



Information Security Group

**IY5601**

**D. Identity management**

Chris Mitchell

<http://www.isg.rhul.ac.uk/~cjm>


[c.mitchell@rhul.ac.uk](mailto:c.mitchell@rhul.ac.uk)

1

All corrections and suggestions for improvement to this course material are gratefully received – please send any comments to:

[c.mitchell@rhul.ac.uk](mailto:c.mitchell@rhul.ac.uk)

Please feel free to re-use all or part of this presentation; however, if you do, it would be much appreciated if an acknowledgement of Royal Holloway as the source of this material could be given. I would also value learning about your experiences with this material.



Information Security Group


## Structure of this part of the course

1. Introduction
2. Single sign-on
3. Kerberos
4. Liberty Alliance
5. Case study: Microsoft InfoCard
6. Further resources

2

This lecture is divided into the following main parts:

1. Introduction
2. Single sign-on
3. Kerberos
4. Liberty Alliance
5. Case study: Microsoft InfoCard
6. Further resources



Information Security Group

## Scope

- Part D of IY5601 is concerned with identity management.
- We focus on single sign-on (SSO) systems, a key component of an identity-management system.
- We give a taxonomy of SSO systems, and look in detail at two examples of such systems.
- We also look at an example of an identity-management system, Microsoft's InfoCard.

3

Part D of IY5601 is concerned with identity management. We focus on single sign-on (SSO) systems, a key component of an identity-management system. We give a taxonomy of SSO systems, and look in detail at two examples of such systems. We also look at an example of an identity-management system, namely Microsoft's InfoCard.



## Background

- When a user wishes to make use of a service, the service will typically wish to be sure who the user is (e.g. for charging purposes).
- This requires the user to provide an *identity*, and also to give the means (via one or more *credentials*) for the service provider to *authenticate* the claimed identity.

4

When a user wishes to make use of a service, the service will typically wish to be sure of the identity of the user (e.g. for charging purposes). This requires the user to provide an *identity*, and also to give the means (via one or more *credentials*) for the service provider to *authenticate* the claimed identity (that is, verify in some way that the user is entitled to use the provided identity).




## Identities

- A user may have many identities (with associated identifiers) for use with different service providers.
- For example:
  - an employee may have an employee number for use with his/her employer;
  - a citizen has one or more numbers for interactions with government;
  - a user of Internet services (e.g. messaging) may have multiple names, each used with a set of service providers.

5

A user may have many identities (with associated identifiers) for use with different service providers. For example:

- an employee may have an employee number for use with his/her employer;
- a citizen has one or more numbers for interactions with government;
- a user of Internet services (e.g. messaging) may have multiple names, each used with a set of service providers (possibly where each set has only one member).



Information Security Group

## Credentials

- To enable a service provider to authenticate a user as a legitimate holder of an identity, the user may be required to provide one or more credentials.
- Possible credentials include:
  - a password;
  - a biometric sample;
  - a public key certificate;
  - a MAC computed using a shared secret key;
  - a signature on a challenge provided by the service provider.

6

To enable a service provider to authenticate a user as a legitimate holder of an identity, the user may be required to provide one or more credentials. Possible credentials include:

- a password;
- a biometric sample;
- a public key certificate;
- a MAC (message authentication code) computed using a shared secret key;
- a signature on a challenge provided by the service provider.



Information Security Group


## Authorisation

- Once an entity has been authenticated, the service provider needs to decide whether or not to grant the requested service.
- This is referred to as *authorisation* (i.e. is the holder of this identity authorised to access this service?).
- This could, for example, be supported using server-held Access Control Lists (ACLs).

7

Once an entity has been authenticated, the service provider needs to decide whether or not to grant the requested service. This is referred to as *authorisation* (i.e. is the holder of this identity authorised to access this service?).

This could, for example, be supported using server-held Access Control Lists (ACLs). Alternatively, the requester of service might provide a statement signed by the resource-owner, saying that requester should be granted access.



Information Security Group

## Privacy

- In some cases, the requester of the service may wish to have a degree of privacy provided.
- For example, the requester may not wish his/her identity to become known to other entities.
- We next consider three different aspects of privacy.

8

In some cases, the requester of the service may wish to have a degree of privacy provided. For example, the requester may not wish his/her identity to become known to other entities. We next consider three different aspects of privacy.




## Anonymity

- A user may wish to be able to access a service in an *anonymous* way.
- Anonymity means that no party will learn any of the identities of the user.
- Providing anonymity for free services is relatively simple.
- If payment is needed, then an anonymous payment system is needed, e.g. cash or e-cash.
- True ('absolute') anonymity is difficult to achieve, since even revealing an IP address to some extent compromises it.

9

A user may wish to be able to access a service in an *anonymous* way. Anonymity means that no party will learn any of the identities of the user. Providing anonymity for free services is relatively simple. If payment is needed, then an anonymous payment system is needed, e.g. cash or e-cash. True ('absolute') anonymity is difficult to achieve, since even revealing an IP address to some extent compromises it.



