ABSTRACT: We describe a role based, policy driven, Privilege Management Infrastructure, in which the authorisation tokens are roles held as X.509 attribute certificate stored in LDAP directories. Users are assigned roles, and roles are granted privileges. The authorisation policy says which roles and attribute certificates are to be trusted, and what access rights are to be granted to each role. The authorisation policy is written in XML by the service provider. The access control decision function (ADF) is a policy driven engine that makes the granted or denied access decisions. The ADF is written in Java, and is completely generic so that it can be built into any e-construction application. We have currently built it into two construction applications, E-tendering and E-planning, and these are described.

1. INTRODUCTION

Secure electronic transactions require both authentication and authorization. Authentication is the process of determining who the communicating parties really are. With weak or no authentication, callers can pretend to be someone else, and this is masquerading. Authorisation is the process of determining what a party is allowed to do. With weak or no authorisation, a caller may gain access to resources (s)he is not allowed to access. Thus both authentication and authorisation are important in electronic transactions to make sure that a) only known parties are allowed to participate and b) each party can only perform the transactions they are authorised to perform.

The more valuable or risky the transaction, the stronger the security needs to be. The Internet is inherently a risky medium for conducting electronic transactions, and so security needs to be relatively strong. If multi million pound contracts, or confidential plans for new construction projects are to be transacted via the Internet, then security needs to be very strong. Typically construction companies have shied away from using the Internet to conduct the confidential or valuable parts of their business processes due to their security concerns, preferring rather to use the Internet for non-confidential
everyday traffic such as email and Web marketing. Thus the Internet is not being used to its full potential.

This paper shows how two organisations, the City of Salford and the Comune di Bologna, have overcome their security concerns of the Internet, and are using it for valuable parts of the construction process. Salford City council is building an E-tendering system, whereby electronic tenders can be submitted digitally signed and encrypted via the Internet. Only authorized companies are allowed to submit tenders up to the close of tender, whilst only authorized city officials are allowed to retrieve the tenders after the close of tender. The Comune di Bologna is providing access to city plans via the Internet to qualified architects, and is allowing them to submit their revised plans back to the city for planning approval. Eventually approved plans will be returned electronically to the architects.

Both cities are using the same authentication technology, namely public key cryptography and digital signatures, to authenticate the users, and both are using the same generic Privilege Management Infrastructure (PMI) to provide the authorisation half of their security. The authorisation PMI was built and piloted under the sponsorship of the EC PrivilEge and Role Management Infrastructure Standards validation (PERMIS) project (see http://www.permis.org), and it can be used for any part of the construction process where strong authorisation is a requirement. The authentication and authorisation components described in this paper act independently of each other, and can be performed by different computer processes and managed by different people in the organisation.

2. PUBLIC KEY CRYPTOGRAPHY

Public key cryptography is based on mathematical formulae in which two different variables are involved. One variable, the encryption key, is used when plain data is converted into encrypted data. The other variable, the decryption key, is used when encrypted data is converted back into plain data. Both the encryption key and decryption key are mathematically related to each other, but it is not possible to determine the other key when only one key is known. Public key cryptography gets its name because one of the keys is usually made public whilst the other key is kept private to the key pair holder. Thus if the encryption key is made public, anyone with this key can encrypt a message, but only the key pair holder can decrypt it using his private key. Conversely, if the decryption key is made public, only the key pair holder can encrypt a message (using his private key), but anyone who receives the message can decrypt it using the corresponding public decryption key. Whilst the latter may not seem very secure, it is in fact the basis of digital signatures, since any person receiving such an encrypted message knows that it can only have come from the person holding the corresponding private encryption key. It therefore follows that when using public key cryptography, the holder of a private key must ensure that (s)he is the only person able to use it, otherwise masquerade or loss of confidentiality is possible. Another requirement of public key cryptography is that the binding of the name of the key pair holder to the public key must also be secure; otherwise again masquerade or loss of confidentiality can ensue. For example if someone can be duped into believing that Jane’s public key actually belongs to Peter then Jane can pretend to be Peter when digitally signing messages, or can decrypt confidential messages intended for Peter. Public keys are therefore stored as public key certificates, which are digital constructs containing the name of the key pair holder, their public key and the name of the certification authority (CA) testifying to this binding. The whole construct is digitally signed by the CA.

