Citation for published version:
Young, A, Kirstein, P & Ibbetson, A 1996, *Technologies to Support Authentication in Higher Education: A Study for the UK Joint Information Systems Committee, August 21th, 1996*. UKOLN, University of Bath, Bath.

Publication date:
1996

Document Version
Early version, also known as pre-print

Link to publication

Publisher Rights
Unspecified

University of Bath

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 07. Dec. 2019

Technologies to Support Authentication in Higher Education

v5

A. Young, I.T.Institute, University of Salford
P.T. Kirstein, Department of Computer Science, University College London
A. Ibbetson, Computing Laboratory, University of Kent at Canterbury

A Study for the UK Joint Information Systems Committee
August 21th, 1996

Executive Summary

This report provides a short and limited study, commissioned by JISC, of the technologies available to support authentication, reviews the needs expressed by a set of people contacted for the study, and provides the beginnings of a road-map on how a National system might be established.

First a brief overview of the fundamentals of Security technology is provided. As part of the study, we were asked to consult a number of people - particularly from the set of those supported under the JISC Electronic Library initiative. These were supplemented by some people at UKERNA and in Information Services departments in the universities. We present our impressions of the requirements envisioned by the people consulted, and their proposed solutions; with very few exceptions, the needs expressed were very limited, and the solutions limited to specific applications. This reflects, we believe, more the selection of the people consulted, than the true needs of the area. It was also coloured, in our view, by the fact that there was no indication that any finance for a wider initiative might be available.

A more detailed review of the current methods of authentication, the needs and the status of different applications follows. This includes a brief discussion about the Standards being developed in the Internet Engineering Task Force in conjunction with the wider deployment of the Internet and the status of infrastructure standardisation and deployment. We consider also a number of applications: electronic mail, the World Wide Web, remote log-in, document security, multimedia conferencing, directories, general network facilities and electronic commerce. A brief discussion of a number of ancillary technical and legal issues follow: this includes smart-cards, directory systems and key escrow. The existence of legal considerations is indicated, but little argument is developed other than the appending of proposed Government legislation.

As a final section, we start on a Road Map of how we might proceed to a National authentication infrastructure for Higher Education. We believe that such a system should be distributed in nature, and could well leverage on the investment already made in an X.500 distributed directory system. It is clear that the current technology would need considerable updating; much broader involvement must be achieved from other sectors of the universities for such an initiative to have broad impact. We mention some of the measures that should be undertaken to enable a successful broader applicability. Based on the existence of a National directory system, we then propose a National authentication infrastructure by proposing a system of Certification Authorities, distributed registration and update, and the retention of the certificates in the National directory system. We propose that existing projects in secure E-mail and electronic libraries be asked how they might be modified if such an infrastructure was developed.

A substantial distributed infrastructure for authentication could have implication well beyond the university sector. For this reason, it may be possible to co-fund the development and many of the earlier trials from sources outside JISC. We propose that we explore avenues of co-funding both from the British Foresight Programme and from the European Union Telematics programme.

Table of Contents

1. Introduction

This draft report gives preliminary results of a three month study, funded by the Joint Information Systems Committee (JISC), to define requirements for authentication of users of distributed services across the Joint Academic Network (JANET), and to identify the technology that that is needed in order to deploy these services. Due to constraints in budget and time-scales, the study cannot be exhaustive or definitive; especially given the speed at which the research area is moving. Nonetheless, it aims to provide a thorough understanding of the principal current research activities and commercial trends, and some insight into the outstanding problems still to be solved. The present document is an update of one submitted for review at a meeting held at UCL at the end of July.

This study was not intended to focus only on the technology; it was supposed also to consider specifically the needs of the customers JISC . For this reason, the initiator of the study, Chris Rushbridge (Programme Director of the JISC-funded Electronic Libraries Programme [ELP]) proposed that we assess the security desires of a number of British organisations - particularly those funded under ELP. We consulted, therefore, not only the people proposed, but also some people in Academic Information Services Divisions, and UKERNA - who have initiated a programme on secure E-mail.

Authentication technology will comprise both hardware and software for use by people operating authentication services, and also end user applications which make use of these services for enhancing facilities such as

- electronic mail;
- World Wide Web access;
- multimedia -conferencing;
- electronic document access; and
- remote login to central facilities.

The current rapid expansion in the use of and interest in open networking services, such as the Internet, has created an enormous new market in electronic information. Current facilities aim only at the distribution of this information and do not consider the mechanisms that will be needed if access to the information is to be controlled, or if the information is to be sold rather than given away. This requires a number of new facilities related to data security:

- strong levels of authentication to support access control;
- digital signatures to support data integrity;
- encryption to support data confidentiality (privacy); and
- non-repudiation to support billing.

This study concentrates primarily on the provision of authentication and non-repudiation, but it is worth noting that exactly the same mechanisms can be used to provide the facilities required for confidentiality.

The use of security extends beyond simple information trading. Typical users who require security services in open networks are: commercial companies, which want to make business over open networks; their clients, administrations, public medical and social services, for whom it is vital that only approved groups are able to participate in their operations; organisations for their external and internal network communication; and communities such as the European research community, which is using intensively open communication networks and which needs secured E-mail, secured directory access and secured file transfer. The rapidly growing use of the Internet for commercial purposes creates an evident need for secured versions of the World Wide Web and for systems supporting electronic payment.

There are many potential uses of JANET and the Internet that are not currently possible without adequate security. If appending an electronic signature to an email document could be given the same legal weight, or popular respect, as appending a hand-written signature to the bottom of a letter then many possibilities open up to the use of email for commercial purposes. This usage is currently

impractical, despite the fact that such a digitally signed document can be made much more difficult to forge than a conventional one. The use of confidentiality to prevent unwanted disclosures allows electronic means to be used for the transfer of commercial documents, even via open international networks. The technology therefore has many potential users and an enormous potential market. This has encouraged the development of solutions and applications, but also politics between the major players.

The organisation is of this report is as follows. In Section 2, a brief overview of security is provided. In Section 3, we summarise the views of the people we consulted in the academic community on their security needs. Since the main requirement for this community seems to be Authentication, we concentrate, in Section 4, on giving a State of the Art report on Authentication. In Section 5, we consider applications which would benefit from addition of security, and list some of the available products. Some additional subjects related to security are considered in Section 6, which include security infrastructures and legal issues; these issues are vital to the developments of our arguments. In Section 7, we present an outline scenario of how we believe this area should be advanced in the academic community.

For completeness, we provide a list of the persons contacted during this study in Annex A, and include a paper submitted to us by John Simmons, on behalf of BIDS, as Annex B. Finally, during the course of the study, the UK Government has issued a consultation paper on security issues which is clearly relevant; this is appended as Annex C. Finally, a set of references, most of which are available on the WWW, are listed in Annex D.

2. Background

Open networks for information exchange and tele-cooperation are no longer a distant dream. The Internet gives this dream a realistic basis. Electronic mail, information offers over the World Wide Web, and simple file transfer of general digital data have become part of every-day work for many people in Europe, and many more are seeking to use the Internet for their professional and private life. The growth rate is still exponential, especially in Europe where the development has started a little later than in the US.

The Internet is certainly not the only possible realisation of an open communication network. However, it defines the most realistic stereotype. In the past, research and trials were the dominant application areas of the Internet. However, today people and organisations are concerned with the commercial potential of an open communication network like the Internet. Business wants to use the Internet and does not really know how to do that. Electronic mail, multi-media information, and file transfer are there, but how shall they be used for business purposes in a world-wide open marketplace? Is it possible to provide open communication networks with enough security in order to allow people and organisations to use the network for their very sensitive communication and co-operative work? Is the Internet secure enough? A common saying is that "the Internet is insecure".

Within closed environments, like the employees of one organisation, it is a well-established practice to enforce security requirements by a security administration in the hands of a trustworthy group of administrators. They can, for example, maintain a system of password-based authentication and encryption functions. This works, because these people are registered by the security centre before they start using the system. However, on a marketplace, people want to enter a formal co-operation when they have never met before. They need to exchange authentication proofs, without having a common basis of credentials. In general, people on a marketplace come from a variety of different domains, with different issuers of credentials. In closed environments, credentials may be based on secret keys which are administered by one central institution. In an open network environment, this is the weakest point for security attacks. The unauthorised access of such a secret (e.g., a password which is communicated over the network in cleartext) would enable the eavesdropper to masquerade as the original owner. No subsequent authentication or encryption function would provide any security for the related communication partner.

2.1 Security Services

In open networks such as JANET and the Internet, all electronic messages that pass through the network are potentially visible to anyone with the inclination and competence to look at them (it is not necessarily easy, but it is certainly possible). The contents of the electronic messages must be assumed to be able to be:

- **intercepted**, to allow the eavesdropper to observe what people are doing;
- **modified**, to allow the eavesdropper to change a request someone makes or the answer they receive in reply; or
- **replayed**, to allow the eavesdropper to duplicate what people were doing at a later time (this is especially useful if the eavesdropper does not understand the content of the electronic messages, but knows the effect that they produce) .

By intercepting messages from a user to a remote server, acting upon them locally and returning the answer, the eavesdropper is able to masquerade as the remote server, and the user is not generally able to distinguish this. By intercepting messages, modifying them and passing them on to their correct destination, complicated “man-in-the-middle” attacks can be constructed, and these can be even harder to detect than masquerade attacks. By intercepting messages and simply logging them, passive “sniffer programs” are able to capture users passwords and credit card numbers. These scenarios are all real, and are not just theoretical scare stories and, while risk analysis is beneficial to determine that the effort in solving a problem is proportionate with the benefit obtained from solving it, it is clear that managers of university computer networks and national CERTs (Computer Emergency Response Teams) face a significant daily workload caused by security breaches. Although internal problems

within a local organisation caused by user carelessness or ignorance probably account for 90% of the problem, the other 10% is very significant and potentially very serious.

Security services are therefore needed to enhance the use of the Internet, and good security will make possible a whole array of network based services that are simply not possible at the moment.

It is important to define the scope of the problem with which we are dealing. The aim is to ensure that privacy, authenticity and integrity are protected from the amount of effort than can reasonably be expended by a commercial company (while undoubtedly large it is substantially less than the determined efforts of a government security agency, which cannot be countered economically; nor is it our brief here to discuss whether such counters should be made). Therefore, we aim to protect users against the efforts of other users of the network (note this even includes system administrators, who have traditionally enjoyed total privileges), or of the operator of the network (e.g. Internet Service Provider); and to protect users against mis-configuration of the network (whether accidental or malevolent) and hostile attempts to subvert network traffic (e.g. passive monitoring through sniffer programs, interception and replay).

The means to provide strong security services is through the use of cryptography and the use of encryption keys. Strong security services do not rely on exchanges of passwords and assume that all communications can be intercepted, modified and replayed at will by an attacker. Mechanisms that prove that a particular action can only have been carried out by the holder of a particular encryption key provide strong levels of authentication of the owner of that key. Knowledge of a key owned by another user allows data to be encrypted in such a way that no other user will be able to decrypt the data, thereby providing confidentiality.

2.2 A Quick Introduction to Security

There are two cryptographic techniques that all security services are built on. Encryption is the process of scrambling a message so that only people able to unscramble it are able to understand the message. Computation of hash values (also called message digests) is the process of reducing an arbitrary length message to a fixed length summary value (sometimes called a fingerprint and roughly equivalent to what was known, many years ago, as a checksum)

There are two types of encryption, symmetric encryption (also known as secret key encryption) and asymmetric encryption (also known as public key encryption). With symmetric encryption, the people encrypting and decrypting have the same key, one uses it to encrypt and one uses it to decrypt. Sometimes this key may have a short lifetime and be used for one specific purpose. Sometimes it may have a long lifetime and may be used for many exchanges between the same pair of people. But the sharing of these keys (known as secret keys because knowledge of the key gives you access to the encrypted message) causes many problems. DES (Data Encryption Standard) and IDEA (International Data Encryption Algorithm) are examples of a symmetric encryption method. DES uses a 56 bit encryption key, and IDEA uses a 128 bit key. IDEA is, therefore, significantly more secure. There is a variant of DES (called Triple-DES) which uses two 56 bit keys in three operations, and therefore provides the equivalent security to a 112-bit key - which is comparable with the security offered by IDEA.

The following table shows the average amount of time (half the worst case) taken to break an encrypted message by brute force (trying every possible decryption key) assuming one key is tried per microsecond

Key size (bits)	Number of alternative keys	Time required for brute force attack
32	$4.3*10^{9}$	35.8 minutes
40	$1.1*10^{12}$	6.3 days
56	$7.2*10^{16}$	1142 years
128	$3.4*10^{38}$	$5.4*10^{24}$ years

(Note that 40 bit keys are currently the maximum allowed for encryption software exported from the USA).

In practice, it is likely that current and future computers could do a brute force search at a higher rate than one key per microsecond. However, even if these times were reduced by several orders of magnitude, it is clear that IDEA will be a significantly secure algorithm for the foreseeable future whereas DES is likely to become vulnerable to attack by wealthy organisations in the not too distant future. There is no publicly known way to break an IDEA or DES message other than by systematically trying every possible key, but there is also no proof that a faster method does not exist.

The other type of encryption is asymmetric encryption; here each user has two keys and keeps one private while making the other public. A user can encrypt a message with his private key and the other user can decrypt with the corresponding public key, or a user can encrypt a message with another user's public key and that user can decrypt with her private key. This type of encryption is very powerful as it does not involve controlled sharing of information - keys are either totally private or totally public. RSA (Rivest Shamir Adleman) is the most common example of an asymmetric encryption method. It uses keys of arbitrary length, with values between 512 bits and 2048 bits being most common. 1024 bit keys are generally recognised as being the minimum for serious security, with larger values being used for military security. Another example of an asymmetric encryption algorithm is DSA (Digital Signature Algorithm) which has the properly that one of the user's keys can only be used for encrypting, and the other key can only be used for decrypting.

A hash value is a number calculated in some way from the content of the message (the method is called the hashing algorithm). A hashing algorithm is irreversible, and is not possible to reconstruct the original message from its hash value. The corollary to this is that many different messages will have the same hash value. However, it is believed to be extremely unlikely to find two messages with the same hash value and computationally unfeasible to construct a message with a given hash value. Examples of hashing algorithms are MD2, MD4 and MD5 (MD stands for message digest - note that the MD4 algorithm is rarely used nowadays since it has been found to be poor at guaranteeing it will be hard to find two messages with the same hash value; recent research has indicated this is also true of the MD2 algorithm, and may be for MD5) and SHA (secure hash algorithm)

We shall now show how these facilities are used to provide the security services we need.

To prove that a message has not been altered, the sender of a message calculates a hash value of the sent message and transmits this along with the message. The recipient calculates a hash value for the received message and compares this with the value provided by the sender. If they are the same then this shows that the message has not been modified. Of course in an insecure environment, this means nothing as an attacker modifying the message can simply recalculate the hash value and modify that as well.

To improve the security, the sender calculates a hash value for the message and encrypts it with "something" mutually known by the sender and the intended recipient. This value is then appended to the message. The recipient calculates a hash value for the message, determines who the message claims to be from, decrypts the appended hash value using the mutually known "something" and compares the result with the locally calculated value. If they match then the recipient knows both who the message is from and that it has not been altered. If they do not match then either the message has been modified or it is not from the user who it claims to be from (it is not generally possible to tell which of these has caused the failure).

This combination of computing a hash value and encrypting it is called a digital signature. If secret key encryption techniques are used then the hash value would be encrypted with a pre-arranged secret shared between the sender and recipient. If public key encryption techniques are used then the hash value would be encrypted with the private key of the sender and decrypted with the well-known public key of the sender. Note that the public key of the sender is well known to everyone and this means that anyone can verify the authenticity and integrity of the message, whereas with secret key encryption techniques only people who know the secret (i.e. the sender and recipient) can verify the signature.

The use of encryption also provides for confidentiality of messages. With secret key encryption techniques, the body of the message will still be encrypted with a secret value shared between the

sender and intended recipient. However, with public key encryption techniques, the body of a message will be encrypted with the public key of the person who is to receive the message. The sender can be sure that only the person who owns the private key that goes with the public key used for the encryption can decrypt the message (even the sender will not be able to decrypt it once it has been encrypted). As long as the sender trusts the recipient to keep their private key private then the sender can be sure that no-one else will be able to read the message.

2.3 Definitions

This section gives formal definitions of the basic security mechanisms introduced in the previous section, and an example of how they can be used to enhance a typical transaction. Again, we emphasise that security mechanism almost never provide "absolute certainty".

Authentication: In the context of electronic communication between two parties (either of which may be a human or an automated process), "authentication" means that the recipient of an electronic message may be reasonably certain of the identity of the sender of the message.

There are a number of types of authentication.

- **simple authentication** uses shared secrets (passwords) which are exchanged as clear text and which therefore provide very little assurance of the identity of the sender of the message;
- **protected authentication** again uses shared secrets, but does not rely on clear text exchange of these secrets, and therefore protects against interception and replay of communication; and
- **strong authentication** uses cryptographic techniques to provide the possibility of extremely high assurance of user identity.

To summarise: simple authentication occurs through exchange of a shared secret (shared between authenticator and person being authenticated): protected authentication occurs through demonstration of knowledge of a shared secret (though the secret is never divulged); strong authentication occurs through demonstration of knowledge of something known **only** to the person being authenticated.

A related term is "**non-repudiation**" which relates to the recipient of a digitally signed message ensuring that the authenticated sender of the message cannot later deny having sent it, either by claiming never to have had the encryption keys needed to create the signature or by claiming that the encryption key had been stolen. This type of mechanism will typically rely on trusted notarisation procedures and trusted time-stamps and will require long term secure storage of signed data, and is essential if legal agreements and penalty clauses apply to an authentication-based service.

Integrity: In the context of electronic communication between two parties, "integrity" means that the recipient of an electronic message may be reasonably certain that the message received was the same as the message sent (i.e. that it has not been altered in transit).

Confidentiality: In the context of electronic communication between two parties, "confidentiality" means that the sender of an electronic message may be reasonably certain that only a predetermined set of recipients will be able to understand the content of the message.

As an example, let us consider retrieving a file from a commercial repository using currently deployed ftp technology. The user first connects to the ftp server:

```
$ ftp ftp.widget.co.uk
Username: fbloggs
Password: mypassword

You are connected
ftp>
```

At this point, the user believes he has connected to ftp.widget.co.uk and the ftp server believes that the user who has connected is fbloggs. However, neither of them has any real evidence for these beliefs, and they are based purely on faith. Provision of strong authentication would allow each of them to be confident that they were indeed communicating with the correct people/services.

The user then issues a command

ftp> get report.doc

The user wants to be sure that the ftp server receives the correct command, and that the file received in return has not been modified in transit. Provision of integrity mechanisms would allow the user to gain these assurances (though, of course, the ftp server has to be trusted to make the correct association between filename and file).

The server will also want to be sure that fbloggs is allowed to retrieve this file, and the authentication carried out in the first stage will allow the server to provide access control. In addition, if required, a non-repudiation mechanism could be used to give the administrator of the ftp server sufficient evidence to be able to send fbloggs a bill for the transaction with the knowledge that fbloggs will not be able to deny having requested the file (of course, mechanisms would in practice be needed to be sure that fbloggs received the file rather than merely requesting it). If the document is sensitive, then the server may also want to be able to encrypt the document to provide confidentiality and be sure that no-one is able to eavesdrop on the communication and receive the document for free.

This example shows how all of the security mechanisms can be used to enhance a standard user→server, request→response transaction. There are many other places in this example where security technology could be added (for example the user may want the get command to be encrypted so that an eavesdropper cannot even tell what file is being downloaded), but the example demonstrates the fundamentals.

3. Requirements for Users and Administrators

From the discussions that we have had with the people in academia suggested to us as most likely to be using or operating authentication servers, the current use of authentication services breaks down fairly clearly into three groups

a) those who are not concerned with security at all;
b) those who use simple passwords (surprisingly often shared between many users);
c) those who use packet filtering techniques to deny access to unauthorised people on the assumption that they are using unauthorised IP addresses;
d) those preparing large-scale secure E-mail system.

Many schemes operate a combination of types (b) and (c). Most acknowledge that the systems are not really secure, but see little evidence of any abuse of the system - certainly not enough to warrant asking the paymasters to spend any more money on improving security. For (d), there is only one UK group, at Cambridge University but backed by UKERNA, who are providing a largish scale system based on PGP.

Most effort was placed on providing a level of security between different organisations (i.e. the information consuming organisation and the providing organisation), and there was little concern for security within local systems.