Information Security Group


## Pseudonymity

- Pseudonymity is a lesser form of anonymity, in which the user reveals a special type of identity to the service provider known as a *pseudonym*.
- Typically, new pseudonyms will be generated regularly, i.e. pseudonyms are typically short-lived.

10

Pseudonymity is a lesser form of anonymity, in which the user reveals a special type of identity to the service provider known as a *pseudonym*. Typically, new pseudonyms will be generated regularly, i.e. pseudonyms are typically short-lived.

An example of short-lived pseudonyms is provided by the TMSIs used by GSM.



Information Security Group


## Unlinkability

- Unlinkability is a privacy property required to support the use of pseudonyms.
- Two pseudonyms are unlinkable if a third party cannot tell whether or not they belong to the same user.
- In practice, absolute unlinkability is often difficult to achieve, since the authorisation process may reveal information about the user.

11

Unlinkability is a privacy property required to support the use of pseudonyms. Two pseudonyms are unlinkable if a third party cannot tell whether or not they belong to the same user.

In practice, absolute unlinkability is often difficult to achieve, since the authorisation process may reveal information about the user. This is because the access rights given to a user may help identify the user.



Information Security Group

## Structure of this part of the course

1. Introduction
2. Single sign-on
3. Kerberos
4. Liberty Alliance
5. Case study: Microsoft InfoCard
6. Further resources

12

We next provide an introduction to SSO systems. We do this by providing a taxonomy of such systems, which will enable us to characterise the different types of SSO system which can be deployed.



## SSO – topics to cover

- a. Introduction to SSO.
- b. A taxonomy of SSO systems.
- c. Properties of SSO schemes.
- d. Liberty Alliance.
- e. Microsoft Passport.
- f. A local true SSO scheme.
- g. A proxy based pseudo-SSO scheme.
- h. Concluding remarks.

13

This discussion of Single sign-on (SSO) is divided into the following main parts:

- a. Introduction to SSO.
- b. A taxonomy of SSO systems.
- c. Properties of SSO schemes.
- d. Liberty Alliance.
- e. Microsoft Passport.
- f. A local true SSO scheme.
- g. A proxy based pseudo-SSO scheme.
- h. Concluding remarks.



## a. Introduction to SSO

- Single Sign-On (SSO) is a widely used term.
- Refers to the ability to 'log in' just once, and thereafter be automatically logged in to a variety of different services.
- This simplifies password management for end user.

Single Sign-On (SSO) is a widely used term. It refers to the ability to 'log in' just once, and thereafter be automatically logged in to a variety of different services. This simplifies password management for the end user.



## SSO and distributed computing

- Historically, SSO has been applied to managed environments, e.g. within a large company.
- Company provides SSO as a 'security layer' as part of the overall computing infrastructure.
- Products to provide SSO of this type are well-established.

15

Historically, SSO has been applied to managed environments, e.g. within a large company. The company would typically provide SSO as a 'security layer' as part of the overall computing infrastructure. Products to provide SSO of this type are well-established.



## Internet SSO

- In this course, the term SSO is used in a somewhat different context.
- Logging in to Internet Service Providers (SPs) is an everyday event.
- Internet SSO refers to the ability of an Internet user to log in just once to an entity (local or remote), which then avoids the need for Internet SP logins.

16

In this course, the term SSO is used in a somewhat different context. Logging in to Internet Service Providers (SPs) is an everyday event. Internet SSO refers to the ability of an Internet user to log in just once to an entity (local or remote), which then avoids the need for Internet SP logins.



## Why Internet SSO?

- Same reasons as traditional SSO – to make life easier for user.
- However, apart from avoiding use of trivial or written down passwords, also addresses a trust issue not arising in the corporate environment.
- If same password used with multiple SPs, this potentially enables one SP to impersonate user to another SP.

17

Internet SSO has been introduced for the same reasons as traditional SSO – to make life easier for the user. However, apart from avoiding the use of trivial or written down passwords, it also addresses a trust issue not arising in the corporate environment. That is, if the same password is used with multiple SPs, then this potentially enables one SP to impersonate the user to another SP.



## Internet SSO – where are we?

- Microsoft has introduced Passport, which provides an SSO service for Passport-registered users to Passport-registered SPs.
- Liberty Alliance formed from consortium of leading vendors to provide open specifications for Internet SSO.

18

Microsoft has introduced the Passport scheme, which provides an SSO service for Passport-registered users to Passport-registered SPs.

In parallel with the above, the Liberty Alliance has been formed from a consortium of leading vendors to provide open specifications for Internet SSO.




## b. A taxonomy of SSO systems

- With the sudden growth of interest in Internet SSO, seems worthwhile to take a step back and consider existing schemes within space of all possible types of SSO scheme.
- Enables properties of different schemes to be examined.
- Enables new schemes to be proposed.

19

With the sudden growth of interest in Internet SSO, it seems worthwhile to take a step back and consider the existing schemes within the space of all possible types of SSO scheme. A taxonomy of the type we now introduce enables the properties of a variety of different schemes to be examined. It also enables new schemes to be proposed.