The X.509 standard [X509, 2000] defines a standard format for public key certificates and specifies the components needed to create a Public Key Infrastructure (PKI). The X.509 standard thus provides the foundation for delivering strong authentication (based on digital signatures and public key certificates) on the Internet. The well-known Secure Sockets Layer protocol [SSL, 1996] uses X.509 certificates to authenticate web servers on the Internet. There are many companies offering X.509 PKI products and services today, and so we will not discuss them further. Readers wishing to know more
about PKIs can consult texts such as [Adams, 1999] and [Austin, 2000]. The main focus of this paper is on secure authorisation.

3. PRIVILEGE MANAGEMENT

In a paper based privilege management system, a resource owner (e.g. the Financial Director of a company, or the Computer Centre Manager) will typically sign a form to say that a particular person (the privilege holder) is allowed to use a particular resource in a particular way. E.g. The Financial Director may say that Mr Smith can sign orders up to the value of so many thousand Euros. In a role based scheme, Mr Smith may be allocated the role of Department Manager, and Department Managers may be authorised to sign orders up to so many thousand Euros.

Paper based systems may also support delegation of authority, whereby a privilege holder is allowed to delegate the use of the resources currently under his control, to one or more other people. E.g. Mr Smith authorises a project manager to sign orders for his project up to a pre-determined sum.

When the privilege is asserted (or exercised) by the privilege holder, the privilege verifier needs to be sure that the privilege really does belong to the user claiming the privileges, and that the privilege conforms to the company policy. For example, a purchase order clerk (privilege verifier) will need to check such things as

i) is this user allowed to sign Internal Requisitions

ii) who gave permission for this user to sign Internal Requisitions

iii) is this Internal Requisition within the limits of what he is allowed to purchase

iv) are all the signatures valid or are they forgeries

v) has the user’s privilege been withdrawn since the last time I did the checking

Traditionally computer systems have implemented privilege management as application based access control lists, saying which users are allowed which types of access to which data items. Each application holds its own access control lists. This system is difficult to manage, and often results in users keeping their access rights long after they should have been withdrawn, often long after they have left the organisation. A better application independent authorisation system is needed – one that more closely resembles existing organisation authorisation mechanisms. A Privilege Management Infrastructure (PMI) based on attribute certificates is one answer to this.

An attribute certificate (AC) is a data structure that confers privileges (in the form of attributes) on the holder, and the whole construct is digitally signed by the issuing authority, thereby providing authenticity, integrity and tamper resistance to the AC. For example, an AC could be an electronic airline frequent flyer card in which the attribute type is FrequentFlyer, the attribute value is Gold Member, and the AC is signed by the airline issuing the card; or the AC could be a credit card in which the attribute type is CreditCard, the attribute value is the type of card (Visa, Amex etc.) and the AC is signed by the credit card issuer. In a role based access control scheme, the attribute type could be Role, the attribute value could be Project Manager, and the AC could be signed by Mr Smith, the Departmental Manager. In a construction project, the Role attribute could have a value of Construction Site Supervisor, or Purchaser, or Foreman etc, and the AC could be signed by the person who is authorised to allocate these roles.

The latest 2000 version of the X.509 standard now defines the data structures, infrastructure components, and processing mechanisms that contribute to the creation of an electronic Privilege Management Infrastructure based on attribute certificates. The EC PERMIS project was set up to validate that the X.509 standard can indeed be used to manage role based electronic privileges, and that such a system, once built, can be used by many different organisations for many different types of
electronic transaction. In other words, that the standard is completely generic and can be used to support a wide variety of applications and organisations. Two of the applications built under the PERMIS project to prove that this is so, are construction related; namely electronic tendering at Salford City Council, and electronic planning at the Comune di Bologna, Italy. (The other applications currently being piloted are electronic prescription processing at the University of Salford, and electronic access to a car parking fine database in the City of Barcelona.)

4. THE PERMIS PMI

The PERMIS PMI is a standards based, policy driven, role based access control infrastructure. The components of the infrastructure are shown in Figure 1. All access control decisions are governed by an authorization policy, which is written by the administrator of the target resource(s) being accessed. The policy states which roles are to be granted which types of access to which target resources. Authorization policies are written in XML according to a DTD that has been published at XML.ORG (see www.xml.org). A full description of the policy can be found in [Chadwick & Otenko, 2002a].