It is fair to say that most organisations seeing little or no need for security were not yet involved in any real commercial activity, and acknowledged that they may eventually have to face this issue. Most people were aware that password sniffing was a possibility. Most of the organisations running E-lib applications were particularly unconcerned by the limited security offered by the current systems; it matched their perceived risks of loss. In the case of the journal publishers, they are really concerned that there is no systematic substitution of subscription as a result of electronic network publishing and access. Their current policy is often to include the price for electronic access into the subscription price. Password sniffing might lose them one personal subscription - but most of their revenue is from institutional subscriptions.

In the long term, there is a widespread view that more than this will be needed. In particular, the following reasons for using security were noted (in alphabetical order, not of importance)

- allowing access from alternative locations (e.g. from home);
- authentication because making copyrighted material available to a target community;
- authentication for administrative purposes (i.e. for people updating databases, directories or mailing lists) - this is usually small scale;
- authentication for controlling access to scarce resources, while others stay open;
- authentication in order to charge for document supply (charging usually on a departmental or project basis);
- authentication to manage closed mailing lists;
- cryptographic checksums of served data to prevent tampering and maybe to spot illegal copying of resources;
- different classes of user can retrieve different views of the objects accessible from a whois++ server;
- encryption of electronic mailing lists;
- infrastructure must scale to individuals for some applications, larger granularity (domain based) is OK for others;
- mechanisms must keep subscribers, service operators and paymasters 'happy', where 'happy' is closely connected with ease of use and ease of operation;
- prevent random passers-by from seeing unfinished work and gaining the wrong impression of a service;
- restrict access to course material;
- security against hacking attacks from the Internet;
- uniform access control and authentication mechanisms across information service providers;
- uniform mechanisms across applications;
- want to practise what we teach;

The main applications seen as needing security were (in roughly decreasing order of perceived importance)

- World Wide Web
- Remote Login
- Directories
- Email
- EDI
- Multimedia conferencing

One requirement which was not mentioned, but which will become evident, is that the potential marketplace for information provided by UK academic sources goes beyond UK academia and, indeed, beyond academia itself. The need to communicate securely with anyone in the commercial or academic world, wherever they are in the world, is extremely important; as shown by the trend within Framework-4 for proposals and deliverables to be submitted electronically. EPSRC are also moving in this direction and it seem likely to be the one of the first multi-institutional uses of EDI in UK academia. Mechanisms therefore need an international capability, and any mechanism that provides fixed size communities is of limited short-term use.

The following issues were seen as major concerns to people considering deploying security technology:

- The scale of the problem of registering and handling the intended number of users was seen as a major hurdle to be overcome, and many current authentication schemes are merely a compromise between the need to have some level of authentication and the need to have a manageable scheme. No one information service is going to want to register every possible user, possibly numbering half a million to a million. However, all of these concerns are rooted in the assumption that the information provider needs to provide all the mechanisms needed to manage control of the information they provide. No-one was approaching the problem from the direction of deploying a wide-scale infrastructure that is managed locally and that information providers would plug into in order to obtain authentication services. If this encompassing infrastructure can be provided then

most, if not all, of the scalability concerns can be removed and so compromise solutions will no longer be necessary. The largest single effort in this area seems to be the NISS Authentication Server (cf. Annex B). This again is a centralised service in its concept, though some of the administration can be devolved to libraries in other institutions to manage User Ids. At the same time this is regarded as a server for one set of Electronic Library applications, from the way its management is proposed.

- The attempt to provide a PGP infrastructure for secured electronic mail (c.f. Section 5.1.3) was entirely centralised. The group in question is now trying to set up a large centralised Key Server, able to scale to the 1M names of the UK academic and research community. The main reason for the stated requirement for a centralised service was the perceived need for key-to-name mapping, which is difficult to achieve in a distributed environment. If this requirement is removed, and it is principally a consequence of the way PGP authentication is done, then the it would be possible to use PGP in a distributed environment.

- US export regulations were seen as a major impediment to deployment of any security infrastructure, although in fact the only effect of them is to deny non-American users access to some products - Netscape with 128 bit SSL may not be available outside America, but Mosaic with 128 bit SSL is, so a user with a specific need can get the required functionality at the cost of not being able to use the preferred user agent. It is important to note that this impacts only some User agents, not the security infrastructure; tools exist in the UK which would allow the deployment of a security infrastructure of any strength desired.

- Some of those contacted saw the advantages of a distributed authentication infrastructure - but nobody had any concept how it could be achieved.

- None of those we contacted had any great hopes of the X.500 Directory Infrastructure that has been set up so far. It is clearly ageing (as will be discussed later, the commercial offerings for this service are now incompatible with those set up in the UK under the PARADISE/NAMEFLOW project), and there has been little attention devoted to ensuring the completeness of the data. Clearly the access to such services should now be provided via the WWW. In addition, it was pointed out that there is a problem of data replication. For example, student records are often held on Oracle or similar databases; a WWW approach based on direct interrogation of such a database via a CGI script was considered more feasible.

- Practitioners of Secured Multimedia-conferencing did see the need for an infrastructure; authentication of conference announcements were a particular problem. However since the main practitioners here were in a project managed by one of the authors of this document, it is difficult to weight their views very heavily.

One class of users whose needs require special attention is that of university administrators. Administrators routinely handle information that requires high levels of assurance over security for financial or legal reasons or more commonly to comply with accepted good practice. Data handled is financial, personal and "company confidential". The activities of purchasing offices, buildings and estate offices, or conference offices can be likened to straightforward trading activities. Student and staff data, including areas where returns will be made electronically that enable persons to be identified, might for instance fall into the legal area. The transmission of financial and planning documents to the funding bodies would fall into the last category.

These uses are by no means restricted to off-campus activities. MIS managers used to talk of having separate physical networks distributed to departmental offices and in Registry buildings. Fortunately, these have not happened through a mixture of financial considerations, staff development and sense. However, most senior administrators in universities seem now to believe that facilities available to them for security over academic networks are inadequate and will stay so for some time. They see little evidence of much attempt to support them and are, therefore, often content to discourage administrative use of networks in any sensitive area. They require many of the functions listed above but are especially enthusiastic about all the authentication requests, encryption of lists and messages and security against hacking.

Outside the medical field there is a relatively small body of confidential material transmitted between institutions but trading with third parties is on the increase. A growing requirement is where an HEI has more than one site that is JANET-cnnected, especially with the rapid growth of MANs. In this case there will be many more sensitive transfers or accesses.

External bodies within the community, such as ERASMUS, HESA, UCAS and the funding bodies, are considering making available material over networks. An interesting case is UCAS where the possibility of providing the electronic equivalents of UCAS forms and personal records of achievement is under debate, possibly as a national CV register. Here the requirement is for authentication of use but also for reliable secure mechanisms to allow permissions to be handled rapidly. This gives a good example of the need for trade-off between efficiency and security which is endemic in all such applications.

4. Current Methods of Authentication

This section describes the state of the art of authentication - techniques currently being used or currently being developed.

4.1 Simple Authentication

Most authentication currently taking place within JANET and the Internet is based on the user presenting a password to the target system (the system is never required to authenticate itself to the user). Examples of this are the telnet protocol for remotely logging in over the network, or the htaccess method of restricting access to WWW pages.

There are numerous problems with use of passwords:

- In order to remember them, users tend to write passwords down. This is especially true if the user has many different passwords for many different systems or if a system requires the user to change passwords frequently. There is no solution to this problem except user education.
- Users tend to pick obvious passwords, such as names of friends, loved ones or pets. One solution to this is for a central administrator to issue passwords, but if they are not memorable then users are more likely to write them down.
- Passwords are transmitted from the user to the system in clear text on the communication lines. This makes it very easy for a rogue system inserted between the user or the system (or more likely on the user's terminal itself) to observe the communication and observe passwords and where they are being used.
- Password files tend to store passwords in encrypted (or hashed) form, but often there is no restriction on reading the password file. This makes them very amenable to dictionary attacks (where you encrypt/hash a dictionary of words and compare them to the values stored in the password file) or brute force cracking techniques where (off-line) you systematically try all possible combinations of password until you guess right. An eight character case sensitive alphanumeric password has around $2*10^{14}$ permutations, making it weaker than a 56 bit DES key (which is known to be not very secure).

In summary, passwords do not provide an adequate mechanism for authentication.

4.2 Packet Filtering

Another commonly used method of restricting access to data or services is through a filtering mechanism, often as part of a firewall system. Firewalls sit between a trusted network (e.g. a organisational internal LAN) and an untrusted network (e.g. the Internet) and inspect communication in both directions according to a security policy. They can block access to certain services or they can redirect access to ensure one internal machine handles all external interaction.

A related technique is used with by WWW servers, which can restrict access to WWW pages on the basis of pattern matching of the client's IP address or domain name. This technique is used to restrict pages to use within one organisation (*.salford.ac.uk) or all of UK academia (*.ac.uk)

It is possible to fool filtering mechanisms (though very difficult), and techniques of spoofing IP addresses have been demonstrated. While it is a useful part of a security strategy, it is not adequate in itself.

4.3 Obscurity

Another commonly used technique is to rely on obscurity to protect your data, and to rely on the fact that the Internet is a pretty big place and people will not find your data unless they have been told where it is. An example of this is the removal of read permissions on directories on ftp servers. This has the effect that users can retrieve a file only if they know its name (and cannot use *ls* or *mget* to help) so the user is assumed to be authenticated by the fact that they know the name of the file to be retrieved.

Obscurity is a very bad strategy for security and should not be considered under any circumstances.

4.4 Protected Authentication

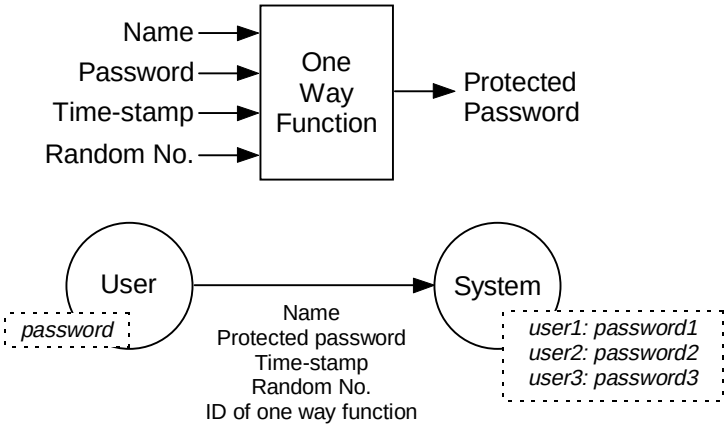
The following authentication mechanisms are based on shared secrets, but do not rely on clear text exchange of those secrets. They provide significantly better security than passwords on their own.

4.4.1 X.509 Protected Passwords

ITU-T X.509 (formerly known as CCITT recommendation X.509) gives details of a method of allowing a directory user agent (i.e. person using the directory) to send a password to a directory system agent (the computer that is part of the directory infrastructure) in a safe manner that is not vulnerable to interception or replay.

With simple authentication, the user simply sends a name and password to the system in clear text. However, with protected authentication the name and password are first given as input to a one way function, such as one of the hashing algorithms discussed earlier. The output of the hashing function is a protected password, and the user passes this to the server along with their name and details of which hashing function was used. The system then gets the local copy of the user's passwords and re-computes the protected password using the same hashing algorithm. If the protected passwords match then the original passwords must have matched and so the user is authenticated.

This mechanism can be protected against replay by including a random number and/or a time-stamp in the data used to calculate the protected password. This information would be included in the data sent from the client to the server, and the server would additionally check that there were no repeats in the random number (within a predefined period of time) and that the time-stamp was monotonically increasing.



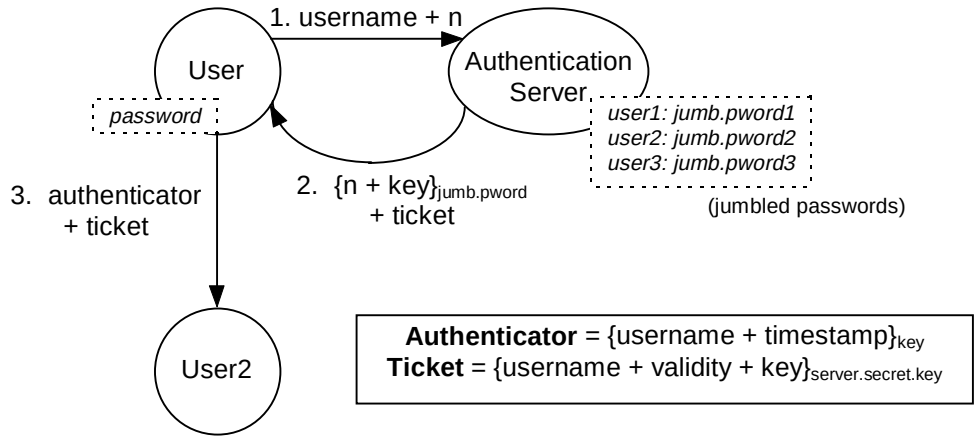
Although this mechanism is specified for directories, it generally applies to all mechanisms that need to send passwords over untrusted networks.

4.4.2 Kerberos Authentication

The Kerberos authentication mechanism introduces the idea of an authentication server. The authentication server holds passwords for all users within a particular community - the passwords have been jumbled (passed through a one-way hashing function) before storage.

To use Kerberos to authenticate a user, the following sequence of actions are performed:

1. The user sends an authentication request to the authentication server, including with the request its name and a random number.
2. The server invents a one-off session key, appends the random number and encrypts this with the jumbled password of the user. This is returned to the user along with a ticket which contains the user's name, a validity period and the one-off session key, all encrypted with a secret key known only to the authentication server.
3. The user then jumbles their own password and uses the result to decrypt the one-off session key (checking that the random number returned is the same as was sent). The user then constructs an authenticator which is the user's name and a timestamp, encrypted with the session key.
4. The combination of ticket and authenticator can then be used by the user to be authenticated to any other person or process that uses the same authentication server. The ticket and authenticator are sent along with any communication, and the recipient passes them back to the authentication server to determine the authenticity of the user.
5. The authentication server decrypts the ticket using its secret key (only it can do this) and this reveals the one-off session key and the user's name. The server then decrypts the authenticator using the session key and reveals the user's name. If they match then the user is authenticated because it proves he knew the correct password to decrypt the session key in the first place.



Kerberos authentication is a very elegant method, especially as the authentication server does not need to know the password of the users but merely a jumbled form. It is extremely suitable to small organisations operating a single LAN (and has been used very successfully in many applications this environment). However, there is no mechanism for multiple inter-operating authentication servers in a WAN environment unless all authentication servers are required to trust each other.

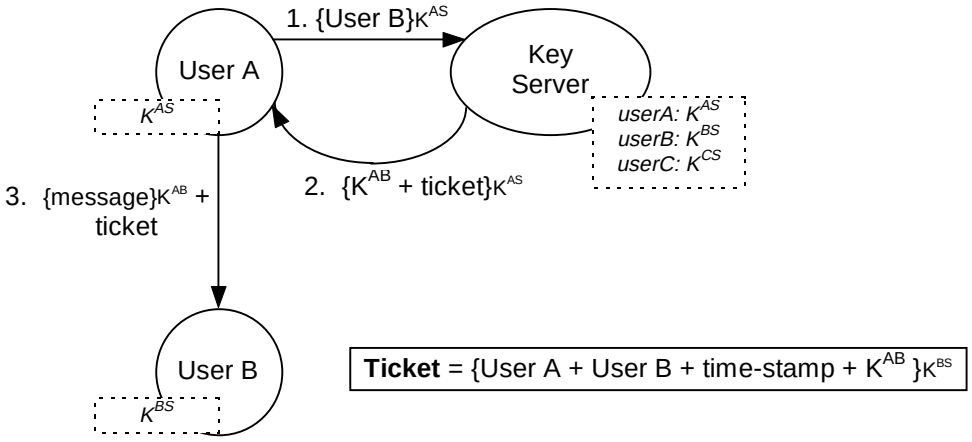
Kerberos Confidentiality

Although this study is focusing on authentication, it is worth briefly examining the mechanisms that Kerberos provides for providing confidential communication between two users. For confidentiality services, Kerberos introduces the idea of a key server. Every user of the key server has a secret key that is known only to them and to the server.

To use Kerberos to establish secure communication between User A and User B, the following sequence of actions are performed:

1. User A sends a session-key request to the server, containing User B's name encrypted with the key shared between the server and User A (we'll call this K^{AS})

2. The server invents a session key for User A and user B to use (K^{AB}), and returns this to User A along with a ticket, all encrypted with K^{AS}. The ticket contains the session key, the names of User A and User B and a timestamp, all encrypted with K^{BS}.
3. User A decrypts the session key K^{AB} and the ticket using K^{AS}. User A then sends a message to User B encrypted with the session key, and appends the ticket.
4. User B decrypts the ticket using K^{BS}, and reveals the session key K^{AB}. B can then decipher the message from A



4.4.3 One-time Passwords

A one-time password is a password that is used once for authentication. Capture and replay are not effective because once a password has been used once, it is useless. There are a number of novel hardware and software implementations of one-time passwords and these tend to fall into two categories: time synchronised and challenge-response. Hardware implementations will typically require the user to carry a small credit-card sized password card, access to which typically requires the entry of a PIN number.

Time Synchronised

With a typical time-synchronised method of authentication, both the password card and the target system compute a new password every 30 seconds, according to a pre-defined algorithm which uses the date, the time and a shared secret to seed a random number generator. The user calculates the current password and sends this to the target system along with their name. The system uses the shared secret to calculate the current password and check that the same value is obtained.

There are several problems with time-synchronised authentication, the largest of which is clock drift. It is very difficult to maintain the clocks in the card and the target system with sufficient accuracy, and if the clocks drift by 5 seconds then it is only possible to be successfully authenticated for 25 seconds out of 30. Furthermore, if an authentication attempt fails then the user must wait 30 seconds to re-attempt (as it is not possible to attempt twice with the same password). These drawbacks have made the technique less useful in some environments.

An example of a time-synchronised one-time password-based product is “SecureID” from Security Dynamics. The “Gauntlet” firewall from Trusted Information Systems is also capable of this type of authentication.

Challenge-Response

With a hardware-based challenge-response method of authentication, the target system sends a challenge to the user (usually a numeric or an alphanumeric string), the user types this into their password card which calculates a response based on a pre-defined algorithm involving a shared secret value. The user returns this response, and the target system compares it against a locally calculated value. If they match then the user is authenticated.

An example of a challenge-response one-time password-based product is “SecureNet” from Digital Pathways. The “Gauntlet” firewall from Trusted Information Systems is also capable of this type of authentication.

S/Key

S/Key (developed by Bellcore and published as Internet RFC1760) is a software based challenge-response one-time password system with some quite novel characteristics. It is based on the fact that the output of a hashing function is a 128 bit summary of the input, that can be hashed just like anything else.

So, to initialise the S/Key system, the user and the target system on a password as a shared secret; and the target system decides on the number of times this password will be allowed to be used (say 1000 for this example) and invents a random seed for each user. It combines the password and the seed in a predefined way, and hashes this 1000 times. In its records for the user, it therefore stores $\{pw,seed\}HASH^{1000}$ and the number 1000.

When the user needs to be authenticated to the system, the system issues a challenge {seed,999} (i.e. the seed and a number one less than the number held in the user’s record). The user calculates a response by combining the password and the seed in the standard way, and hashing the result 999 times to get $\{pw,seed\}HASH^{999}$. This is passed back to the target system, which hashes the result once more and compares it to the value held in the user’s record. If it matches then the user must have known the correct password and so is authenticated. The target system then overwrites the hash value held in the user’s record with the value $\{pw,seed\}HASH^{999}$ sent back by the user, and decrements the required number.

The next time the user authenticates, the challenge is {seed,998} (again, one less than the number held in the user’s record), and the response is $\{pw,seed\}HASH^{998}$.The target system hashes the response once more, compares the result and, if successful, overwrites the users record again with $\{pw,seed\}HASH^{998}$ and decrements the counter again.

So the sequence continues until the counter reaches zero, after which the user can no longer be authenticated and will need to get the system re-initialised

S/Key is a very powerful means for controlling access to systems, as it allows the user to be charged in advance for a predetermined number of accesses. The bulk of the computation work takes place in the user’s software and so leaves the server free to accept a high load.

