is proposed to support anonymity and enhance the scalability of access control systems. The anonymous attribute certificate is designed to provide dynamic authorisation suitable for applications in open environments, such as e-commerce on the Internet. The proposed authorisation scheme assures that the user receives the required service with his/her privacy protected from the service provider.

As further work, several new challenges are particularly worth research efforts:

- *Standardisation of anonymous attribute certificate* - While multiple business parties across different security domains are involved, the design of the attributes and other fields in the anonymous attribute certificates should provide effective translation of policies and management of trust between these related communities.

- *Security architecture and protocols* - The design of efficient, flexible and secure communications between different entities in the architecture is crucial in providing authorisation services.

## 10. REFERENCES


[1] P. Ashley, S. Hada, and G. Karjoth. E-P3P Privacy Policies and Privacy Authorisation. In *Proceedings of ACM Workshop on Privacy in Electronic Society*, pages 103–109, 2002.

[2] P. Ashley and M. Vandenwauver. *Practical Intranet Security : An Overview of the State of the Art and Available Technolgies*. Kluwer Academic Publishers, 1999.

[3] R. Au, M. Looi, and P. Ashley. Cross Domain One-Shot Authorisation using Smart Card. In *Proceedings of 7th ACM Conference on Computer and Communication Security (CCS' 2000)*, pages 220–227, 2000.

[4] T. Aura and C. Ellison. Privacy and Accountability in Certificate Systems. In *Helsinki University of Technology Laboratory for Theoretical Computer Science Research Report 61*, 2000.

[5] M. Bellare, A. Boldyreva, and S. Micali. Public-key Encryption in a Multi-User Setting: Security Proofs and Improvements. In *Advances in Cryptology – Eurocrypt*, pages 259 – 274. Springer-Verlag, 2000. Lecture Notes in Computer Science Volume 1807.

[6] M. Bellare, R. Canetti, and H. Krawczyk. Keying Hash Functions for Message Authentication. In *Advances in Cryptology - Crypto 96*, pages 1 – 15. Springer-Verlag, 1996. Lecture Notes in Computer Science Volume 1109.

[7] M. Bellare, R. Guerin, and H. Krawczyk. XOR MACs: New Methods For Message Authentication Using Finite Pseudorandom Functions. In *16th Annual International Cryptology Conference on Advances in Cryptology*, pages 15 – 28. Springer-Verlag, 1995. Lecture Notes in Computer Science Volume 963.

[8] M. Bellare and Phillip Rogaway. Provably Secure Session Key Distribution: The Three Party Case. In *27th ACM Symposium on the Theory of Computing*, pages 57–66. ACM Press, 1995.

[9] L. Cardelli. Abstractions for Mobile Computation. In *Secure Internet Programming: Security Issues for Mobile and Distributed Objects, LNCS 1603*, pages 51–79. Springer Verlag, 1999.

[10] D. Chaum. Untraceable Electronic Mail, Return addresses and digital Pseudonyms. In *Communications of the ACM*, volume 24, pages 84–88, 1981.

[11] D. Chaum. Security Without Identification: Transaction Systems to Make Big Brother Obsolete. In *Communications of the ACM*, volume 28, pages 1030–1044, 1985.

[12] R. Clarke. Identified, Anonymous and Pseudonymous Transactions: The Spectrum of Choice. In *Proceedings of User Identification & Privacy Protection Conference*, 1999.

[13] R. Cramer and V. Shoup. Design and Analysis of Practical Public-Key Encryption Schemes Secure Against Adaptive Chosen Ciphertext Attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

[14] C. Ellison. Improvements on Conventional PKI Wisdom. In *Proceedings of the First Annual PKI Research Workshop*, pages 165–175, 2003.

[15] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. SPKI Certificate Theory. In *RFC 2693, Internet Engineering Task Force*, 1999.

[16] C. Farkas, G. Ziegler, A. Meretei, and A. Lorincz. Anonymity and Accountability in Self-organising Electronic Communities. In *Proceedings of ACM Workshop on Privacy in Electronic Society*, pages 81–90, 2002.

[17] S. Farrell and R. Housley. An Internet Attribute Certificate for Authorisation. In *RFC 3281, Internet Engineering Task Force*, 2002.

[18] B. Friedman, P.H. Khan, and D.C. Howe. Trust Online. In *Communications of the ACM*, volume 43, pages 34–40, 2000.

[19] S. Goldwasser and S. Micali. Probabilisitic Encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.

[20] S. Goldwasser, S. Micali, and R. L. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. *SIAM Journal on Computing*, 17(2):281 – 308, 1988.

[21] D.L. Hoffman, T.P. Novak, and T. Peralta. Building Consumer Trust Online. In *Communications of the ACM*, volume 42, pages 80–85, 1999.

[22] G. Karjoth, M. Schunter, and M. Waidner. Platform for Enterprise Privacy Practices - Privacy-enabled Management of Customer Data. In *Proceedings of the Privacy Enhancing Technologies Conference*, volume LNCS 2482, pages 69–84, 2003.

[23] A. Pfitzmann and M. Kohntopp. Anonymity, Unobservability and Pseudonymity - A proposal for Terminology. In *Proceedings of the workshop on Design issues in anonymity and unobservability, LNCS 2009, Springer-Verlag*, 2000.

[24] M.G. Reed, P. F. Syverson, and D. Goldschlag. Anonymous Connections and Onion Routing. In *IEEE Journal on Selected Areas in Communications*, volume 16, pages 482–494, 1998.

[25] M.K. Reiter and A.D. Rubin. Crowds: Anonymity for Web Transactions. In *ACM Transactions on Information and System Security*, volume 1, pages 66–92, 1998.

[26] W3C. Platform for Privacy Preferences. In *URL: www.w3.org/P3P*.

[27] ITU-T Recommendation X.509. In *Information technology - Open systems interconnection - the directory: Public-key and attribute certificate frameworks*, 2000.

[28] ITU-T Recommendation X.812. *ISO 10181-3: Information Technology - Open Systems Interconnection - Security Frameworks for Open Systems : Access Control Framework*. International Organisation for Standarisation, 1996.