Users are given roles by appropriate authorities, e.g. their employer and/or professional bodies, and we provide a software tool, the Privilege Allocator that aids the issuer in this task. Privilege management can thus be delegated and distributed to various authorities throughout the world. For example, the role of architect can be assigned by RIBA, or the American Institute of Architects, or the Royal Institution of Naval Architects (RINA) etc. and they could issue the appropriate attribute certificates to professionally qualified people. Likewise each company can undertake organisational role allocation to its employees. All the roles can be stored as X.509 ACs in one or more LDAP directories [LDAP, 1997] thereby making them globally available over the Internet. Thus RIBA could publish the ACs for the architects it has accredited, whilst RINA could do likewise, and the PERMIS API would be able to access them all. Since all X.509 ACs are digitally signed there is no authorisation risk in making them publicly available, since they are tamperproof and not forgeable.

Each authorization policy states which authorities it trusts to issue ACs, so ACs can be in existence that are unusable in practice as no-one trusts them, for example, architect roles issued by the Architects School for Dummies. The authorization policy is digitally signed by whoever is acting as the Source of Authority (SOA) for the targets being accessed, thus guaranteeing its authenticity, and ensuring that all access control decisions are based on a policy issued by the trusted SOA. The policy is stored in an LDAP directory after its creation, making it available to the PERMIS API at runtime.
The access control infrastructure is designed according to the ISO 10181 Access control framework [ISO, 1996]. It consists of an application dependent Access Control Enforcement Function (AEF) and an application independent Access Control Decision Function (ADF). All access decisions are made by the ADF according to the authorization policy that is retrieved from the LDAP directory at initialisation time. Thus all targets in a domain can be controlled by the same policy, which will significantly simplify the task of access control management, and ensure authorisation consistency throughout the domain. It is the responsibility of each application AEF writer to correctly authenticate each user before calling the ADF, but the PERMIS ADF is authentication agnostic i.e. any authentication mechanism can be used as determined by the application.

The ADF is written in Java, and the PERMIS API used to call the ADF is simple to use, comprising of just 3 methods (GetCreds, Decision and Finalize) and a constructor.

5. THE PERMIS API

A constructor prepares the PERMIS API implementation object ready to operate according to a given policy signed by the trusted SOA. After completion the PERMIS API will have read in the correct policy AC (each policy is uniquely identified by an object identifier (OID) so that an SOA can have multiple policies for activation under different conditions), validated that the signature on the policy AC was created by the SOA, parsed the policy and validated that it is syntactically correct. The parameters passed to the constructor contain the SOA name, the policy OID, the URLs of the LDAP repositories to be contacted and the PKI object that is to be used to validate signatures on the retrieved ACs.

The Finalize method informs the PERMIS API implementation object that the Policy and PMI settings are not to be used anymore. This method is called by gc (JVM's Garbage Collector) when the PERMIS API is no longer referenced by the AEF. This will be the case when the application is closing down, but it can also be the case when the application is directed to dynamically change policies mid stream (for example when the SOA wishes to switch a target into testing mode from operational mode of use).
GetCreds is a family of methods used to retrieve a user’s privileges. The methods are differentiated by whether they are passed the name of the user or the public key certificate of the user, whether they are called in pull mode or push mode and whether a session time out is required or not. Pull mode retrieves the user’s role ACs from the set of LDAP directories provided at construction time. Push mode is provided with the set of role ACs to use by the AEF. In either mode, the role ACs are validated, and the valid credentials/roles (according to the policy at the time this method is called) are extracted. These credentials will be used for decision-making until Finalize is invoked or the session times out.

The Decision method is passed a pre-existing set of credentials, a description of an action the user wants to perform, the name of the target the user wishes to access and other contextual parameters e.g. the time of day. Decision returns a Boolean reply (True for Granted, False for Denied) based on the policy.