4.4.4 Message Authentication Code

A Message Authentication Code (MAC) is a cryptographically calculated checksum of an electronic message which, when transmitted along with the message, proves the authenticity and integrity of the message to the intended recipient. A common method of calculating MACs is to use DES in cipher-block-chaining (DES-CBC) mode (where each block of plain-text is exclusively-ORed with the cipher-text from the previous block, to ensure that blocks cannot be inserted, removed or reordered and to ensure that repeating characters get differently enciphered).

To calculate a MAC, both the sender and the intended recipient of the message share a 56 bit DES key as a shared secret. The message is divided into 64 bit blocks, and the shared key is used to encrypt the first block. The result of this is used as the key to encrypt the second block, the result of that is used as the key to encrypt the third block, and so on. After all blocks have been encrypted in this way the result is a value characteristic of both message content and sender.

Protection against replay can be added if the message includes a time-stamp and/or a random number.

4.4.5 Message Digest Authentication

Integrity of a message can be proved by calculating a hash value of the entire message and securely transmitting this along with the message. However, if the sender and recipient of the message have a

shared secret (e.g. a password or phrase) then this can be appended to the message before the sender and recipient calculate the hash, thereby providing authentication as well. An interceptor would not be able to modify the message or forge an identity, as the shared secret would be needed in order to calculate the correct hash value.

Protection against replay can be added if the message includes a time-stamp and/or a random number.

There is an Internet-draft describing a mechanism for a protected authentication scheme for the HTTP protocol to replace the cleartext username/password that is used in HTTP/1.0. The mechanism is called “Digest Access Authentication”, though it is more of a challenge-response one-time-password scheme rather than a message digest authentication scheme. The WWW server issues a random number (a nonce) as a challenge, and the user responds with an MD5 hashed combination of username, password, nonce, HTTP method and requested URL. Therefore, the password is never sent in the clear, and the server can reconstruct the hashed value using its locally held version of the password.

4.5 Strong Authentication and Public Key Infrastructures

The use of public key cryptography has made possible many developments in security technology. If each user can make their public key completely public (and can keep their private key completely private) then strong authentication, integrity, confidentiality and non-repudiation are possible in many different applications, removing many of the threats posed by the use of open networks.

However, this does not come for free, and some additional infrastructure is needed for publishing a public key in a safe and reliable manner. It sounds like a relatively simple matter, but in practice it introduces many new problems of its own. The problem is that all of the mechanisms available for advertising public keys are insecure themselves (X.500 directories, finger files, WWW pages and DNS are mechanisms that are commonly used or considered). It is not possible merely to store a public key in these mechanisms as the key can be modified in several ways that would breach security (if I can make the world believe that a private key I have belongs to someone else then I can create digital signatures that people will believe were signed by them, and I can decrypt their encrypted email).

The solution to this problem is certification, where rather than advertising your public key you advertise your public key certificate. This is a package combining your identity, your public key, validity information and a digital signature appended by a third party (a certification authority (CA)) confirming their belief in the binding between the identity and the public key for the interval of the validity. If a user requiring someone else's public key knows something about the CA that signed the certificate, then they can determine if the certificate has been maliciously or accidentally modified.

This certification mechanism introduces the extra problem of Trust Models: what individual users are expected to know about these certification authorities; who (if anyone) individual users have to trust in order to obtain useful results; and what mechanisms are needed to allow two users who have never previously met and have not prearranged the conversation to establish a useful level of assurance about the identity of each other.

4.5.1 X.509 Certificates

ITU-T X.509 (formerly known as CCITT recommendation X.509) defines a format of certificates that is widely used in commercial and research environments. Early versions of the standard (version 1 was published in 1988) have been superseded and version 3 (to be published in 1997) is now the preferred version, even though it is not yet stable.

The standard has a discussion on how to use certificates to obtain public keys. For a general communication between two users A and B, A wishes to know the public key of B. A therefore retrieves B’s certificate signed by B's certification authority (we can call this CA(B)). The problem can now be seen to be recursive, as the problem has now become one of discovering the public key of CA(B). A can retrieve a certificate of CA(B), CA(CA(B) and CA(CA(CA(B))) and so on until it finds a certificate that was issued by a CA for which A knows the public key (this typically means CA(A),

the CA that issued A's own certificate). The list of CAs between A and B is known as a certification path. In order to be sure of the binding between the identity of B and the public key of B, user A simply has to trust the CA that issued A's certificate. A must also be sure that A holds a correct and up to date version of CA(A)'s public key (this is always assumed to have been passed securely and out-of-band (i.e. not via electronic means)).

A certification path logically forms an unbroken chain of trusted points between two users wishing to communicate. User A may end up with a mechanism for determining that user B owns a particular public key, but extra information is needed in order to know how much assurance can be placed in this information. Simply checking certificates is merely a mechanical process with an end result of saying either “I do believe that Public Key K belongs to someone claiming to be ‘User B at Organisation X’” or “I do not believe it” (i.e. a simple Boolean answer). However, there is an extra vital stage in public key authentication which produces an answer “I have the following degree of confidence that ‘User B at Organisation X’ is an accurate description of the person who actually owns this public key” and the level of confidence can be absolutely any value between “None at all - the name is a pseudonym” to “Very high indeed”.

In order to calculate this degree of confidence, User A needs to know what the policies of all of the CAs that form the certification path are towards proving the identity of the user. If one of the CAs has a policy that it will issue certificates to anyone who asks without checking who they are then the chain of trust is meaningless as people can be certified as being someone who they are not. If User A needs to be really sure of who User B actually is then only CAs that implement the most stringent possible checks will be trusted

In general, establishing a certification path in this way between two arbitrary users who have never met will be difficult. Unless each CA issues many certificates for other CAs (these are called cross-certificates), such a path is likely not even to exist (and is rarely guaranteed to exist). If CAs do issue a lot of cross-certificates then determining which to use at any time is a non-trivial matter. If you also want to only use CAs which provide a minimum level of assurance as to the identity of the users they certify then this complicates the process further.

A useful simplification mentioned in the X.509 recommendations is to arrange CAs in a hierarchy, as this makes certification paths between two users much easier to determine. A hierarchical approach has been formalised in RFC1422 for use by the PEM secure email protocol.

4.5.2 RFC1422 – The PEM Trust Model

The Privacy Enhanced Mail (PEM) secure electronic mail protocol (defined in RFCs 1421 to 1424) defines a strict hierarchy with three or more levels. RFC1422 defined the model for determining trust, requiring the use of X.509 version 1 certificates.

At the top level of the hierarchy is a single certification authority which merely ties together all of the CAs at the level below without making any statements about what they do. The Internet Society has set up the 'Internet Policy Registration Authority' (IPRA) to fulfil this service. The EC PASSWORD project also set up a 'Top Level CA' (TLCA) for its piloting phase.

The second level of the RFC1422 hierarchy comprises Policy CAs (PCA). These are CAs which certify other CAs, guaranteeing that they only certify CAs that enforce a minimum security policy among the things (users or other CAs) that they certify, where elements of specifiable policy include allowable encryption and hashing algorithms, minimum key sizes, maximum certificate validity periods, procedures for identifying users, procedures for revoking certificates. PCAs form certification islands and these were typically expected to coincide with national boundaries, sensitive organisations or vertical market sectors.

At the third level of the RFC1422 hierarchy, you get Organisational CAs. These issue certificates for other users and possibly lower level departmental CAs according to their stated policy which is enforced by the PCA which certifies the organisational CA. Some CAs may be high assurance CAs which take enormous care to verify that a user is who they claim to be before signing their certificate,

including possible DNA fingerprinting or other techniques. Some organisational CAs may be medium assurance, requiring a visit to a particular office and a visual inspection by someone who both knows you and is trusted by the CA to authorise the issue of the certificate. Some low assurance CAs may just require a telephone call or an email message, and some persona CAs may explicitly issue certificates based on pseudonyms or anonymity.

So, in order to verify the public key of user B, user A must obtain the certificate for B issued by CA(B), the certificate for CA(B) issued by PCA(CA(B)) and the certificate for this PCA issued by the CA at the top of the hierarchy. Because this certification path is so well defined, it can be provided by all users as a matter of course to allow other users do verification without needing to look up certificates in directories. User A is assumed to know the public key of the CA at the top of the hierarchy (this is assumed to be published in so many places that modification of them all is not practical) and to trust the CA at the top of the hierarchy (which, after all, is not doing anything other than tying together the second level of the hierarchy).

User A now has a certified copy of the public key of user B, and can make an assessment of how much assurance can be placed in the identity of user B by examining the policy of PCA(CA(B)). This means that user A needs to be able to find out the policy of that PCA, and user A also needs to trust PCA(CA(B)) to correctly enforce the policies it claims to enforce.

The process of verifying a signed message therefore becomes:

This is certified with medium assurance as from "Andrew Young, Salford University" because.....

1. The message can be verified using public key xxxxx
2. The certificate shows that "Andrew Young, Salford University " owns key xxxxx
3. "Salford CA" signed certificate with key yyyyy
4. The CA certificate shows that CA "Salford CA" owns key yyyyy
5. "UK Academic PCA" signed Salford's CA certificate with key zzzzz
6. The CA certificate shows that CA "UK Academic PCA" owns key zzzzz
7. "Top Level CA" signed CA certificate with key qqqqq.
8. I believe that qqqqq is the key of the "Top Level CA"
9. I trust the "Top Level CA
10. I believe that CAs certified by the "UK Academic PCA" provide at least medium assurance as to the identities of the users they certify.

4.5.3 The PGP Trust Model

A different model of trust is used in the **Pretty Good Privacy** (PGP) program. PGP assumes that only individual users are competent to decide who to trust and consequently that all users are competent to decide the implications of trusting someone and to assess the levels of assurance that can be inferred. PGP has defined its own certificate format which is incompatible with the X.509 standard (though there are rumours that a version of PGP that uses X.509 certificates will be produced in the future).

If user A wishes to communicate with user B then each sends the other their public key by insecure means, packaged in a proprietary certificate format. It is at least self-signed, and may possible be signed by any number of other people as well. Users A and B then compute a fingerprint value from their keys (a hash value or checksum) and communicate this to each other via a different insecure method as was used for communicating the key (different means the difference between via email and via telephone, rather than via email and via finger). Users A and B then compare the fingerprints that they have been sent with the fingerprints they have computed from the key they have been sent and, if they match, they assume that the key is good. Each then signs the key of the other user with their own key and adds it to their personal security environment (known as a public keyring) so that further secure communication between users A and B will not need this preamble. User A can optionally send back to user B the key of user B signed by user A (and vice versa of course). The users can than add this new signature to their public key certificate, and we will see how this can be used shortly.

The above description bears a great deal of similarity to a description of how two CA operators issuing X.509 certificates would cross-certify each other, or how a CA would issue a certificate for a user. In PGP's case, the cross-certification is being done at the user level, the assurance of user

identity is almost always either low or persona (anonymous), and the protection against security attack during key exchange is as strong as the two distinct methods of distribution of key and fingerprint. CAs cross certifying each other need to solve these problems as well, and the only hope is that people operating commercial CA services know precisely what the issues and implications of the parts of the process are. Two users who have simply downloaded PGP from an FTP site may not have this expertise.

The reason that users get their keys signed by other users and add these signatures to their own certificates is due to the way in which the PGP trust model works. If user A has obtained a copy of user B's key that is known to be good, and if A knows B well enough to make an assessment of B's competence in security matters then A can nominate B to be either completely trusted or semi-trusted. A can then configure PGP to accept all certificates that are signed by a certain number of completely trusted people or a certain (presumably greater) number of semi-trusted people. So if user A wants to communicate with user C and has obtained a copy of user C's certificate from a key server (see below) or a directory, then if user C's certificate is signed by user B then user A will trust the user C's certificate since user B has vouched for the identity and key of user C. If user C has also declared user B to be trustworthy then user C will trust user A's certificate since in the exchange described at the top of this section user B signed the key of user A, and user A added this to A's certificate.

Note that if A tries to communicate with user D, and user D's certificate is signed by user C, then user A will not trust that certificate as trust is not transitive. User A must be convinced of user C's competence before explicitly adding user C to the list of trusted users.

The process of verifying a signed message therefore becomes

1. The message can be verified using public key xxxxx which claims to be "Andrew Young" because.....
    2. Key xxxxx is signed by user "A.Other" with key yyyyy
    3. I trust user "A.Other" to be competent to vouch for the identity of others
    4. I believe that yyyyy is the key of "A.Other"
    5. I require one trusted user in order to believe a signed certificate

This approach to trust works well in very small groups of users, for example a small company where one person signs the certificates for people in the company and everyone declares that person to be trusted. Of course, what we have here is an organisational CA which signs everyone's keys and which people within the organisation trust. The PGP model can be made to greatly resemble the PEM model. However, scaling the model to arbitrary communities is not easy, since user A from company AA and user E from company EE have no common point of trust. What is needed is for the completely trusted person within company AA and the completely trusted person within company EE to both be certified by someone else, an equivalent of a higher level CA, perhaps a PCA. However, current versions of PGP only allow a single certificate to be attached to a message and so these certification paths cannot be conveyed with the message.

The PGP trust model is supported by a simple infrastructure, where "**Public Key Servers**" are placed in well defined places on the Internet, and these servers respond to email requests to add a key or to retrieve a key for a named user. The implementation of these key servers is entirely centralised, with all servers are assumed to hold identical data and to synchronise among themselves. The key servers currently hold over 25000 keys, and the performance under this load is about to degrade to a level where they are no longer usable. This is clearly a non-starter in an environment where you would hope for tens of millions of keys to be made available. A proposed extension to the UKERNA 'Secure Email Project' aims to develop a new centralised key server capable of handling a million keys without performance problems. However, a scaleable distributed approach seems the only way forward in the long term; though this seem to be hampered by the apparent need of the infrastructure to look-up information based on key value as well as user name.

4.5.4 PEM/PGP Trust Model Comparison

Over the last few years, both of these trust models have been used for certain applications and each has gathered a number of advocates and detractors. Neither of the models has met the needs of the

general user community. However, experience from using the models forms a useful body of knowledge to be used when considering the problem of designing a trust model that can be more generally useful.

In use, a number of observations can be made about the PGP trust model. Almost all trust between users is conveyed by the mutual exchange of keys prior to the initiation of the first communication. The “trusted introducer” idea is rarely if ever used, for two reasons. Firstly, many people are not willing to add their signature to a certificate that will be lodged in a public place for use by others because of the fear that some liability could be attached to that signature if it is misused for any reason. Secondly, use of the PGP public key servers is discouraged by their poor performance and so even if you declare someone to be a trusted introducer, the chances of finding a certificate signed by them are low. It turns out to be easiest to exchange keys directly with the person you want to communicate with and add their key directly onto your own public keyring after verifying its fingerprint, and so this is what is normally done.

The PGP model therefore leads to a graph with arbitrary user to user links where certificate chains are either of length 1 or do not exist. Its use to create a security domain within a small organisation is efficient, its use to create a security domain within a large company is rather inefficient, its use to create a link between two separate security domains is more inefficient, and the problems multiply as the size of security domains grows. The model therefore offers great power to users and little power to organisations.

A number of observations can also be made about experience from use of the PEM RFC1422 trust model. One reason why the model has not been more widely used is that a considerable infrastructure is required to exist and be functioning before users can start to make meaningful use of the services. In particular, a root CA must act as a registry of information regarding Policy CAs and a starting point for certification paths. For a variety of reasons, the proposed Internet Society IPRA (top level CA) did not start at the advertised time and has not been widely used. The lack of this service has meant that a number of other hierarchies have come into existence for specialised applications and the IPRA hierarchy has been correspondingly less useful.

As well as requiring the pre-existence of the top level registration authority, RFC1422 users are required to use the services of pre-existing Policy CAs (or to set up and register their own). Two users wishing to make use of security services would need to create organisational CAs and register these with a PCA before issuing themselves with certificates. This significant start-up cost makes use of the model impossible for some and impractical for others.

The RFC1422 model leads to a graph with links between organisations and PCAs and (in theory) between PCAs and a root. PCAs naturally create certification islands where a common security policy is enforced, and organisations' CAs naturally create security domains of arbitrary size and complexity. The model therefore offers great power to organisations and little power to users.

The two models can therefore be seen as user-centric and organisation-centric, and there are few options for interoperability between the two. However there are clear market niches for each model, and neither model will suffice on its own.

4.5.5 SESAME

The SESAME project opted for a slightly different approach to managing a public key infrastructure, and proposed mechanisms to let a single certification authority serve an arbitrarily large community of users by distributing the functionality of the CA. A centralised CA can easily get overwhelmed by the job of managing the certificates for a large community of users, and this is especially important in an academic environment where certificates for all students must potentially be reissued every year (if it takes three minutes to take a public key from each student (via a floppy disc), examine the student’s identity card, sign the public key, return the certificate (via the same floppy disc) and add it to a directory service or key server, and if an institution has 10000 students who are each re-certified each year on payment of tuition fees, then it would take 72 working days to issue all certificates).

Distribution of this effort is clearly needed, and the SESAME mechanism provides an alternative to a hierarchical approach (and, of course, the two can be easily combined as well).

With SESAME, each CA can delegate responsibility to any number of Local Registration Authorities (LRA) (in an academic environment, each department could have its own LRA).
The LRA is responsible for identifying the users requesting certification, and may also be able to issue and escrow key-pairs for the user if the user has not done this him/herself. The LRA takes the public key of the user and sends a signed copy of this to the CA Administrator (CAA) (authenticated either via a shared secret or via the CAA issuing a certificate for the LRA). The CAA then has the purely mechanical job of issuing the certificates, returning them to the LRA and publishing them in the directory (this latter task can, of course, be delegated to the manager of the local directory system). The actual CA itself would normally be located on a machine physically removed from the network in order to make sure the important private key of the CA was not vulnerable. The CAA would periodically transfer (via floppy disc) a batch of certification requests signed by LRAs to the CA machine where they would be mechanically signed. Automated software can make this part of the CAA's job quite straightforward.

4.5.6 X.509v3 Certificates and X.509v2 CRLs

The original version of X.509 was published in 1988, and this described so called "version 1 certificates". In these certificates, the CA signed a package of information comprising:

- Issuer (i.e. CA name);
- Serial Number (i.e. unique sequence for each issuer);
- Validity (two dates giving start and end of validity period);
- Subject (i.e. name of user of CA who the certificate is for);
- Subject public key (the key that is being certified);
- Algorithm used for the signature.

These certificates gained widespread use. In use, a few shortcomings became apparent (especially with regard to the naming of CAs and users, which was constrained to the X.500 naming scheme), and the 1993 version of the X.509 standard defined "version 2 certificates" which allowed the optional specification of unique numeric IDs for the subject and issuer of a certificate. These fields provide a 'unique name' functionality in the presence of re-use of conventional names, which had previously been done solely through the names themselves (this allows naming administrators to reuse a name without compromising the uniqueness of an identifier, for example to cope with multiple employees called "J.Smith").

Even before this was published, work was starting on "version 3 certificates", and so there has been no significant deployment of version 2 certificates with most people waiting for version 3 which is due to be formally published in 1997 (and is still not completely stable at the time of writing in June 1996). Version 3 adds a very important mechanism called "extensions" which allow communities of users to specify extra information to be included in the certificate and to mark important information as 'critical' (so that other users who cannot process that information know that this constitutes a failure of the verification process). The standard includes a number of predefined extensions, and communities of users are able to define and use their own.

The standard set of extensions include the following:

1. **Key and Policy Information Extensions**
    - **Authority Key Identifier**: this extension provides unique identification of the public key to be used to verify the signature on the certificate, with identification either via a unique key identifier or by an issuer name/serial number pair.
    - **Key Usage**: this extension indicates the purpose(s) of the certified public key. If flagged critical then the key may only be used for that purpose, otherwise it just provides an indication of the purpose of the key. Defined uses are
        - creating digital signatures;
        - signing information intended to support non-repudiation claims;
        - enciphering encryption keys;
        - enciphering data;
        - key agreement protocols;

    - signing certificates;
    - signing CRLs.
  - **Certificate Policies**: this extension indicates the certification policy of the certificate issuer (what is given is the object identifier of the policy rather then the policy itself or even the location of it). If it is flagged as critical then the certified public key shall only be used for the purpose and rules described in one of the indicated policies. If it is flagged as non-critical then use of the certificate is not necessarily constrained to the policies listed.
2. **Subject and Issuer Attributes Extensions**
  - Subject Alternative Name: this extension gives one or more alternative name forms for the subject of the certificate. Defined name forms include:
    - RFC822 SMTP email address;
    - X.400 email address;
    - directory name;
    - EDI party name;
    - WWW URI;
    - DNS name;
    - IP address
  - Issuer Alternative Name: this extension gives one or more alternative name forms for the issuer of the certificate, with the same name forms as given above.
3. **Certificate Path Constraints Extensions**
  - Basic Constraints: this extension indicates whether the subject may act as a CA and issue certificates to other users and CAs. It specifies whether the certified public key can be used to verify certificate signatures, and limits the depth of a certification subtree under a subject CA.
4. **CRL Distribution Points Extensions**
  - This extension contains information about where to retrieve the CRL that is to be used in the verification of this certificate (using the name forms as indicated above for issuer and subject). If this extension is flagged critical then the certificate shall not be verified without first retrieving and verifying the corresponding CRL.

The version of X.509 published in 1988 contained no description of Certificate Revocation Lists (CRL). The version published in 1993 contained a specification of “Version 1 CRLs”. In these, the CA signed a package of information comprising:

- Issuer (i.e. CA name);
- Time and date of issue of this CRL
- Time and date of planned issue of next CRL
- Algorithm used for the signature.
- List of serial numbers of revoked certificates with time and date of revocation.

However, deployment of this standard did not really happen for the same reason as for version 2 certificates. The forthcoming version of X.509 due to be published in 1997 contains a specification of “version 2 CRLs” which mirrors the extensibility mechanisms in version 3 certificates. Again, the standard includes a number of predefined extensions, and communities of users are able to define and use their own.

The standard set of extensions include the following:

1. **CRL Extensions**
  - CRL Number: this extension contains a sequentially increasing sequence number for each CRL issued by a given CRL issuer. It allows a CRL user to detect whether CRLs issued prior to the one being processed were also retrieved and processed.
2. **CRL Entry Extensions**
  - Reason Code: this extension describes the reason for revocation of the certificate. Defined reasons include:
    - key compromise;
    - CA key compromise;
    - user’s affiliation changed;
    - certificate superseded;
    - CA cessation of operation;

- certificate hold (use is deprecated)
- remove from CRL (use is deprecated).

Use of X.509v3 certificates and X.509v2 CRLs provides the users of a Public Key Infrastructure a lot more information on which to base their decisions regarding user identity, and gives system administrators much more control over the use that keys can be put to, and all current research in X.509 is assuming the use of these formats (even though there are currently very few implementations)

4.5.7 The IETF PKIX Working Group

The Internet Engineering task Force (IETF) formed a working group in 1995 to design an infrastructure for the use of X.509v3 certificates and X.509v2 CRLs in the Internet.

The PKIX group is currently writing a series of Internet Drafts on various aspects of the problem, and there are currently four documents scheduled:

Part 1: X.509 Certificate and CRL Profile
Part 2: Distribution and Access to Certificates and CRLs
Part 3: Certificate Management Protocols
Part 4: Policies

with parts 1 and 3 currently in draft form.

The working group does not plan to do any work on integrating the infrastructure with secure applications and, more surprisingly, does not plan any concrete proposals on a trust model describing how a collection of certificates and policies are concerted into a level of assurance of identity (though the ICE-TEL project is contributing its trust model to the working group as a contribution - see the section below).

4.5.8 The IETF SPKI Working Group

In 1996, a new working group was formed within the IETF to design a “simple certificate format and public key infrastructure”. The motives for forming the group seem to be largely political (anti-ASN.1 and anti-OSI seem to be the prime motivating factors), although in the area of adding security to email and the World Wide Web, the Internet Community has shown a remarkable lack of success in its efforts to design and deploy effective de-jure standards and it is possible there is a prevailing view that the SPKI working group will fall into a similar impasse. The view of the SPKI group seems to be that the Internet has always thrived on de-facto standards with simple (usually ASCII-based) formats (with SMTP, PGP and WWW as the three main successes) and so that is the best way to work.

So far, the SPKI group has not published any draft documents. The basis of the discussion so far has been to argue that there is a need for more than the “identity certificates” that X.509 is seen to provide, and that “authorisation certificates” are needed to convey privileges. In terms of the old operating-system arguments from the 1970s, it seems that they are moving towards a “capability” model of certification rather than an “access control list” model. Current thoughts are for using SDSI (Simple Distributed Security Infrastructure) by Ron Rivest and Butler Lampson as a public key infrastructure. The motivation of this is that current global certification infrastructures are impractical because they rely on a global name-space which cannot be guaranteed. SDSI includes a means of defining groups and issuing group membership certificates, emphasising linked local name spaces rather than a hierarchical global name space, and provides terminology for defining ACLs and security policies.

4.5.9 The EC ICE-TEL Project

As part of the EC Framework-4 programme, the ICE-TEL project (Interworking Public Key Certification Infrastructure for Europe), which started in December 1995, aims to provide a large scale public key certification infrastructure in a number of European countries for the use of public key based security services, and to provide all technology components which allow the deployment of user tools and applications with a common integrated public key security technology. The ICE-TEL consortium includes many of the partners from the EC VALUE programme PASSWORD project (which ran during 1992-3 and was followed by a year long pilot phase) and it intends to use

components from PASSWORD as well as from the RACE SESAME project which is also represented among the consortium members.

The PASSWORD project created a pilot security infrastructure based on X.509v1 technology and RFC1422/PEM trust models for the European research community, and piloted secure applications with users. The project developed three interworking security toolkits and produced documents defining the requirements and policies necessary for such a pilot scheme. An important aim was to use the existing X.500 global directory infrastructure for the registration of Certification Authorities and lodging of public keys. While not directly relevant to the pilot, this was crucial for large-scale open deployment.

The improvements to be made to the original PASSWORD infrastructure by the ICE-TEL project are as follows:

- address the needs of inexperienced users;
- incorporate recent improvements in security technology, such as smart-cards and X.509v3 certificates;
- incorporate developments in related areas, such as the latest X.500(93) directory standards;
- develop a new trust model to be applicable to a wide range of users and organisations, and to be able to interwork with other trust models;
- support as many different platforms as is practical;
- integrate security into as many popular user agents as possible;
- provision of serious CA services rather than pilot services;
- provision of secure WWW based applications;
- provision of secure MIME email support.

The ICE-TEL trust model was designed to allow an arbitrary combination of PEM style hierarchies, PGP style webs of trust and other intermediate combinations to interwork. A major requirement was for organic growth, where an organisation with small security requirements could set up a small infrastructure at very low cost; and could grow this into a larger infrastructure when required, with minimum inconvenience to other users. The trust model was specified on the basis of the following six requirements:

1. The trust model shall be capable of operating without the use of certificates or CRLs, through trusted exchange of public keys.
2. Where certificates are used, the trust model shall require the use of the X.509 standard v3 certificates and v2 CRLs expected to be ratified in 1997. Earlier versions of the certificate and CRL may be supported by implementations while they remain in widespread use. Proprietary certificate and CRL formats shall not be supported.
3. The trust model shall allow for the creation of security domains encompassing
   - single users
   - multiple users (small/simple organisations); and
   - arbitrarily complicated organisations.
4. The trust model shall allow organic growth among the users and organisations; and shall allow security domains to grow, shrink and reorganise at any time with minimum inconvenience to the users of the domain and to people communicating with those users.
5. The trust model shall allow the administrator of a security domain to choose which other domains are to be trusted. There shall be no requirement that inter-domain trust be mutual and there shall be no points that any domain is required to trust. Inter-domain trust shall not be transitive.
6. The operation of the trust model as a whole shall not depend on the existence and operation of any single part of the deployed infrastructure. In particular, there need not be a central top level registration authority like the PEM IPRA

The model allows a network of users and CAs to be built up where each CA has a clearly defined policy of who it is allowed to certify. CA hierarchies can be built up, and users in different hierarchies can certify each other through cross-certification between the hierarchies. A CA has a stated set of security policies it requires from other CAs before it will be allowed to cross-certify. Therefore, a user joining a particular hierarchy knows in advance the minimum assurance that can be expected from a user being authenticated via certification or via cross-certification. Transitive cross-certification is

explicitly prohibited, thereby ensuring that the administrator of a particular security domain has complete control over who is trusted by users in that domain.

The ICE-TEL project is working closely with the international community and the World Wide Web Consortium in the area of WWW security, and intends to produce a selection of browsers and servers allowing signed and encrypted documents as well as authenticated retrieval. It has already done work in the area of secure email, and has implementations of both the PEM and MOSS protocols with integration into a number of user agents.

The first stage of an ICE-TEL certification infrastructure is due to be deployed in the summer of 1996, though it will initially be based on existing technology implementing X.509 version 1 certificates as implementations of version 3 certificates are not yet ready. The initial trust model will be a single project root, a second level CA per country and organisations at the third level (i.e. very much in the RFC1422 style). As the security toolkits and applications implement the new technology and trust model, this infrastructure will interwork with other pre-existing hierarchies and with new small scale security infrastructures as specified in the ICE-TEL trust model.

4.6 GSS-API

As well as the two working groups mentioned above looking at various aspects of public key infrastructure, another important working group within the IETF is the Common Authentication Technology (CAT) Working Group. The CAT Working Group is engaged in several activities involving definition of service interfaces as well as protocols. A prime goal is to separate security implementation tasks from integration of security data elements into caller protocols, enabling those tasks to be partitioned and performed separately by implementers with different areas of expertise. This strategy is intended to allow a single security implementation to be integrated with (and used by) multiple caller protocols and to define an abstract service which multiple mechanisms can realise. Therefore, it allows protocol implementers to focus on the functions that their protocols are designed to provide rather than on the characteristics of particular security mechanisms.

The CAT WG has standardised a common service interface - GSS-API (the Generic Security Service Application Program Interface) - and also a common security token format incorporating the means to identify the mechanism type by which security data elements should be interpreted. The GSS-API, comprising a mechanism-independent model for security integration, provides authentication services (peer entity authentication) to a variety of protocol callers in a manner which insulates those callers from the specifics of underlying security mechanisms. With certain underlying mechanisms, per-message protection facilities (data origin authentication, data integrity, and data confidentiality) can also be provided. This is published in a number of RFCs, with RFC1508 being the most important.

The CAT WG has also worked on underlying security technologies (and their associated protocols) implementing the GSS-API model; including Kerberos (RFC1510), the X.509 based Distributed Authentication Services (RFC1507) and the Simple Public Key Mechanism (currently at Internet Draft stage). Currently, the CAT WG are updating GSS-API and will shortly publish a much enlarged (and considerably more complex) version 2.

5. Current State of Authentication in Applications

This section describes the applications which would benefit from the addition of security, and looks at the current state of the art for each application.

5.1 Electronic Mail

Sending an email message between two people involves the following steps

1. The sender composes the message on the local machine. At this point, the message text can be observed in machine memory, temporary files or on the screen.
2. The sender submits the message to the message transfer system. At this point, the message can be observed or modified while it is being transmitted between two machines or while it is held in a temporary storage area on a intermediate mail spooling machine

3. The recipient displays the message. Message text can be observed in machine memory, temporary files or on the screen
4. The sender or recipient may also store a local copy of the message in some folder structure.

There are therefore many potential places where security of messages can be compromised, and where security will be required if email is to be used for serious purposes. Many secure email techniques have been developed, and these all solve the problems of step 2, while doing little to address steps 1 and 3, and even making step 4 rather more complicated.

Another potential security problem with email is the accidental misdirection of email caused by the sender typing in the wrong address. This is a problem both for graphical user interfaces as it is easy for mouse buttons to stick or slip, and for textual user interfaces where key transposition can produce another valid user name. This is especially a problem if organisations have short email addresses based on initials or allow the user to specify short aliases for long email addresses, as it is then more likely that a transposition will produce a valid name. Security mechanisms will not help here, as the sending mail system will be perfectly happy to encrypt a confidential message to the wrong person as long as it can find a public key for them. User education is the only answer here.

There are currently seven different and incompatible secure email protocols that have been proposed; and deployment of these has met with varying degrees of success:

5.1.1 PEM (Privacy Enhanced Mail)

PEM was designed by the IETF PEM working group, which formed in 1986 and delivered a series of drafts culminating in the publication of PEM version 4 as RFCs 1421-1424 in 1993. PEM has the following characteristics:

- It operates only on ASCII messages - of course binary files can be UUENCODEd, but this relies on the recipient knowing what to do with an encoded file. According to the standard, it operates on entire messages only, though most implementations can pick a PEM message out of the middle of a larger message and can even cope with nested PEM messages.
- Messages can be signed or signed and encrypted, but not just encrypted;
- Signed messages can be made clear if there is the possibility that the recipient is not a PEM user, but at the cost that there is a small chance they will not be able to verify the signature if they are.
- The standard includes the following components:
  - RFC1421 - Message Encryption and Authentication Procedures
  - RFC1422 - Certificate Based Key Management
  - RFC1423 - Algorithms, Modes and Identifiers
  - RFC1424 - Key Certification and Related Services
- RFC1422 includes a complete specification of the infrastructure and trust model that PEM is expected to operate in.

There are a reasonably large number of inter-operating PEM implementations around the world, but there are very few users of these implementations. It is fair to say that PEM took too long arriving and had missed the boat by the time it arrived, and the long drawn-out design process gave the protocol a bad name and many enemies before it was even finished. There were also technical problems with the deployment of the required certification infrastructure.

5.1.2 MOSS (MIME Object Security Services)

MOSS was designed by the IETF PEM working group and was originally intended to be PEM version 5 but the name was changed to avoid the new protocol being unfairly tarred with the same brush as its failed predecessor. MOSS was published as RFC1847 and RFC1848 in 1995, after which the PEM working group disbanded. MOSS has the following characteristics:

- A new MIME bodypart “multipart/signed” has been introduced to add a signature to an arbitrary MIME bodypart.

- A new MIME bodypart “multipart/encrypted” has been introduced to encrypt an arbitrary MIME bodypart.
- These two bodyparts may be arbitrarily combined and nested in any way. Thus, messages can be signed, encrypted or both signed and encrypted (in either order), and this may be done to whole messages or small parts of messages as required. As any bodypart can be signed (including multiparts of course), the security services are not limited to just text messages and the MIME user agent will automatically take care of displaying the protected bodypart in the appropriate way.
- Signed messages will always be in a form that is readable to a recipient who does not use MOSS (though they will, of course, not be able to take advantage of the security). MIME takes care of any end-to-end problems and ensures that this does not give a chance that the signature will be made unverifiable my mail gateway transformations.
- The standard includes the following components:
  - RFC1847 - Security Multiparts for MIME
  - RFC1848 - MIME Object Security Services

  The intention is that RFC1847 will describe a general mechanism for carrying security information in MIME, and that other MIME security protocols will use it and merely add in their own set of details.
- The MOSS specifications have removed the dependence on certificate based infrastructures and X.500 style naming, and it is now left to the user to plug into whatever key management scheme is available (and the PKIX and SPKI working groups are attempting to describe such a scheme)
- Two new MIME bodyparts have been defined for requesting public key information (application/mosskey-request) and for sending public key information (application/mosskey-data).

There are a very small number of inter-operating MOSS implementations around the world, with very few users of these implementations. The reasons that MOSS seems to be failing are not quite clear, though a lack of implementations is certainly contributing to it. While implementation of RFC1848 is straightforward, modification of MIME aware user agents to implement RFC1847 is rather difficult and this must be done once for every user agent in use.

5.1.3 PGP (Pretty Good Privacy)

PGP is a de-facto standard that originally started life as a standalone program for encrypting binary files, but has acquired capabilities for handling email over the years and has now become synonymous with secure email. PGP has the following characteristics:

- Messages can be signed, encrypted or signed-then-encrypted;
- Messages are compressed prior to encryption, as this not only saves space but makes the message much more resistant to crypto-analytic attack.
- Signed text messages can be made clear if there is the possibility that the recipient is not a PGP user, but at the cost that there is a small chance they will not be able to verify the signature if they are.
- The PGP documentation includes a complete specification of the infrastructure and trust model that PGP is expected to operate in.

PGP has been enormously successful among a certain class of Internet users. PGP places the user at the centre of the trust model and requires the user to manage their own trust relationships and (largely) to perform their own key management. There are many for whom this user-centric model is attractive and conforms to the Internet ethos and so, among the people who may be termed “power users” of the Internet, there is a substantial usage of PGP and a critical mass that will ensure it survives whatever the competition does. Among organisations, it is less popular as it is often seen to place too much power in the hands of the users and too little in the hands of the organisation:

- users are easily able to create their own keys and encrypt data belonging to the organisation (PGP encrypted data is, to all intents and purposes, unbreakable as it uses 128 bit IDEA encryption as opposed to the 56 bit DES-CBC encryption usually found in PEM and MOSS implementations)
- organisations are not able to enforce security policies, especially regarding the trust relationships that users are able to build up.

There is little commercial support for PGP products, though this is slowly changing..

5.1.4 PGP/MIME

There have been a number of attempts to integrate PGP and MIME, and these have largely mirrored the earlier attempts to integrate PEM and MIME. One common method is to define an "application/pgp" MIME bodypart, so that MIME may be used to carry a PGP message. The current favoured method (published as an Internet Draft and being considered as an IETF Proposed Standard) is to use the two security multiparts defined in RFC1847 and to use these as a building block for signing and encrypting using PGP. This re-use of RFC1847 is, of course, exactly what the MOSS designers had expected and it is very refreshing (and somewhat unusual!) to see competing security protocols working together in this way.

An integration of PGP and MIME is guaranteed to be a success in the secure email marketplace, though there is no guarantee that the current draft is the exact form it will take when it does appear. The advantages of secure MIME over secure SMTP are manifold and, although the effort needed to modify existing MIME mail user agents is great, PGP/MIME could provide the impetus needed to get it done. Ironically, this could help MOSS survive, as a MIME aware mail user agent able to handle security multiparts would be able to handle both PGP/MIME and MOSS.

5.1.5 S/MIME

S/MIME is a de-facto standard developed by RDS Data Security (the company formed to commercially exploit the RSA algorithm) as a way of carrying PKCS#7 and PKCS#10 format messages using MIME (The PKCS series are the Public Key Cryptographic Standards, defined by RSA)

S/MIME has the following characteristics

- A new bodypart (application/x-pkcs7-mime) has been defined for sending secure messages (these may be signed, encrypted or both)
- A new bodypart (application/x-pkcs10) has been defined for sending certification information.
- Signed text messages are not visible to recipients who do not have S/MIME. The recommended workaround for this is to use an outer multipart/alternative structure where the first part is the unsigned message and the second part is the signed message. This doubles the length of all messages, which may not be a significant problem for text messages but is a bad idea for signed multimedia messages which can be very large. There is also no requirement that the signed message and the unsigned message are related, leaving the possibility of some quite subtle security attacks.

S/MIME has significant industry backing (from Microsoft, Netscape and Lotus among others, RSA is one of the industry heavyweights anyway) and is easy to implement and integrate into a MIME user agent. The 100% overhead for signed messages is a problem, and there have been reports that RSA are looking at how RFC1847 security multiparts can be used to alleviate this (though this would make it as hard to integrate into user agents as MOSS and PGP/MIME)

5.1.6 MSP (Message Security Protocol)

MSP will be used to protect messages in the American DoD's massive Defence Message System (DMS) project. It is also being considered for use by the UK MoD. The documentation is hard to obtain (and could not be printed when we did) so very little is known about this protocol. MSP has the following characteristics:

- It provides interworking between X.400, MIME and vanilla SMTP environments by being completely independent of the content that is protected (this is done by providing enough redundant information that it will survive usual gateway transformations);

- As well as the usual authentication, integrity and confidentiality services, MSP offers signed receipts. With an appropriate infrastructure, signed receipts provide non-repudiation with proof of delivery, and none of the other protocols offer this useful service (i.e. proof that a supplier received a particular purchase order)
- It seems that signed messages are made transparent by including both an unsigned message and a signed message, thereby doubling the size of the message in a similar way to S/MIME. It is not known whether there is any linkage between the signed message and the unsigned message to ensure they are the same.
- Akin to PEM or PGP, the encryption mechanism is provided by encrypting the message once in a Message Encryption Key (MEK) and then providing a separate token in the security header for each recipient. The token consists of two parts: a cleartext tag that identifies who the token in for, and the encrypted MEK, encrypted in a key formed through a key generation process that uses the intended recipients public key and the sender's private key. The key generation process thus is somewhat different than PEM, as it provides data origin authentication for the message, even if the message had not, for example, been signed.

There is strong support by US Defence contractors, but little other uptake of this technology.