GetCreds is comparatively slow to execute because of network communications to the LDAP repositories, and using public key cryptography to validate the role ACs. Decision on the other hand executes much faster. For this reason GetCreds and Decision are separate methods rather than one combined method. GetCreds should be called during the process of user authentication, whilst Decision should be called for each action the user wishes to invoke.

The PERMIS API has been incorporated into E-tendering and E-planning applications, and the following sections describe these applications, and the roles that are assigned and the policies that have been defined for granting access to the various targets in the applications. A fuller description of the PERMIS PMI can be found in [Chadwick & Otenko, 2002b].

6. E-TENDERING IN THE CITY OF SALFORD

In present times pressure is being put on the Public Sector to become more cost effective and customer centric. To fulfil this social requirement governments has spotted a clear opportunity in providing e-based solutions and have committed to put many of their services on-line. The City of Salford identified tendering as a process that would benefit from the use of modern technology.

The Salford E-tendering application fully automates the tendering process. From the initial advertisement of the tender (when the tender is ‘called’) to the opening of the submissions, the whole process is electronic, thus eliminating the massive paper trails and providing a more efficient and cost effective service.

The features include:

- Tender documents are held on the Authorities Internet site.
- Invitations to tender are posted on the Website or sent via email to prospective suppliers.
- Suppliers submit binding digitally signed contracts electronically.
- Historic information on tenders and quotations is captured.
- The supplier database is updated by the suppliers themselves
- Suppliers can track the progress of their tenders.

6.1. Process

The application centres on the internal purchasing / procurement section. This section receives an email from an internal party identifying the requirement to begin a tendering process. The internal purchasing / procurement section in conjunction with the issuing directorate produce a needs analysis report and allocate a TenderID to the tender. The Tender Officer co-ordinates with the issuing directorate to establish the estimated value of the tender, the start date of the tender process, its end date and identifies the lead member roles and tender officer roles that are able to open the received
tender submissions. This information forms the basis of the authorisation policy and the attribute certificates that will be used – a different policy is required for each tender. All relevant documentation is added to the document management system i.e. building plans, covering letter, images, etc. These are all downloadable separately or as a compressed zip file by potential suppliers.

A potential supplier views the open tenders on Salford’s e-tender site, downloading the specific tender information that they are interested in. If they decide to progress they must first register for the service and download and set up the Entrust Desktop Solution, the software for signing and encryption, and the E-tender client. Registration only needs to be performed once, prior to the first tender submission. Thereafter the software can be used for all subsequent tender submissions. On registration the supplier’s LDAP Distinguished Name (DN) is allocated. This is a globally unique name that will be inserted into the suppliers public key and attribute certificates.

The tender client is a Visual Basic wizard – providing a simple user-friendly submission process. The supplier answers the prompts supplying their supplier ID, Entrust Password and answers to tender specific questions. Suppliers can also attach extra files for use in answers to the tender specific questions e.g. a diagram of the proposed structure of the building. The selection of the tender lets the E-tender application know what tender specific questions to ask and also which staff within Salford are allowed to open the submission files (this is sent to the client in XML). The submission file and any additional files are digitally signed by the supplier and then encrypted to the Salford staff and then the supplier signs the whole file again. This triple wrapping is necessary to ensure that only authorised suppliers are allowed to submit the tender, then the outer signature is discarded by the tender store leaving encrypted tenders that cannot be read by anyone other than the Salford staff after the close of tender. The inner signature is the legally binding one that proves who has submitted the tender.

The submission is uploaded into the tender store. The tender store is simply a directory in a filestore. The submission is placed in a sub-directory of the tender store with the name of the TenderID. The submission process is add only, so no submission is overwritten – only new ones are added. Once the submission is fully uploaded and saved to filestore, it is reloaded and the outer signature of the submission file is checked. If the signature cannot be verified – the submission is deleted. If the signature is verified and the DN matches the DN in the file (in the registration record) the submission is kept and the signature discarded. The PERMIS PMI checks that the supplier has a role that gives him permission to upload the submission. If permission is granted, an email is sent to the supplier confirming receipt of the submission. If permission is denied the submission is rejected and a rejection email is sent to the supplier and the submission deleted from the tender store.