5.1.7 Secure X.400

Since 1988, X.400 email has had security, providing the following services
- Origin authentication for all types of messages
- Authentication of each pair of communicating parties throughout the MHS
- Proof of (and, with infrastructure, non-repudiation of) submission and delivery
- Confidentiality
- Message Integrity
- Sequencing
- Message security labelling

In 1994, the PASSWORD project performed some testing on secure X.400, and found that the deployed international X.400 infrastructure was unable to support it (due to being based on implementations of the 1984 standard) and catastrophically stripped the security information from the message. It is not clear the extent to which this will now have changed.

5.1.8 Conclusion

The main conclusion is that there are enough candidate protocols, and no new ones should be invented. However, it is not at all clear which protocols with thrive, which will survive and which will die (except that it is obvious PGP will survive in some form and that PEM will die in due course). MOSS, PGP/MIME and S/MIME are basically similar and their success will depend on sufficient implementations, good integration into user agents and the availability of a good certification infrastructure to make them applicable outside a small community. Given these three conditions, it is quite possible that all could survive and that multi-protocol user agents would be able to automatically handle all of them.

5.2 World Wide Web

Security of the World Wide Web is an area of even greater commercial and research interest than secure email, and the proliferation of competing standards is just as great and just as confusing. One significant difference is that there are two major players (Netscape and Microsoft) who are forcing the issue to be resolved (admittedly in their direction), and there is little possibility that it will drag on for 10 years as secure email has done.

The areas where security is required in the WWW are:
- The need to restrict access to specific WWW pages based on a strongly authenticated identity;
- The need to strongly authenticate users who desire to make commercial transactions or access data;

- The need to bind commercial terms from one source (e.g. a supplier) to a user, and have the resulting transaction confidential and non-repudiable with all parties authenticated;

Currently there are the two main approaches to WWW security:

- transport level security - this leaves the actual WWW mechanisms the same but routes them over a secured TCP channel. This method is actually applicable to all socket based Internet applications, and secured versions of telnet and ftp have been demonstrated using it. Example protocols of this type are SSL ("Secure Sockets Layer" from Netscape), PCT ("Private Communication Technology" from Microsoft) and GSS-API (from the IETF CAT and WTS WGs)
- message-based security - this leaves the network layer protocols the same but makes modifications to the application layer protocols. Example protocols of this type are SHTTP ("Secure HTTP" by EIT for the IETF web transaction security working group) and SEA ("Security Extension Architecture" from the World Wide Web Consortium).

Additionally, some researchers have approaches that just modify the client to know how to handle signed and encrypted documents when they have been downloaded (Mosaic are following this approach, currently using PGP as the document security mechanism but considering a secure MIME approach)

So far, only transport level security has been implemented and deployed successfully; this has been mainly through releases of the Netscape server and browser, and as add-ons to free servers (Apache and CERN httpd) and browsers (Mosaic) using toolkits such as the SSLeay Reference Implementation. Version 2 of Netscape's SSL protocol has been deployed widely based on draft specifications, but many people have found problems with the specification and these have been addressed both by Microsoft's PCT specification and Netscape's own SSL version 3 protocol (both of which are largely derived from SSL version 2). While both of the new proposed standards are incompatible with each other, there are significant the similarities that would be expected from their common ancestry; it is very probable (and highly desirable) that a unified specification will be produced as the best way forward for transport level security.

Message-based WWW security mechanisms have found specification and deployment difficult (and possibly have, in the past, been advocated by people with insufficient commercial weight, although IBM are supporting it and the next releases of the NCSA httpd server and Mosaic browser are expected to support it). Some people advocate a combination solution (with overall transport level security plus mechanisms for signed and encrypted documents) as the best chance for the near future.

One big problem is that transport level security cannot cope with WWW caching proxy servers that are so necessary in the Internet to relieve congestion on bottlenecks such as on the UK - US transatlantic links. A proxy server will perform http retrievals on behalf of the user, but a side-effect is that the ultimate owner of the pages sees the proxy as the person retrieving the page and can only apply access control based on this identity. This means that if your organisation has pages that are only accessible from inside the organisation (such as a departmental telephone directory) then users of an external WWW cache will not be able to retrieve these pages, and external people using a cache inside the organisation will be able to (it is undoubtedly possible to configure the cache not to act on behalf of external users but the default configuration is almost certainly insecure). Of course proxy servers also make more difficult both the introduction of advertising, and the gathering of targeted usage data - much to the concern of some commercial interests.

Another area that needs to be carefully considered is the area of CGI scripts and Java scripts. These are both fundamental to the flexibility of the Internet but are potential security holes, and it is not clear how current security research addresses these .

There are many significant research projects and commercial organisations that are looking at the overall field of authentication and privacy within WWW applications, and the European ICE-TEL project is doing work in exactly this area. It is fair to say that no project or organisation knows the answers to the long-term question of where this technology is heading, and most people are sitting on the fence hedging their bets between SSL versus PCT, SHTTP versus SEA and in the final contest between the winners of each contest. One problem is that organisations such as the IETF and the World Wide Web Consortium do not have sufficient resources or manpower to provide a lead in the

area, and it is left to commercial companies (mainly Netscape and Microsoft) to set the agenda. This tends to mean that a result will come more from political considerations rather than technical considerations.

WWW security research and deployment also needs to consider one feature that no other technology faces - the rapid development cycle of user agents and users' insatiable desire to use the latest and slickest commercial user agent. Researchers must face the fact of life that users want to use the latest Netscape release, and that they will shun a more secure version of an older and less impressive browser. This is unfortunate, as researchers developing security protocols will almost always have to work with old versions of the Mosaic browser for which source code is available.

Secure electronic mail research does face a similar issue, as users become very attached to their customary user agent (and the archive of many years email messages they have built up using it), however clunky it is; and are reluctant to change just to acquire a security functionality. However, at least here the market is relatively static and deployment does not have to cope with a quarterly release cycle and the feature-catch-up game; and although there is an initial effort in adding security to commonly used mail user agents, there is the expectation that once a critical mass of user agents has been built up then the technology will drive itself, and not the fear that all work will be obsolete with the next three months.

5.3 Remote Login

Problems caused by password compromise is the largest cause of unnecessary work by university computer services and by national CERTs. The problems come from two areas:

- Snooping of Ethernet packets by uncontrolled PCs - MSDOS based PCs (including Windows95) allow the user great control over the hardware, and so users are able to install and monitor Ethernet probes to see what other users are doing. This is less of a problem for Windows NT and UNIX systems as, without having administrator/root privileges such intimate contact with the hardware is not possible (of course if there is a security loophole allowing a user to gain root privilege then this safeguard does not apply).
- user carelessness - users watching other people typing in their passwords, users walking away and leaving themselves logged in, users telling friends what their password is, users choosing obvious passwords, users writing their password down on a post-it note and sticking it to the side of their machine.

With many systems, it is possible to close password loopholes by setting minimum password lengths or minimum password change interval, insisting that new passwords are not in a dictionary or do not repeat old passwords, or mandating that passwords must contain mixtures of upper and lower case letters, numbers and punctuation. However, most of these measures simply make it more likely that people will write down their passwords, thereby compromising any security the measures may have added.

These are concerns for people logging in to a local machine. Authentication and confidentiality when logging in remotely over the Internet are a greater concern to the travelling email user. More and more conferences these days provide an Internet connection so that delegates can log in to their home machines and read their daily dose of email or news. However, this invariably uses simple (password-based) authentication and the user will have no idea how the TCP/IP packets will be routed from the conference location to the home network. Therefore, there can be no assumption that the packets are safe from packet sniffers or that the remote workstation is free from password grabbing software. It is worth commenting that, at the quarterly meetings of the IETF, delegates are routinely asked who uses plain text passwords for logging into their home machines, and it is most unusual for an attendee from the UK to avoid raising their hand to this question.

There are a number of solutions to the password problem, as we saw in the above section on protected authentication, and one-time-password cards provide an excellent means of logging in without passing a clear text passwords and without requiring special hardware or making any assumptions about the environment from which the user will be logging on. Similar means can be used to negotiate a one-time session key to encrypt the traffic.

Another option is to use a protocol such as SSL or PCT to make a secure TCP channel, and then to use an enhanced version of telnet over this channel. The Ssh (secure shell) program provides an alternative implementation, which is free and is widely available for a large number of computer platforms. Ssh is a program to log into another computer over a network, to execute commands in a remote machine, and to move files from one machine to another. It provides strong authentication and secure communications over insecure channels. Its features include the following:

- Strong authentication: it closes several security holes (e.g. IP, routing, DNS spoofing and listening for passwords from the network); and adds new authentication methods: UNIX-style .rhosts together with RSA based host authentication, and pure RSA authentication.
- All communications are automatically and transparently encrypted. Encryption is also used to protect against spoofed packets and hijacked connections.
- X11 connection forwarding provides secure X11 sessions.
- Arbitrary TCP/IP ports can be redirected over the encrypted channel in both directions.
- The client RSA-authenticates the server machine in the beginning of every connection to prevent Trojan horses (by routing or DNS spoofing) and man-in-the-middle attacks. The server RSA-authenticates the client machine before accepting .rhosts or /etc/hosts.equiv authentication (to prevent DNS, routing or IP spoofing).
- An authentication agent, running in the user's local workstation or laptop, can be used to hold the user's RSA authentication keys.

Ssh is intended as a complete replacement for rlogin, rsh, rcp, and rdist. It can also replace telnet in many cases.

Another problem that mobile users face is that, if firewalls or packet filtering mechanisms are used in the local network, the travelling user will be unable to use certain local facilities as these will be filtered by the firewall. A POP3 mail server, for example, may only accept connections from machines in the local network. Therefore, the only option for the travelling user is to directly dial to a modem inside the local organisation, and access email via a very expensive (and unreliable) international telephone call.

The various solutions to this problem involve tunnelling through the network to make the local application think you are a local user. The remote user is strongly authenticated with a server inside the local network, and this server relays communication securely in both directions between remote user and local application. Therefore, a user who is outside a firewall can securely have the privileges of a user who is inside the firewall. The travelling user just needs to find a network connection and a local IP address at the remote institution, and then to use secure access software on his/her notebook PC. This approach is being followed in a project proposed to EPSRC by Salford University.

5.4 Document Security

There are two aspects to secure document retrieval - securing the process of retrieving the document, and securing the document itself.

Document retrieval typically uses mechanisms such as ftp, http or WAIS to download the document, and the user then invokes the correct document viewer to examine the document. These protocols can both be modified (either explicitly or by routing them over a TCP channel secured using a mechanism such as SSL) such that the user and the server mutually authenticate each other and negotiate a session key to make the data request and the data download confidential and with integrity. The “owner” of the downloaded document then loses control over the data and the downloader can manipulate it as required.

Another option is to apply the security to the document itself, and have either the entire document or just parts of it either signed or visible only to predefined people. The ODA (Open Document Architecture) standard has an addendum on document security that describes the application of security to any part of a document profile or of a document body, and UCL have implemented this and seamlessly integrated it with the BBN Slate document editor to allow secure documents to be created and viewed (a user simply does not see parts of the document that are not authorised).

An alternative approach is to note that exactly the same result can be achieved by using MIME to structure the document and RFC1847 security multiparts to add the security. A suitable MIME viewer could then display the document with security unfolded, and this approach is very compatible with some of the mechanisms for message or document based WWW security described above.

Of course, even with these mechanisms the "owner" of the downloaded document loses control over the data as an authorised downloader can manipulate the document after the security has been removed by the secure document viewer. There does not seem to be a technical solution to this problem.

5.5 Multimedia conferencing

With the increased power of user's workstations, and the increased bandwidth available from modern networks such as ATM, the use of open networks for multimedia conferencing is becoming more and more popular. Using technology developed by projects such as the EC MICE and MERCI projects, users with PCs or UNIX workstations are able to participate in live conferences with multicast audio and video channels and a shared whiteboard for drawing. For audio participation, software and a microphone is all that is needed. For video, a camera and a special video card is needed, but the price of these is not prohibitive.

The normal architecture for multicast multimedia conferencing is to openly advertise the conferences that are available and once they are running, the content of the conference and the people participating in it is easily visible to anyone who wants to look. This is clearly useful only for very general conferences. The technology offers great possibility for holding international project meetings with everyone in the comfort of their own office, and such an open architecture is clearly undesirable for this. Security requirements include:

- Restricting the distribution of the conference announcement;
- Authenticating conference announcements, and even more significantly changes to such announcements;
- Authenticating people applying to join a conference;
- Ensuring that only invited people can participate, actively or passively (for instance by restricting the distribution of encryption keys);
- Keeping the list of conference participants confidential.

Research here is focusing on both the addition of security technology to the software tools used for multimedia conferencing; and on the key distribution problem, with small-scale tightly bound groups of users using secure E-mail for key exchange, and group keys stored in directories providing larger granularity. It now seems that a complete system for Session Announcement and Session Invitation will come out of the IETF deliberations; the system can be made to work without a large infrastructure, but would operate better if a solid authentication infrastructure existed. The EC Framework-IV MERCI project is looking at the issues raised, and will certainly be deploying Secured multimedia conferencing

5.6 Directories

There is fairly wide deployment of X.500 directory technology, especially among the European academic community under the umbrella of the NAMEFLOW-PARADISE project. The Paradise directory currently contains around two million people (most of these are academics) and allows information to be almost instantly retrieved for anyone with an entry. Searches can be made on a departmental basis, an organisational basis or even by power-searching through an entire country (this is an expensive operation as the contents of the directory are distributed among many servers, and they all have to be visited for a spanning search to work).

Deployment of the X.500 DIT (Directory Information Tree) is divided between DSAs (Directory Service Agents) which each hold a portion of the DIT and communicate with each other to obtain information they do not have; and DUAs (Directory User Agents) which are programs operated by end users to interrogate the DSAs and retrieve information. The X.500(1988) standard defined security extensions for all communication, allowing signed operations and signed results (no confidentiality was included).

The vast majority of DSAs currently deployed in the UK are based on the public domain implementation of QUIPU, an implementation of the X.500(1988) standard that does not include any support for strong authentication. In 1993, the European PASSWORD project enhanced QUIPU using the OSISEC and SECUDE security toolkits to provide secure operations between the DUA and the DSA, though not between multiple DSAs exchanging information (this would not have been generally possible as the installed QUIPU infrastructure had no security support). These implementations were piloted, but the installed base remained predominantly insecure and so there was little benefit to be gained by upgrading.

Recently, the first implementations of the X.500(1993) standard are appearing, and these are far more likely to include facilities for strong authentication. Therefore, assuming that existing 1988 standard installations are updated to the 1993 standard (which is by no means guaranteed, as 1988 standard implementations tend to be free while 1993 standard implementations tend to be commercial products), a directory infrastructure incorporating security will begin to evolve.

The situation is similar with DUAs - most currently deployed user agents do not contain facilities for signing operations nor for verifying signed results, though the PASSWORD project added these security facilities to the “dish” and “de” user agents. A secure DUA is no use without a secure DSA, and so the securing of the two will go hand in hand.

An alternative architecture for a distributed directory is WHOIS++ defined in RFC1834 and RFC1835. The original WHOIS service (defined in RFC954 in 1985) provided a very limited directory service, serving information about a small number in Internet users registered with the DDN NIC. The basic service was eventually enhanced, and similar services were set up elsewhere, but there was little or no co-ordination between these activities. The service was also centralised, and therefore could not serve as a white pages directory service for the entire Internet, as was required. WHOIS++ was therefore designed as a simple, distributed and extensible information lookup service. As well as extending the original data model and query protocol, WHOIS++ adds a distributed indexing service and more powerful search constraints and search methods. It also structures the data around a series of standardised information templates to make the client and server dialogue more parseable. An optional authentication mechanism for access control is also introduced, based solely on passwords in the current specifications. The authentication mechanism does not actually dictate schemes to be used, but just provides a framework for indicating that a transaction is to be authenticated.

The current state of deployment of WHOIS++ is not clear, and the extent to which adequate authentication mechanisms are implemented even less so.

A recent development in directories is LDAP (Lightweight Directory Access Protocol) which is an ASCII variant of the protocol used for communication between the DUA and the DSA. With LDAP, a program communicates with an LDAP server using an exchange of clear text messages in the same way as a mail program communicates with an SMTP server. The LDAP server then makes regular directory queries on their behalf and returns the results in a standard form. This approach means that the directory using application does not need to understand the intricacies of the X.500 communication protocols. Another advantage is that the LDAP server can use the same query over multiple directory technologies and this is transparent to the user. So a user can request information on “J.Smith” from an LDAP server, and the information that is returned could have come from an X.500 DSA, a Whois++ server or any other directory mechanism. End user applications only need to know one protocol, and the LDAP server can be extended to cope with extensions to the directory standards as they become available.

5.7 General Network Facilities

This section introduces a brief discussion of the security risks and needs of common network applications, many of which present subtle security threats.

NFS (network file system) and NIS (network information system) are examples of RPC (remote procedure call) based protocols. Each are vulnerable protocols from a security perspective: NFS

because an attacker with access to an NFS server can read any file on the server; and NIS because it is used to distribute important information such as password files, host tables and the public and private key databases used by secure RPC. Because these protocols are designed to be primarily efficient (NFS can make a remotely mounted server seem almost as fast as a local disk drive), there is little authentication and so control messages are easily forged. Also, the default way these protocols are set up on unix systems is often insecure, and an improperly configured NFS server can allow any othe rhost to simply mount it's file system.

NTP (network time protocol) is used to synchronise the clock on a computer to within 10ms of a reference clock. It is vulnerable to attacks aimed at altering a host's time and, because of the importance of synchronised log files in many applications, this is a serious consideration.

DNS is a distributed database used to match host names with IP addresses. There are several security weaknesses in the protocol that can allow mis-routing of TCP or UDP messages or which can allow domain name spoofing. This implies that authenticating users based on their domain name is a significantly dangerous thing to do - the use of IP addresses is much better, although it is also possible to spoof these as well and so it should not be regarded as an adequate security strategy. DNS also holds a wealth of information that is valuable to a potential attacker (the finger protocol is also guilty of this, and other distributed directory services can also be accused of it) although this is only really a significant danger for people relying on obscurity or weak security.

X Windows is a network oriented windowing system that uses the network to communicate between the running application and the window displaying software. X-client applications that have connected to an X-server can do many undesirable things, such as reading screen contents and both detecting and generating key presses. The authentication mechanisms built into X11 are weak and inadequate.

5.8 Electronic Commerce

5.8.1 Introduction

There is an increasing overlap between academic requirements and commercial requirements, especially in the area of information provision. Moreover, consideration is being given to the use of electronic mechanisms for purchasing in Higher Education institutes. Therefore, it is appropriate to look here at the various schemes proposed for electronic commerce. The security requirements for this are:

- authentication of customer, card acceptor and transaction acquirer
- transaction integrity, confidentiality and non-repudiation.

This section includes outline descriptions of a number of proposed payment models. It is worth emphasising that commercial applications in the banking and finance world will almost certainly define their own trust models and infrastructures, and will have no interest in inter-working with other applications or other general security infrastructures (i.e. a Mastercard SET certificate will not be useful for anything other than a Mastercard transaction).

An overview of electronic payment schemes can be found at
http://www.w3.org/pub/WWW/Payments/roadmap.html
Note that some of these payment schemes are proprietary and commercial, and so few details of how they work internally are available

5.8.2 Credit card models

- Secure Electronic Transactions - SET (Visa/Mastercard)
  http://www.mastercard.com/set/set.htm

The SET protocol is being developed by Visa and Mastercard with the assistance of GTE, IBM, Microsoft, Netscape, SAIC, Terisa, and Verisign. It uses an X.509 CA hierarchy with a single, globally trusted, root. This exists to authenticate certificates held by the three types of leaf entity: cardholders, merchants, and Acquirer Payment Gateways (these are the interface between merchants and the acquiring bank). The protocol uses existing financial networks to communicate authorisation requests between the acquiring bank and the issuing bank. SET has the characteristic that a cardholder can charge a purchase to a merchant without allowing the merchant to learn the credit card number used.