The withdrawal client is used by internal Salford staff i.e. the Lead Member or Tender Officer, who wish to review the tenders. The user logs onto the client to issue the withdrawal command. By successfully logging into the client, authentication has taken place. A withdrawal command can then be issued. The PERMIS API checks that the user is authorised to withdraw tenders, and either grants or denies access to the tender store. Once two withdrawal commands are successfully issued by two separate staff, the tender submissions are returned to the client. If one user issues a withdrawal command an email is sent to the other users (who are authorised to withdraw the submission) asking him to issue a withdrawal command. Once the tender submissions have been returned back to the users, the withdrawal client decrypts them, checks their innermost signatures, and places them in a decryption directory.

Salford integrated the PERMIS API by using the SUN COM/CAS Active X Java bridge. A COM object was produced that enabled integration with other Microsoft products. The document management sub-system was created with Microsoft Share Point portal server.

6.2. Roles and Policy

The roles for the users of Salford’s E-tendering system, and their associated permissions are as follows: -

A supplier has two possible roles:
The open supplier role that allows any supplier to submit for an open tender.

The restricted supplier role a,b,c,d,e,f,g, etc. that allows groups of suppliers to be selected for restricted tenders

The Lead Member role has two variants:

The simple Lead Member role allows any lead member to issue a withdrawal command for any tender

The restricted Lead Member role a,b,c,d,e,f,g, etc. allows individual Lead Members to issue a withdrawal command for specific tenders

The Tender Officer role has two variants:

The simple Tender Officer role allows any Tender Officer to issue a withdrawal command for any tender

The restricted Tender Officer role a,b,c,d,e,f,g, etc. allows individual Tender Officers to issue a withdrawal command for specific tenders

The Tender Team Leader has one role that allows the user to create tenders, tender policies, attribute certificates for suppliers etc. The Tender Team Leader is also the only role that has access to the Privilege Allocator for digitally signing attribute certificates and policies.

The Privilege Allocator gives Salford’s Tender Team Leader the ability to allocate roles to specific users based on the tender that is being processed.

The roles are transparent to the suppliers, they only know that their submission has worked or failed and why the submission has failed.

E-tendering gives both the City of Salford and the suppliers obvious benefits such as: ease of access and notification, improved download and submission of documents, and of course time and financial savings. After the successful completion of the PERMIS project, it is the aim of Salford to roll out the application to a wider audience within the authority.

7. E-PLANNING IN THE COMUNE DI BOLOGNA

Bologna’s E-planning application is about digital cartography, and it’s named FreeCart. In the FreeCart application, users may browse, download digital maps of the city and even submit modified maps back to the city in order to request building licenses.

The FreeCart application can be accessed today at http://sit.comune.bologna.it

7.1. Application Access Policy

Application access is ruled by a simple policy. The following table summarizes the user roles that the FreeCart Application supports, and the access rights granted to each role:

<table>
<thead>
<tr>
<th>User role</th>
<th>Browse city map, locate city address</th>
<th>Download selected map sections in DXF format</th>
<th>Upload modified DXF maps and other documents in order to get a building license</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Trusted</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Professional</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
The following table summarizes how the user roles will be mapped in the pilot phase onto actual Internet users:

<table>
<thead>
<tr>
<th>User role</th>
<th>Assigned to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>Everybody</td>
</tr>
<tr>
<td>Trusted</td>
<td>Bologna Citizens who have requested to use the service (Digital Certificate necessary)</td>
</tr>
<tr>
<td>Professional</td>
<td>Bologna Professionals who have requested the service (Digital Certificate necessary)</td>
</tr>
</tbody>
</table>

All roles will be allocated by the City of Bologna. Each user is required to possess a digital certificate in order to be able to digitally sign their requests to access the FreeCart application.

Once the application is validated, a natural evolution of this mapping would be:

<table>
<thead>
<tr>
<th>User role</th>
<th>Assigned to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>Everybody</td>
</tr>
<tr>
<td>Trusted</td>
<td>Citizens from Bologna and Other Cities who have requested to use the service (Digital Certificate necessary)</td>
</tr>
<tr>
<td></td>
<td>(1)</td>
</tr>
<tr>
<td>Professional</td>
<td>Bologna Professionals who have requested the service (Digital Certificate necessary)</td>
</tr>
<tr>
<td></td>
<td>(2)</td>
</tr>
</tbody>
</table>

(1)=Privilege granted by Bologna and other partner cities
(2)=Privilege granted by the appropriate Professionals’ Association

The above allows users from other cities to access the FreeCart application, using the Trusted role assigned to them by either Bologna or other cities that Bologna trusts. Also the Professional role can be allocated by the appropriate Professional Association once they have implemented Attribute Certificates, rather than by the City of Bologna.