The SET procedure can be summarised as follows:

1. **shopping**: the shopper (cardholder) contacts a merchant to request a list of the goods and services offered. The customer shops from this list.
2. **ordering**: the cardholder selects items, prepares an order and sends it to the merchant, who processes it.
3. **inventory**: the merchant checks its inventory to determine if the goods and services ordered by the customer needs to be back-ordered. The merchant may decide to handle the order as a split shipment.
4. **authorisation request**: the merchant sends an authorization request to its financial institution (Acquirer). The Acquirer reformats the data into a request that is enriched and sent via a payment card network to be processed by the financial institution (Issuer) that issued the payment card to the cardholder.
5. **authorisation request**: the Issuer responds via the payment card network with an authorization response, which includes an indication of whether the authorization request has been approved. The network sends it to the Acquirer who responds to the merchant with the outcome of processing.
6. **shipping**: the merchant delivers the goods and services to the customer. The time delay between authorization and shipment (which shall precede capture) can legitimately be several days. Many MOTO merchants are not able to check inventory before authorization. If goods are not available in the warehouse, the shipment is held up waiting for the warehouse to be replenished.
7. **capture processing**: the merchant submits a capture request to the Acquirer in order to obtain payment. This request is sent through the payment card network to the Issuer.
8. **credit processing**: if a credit is to be issued to a customer, such as when the goods are returned or defective, the merchant sends a message to the Acquirer requesting a credit be issued to the cardholder's account.

SET includes a secure protocol for cardholders to get their certificates through network web sites. The bootstrapping process is as follows: the cardholder receives a plastic card in the usual way, along with software allowing a WWW browser to handle the SET MIME type (it also works via MIME email). The software has a certificate for the root CA compiled into it.. The software sends a request for a cardholder certificate, and the WWW server replies with a form to fill in, along with the complete certification chain from the root CA to the server's certificate. The software validates the WWW server certificate and the user fills in the form, including a shared secret such as "Mother's maiden name" The MIME handler then generate a public/private key pair and a random DES key, encrypts the DES key with the public key of the server, encrypts the form data and the cardholder's public key with the DES key and sends the whole package to the server. The server now has enough information to validate the user and issue a certificate.

The protocol includes a detailed consideration of revocation, with examples of time based, CRL based, physical, and capability revocation in one protocol. Cardholder certificates never appear in a CRL - a cardholder certificate is revoked by revoking the card. Merchant certificates may be revoked either by distributing a CRL to the Acquirer Payment Gateways, or by marking them revoked in the acquiring bank's database.

- Micro Payment Transfer Protocol - MPTP (World Wide Web Consortium)
  http://www.w3.org/pub/WWW/TR/WD-mptp-951122

"MPTP is a protocol for transfer of payments through the services of a common broker. The processing demands of the protocol make it practical for small payment amounts while the latency makes it practical for use in interactive applications. Therefore, the scheme satisfies the two key criteria for a micropayments scheme. For efficiency it is desirable to be able to combine transfer of payments instructions with those accomplishing the delivery of goods. For this reason MPTP may be layered on a variety of Internet protocols including HTTP and SMTP/MIME.

With payment systems that support charging relatively small amounts for a unit of information, the speed and cost of processing payments are critical factors in assessing a scheme's usability. Fast user response is essential if the user is to be encouraged to make a large number of purchases. Processing and storage requirements placed on brokers and vendors must be economic for low value transactions. MPTP is optimized for use for low value transfers between parties who have a relationship over a period of time. It also provides a high degree of protection against fraud making it applicable in wider scenarios, including sale of tangible goods.

In the Pay Word scheme a payment order consists of two parts, a digitally signed payment authority and a separate payment token which determines the amount. A chained hash function,is used to authenticate the token (in the same way as in the S/Key mechanism described earlier in the document). To create the payment authority the customer first chooses a value w_n at random. The customer then calculates a chain of payment tokens (or paychain) w_0, w_1, ... w_n by computing

w_i = hash (w_i+1)

where h is a cryptographically secure one way hash function such as MD5 or SHA.

The signed payment authority contains w_0, the root of the payment chain and defines a value for each link in the chain. Payments are made by revealing successive paychain tokens. Once the vendor or broker has authenticated a payment authority an arbitrary payment token may be authenticated by performing successive hash functions and comparing against the root value. It should be noted however that the broker is only presented with the final payment order. It is therefore unnecessary for the broker to maintain large online databases.

MPTP permits use of double payment chains. This allows implementation of a broker mediated satisfaction guarantee scheme. The pair of payment chains represent the high and low watermarks for the payment order. The low watermark chain represents the amount that the customer has fully committed to pay. The high watermark chain represents partial commitments. The vendor exposure is the difference between the counter values. "

- Anonymous Internet Mercantile Protocol - AIMP (AT&T)
  ftp://ftp.research.att.com/dist/anoncc/accinet.ps.Z

  "A card model protocol which implements a policy which balances strong guarantees of confidentiality with the needs of law enforcement. A formal approach is employed with comprehensive details of mechanism and data flow."

  No further details available.

- Simple Green Commerce Protocol - SGCP (First Virtual)
  http://www.fv.com/info

  First Virtual's Green Commerce payments model is one of the first payments schemes to become established on the Internet. The major novel feature of this scheme is its 'satisfaction guaranteed' policy which protects customers from dishonest merchants by allowing them an unconditional right to refuse payment for individual items. A statistical mechanism is used to identify over frequent use of this option and exclude habitual non-payers. Identification of customers is via an email call back loop scheme.

  The FV approach is unique in that it uses no cryptography (and therefore achieves a lower level of commercial security). If a buyer and a seller both have an FV account then the following six stage protocol is used:

1. the buyer emails the sender requesting the services
2. the seller emails FV asking for authorisation
3. FV emails the buyer asking for confirmation
4. the buyer replies with a one word answer
   - “yes” - confirm the transaction
   - “no” - decline the transaction (e.g. buyer changed mind) no reason is required
   - “fraud” - declare the transaction suspicious, and irrevocably terminate the FV account.
5. if the answer is yes then FV make appropriate credit and debit actions using standard banking and credit card facilities.
6. if the bank approves the transaction then FV notify the seller

The goods can be shipped either at stage 2 or stage 6, depending on seller policy.

If the answer at stage 4 is “fraud” then the buyers account will be terminated and, in order to continue using FV facilities, a new account must be opened in the normal way (which involves some telephone authentication and payment of a modest fee).

- CyberCash
  http://www.cybercash.com/cybercash/news

  Cybercash provides real-time credit card transactions on the Internet, using 1024-bit RSA technology worldwide. A micro-payment service (electronic coin) is also available. CyberCash transactions move between three separate software programs: one program that resides on the consumer's PC (called a wallet), one that operates as part of the merchant server, and the third part that operates within the CyberCash servers. The merchant and consumer software is free.

  A six stage payment scheme is described:

  1. Consumer has shopped at the merchant's site and decided what it is they wish to purchase, where they want it shipped, etc.... Merchant server returns a summary of the item, price, transaction ID, etc. to consumer.
  2. Consumer clicks on the "Pay" Button which launches the CyberCash, Checkfree or Compuserve Wallet and chooses which credit card from their "wallet" they wish to pay with and clicks OK to forward the order and encrypted payment information to the merchant.
  3. Merchant receives the packet, strips off the order and forwards the encrypted payment information digitally signed and encrypted with his private key to the CyberCash server. The merchant cannot see the consumer's credit card information.
  4. CyberCash server receives the packet, takes the transaction behind its firewall and off the Internet, unwraps data within a hardware based crypto box (the same ones the banks use to handle PINs as they are shipped from ATM network to ATM network), reformats the transaction and forwards it to the merchant's bank over dedicated lines.
  5. The merchant's bank then forwards the authorization request to the issuing bank via the card associations or directly to American Express or Discover in those cases. The approval or denial code then is sent back to CyberCash.
  6. CyberCash then returns the approval or denial code to the merchant who then passes it on to the consumer. From Step 1 to Step 6 takes approximately 15-20 seconds. All encryption is at the message level and is therefore independent of the WWW browser technology used.

5.8.3 Cash models

- E-Cash (DigiCash)
  http://www.digicash.com/ecash/ecash-home.html

  “With the ecash client software a customer withdraws ecash (a form of digital money) from a bank and stores it on his local computer. The user can spend the digital money at any shop accepting ecash, without the trouble of having to open an account there first, or having to transmit credit

card numbers. Because the received ecash is the value involved with the transaction, shops can instantly provide the goods or services requested. Person to person payments can also be performed with ecash.

One of the unique features of ecash is payer anonymity. When paying with ecash the identity of the payer is not revealed automatically. Ecash offers one-sided anonymity; when clearing a transaction the payee is identified by the bank. Additional security features of ecash include the protection of ecash withdrawals from your account with a password that is only known to you; not even to your bank."

- NetCash (USC)
  http://nii-server.isi.edu/info/netcash/

  "NetCash is a framework for electronic currency being developed at the Information Sciences Institute of the University of Southern California. NetCash will enable new types of services on the Internet by providing a real-time electronic payment system that satisfies the diverse requirements of service providers and their users. Among the properties of the NetCash framework are: security, anonymity, scalability, acceptability, and interoperability.

  NetCash was designed to facilitate anonymous electronic payments over an unsecure network without requiring the use of tamper-proof hardware. NetCash provides secure transactions in an environment where attempts at illegal creation, copying, and reuse of electronic currency are likely. In order to protect the privacy of parties to a transaction, NetCash implements financial instruments that prevent traceability and preserve the anonymity of users."

  There is also NetCheque, from the same source.

5.8.4 Cheque models

- Electronic Cheque (Financial Services Technology Consortium)
  http://www.fstc.org/projects/echeck/index.shtml

  "The FSTC Electronic Check is an innovative, all-electronic, payments and deposit gathering instrument that can be initiated from a variety of devices, such as a personal computer, screen phone, ATM, or accounting system. Electronic Check provides rapid and secure settlement of financial accounts between trading partners over open public or proprietary networks, without requiring pre-arrangement, by interconnection with the existing bank clearing and settlement systems infrastructure.

  The Electronic Check is modelled on the paper check, except that it is initiated electronically, uses digital signatures for signing and endorsing, and digital certificates to authenticate the payer, the payer's bank and bank account. However, unlike the paper check, through the use of an issuer defined parameter, the Electronic Check can resemble other financial payments instruments, such as electronic charge card slips, travellers checks, or certified checks. Although Electronic Check's primary use is to make electronic payments on public networks, the project design will enable Electronic Check to be used in any situation where paper check is used today. For example, banks will use Electronic Checks to gather electronic deposits from public network users, thus opening the opportunity for complete full service electronic remote banking, anywhere the customer is connected. Later, point-of-sale implementations are possible, if the marketplace demands.

  The Electronic Check is delivered by either direct transmission or by public electronic mail systems. Payments (deposits) consisting of Electronic Checks are gathered by banks via e-mail and cleared through existing banking channels, such as Electronic Check Presentment (ECP) or ACH networks. This integration of the existing banking infrastructure with the new, rapidly growing public networks in a secure fashion provides a powerful implementation and acceptance path for banking, industry, and consumers."

- Millicent (DEC System Research Centre)
  http://www.research.digital.com/SRC/millicent/

  "Millicent is a lightweight and secure protocol for electronic commerce over the Internet. As a microcommerce system, Millicent is designed to support purchases costing fractions of a cent (the name Millicent is derived from the underlying cost associated with each micropayment transaction). Millicent is based on decentralized validation of electronic cash at the content providers Web server without any additional communication, expensive encryption, or off-line processing.

  The key innovations of Millicent are its use of brokers and of scrip. Brokers take care of account management, billing, connection maintenance, and establishing accounts with vendors. Scrip is microcurrency that is only valid within the Millicent-enabled world."

- NetBill (Carnegie Mellon University)
  http://www.ini.cmu.edu/NETBILL

  NetBill implements of a cheque payment model employing a symmetric key cryptography mechanism based on Kerberos. It acts as a third party to provide authentication, account management, transaction processing, billing, and reporting services for network-based clients and users. With a NetBill account and client software, users can buy information, software, CPU cycles, or other services from NetBill-authorized service providers, under a variety of payment schemes.

  The transaction protocol is especially designed to handle low cost items; for example, journal articles at 10 cents a page. It ensures that both the consumer and merchant are protected: the consumer is guaranteed of the certified delivery of goods before payment is processed, and the merchant is guaranteed that the consumer cannot access the goods until payment has been received.

  The protocol is abstracted around a client library, called the checkbook, and a server library, called the till. An eight stage protocol is defined:

  1. The customer's client application indicates to the checkbook that it would like a price quote from a particular merchant for a specified product. The checkbook sends an authenticated request for a quote to the till which forwards it to the merchant's application.
  2. The merchant then determines a price for the authenticated user and returns the digitally signed price quote through the till, to the checkbook, and on to the customer's application.
  3. If the customer's application accepts the price quote, the checkbook sends a digitally signed purchase request to the merchant's till.
  4. The till then requests the information goods from the merchant's application and sends them to the customer's checkbook encrypted in a one-time key, and computes a cryptographic checksum on the encrypted message.
  5. The checkbook computes its own cryptographic checksum on the encrypted goods and returns to the till a digitally signed message specifying the product identifier, the accepted price, the cryptographic checksum, and a timeout stamp: this is called the electronic payment order (EPO). Note that, at this point, the customer can not decrypt the goods; neither has the customer been charged.
  6. Upon receipt of the EPO, the till checks its checksum against the one computed by the checkbook. If they match then the encrypted goods were received without error and the merchant's application creates a digitally signed invoice consisting of price quote, checksum, and the decryption key for the goods. The application sends both the EPO and the invoice to the NetBill server.
  7. The NetBill server verifies that the product identifiers, prices and checksums are all in agreement. If the customer has the necessary funds or credit, the NetBill server debits the customer's account and credits the merchant's account, logs the transaction, and saves a copy of

the decryption key. The NetBill server then returns to the merchant a digitally signed message containing an approval or an error code.

8. The merchant's application forwards the NetBill server's reply and the decryption key to the checkbook (it is a weakness that the merchant does this and not the neutral NetBill server).

5.9 Products Supporting Security Services

This section looks briefly at the security products (CAs, secure applications, toolkits) of some of the main players in the security market.

5.9.1 Europe

- **ISODE Consortium (http://www.isode.com)**

  As part of the ICR3 commercial ISODE toolkit, IC has incorporated a security toolkit implementing X.509v1, with support for X.509v3 coming very soon (extensions are currently recognised but not acted on). ICR3 includes an X.500(1993) DSA and DUA and these include some signed operations and results - full security will be included in later subreleases. ICR3 is for unix platforms only, though an Windows NT version is promised in the medium term future.

  The ISODE Consortium business model is that they provide core technology to their members in source code form, and these members than make products out of these and sell them in binary form in return for a royalty. The X.509 toolkit is too new within the IC range to have been incorporated into member's products yet.

- **SSE (http://www.broadcom.ie/telecom/dupjmc/sse/welcome.html)**

  SSE markets a range of secure messaging products under the OpenPath name. One of these products, OpenPath CA, is a SESAME compatible X.509v1 Certification Authority wich runs on PC and UNIX platforms.

- **COST (http://www.cost.se)**

  COST provides security services and products in the following areas (all run on PCs):

  - secure PC/smart-card - provides full protection at a single PC against intruders and illegal users. It may be used for protection of any kind of PC resource: text files, source and executable programs, documents, etc. It may be activated at the start-up of Windows or invoked separately for each application whose resources must be protected.
  - secure DCE - is a comprehensive security system for client/server distributed environments. It is based on the Internet standard GSS (Generic Security System) which provides user authentication based either on usage of Kerberos tickets or strong authentication with a challenge/response protocol. The product also provides full control of access to application servers, and also encryption and integrity of communication messages.
  - certification systems - a strict RFC1422 compliant CA hierarchy with X.509v1 certificates, with the IPRA at the root, the COST PCA underneath, one subsidiary CA per country underneath and then organisational CAs underneath.
  - secure email - PEM
  - secure EDIFACT - provides full protection of EDIFACT documents against illegal reading or accidental modifications. Documents are protected during transfer through communication networks or while stored in local archives. Sender's and receiver's authenticity and sender's non-repudiation are also provided by usage of public key cryptography.
  - security platform - provides various security functions for user developed applications which need security enhancements. The platform consists of different security modules, functions and protocols, all accessible via security Application Programming Interfaces (APIs).

- **ICE-TEL (http://www.darmstadt.gmd.de/ice-tel)**

  ICE-TEL partners are contributing five different security toolkits (OSISEC from UCL, SecuDE from GMD, ICR3 from IC, SESAME from SSE and COST from COST) and are enhancing these to include x.509v3 certificates, support for a new trust model, support for a wide range of applications (including CA tools, MOSS email, WWW and directories) and platforms, and integration into popular user agents. These will then be made available for piloting, and will be used for the operation of a certification infrastructure within Europe. By the end of the project, the full range of required applications should be available for PC, Macintosh and UNIX platforms.

5.9.2 North America

- **TIS (http://www.tis.com)**

  Trusted Information Systems, Inc. (TIS) has been dedicated to providing computer and communications security solutions for information systems for over a decade. The most relevant of their products for this study are the TIS/PEM and TIS/MOSS secure email implementations (it is believed there is an exportable version of TIS/MOSS although I am not sure how this has been done). They run on UNIX and PC, and have also been ported to Macintosh.

  TIS has undertaken a comprehensive survey of over 1000 security products available world-wide from 30 countries, and the findings are given on the WWW page. One very interesting finding is that no significant difference in product quality was found between products from within the USA and from outside. This proves that the USA export ban on cryptographic software does not significantly impact people outside the USA and is therefore not a significant reason for the slow world-wide take-up of security technology.

- **NorTel (http://www.nortel.com/entrust)**

  Nortel produce ENTRUST, a family of public-key cryptography products for digital signature and encryption on computer networks with automatic key management on PC, Macintosh and Unix supported platforms. Entrust provides a network infrastructure which scales to an organisation size of tens of thousands of users. Entrust includes a toolkit that allows developers to incorporate security into other applications. Hardware encryption solutions will soon be available for increased security. However, the current infrastructure of certificates supported is somewhat limited, and the use of encryption is deliberately restricted in ways probably not appropriate to the academic community.

- **VeriSign (http://www.verisign.com)**

  VeriSign is a commercial CA that issues DigitalIDs (certificates) to individuals for use with web client software, secure email packages and other end-user applications' and to entities such as web servers, EDI hosts and firewalls that need to be authenticated by users or by other entities. Several classes of DigitalID are available, differentiated by the level of assurance or trust associated with the DigitalID (or, more precisely, by the amount of liability VeriSign will accept if they are wrong) which is determined by the method VeriSign uses to verify an applicant's identity.

  In addition to this, VeriSign also offers private-label certificate services, which are suitable for conveying information about authorisation, permission and access rights as well as basic identification information. These are restricted use certificates, typically only recognised by a specific application or service.

  VeriSign also market a CIS (certificate issuing system) to allow companies to set up their own CAs. The CIS uses the SafeKeyper hardware signing unit from BBN, which is a tamperproof electromagnetically shielded hardware device that signs certificates and meters certificate serial numbers. CIS also includes sophisticated certificate management software and an integrated relational database and reporting software.

- **Xcert (http://x509.com)**
  The Xcert Software Sentry supports the authentication of electronically mediated transactions, enabling organisations to set up their own X.509 CAs. It issues certificates in a form compatible with Netscape and Microsoft products, and any other application accepting DER format certificates.

6. Legal Issues and Security Infrastructures

This section covers both security infrastructures and legal issues. This is not because they are really related to each other, but consideration of both is needed in this report.

6.1 Security Infrastructures

A community the size of the UK academic community cannot be adequately served by centralised mechanisms that cater for all members of the community. It relies on devolved management by each academic institution, and the institutions almost always rely on devolved management by colleges or departments. Therefore, a community of around a million where one third change every year is managed in a highly distributed fashion.

The same is required for an infrastructure supporting authentication for the academic community. If a central facility was required to serve the whole academic community then management would be very difficult, and the system would probably break down every October when the new students started. Local management is therefore essential.

6.1.1 Directories

There are several standard models for white pages directories which all meet the requirement of local management of data. X.500, Whois++ and DNS all have distributed data models and, for each, a user connects to their nearest access point and sees exactly the same view of the entire directory. A directory-based approach therefore scales to a community of arbitrary size. Because the data model is also distributed, the system is also robust to network failures and only the data held by the systems affected by the network failure are affected (the directory standards define replication methods and so may even be resistant to partial failures.).

Directories have many more advantages as an infrastructure for the dissemination of authentication information. Because they are widely deployed, they allow a security infrastructure to grow beyond the confines of the UK academic community and allow the possibility of interworking with industry and other countries. Because access methods such as LDAP allow a user to transparently interrogate directories implementing diverse technologies, there is no requirement that one technology dominates. There are now WWW to X.500 gateways; these ensure that access to directory systems from arbitrary applications is now much easier both to implement and to use..

Finally, if certificates are to be stored in directories, then these are signed by the CA before being stored in the directory and so there is no requirement that the directory technology itself be secure, as tampering will be detected when certificate chains are verified.

Directories are, therefore, an ideal infrastructure for storing whatever information is required to support authentication technology.

6.1.2 Smart cards

There are currently a wide variety of smart-cards available on the international market, with a wide variety of capabilities. Some are designed merely as a place to hold a user's private key (requiring entry of a PIN number to retrieve it). Others have enough on-card memory to store a cache of certificates and even CRLs. Others actually have an encryption capability and generate digital signatures on the card (so the private key never leaves the card, providing the maximum protection against security breaches possible if the key is read into memory and another process scans the memory looking for it). Smart-cards therefore provide increased convenience at the very least, and increased security at the best.

The wide variety of cards comes with a wide variety of cost, and each has advantages and disadvantages. There would be significant advantages in choosing a single smart-card as the "JISC recommended smart-card" as any software development activity funded by JISC or undertaken by related projects such as ICE-TEL could then incorporate the card making the software immediately useful to the UK academic community.

In order to achieve this, a survey of the current smart-card market is needed, and an assessment of the capabilities and cost of current and near-future cards made. This is outside the scope of the current project, and is recommended for future work.

6.1.3 Key Escrow

Key escrow is the process of keeping a backup copy of a user's private key. It is normally divided into several parts which are kept independently, and these parts can then be bought together under certain well-defined circumstances. One of these circumstances is usually to do with national-security or law enforcement agencies applying for a court order to have the key bought together in the case of a criminal or national-security investigation. The mandated use of key escrow in schemes like the US Government "Clipper" project has raised fears of snooping governments 'steaming open' your encrypted email, and has given the technique a bad name in some circles.

There are also, undoubtedly, some good aspects to key escrow as well. Users are notorious for forgetting passwords, and in a system where access to a private key is protected by a password, forgetting the password means that you no longer can use the private key and therefore no longer can decrypt any messages encrypted with the corresponding public key (this could be a problem if you have an archive of encrypted email messages). So organisational key escrow schemes are a very valuable safeguard against this. Normally, participation would be voluntary and it would be made clear that a user who did not join in was taking full responsibility for their own actions.

A situation where you may not want key escrow to be voluntary would be in the case of a commercial organisation. No commercial organisation would want an employee to encrypt company data with a key known only to them as, if they suffered as fatal accident, the data would effectively be unrecoverable. Commercial companies view the data manipulated by their employees as belonging to the company, and so many companies would insist on holding a copy of their employees' private keys in escrow to cover eventualities such as these.

This has lead to the idea that separate keys should be used for signing data and encrypting data. When data is encrypted with a public key, the owner of the data should know the private key needed to decrypt it; and the government may also consider it has a need to this information under some circumstances. However, when signing information with a private key, there is never a need for anyone to know this key - the verification key is public knowledge anyway, and there is no adverse effect if the private key is lost. So the only effect of keeping a copy of a signature key in escrow is to allow someone reconstructing that key to impersonate the actual owner of the key. No company has the right to sign things as though they are the employee themselves, and the government has no legitimate need for this data either. So the modern practice is to generate one key pair for encryption (this key pair may be generated by the user or by a trusted third party, and may be kept in escrow), and another key pair for signature (this key pair is always generated by the user themselves and is never passed to anyone else).

6.2 Legal issues

There are three legal aspects relevant to this study:

- Export controls by another government limiting access to products;
- Domestic legislation restricting use of cryptography;
- The legal significance of a digital signature.

The authors do not feel that this short and necessarily limited study report is an appropriate place to raise these complex issues in detail, and limit themselves to brief discussion of each. Matters of law are beyond the professional competence of the authors, and a more appropriate forum must be sought.

Export controls by another government are an issue. Of particular importance are the United States government's ITAR regulations, which classify software and hardware implementing bulk data encryption as a munition and therefore subject to export controls. Currently, systems performing encryption using keys larger than 40 bits are prohibited from export beyond the USA and Canada. Therefore, the only security products successfully deployed internationally have either used weak security (Netscape) or have been the subject of government investigation to discover how they left the country (PGP). It should be noted that there is no corresponding ban on the import of encryption technology into the USA (other than issues regarding patenting of cryptographic algorithms), and so a security product developed outside the USA could currently be deployed world-wide using secure 128 bit encryption. Nevertheless, US-owned companies may be unwilling to go against the spirit of the US regulations in their provision of such products; the recent US laws covering the activities of foreign-owned countries outside the US may also discourage Non-US firms from similar activities. There are signs that these regulations may be relaxed in the near future, but they are still in operation at the time of writing this report.

The regulations apply only to systems performing bulk encryption of data; and systems that provide only authentication are not covered. However, there has recently been some debate on the SPKI mailing list as to exactly what is permitted. Some parties claim that a Commerce Jurisdiction Ruling is needed before things can be exported as something is only exportable if it **cannot** be used for bulk encryption (and this effectively means that only binary executables can be exported, and not source code). Other people claim that you do not need permission or approval of any kind for pure authentication technology of any strength.

It should be noted that there is considerable industry pressure to relax or remove the export ban, and there is also evidence that it has no significant effect on the world-wide availability of security software.

In the UK, there seems to be no current legislation restricting the use of cryptographic products for encryption (as there is in France and Belgium). The UK Goverment has, in the course of this study, issued a consultation paper on security issues which is clearly relevant; this is appended as Annex C of the report.

Finally, if strong authentication technology is deployed in order to support commercial electronic transactions and strict access control with penalty clauses for breaches of security, then the technology must have sufficient backing in law to make the theory possible in practice. The crucial security service involved is "non-repudiation", which we defined as involving the recipient of a digitally signed message gathering sufficient evidence to ensure that the authenticated sender of the message cannot later deny having sent it, with trusted notarisation procedures and trusted time-stamps as the mechanisms to achieve this. The legal standing of the trusted services required to support this must be established (presumably involving licensing, audit and regulation) and the procedures for arbitrating disputes must be clearly laid down.

7. Outline Scenario for Solution

7.1 What should be Provided

There is a clear need for widespread deployment of technology to improve security of users logging into their computing systems, whether local login from their desk or remote login from the other side of the world. The two scenarios have different needs, as with remote login it is not possible to make any assumptions about the computing facilities and infrastructure that will be available. Challenge-response one-time-passwords based on users carrying password cards provides a mechanism that will work whatever the computer equipment and bandwidth available, and this technology is therefore ideal for travelling users. The cost of introducing this technology could largely be covered by the reduced time wasted by computer operations staff anticipating and dealing with the after-effects of password compromises.

Within the UK academic community, it is possible to do much better given an authentication infrastructure (and use of an approved smart-card where possible). The following gives an outline of what an infrastructure can and should do, for discussion purposes.

An authentication infrastructure requires

- deployment of directory systems, by whatever technology;
- organisation of local management in order to register and manage users.

An authentication infrastructure provides

- a means for advertising security credentials;
- a means for revoking security credentials;
- a model for how you convert credentials into an assurance of identity.

Options for an authentication infrastructure are

- A Kerberos infrastructure - as was discussed earlier, Kerberos authentication is excellent for small scale use in local networks, but does not scale well to a larger community. Either a central authentication server would be needed (and this would be a management nightmare) or a means to interlink local Kerberos domains would be needed. The infrastructure would also only be useful for negotiation based protocols (in other words it would not be useful for store-and-forward protocols such as email or secure documents) and may not provide non-repudiation. Therefore, this is not a useful infrastructure for the needs of the entire academic community. However, there are attempts to link Kerberos also to a Public Key technology; if that occurs, then much the same infrastructure would still be needed here too.
- Public key - public key authentication techniques inherently provide our scalability and local management requirements. They are equally applicable to all communications protocols and can provide the evidence required for non-repudiation. Public key based schemes are being deployed world-wide, and so the scheme provides interworking with all other communities. There are currently three flavours of public key infrastructure:
  - x509v3 certificates - as used by either the PKIX working group or the ICE-TEL project. Both of these define models for using X.509v3 to achieve useful results with all applications.
  - PGP certificates - as used by the PGP security package. Although normally used with email, the PGP certificate format could be used by any secure communication that understood the PGP trust model, and so reuse of PGP format is possible. However, it is rumoured that a future version of PGP will use X.509 certificates, thereby obsoleting the certificate format. Additionally, I believe that, under some circumstances, it may be possible to convert from X.509 certificates to PGP certificates (since they are both just a public key encrypted with a private key there is no fundamental difference)
  - SPKI - no-one really knows what this means yet, though it is likely to be just another certificate format.

  It should be noted that a public key authentication infrastructure does not necessarily require you to choose from these three possibilities. It is perfectly normal for a user to have multiple certificates for distinct purposes and it would not be unreasonable for different certificates to come from different technologies, as long as the impact on who could authenticate you under what circumstances was understood.

An infrastructure must be:

- scaleable to a community the size of UK academic community
- managed locally
- able to inter-work with infrastructures of other communities and in other countries (this could mean compatible or it could mean part of the same infrastructure)
- include support for many applications to use it

The various directory technologies meet these requirements. Use of a standard protocol and certificate format means that commercial products will contain support for the infrastructure (such as Netscape, which has announced support for LDAP and which uses X.509 certificates for its SSL protocol), thereby increasing the range of applications that can use the infrastructure and (in this case) making sure the popular ones are included.

An infrastructure does not need to:

- mandate applications and technologies - A user who wants to use PGP and SSL can easily have both a PGP certificate and an X.509 certificate, and can store one in an X.500 directory and one in a Whois++ directory. The user will, of course, not be able to communicate securely with a MOSS user who just has an X.509 certificate. So while there are definite advantages to widespread use of one technology, it can be left to user choice as long as the issues are explained.
- require major modification to applications - an example of what is meant here is that PGP can be integrated with an X.500 based architecture by writing an LDAP application that will take a certificate from the user's public keyring and store it in the directory, or will retrieve a certificate from the directory and add it to the user's keyring. Therefore, the user can use PGP unaltered and still take advantage of the infrastructure, though the basic PGP trust model will be unaltered.

An infrastructure must not:

- be specific to an application - we should avoid investing in an infrastructure for one application and then have to redo it for another implementation.

The requirement for uniformity of infrastructure is much more important than any desire for uniformity of certificate format or even uniformity of trust model; usually applications can cope with the existence of multiple certificates. It should be noted that the need for such an infrastructure is well-recognised in the IEFT and related bodies – but all the mechanisms for deploying it is not well-understood.

Thus, the overall impact of the infrastructure will be:

- central services required: a UKHE Certification Authority will be needed to certify each participating institution. This CA will form the root of a UKHE hierarchy and will (according to the ICE-TEL trust model) allow the UKHE hierarchy to be interlinked with other hierarchies thereby achieving interoperability with the wider world community
- local services required: an Organisational Certification Authority will be needed in each participating institution - they may choose to devolve responsibility further down to the departments, and this could be done either by having a SESAME style local registration authority in each department, or having a fully blown certification authority in each department.
- product development or enhancement that may be required: there is currently a serious shortage of systems for running certification authorities (those that do exist tend to be unfriendly), and this should be addressed. Other projects (such as ICE-TEL and the UKERNA secure email project) are looking at enhancing applications, and it is likely that a lot of work can be inherited from here.

Future changes in requirements can be anticipated and handled if the infrastructure and protocols used closely follow patterns of research and deployment seen elsewhere. There is currently a lot of work taking place in the area of distributed public key infrastructures and what finally emerges is undoubtedly going to resemble closely what we have proposed. Use of LDAP to access heterogeneous directories is definitely going to be widely supported in popular user agents. The exact formats of certificates and the specification of trust relationships are still a matter for debate, and several standards will emerge. Whether one will eventually dominate or whether each will coexist is largely irrelevant, as a flexible authentication infrastructure backed with a set of well-written user applications can easily choose the appropriate certificate for any transaction, and some inter-operation may even be possible. By focusing on distributed scaleable certificate distribution mechanisms, the infrastructure will be well placed to adapt to future changing requirements.

Any deployment of a key based infrastructure will falter unless there is investment in a distributed scaleable infrastructure. This requires a commitment by central funding bodies and a willingness by individual institutions to deploy. As has been seen many times in the past (notably with the X.500 PARADISE pilot), this cannot be taken for granted. Moreover, it is unlikely that an *ab initio* deployment will succeed; substantial pilots will be required to ensure that the right facilities are provided.

A security infrastructure is not just a computing activity; it is related to the whole area of use of electronic mechanisms in administration. The principal sources of the data are local data bases, and the principal users will be local services. As has been seen many times in the past (notably, again,

with the X.500 PARADISE pilot and the JNT Directory project), the whole-hearted participation of the institutions cannot be taken for granted – it is essential to secure the co-operation of the senior administrators in each of the higher education institutions.

As well as reducing the costs incurred by the need to anticipate and deal with the effect of security breaches, the cost of the infrastructure can be offset against the considerable savings that will be possible with the increased use of electronic data that that would occur but that is not possible with current weak security. It would not be too strong a statement to say that an authentication infrastructure (of some description) is a pre-requisite for making progress in the area of information provision and administration involving sensitive or financial information. An infrastructure that is inter-operable with the rest of the world opens up even more possibilities. This infrastructure would have many benefits for the whole higher education community and is in no way a purely technical matter for computer centres and scientific staff.

7.2 An Implementation Approach

7.2.1 The Aims and Justifications

We believe that an authentication infrastructure should be multi-purpose and distributed. We would propose that each HEFC institution already has a need for E-mail Directories for every staff member and student, Telephone Directories for all staff and most graduate students, Payroll Databases for all staff, Registration Databases for all students, Examination databases for all students, User Ids for electronic libraries and information services, identification mechanisms for each member of staff for building access. Moreover, many staff and students already have WWW access - and eventually all will have. I believe that JISC should capitalise on this by encouraging the universities to derive all these services from interacting databases. It is unlikely that the administrators would be prepared to see wide access to the databases themselves, there would have to be severe constraints on such access.

For this reason, we believe that the encouragement of moving into the electronic age in these various areas would immediately result in the need for on a need for distributed Directories. With the current move towards a UK-wide tied system of telephone lines for the Universities, but university extensions being unknown outside the site, a UK-wide telephone directory system for the universities is essential. The telephone numbers - being dependent on the room location, and E-mail systems - often providing mailboxes on Departmental systems - alone justify the establishment of distributed directories of all staff and research students. Moreover, these resources should be available at least on a country-wide basis, if not internationally. Once we have put in a distributed directory system covering the requisite staff - containing both telephone numbers and E-mail addresses as a minimum, it is straightforward to issue also X.509 Certificates for the relevant staff. The attributes of the Distinguished Names used can contain as much information as is needed for E-Lib billing purposes; this may be only at the Institutional level, but may be also Departmental or even personal.

It is not necessary to start from scratch; as a result of the PARADISE/NAMEFLOW project we have already gone a long way in establishing a structure for a National X.500 Directory system. However, this system has had variable support from the different institutions, is not very complete in its coverage, requires updating in the technology it uses, and needs more integrated thinking in how it should be extended and used. We recognise the ambivalent view both inside the Internet community and some parts of the university community to the use of X.500 as a directory technology; it would be comparatively straight-forward to re-process the basic data to use another such technology. It is essential, however, to decide quickly what technology to adopt. It is not even essential to adopt a single technology; it is quite feasible to adopt one technology for internal access, and to provide another partial version of the same date for external access by a different technology.

Certification Authorities (CAs), which issue the Private Keys and the Certificates can be set up on an institutional basis. In some cases there will be single CAs per institution, in some there will be a number of them. It is important that the issuance of certificates includes a proper identification and authorisation step; this is one of the reason some of the institutions may prefer a more distributed CA structure. In addition, one organisation, possibly UKERNA, should run a Top Level CA. That or another one should also be responsible for cross-certification - both with other UK institutions (for instance for Electronic Data Interchange for commercial transactions) and internationally. If it is

necessary from a data protection viewpoint, it would be possible to set up many of the attributes on a strongly protected basis; we would be surprised if this was found necessary.

In general software-based certificates are considered much weaker than hardware based one, but much cheaper. The cost of a Smart Card, which is one of the most important alternatives, would probably be of the order of £30-50 at present prices. This would not be a negligible expense if it was required for all staff and students; it would be less serious if it was limited to staff engaged in serious financial transactions, or in other particular positions of trust. The V3 certificates have a field for CA policies; this could be used to identify which staff had used smart cards, and which only software. If the cards could be used for other purposes - automatic entry into buildings, financial transactions etc., then their use might easily be justified.

7.2.2 The Steps

Regarding recommendations that institutions should improve facilities for users logging in from local or remote systems, we propose that:

- institutions should use software based solutions (such as Kerberos or S/Key) to allow their users to log onto their computers from within the institution;
- institutions should use hardware-based solutions (such as challenge-response password cards) to allow those users that require the facility to log onto their computers while visiting other institutions;
- JISC should commission a study into available smart-card technology to determine the availability and applicability of current products in the area, how they may be integrated into current systems, and the extent to which they can be re-used within other applications (such as building access, library cards, identity cards). The output of such a study should be the recommendation of a "JISC Approved Smart-card" and a simple demonstration of the card for local and remote login within the UK HE community.

Regarding recommendations to introduce a national authentication infrastructure, we propose that:

- a Public Key infrastructure be deployed, with at least one Registration Agent and one Certification Agent in each institution; and one root CA for the whole UK HE community.
- public keys should be made available in some directory structure. Institutions should be encouraged to deploy distributed directory systems in order to support both the authentication infrastructure and also their own internal administration. At this stage, we do not specify or rule out a particular directory technology and point out that the LDAP lightweight directory access protocol (which will be included in future versions of the Netscape WWW browser) is intended to be largely technology independent and interoperate with X.500, WHOIS++ and other technologies. It is clear that some institutions will be unable or unwilling to participate in directory activities immediately - while these institutions will still be able to play some part in the authentication infrastructure, it is likely to be at a reduced level probably based on bilateral key exchange, and this must be defined.
- the infrastructure should contain both X.509v3 and PGP certificates (and any other certificate formats required by important applications, though these are expected to be the principle two for the foreseeable future). X.509v3 certificates should be handled according to the procedures specified by bodies such as the EC ICE-TEL project and the IETF PKIX working group. PGP certificates should be handled according to procedures specified by the UKERNA secure email project.
- JISC should pilot projects on the following components, using the authentication infrastructure:
  - deployable and supportable CAs and RAs;
  - WWW/directory enabled user agents for popular university admin, Elib & Email applications;
  - sample EDI applications for purchasing;
  - a secured multimedia conferencing system.

Regarding recommendations to move towards a national directory infrastructure to support authentication, we propose that:

- there be a series of consultations with at least three groups in the universities (Directors of Computer Centres, Librarians, and Senior Administrators) and also with national service providers. The aim would be to establish the principles of the need for a Directory infrastructure, the requirements for tools to extract the Directory data from existing databases, and the impact on possible applications caused by the existence of an authentication infrastructure. JISC should commission additional tools required to provide bulk loading of directories from existing databases, if required.
- there be a review of the current NAMEFLOW Directory service, to ascertain whether it provides a basis for a suitable system if upgraded to the X.500(93) standard, if the schemas are modified to meet the needs of the authentication infrastructure and if LDAP access via the WWW is provided.
- consideration be made as to whether the Directory project be integrated with the UKERNA PGP Email project, and JIBS Authentication Server. The PGP project should be encouraged to consider how their key maintenance problems might be eased by the use of the distributed directory structure. At present PGP is the largest user of E-mail, but that initiative is not favoured by the commercial providers. It would not be a problem to see directories hold X.509 certificates, PGP certificates and, if so specified by Electronic Commerce requirements, further types.
- a directory pilot be established to ensure the provision of directories for telephone numbers, email and certificates for a number of institutions. Preference should be given to those institutions who commit to use the directories for other purposes also like buildings access and E-library usage.
- particular effort should be devoted to a pilot between a small number of sites for University Registrars; if such a pilot could include The Association of Heads of University Administrators (AHUA), one of the Higher Education Funding Councils (e.g. HEFCE), and the University Central Admissions System (UCAS), it would be very worthwhile in influencing this important community.
- the major providers of E-Lib services be encouraged to pilot authentication systems using these directories.
- large-scale approaches be made (perhaps via TERENA and DANTE) to some other European Union countries to undertake a similar activity on a pilot basis; it is likely that at least Germany, the Netherlands and Norway would collaborate, but there may well be others.
- approach the European Union (under the Framework programmes) and DTI (under Foresight initiatives) to co-fund this activity first on a Pilot scale, and later as a full scale system.