### 7.2. Application Architecture and Logic

Bologna’s E-planning application is built over a standard Tomcat Application Server, therefore using an ordinary Java-based environment with HTML pages and the HTTP protocol.
It is useful to describe the application by following a typical sequence to read a map, as follows (see Fig. 2):

1. A user connects to the web application, passing it a digitally signed request to download a map. The application receives the request (1), and validates the digital signature.
2. The application passes the request to the Application Gateway, providing it with the user's Distinguished Name extracted from the user’s Digital Certificate.
3. The Privilege Verifier (PERMIS API) inside the Application Gateway checks on the LDAP server for attribute certificates related to the user.
4. As soon as the attribute certificates are back a decision is made based on the role of the user.
5. Supposing the latter is positive, a request is made using a secure intranet connection to the City Maps Server.
6. The map is returned as a byte stream.
7. The byte stream is redirected to the user.
8. Finally, the user receives the map.

The physical blocks in the above Fig. 2 are:
- The users’ PC
- The City web application server
- The LDAP directory of the City’s PKI
- The file server providing maps in vector format.

The authorisation component (“Application Gateway”) has been structured and implemented so that it can be re-used for other applications with only a few minimal modifications.
8. BENEFITS OF THE PERMIS APPROACH

There are several advantages of using the PERMIS PMI approach to authorisation instead of traditional discretionary access control lists (ACLs). ACLs become difficult to manage with large numbers of users, and one can never be sure that some users do not have more privileges than they ought to. One particularly extreme example of this is when an employee leaves an organisation; his access rights may not be removed from an ACL, as the application owner was unaware that the employee had left. By using role based access controls (RBAC), as in PERMIS, only roles have permission to access an application, and it is much easier to enforce the least-privilege principle i.e. that people only have the minimum set of permissions needed to perform their designated tasks. When an employee leaves an organisation, or is transferred between departments, the personnel function simply needs to remove or change the role of the employee and he automatically inherits the privileges necessary for his new role in each application that supports RBAC. Secondly, traditional ACLs have to be managed separately for each application they belong to. This compounds the difficulty of managing them. It also makes it much more difficult to ensure consistency of a user’s privileges between applications. By using policy based authorisation, as in PERMIS, it is possible to have a consistent access control policy across all applications in a domain, so that users only gain the privileges they are required to have throughout the entire domain. Management of access rights is further simplified since only one policy needs to be written and kept up to date, and this covers all applications in the domain.

9. CONCLUSIONS

In this paper we have described a generic and flexible policy based Privilege Management Infrastructure that stores users privileges as roles inside X.509 attribute certificates. We have further shown how the same authorisation infrastructure has been successfully built into two components of the construction process – E-tendering and EPlanning. We have found that the PERMIS PMI is sufficiently generic to allow it to be built into many more applications. For example, at the University of Salford we have incorporated it into an electronic prescription processing application, in which doctors, dentists, pharmacists, patients and prescribing nurses are given appropriate roles and entitlements (some patients for example are entitled to free prescriptions). This has duly been implemented and the results were presented in July 2002 at the Hospital of the Future conference in Chicago [Mundy & Chadwick, 2002]. In Barcelona, the PERMIS PMI has been incorporated into an application to give its citizens and car hire companies on line access to the city’s car parking ticket database. Finally Dartmouth College in the USA are incorporating the PERMIS PMI into an application for performing digital rights management for the US Justice Department's National Institute of Justice. We are therefore confident that the PERMIS PMI can be incorporated into many more components of the construction process, and we hope in due course to be able to demonstrate more of these.

9. ACKNOWLEDGMENTS

This work has been 50% funded by the EC ISIS programme PERMIS project, and partially funded by the EPSRC under grant number GR/M83483. The authors would also like to thank Entrust Inc. for making their PKI security software available to the University on preferential terms.
REFERENCES