Appendix A: List of Contacts

The following people were nominated or suggested as being people to be contacted, and have been consulted by email or by telephone.

SURNAME	FIRST	ORGANISATION	TELEPHONE	E-mail
Blanchet	Louis	MIDAS	0161-275 6062	P.L.Blanchet@mcc.ac.uk,
Brookes	Piete	Cambridge U	01223-334659	Piete.Brooks@cl.cam.ac.uk
Bruce	Rachel	HEFCE	0171-873 2610	
Burnhill	Peter	EDINA	0131-650 3301	p.burnhill@ed.ac.uk
Chiswell	Ron	MIDAS	0161-275 6062	
Cornwall	Trevor	EON	0191-227 3040	Trevor.Cornwall@unn.ac.uk
Davnall	Sarah	COPAC	0161-275 6074	zzacurl@midas.ac.uk
Dempsey	Lorcan	UKOLN	01225-826254	lisld@ukoln.bath.ac.uk
Ferguson	Nicky	SOSIG, FIG	0117-928 8471	nicky.ferguson@bristol.ac.uk
Fitzgerald	Jacqueline	HEFCE	0171-873 2599	J.Fitzgerald@hefce.ac.uk,
Foster	Jill	ESRC	0191-222 8250	Jill.Foster@newcastle.ac.uk,
Gibson	Chris	Academic Press	0171-482 2893	
Hamilton	Martin	ROADS, FIG	01509-228237	martin@net.lut.ac.uk
Islei	Gerd	ERIM, FIG	01865-735422	islei_g@templeton.oxford.ac.uk
Jackson	Denis	UKERNA	01235-822340	Denis.Jackson@ukerna.ac.uk
Johnson	Mike	CHEST	01225-826042	Mike.Johnson @bath.ac.uk,
Knight	Jon	ROADS, FIG	01509-228237	J.P.Knight@lut.ac.uk
Larbey	Dave	EDIS, FIG	01603-592426	D.Larbey@uea.ac.uk,
Murray	Robin	Fretwell Downing	0114-281 6000	
Mumford	Ann	AGOCC	01509-222312	A.M.Mumford@lut.ac.uk,
Nicholson	Dennis	BUBL	0141-552 3701	D.M.Nicholson@strath.ac.uk,
Pentz	Ed	Academic Press	0171-482 2893	
Ramsden	Anne	ERCOMS	01908-834924	ar@dmu.ac.uk,
Robiette	Alan	FIGIT	01203-524459	cudaf@csv.warwick.ac.uk,
Robinson	Brian	RUDI, FIG	01707-284166	B.P.Robinson@hertfordshire.ac.uk
Rushbridge	Chris	Warwick U	01203-524979	C.A.Rushbridge@warwick.ac.uk
Rzepa	H	CLIC, FIG	0171-594 5774	H.Rzepa@ic.ac.uk,
Simmons	John	BIDS	01225-826194	
Slater	John	Kent U	01227-764000	J.Slater@kent.ac.uk
Smethurst	Barry	BIDS	01225-826194	ccsbs@bath.ac.uk
Tedd	Mike	FIGIT	01970-622422	MDT@aber.ac.uk,
Weston	Sue	UKERNA	01235 822253	
Winkworth	Ian	Phoenix, FIG	0191-227 4126	Ian.Winkworth@unn.ac.uk
Wood	Dee		01235-822257	dwood@salixedu.demon.co.uk
Singleton	Alan	IOP	0117-929 7481	
Zedlewski	Eddie	NISS	01225-826042	Eddie.Zedlewski@niss.ac.uk

Appendix B: BIDS Submission

BATH INFORMATION AND DATA SERVICES - DRAFT

Towards a Uniform Framework for JISC Bibliographic Services

User Registration and Authentication

J.A.Simmons

7 June, 1996

1.0 Introduction

The problem of User Registration and Authentication is both complex and vast. There are potentially hundreds of thousands of users of the services offered by the National Datacentres. The problem is exacerbated by the dynamic nature of the user base; each academic year approximately three hundred thousand new users appear and a similar number become ineligible to use services as they leave higher education. The problem is also becoming more complex as new services are created in areas such as electronic journals.

Any mechanism that is put in place must address the needs of several parties; the users themselves, the librarians (or others) who administer access to services from subscribing institutions, the service providers, and the data providers. The users need be able to access services to which they have rights with the minimum of difficulty, and preferably without having to remember different usernames and passwords for each service. The site administrators need to be able to give their users access to services without getting involved in an onerous administrative task. The service providers need to be able to cope with the scale of any scheme and provide secure access for authorised users to the datacentres service machines. The data providers need to be reassured that the scheme is reasonably watertight and that their data is only being accessed by persons who are eligible and that they are abiding by the terms of the licence under which it was provided.

2.0 Background

The three national data centres at Manchester, Edinburgh and Bath have historically had different approaches to the problems of user registration.

Manchester and Edinburgh have existing services which are provided free but with an imperative that individual users sign undertakings regarding use of the data. This has led to schemes for individual user registration, administered by the data centres but with the possibility of having facilities to devolve registration to other authorised persons. The BIDS services at Bath have used a system of shared usernames.

2.1 BIDS

BIDS stepped back from individual registration before the start of its initial ISI service. The potential requirement to register over five hundred thousand users that were constantly changing was too complex to manage at that time. There was a requirement from the data provider, ISI, that individuals should sign a declaration regarding the conditions of use of the data and so the onus was put upon the subscribing institutions to gather such signatures. A shared username scheme was put in place whereby institutions could ask for a reasonable number of usernames (generally less than 100) to be allocated at a departmental or group level. This allows institutions to get usage breakdowns within their site at whatever level of granularity they require (except individuals!).

The shared BIDS usernames are backed up by calling address validation; calls into a service must come from a DNS registered machine (or registered X25 address) from a subscribing site. It is

acknowledged that this is not totally secure. However it does prevent ex students from gaining access when they have left HE, and also prevents non UK access. There is no evidence of any major abuse of the system, and the leakage at the edges appears to be within the bounds that are acceptable to the data providers.

2.2 MIDAS

The MIDAS services at Manchester primarily provide access to non-bibliographic datasets. They provide a range of research datasets covering census data, government surveys, macro-economic time series, digital maps and other miscellaneous datasets. In almost all cases individual user registration is required to gain access. This is in part a requirement of the data suppliers (e.g. the Data Archive or the Office of National Statistics) and in part due to the nature of the data services being provided (i.e. to fully exploit the data, the interface to it is mostly provided within a UNIX environment as opposed to a packaged environment and hence the need for individual UNIX logons).

2.3 EDINA

EDINA 92s first major bibliographic service, BIOSIS, was launched at the beginning of this year. EDINA requires individual user registration for access to its services. Having registered, users are then issued with an individual username/password. The practice is in line with other services provided from the Data Library at Edinburgh. Individual user registration has been adopted for the same reasons as described for MIDAS datasets. EDINA believe the scheme to be simple, robust and ultimately easiest for all concerned. It has the added advantage that statistical feedback on the use of services will be possible at a detailed level. Calling address validation is not used as it is believed that it is inadequate (verifying subscribing institutions, not registered individuals), restrictive (forcing users to fixed geography) and inherently complex (if ad hoc proxy geography is allowed).

On registration, prospective end-users must satisfy their local HEI site representative that they accept the terms and conditions of use of the service. The user is then allocated an EDINA card on which is printed their individual userid. The administration of userids is automated through an online registration database to which HEI site representatives have access. An expiry date can be set on userids and batch registration of groups of users is possible.

2.4 COPAC

Manchester launched their COPAC service in April this year. This national service gives online access to a consolidated catalogue from the Consortium of University Research Libraries (CURL). There is no intention to impose registration on users of the basic OPAC service. However, when the facility for document delivery is included in the service it will be necessary to authenticate users. At the moment, the intention is that authentication is done on the basis of shared usernames, and or, calling address validation.

2.5 CHEST purchased datasets

Since the original CHEST-ISI agreement, CHEST have altered the wordings in other data agreements so that the phrase 'users will sign ...' has been replaced by 'users will have signed ...'. The contract for CHEST-EMBASE is now the only one without the revised wording (and that comes up for renewal next year). Institutions have been able to take advantage of this to incorporate a statement regarding the use of protected data within their standard procedures - either for the use of library or computing services, or at matriculation time. Where institutions include the CHEST code of conduct for the use of software and data (or words equivalent to those in the code regarding the use of copyright data) in the their sign-on of new students and staff, then their users are eligible to use any of the data collections that have been procured via a CHEST agreement without the need for further signatures. Many HEI 92s have already included the CHEST code of conduct in their regulations.

2.6 Other Data Services

There are an increasing number of other suppliers of data services who will need to be involved in any widespread scheme for registration and authentication. For example, the EBSCO and NetFirst services provided by NISS, both require appropriate authentication. A number of eLib projects, and electronic journal services provided by publishers, will also require some form of registration and authentication.

3. WWW based services

The major bibliographic services provided by the National Datacentres are currently provided via Telnet based sessions. As the centres develop their services, and as a result of pressure from the user community, there is a move to provide WWW based interfaces. The mechanisms employed to authorise users described above could also be applied to these WWW services.However, there are difficulties when applying calling address validation to WWW accesses. A lot of users use proxy machines from which to make their WWW connections. This works if the proxy is within the same domain as the calling user; the call will be seen as coming from the correct institution. When the proxy is remote from the user (e.g. one of the national or regional WWW cache sites) then the call is going to appear to come from the wrong place. There is also a growing number of users who are working away from site, either at home (using a commercial Internet provider) or from a remote institution. Again, the calling address of the user will not map onto the subscribing institution.

In the short term, the breakdown of the calling address checking mechanism can be patched-up by restricting users to local proxies and by allocating individual user ids to people working away from site. This requires additional facilities to allow site administrators to have more devolved control of their users. In the longer term, a national authentication scheme for UK Higher Education is needed. Such a scheme would need to have the flexibility to cope with remote and local users.

4. NISS / BIDS Authentication Server

As part of the development of BIDS WWW based services a collaboration has been set up with NISS/CHEST who have been working on an authentication server for some time. The authentication server is designed for a WWW environment and can be accessed from a number of 91service 92 machines. A table of users and their resource entitlements is held by the server; the resources are usually services (e.g. ISI, EMBASE, etc.) but could be defined down to pages of material. There is a set of site administrator facilities to allow devolved administration to the site librarian for setting up and amending user ids. At this stage the user ids are based on existing BIDS and NISS ids and are generally allocated at a departmental level within an institution. The technology will support individual user ids. The Authentication Server could be extended to incorporate other services apart from those offered by BIDS and NISS.

There is a requirement for a national scheme to authenticate users connecting to JISC sponsored services. Otherwise, there will be a proliferation of schemes offered by the service providers and general confusion for users and HEI site administrators.

5. User Requirement

The JIBS User Group is the successor to the BIDS User Group and now has a remit to represent users of the EDINA, BIDS and Manchester COPAC services.

It has recently started an initiative on the issue of registration and authorisation. Firstly the Group sent a message to the discussion list *lis-bids-users* asking for comments on current practice within institutions. Subsequently the Chair of JIBS sent a letter to the data providers and the service providers. The thrust of the letter is that the requirement for signatures from users is an administrative burden for the sites and suggests that entitlement to services is established as part of the user's registration to local computing or library facilities. It then goes on to suggest that a user could be issued with a single user id which would be valid for all JISC services.

In part, the requirement for individual registration for datasets has already been obviated by the CHEST wording in their current contracts with the data suppliers. However, there is still an issue (e.g. at EDINA and MIDAS) where the service provider requires a declaration from the user regarding the use of the service provider computing facilities. This could be addressed by all HEIs universally enforcing the CHEST Code of Conduct when locally registering users for access to computing facilities.

The second issue (single user ids for all services) can be met by either expanding the NISS/BIDS authentication scheme to incorporate the other national services, or to use another national scheme (e.g. based on a user 92s local authorisation and rights being propagated across the network).

6. National Datacentres common requirement

The role of the data centres should be to promote the legitimate use of their data services while maintaining the integrity of their service machines and ensuring the rights of the data suppliers. User registration and authentication are mechanisms to police that process. The subscribing institutions have an important stake in the discussion in that they are expected to administer the registration process. To a first approximation the use of shared usernames has met the needs of most parties. The move to WWW based services, where the user is more likely to be 91removed 92 from his or her home geographic location (either physically or by the nature or their WWW connection), increases the pressure for individual user ids. The scale of the task of individually registering all UK HE users is beyond the resources of any one data centre. The long term aim must be to devise a single scheme for all of UK HE that meets the needs of all parties (users, subscribers, site administrators, service providers and data providers).

Appendix C: UK Government Paper

This section gives a recently published UK government consultation paper on security issues.

PAPER ON REGULATORY INTENT CONCERNING USE OF ENCRYPTION ON PUBLIC NETWORKS

I. Summary

1. The Government recognises the importance of the development of the Global Information Infrastructure (GII) with respect to the continuing competitiveness of UK companies. Its aim is to facilitate the development of electronic commerce by the introduction of measures which recognise the growing demand for encryption services to safeguard the integrity and confidentiality of electronic information transmitted on public telecommunications networks.

2. The policy, which has been decided upon after detailed discussion between Government Departments, involves the licensing and regulation of Trusted Third Parties (hereafter called TTPs) which will provide a range of information security services to their clients, whether they are corporate users or individual citizens. The provision of such information security services will be welcomed by IT users, and will considerably facilitate the establishment of, and industry's participation in, the GII, where trust in the security of communication has been acknowledged to be of paramount importance. The licensing policy will aim to preserve the ability of the intelligence and law enforcement agencies to fight serious crime and terrorism by establishing procedures for disclosure to them of encryption keys, under safeguards similar to those which already exist for warranted interception under the Interception of Communications Act.

3. The Government intends to bring forward proposals for legislation following consultation by the Department of Trade and Industry on detailed policy proposals.

II Background

4. The increased use of IT systems by British business and commerce in the last decade has been a major factor in their improved competitive position in global markets. This reliance on IT systems has, however, brought with it increased security risks; especially concerning the integrity and confidentiality of information passed electronically between trading bodies. The use of encryption services on electronic networks can help solve some of these security problems. In particular TTPs will facilitate secure electronic communications either within a particular trading environment (eg between a bank and its customers) or between companies, especially smaller ones, that do not necessarily have any previous trading relationship.

5. In developing an encryption policy for the information society, we have also considered how the spread and availability of encryption technology will affect the ability of the authorities to continue to fight serious crime and terrorism. In developing policy in this area, the Government has been concerned to balance the commercial requirement for robust encryption services, with the need to protect users and for the intelligence and law enforcement authorities to retain the effectiveness of warranted interception under the Interception of Communications Act (1985).

6. Consideration by Government has also been given to the requirement for business to trade electronically throughout Europe and further afield. The inter-departmental discussions have therefore taken into account draft proposals by the European Commission, concerning information security (which include the promotion of TTPs), and discussions on similar issues taking place within the OECD.

III The Government's Proposals

(a) Licensing

7. By their nature, TTPs, whatever services they may provide, will have to be trusted by their clients. Indeed in a global trading environment there will have to be trust of, and between, the various bodies fulfilling this function. To engender such trust, TTPs providing information security services to the general public will be licensed. The licensing regime would seek to ensure that organisations and bodies desiring to be TTPs will be fit for the purpose. The criteria could include fiduciary requirements (eg appropriate liability cover), competence of employees and adherence to quality management standards. TTPs would also be required to release to the authorities the encryption keys of their clients under similar safeguards to those which already exist. We would expect organisations with existing customers, such as banks, network operators and associations (trade or otherwise) to be prime candidates for TTPs.

8. The Government will consult with organisations such as financial services companies, who have made existing arrangements for the use and provision of encryption services, with the intention of avoiding any adverse effects on their competitiveness. It is not the intention of the Government to regulate the private use of encryption. It will, however, ensure that organisations and bodies wishing to provide encryption services to the public will be appropriately licensed.

(b) Services Offered

9. The services which a TTP may provide for its customers will be a commercial decision. Typically, provision of authentication services may include the verification of a client's public key, time stamping of documents and digital signatures (which secure the integrity of documents). TTPs may also offer a service of key retrieval (typically for documents and files that have been encrypted by employees) in addition to facilitating the real time encryption of a client's communications.

10. Licensed TTPs operating within a common architectural framework, on a European or even a global basis, will be able to facilitate secure communications between potential business partners in different countries. Providing the respective clients trust their TTPs, secure electronic commerce between parties who have not met will become possible because they will have confidence in the security and integrity of their dealings.

(c) Architecture and supporting products

11. It is envisaged that a common architectural framework will be needed to support the information security services being offered by TTPs in different countries. Clearly this will be a matter for negotiation between interested parties taking into account developments in international standards organisations. The architecture would need, however, to support both the provision of integrity and confidentiality and therefore be capable of verifying public encryption keys and escrowing private ones. There is no reason why it should not also support a choice of encryption algorithms, such as those on the ISO (International Standards Organisation) register.

12. In support of such an architectural framework we would envisage manufacturers developing software or hardware products for use by the business community. Such products will need to be consistent with whatever standard (or standards) are arrived at to enable TTPs to interoperate. The type of algorithm used for message encryption, and whether it is implemented in hardware or software, will be a matter of business choice.

(d) European Union

13. The Government is working closely with the European Commission on the development of encryption services through their work on information security. Arrangements concerning lawful interception and the regulation of TTPs in that context are matters for Member States to determine. However, the Commission has an important role in facilitating the establishment of an

environment where developments in the use of TTPs can be fostered. The Commission should soon be in a position to bring forward a programme of work involving, for example, the piloting and testing of TTP networks.

(e) OECD

14. The Government are also participating in discussions at the OECD on encryption matters. Where possible we will encourage the development of networks of TTPs which facilitate secure electronic trading on a global basis.

(f) Export Controls

15. Export controls will remain in place for encryption products (whether in hardware or software form) and for digital encryption algorithms. However, to facilitate the participation of business and commerce in the information society the Government will take steps, with our EU partners, with a view to simplifying the export controls applicable to encryption products which are of use with licensed TTPs.

IV Consultation

16. Officials from the Department of Trade and Industry have already held preliminary discussions with various industry group on the general concepts surrounding the provision of encryption services through TTPs. A more formal consultation on the Government's proposals will be undertaken by the Department of Trade and Industry with all interested parties prior to the bringing forward of legislative proposals. The Government recognises that the successful facilitation of electronic commerce through the introduction of information security services by TTPs either in the UK or in Europe, will, to a significant extent, depend on their widespread use across business. It will therefore be important to secure the broad acceptance of the business community for the Government's proposals. The Department will pay particular attention to this during the consultation process.

Appendix D: References

This section lists URLs where important information can be found. It will be expanded in the final version of this document.

Digest access authentication	ftp://nic.nordu.net/internet-drafts/draft-ietf-http-digest-aa-04.txt
gss-api-V1	ftp://nic.nordu.net/rfc/rfc1508.txt
gss-api-V2	ftp://nic.nordu.net/internet-drafts/draft-ietf-cat-gssv2-06.txt
moss	ftp://nic.nordu.net/rfc/rfc1847.txt
	ftp://nic.nordu.net/rfc/rfc1848.txt
pct	http://pct.microsoft.com
pem	ftp://nic.nordu.net/rfc/rfc1421.txt
	ftp://nic.nordu.net/rfc/rfc1422.txt
	ftp://nic.nordu.net/rfc/rfc1423.txt
	ftp://nic.nordu.net/rfc/rfc1424.txt
pep	http://www.w3.org/pub/WWW/TR/WD-http-pep.html
pgp	http://www.pgp.net
pgp/mime	ftp://nic.nordu.net/internet-drafts/draft-elkins-pem-pgp-04.txt
pkcs	http://www.rsa.com/pub/pkcs/ascii/overview.asc
s/mime	http://www.rsa.com/pub/S-MIME/smimequa.htm
sdsi	http://theory.lcs.mit.edu/~rivest/sdsi.ps
sea	http://www.w3.org/pub/WWW/TR/WD-http-sea.html
set	http://www.visa.com
shttp	ftp://nic.nordu.net/internet-drafts/draft-ietf-wts-shttp-01.txt
ssh	http://www.cs.hut.fi/ssh
ssl	http://www.netscape.com/newsref/std/SSL.html
	ftp://nic.nordu.net/internet-drafts/draft-freier-ssl-version3-01.txt
whois++	ftp://nic.nordu.net/rfc/rfc1835.txt