Information Security Group

## Roles


- Roles in 'general' SSO system:
  - User,
  - Internet Service Provider (SP),
  - Authentication Service Provider (ASP).
- Assumed infrastructure: User host, SP host and ASP host, with Internet connectivity between hosts as necessary.

20

The principal roles in a 'general' SSO system are:

- User,
- Internet Service Provider (SP),
- Authentication Service Provider (ASP).

The assumed infrastructure to support SSO contains: User host, SP host and ASP host, with Internet connectivity between hosts as necessary.



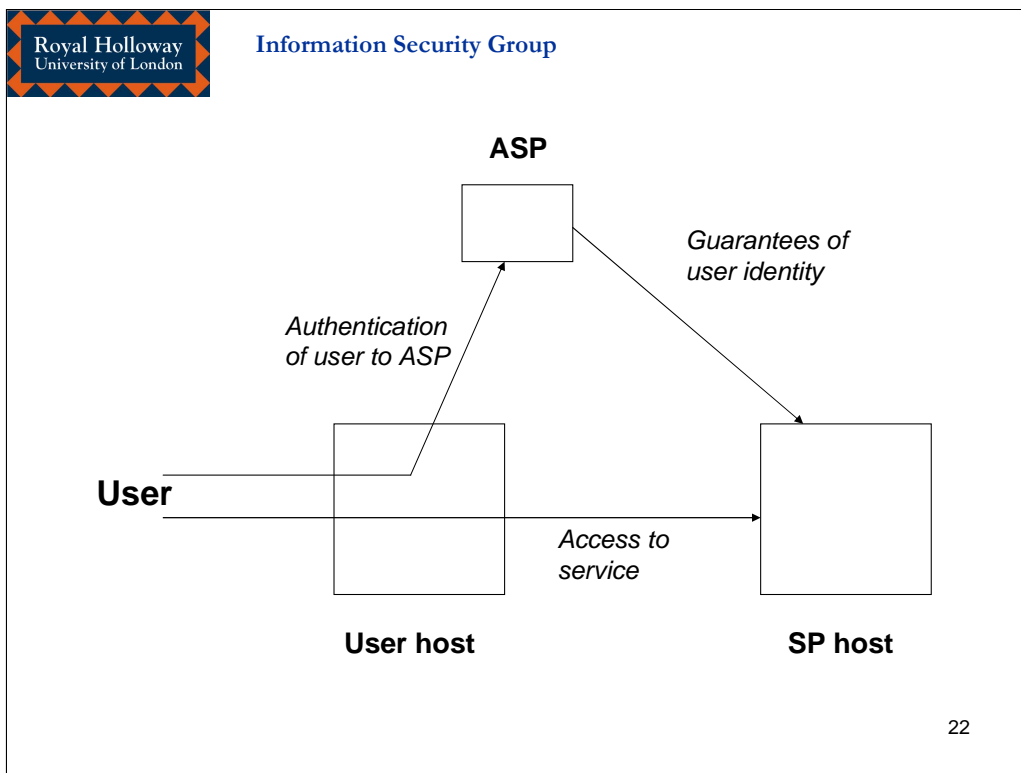
Information Security Group

## SSO operation


- User host and SP host have some kind of session (e.g. an SSL/TLS connection) – i.e. more than stateless http web connectivity.
- User authenticates to the ASP (in context of User/SP session).
- The ASP provides evidence to the SP regarding the identity of the user with whom the SP shares the session.

21

The User host and the SP host are assumed to share some kind of session (e.g. an SSL/TLS connection) – i.e. more than stateless http web connectivity. The User authenticates to the ASP (in the context of a User/SP session). The ASP provides evidence to the SP regarding the identity of the user with whom the SP shares the session.



The diagram shows the assumed model for SSO.



Information Security Group

## SSO identities and privacy

- ASP can use different identifiers for a User depending on which SP is involved.
- These identifiers could be SSO-specific, i.e. to provide User pseudonymity.

23

An ASP can use different identifiers for a User, depending on which SP is involved. These identifiers could be SSO-specific, i.e. to provide user pseudonymity.



## True and pseudo SSO (1)

- We divide SSO systems into two classes depending on nature of ASP/SP interaction.
- In pseudo-SSO, ASP simply manages the User authentication credentials for each SP.
- That is, the ASP simply performs the SP authentication process on behalf of the user.

24

We divide SSO systems into two classes depending on the nature of the ASP/SP interaction. In pseudo-SSO schemes, the ASP simply manages the User authentication credentials for each SP. That is, the ASP simply performs the SP authentication process on behalf of the user.



## True and pseudo SSO (2)

- In true SSO, the ASP has an explicit relationship with the SP.
- The ASP is authenticated by the SP and provides information about the User (e.g. including how the User was authenticated to the ASP).
- Note contrast with pseudo SSO where the SP need not be aware of the ASP.

25

In true SSO, the ASP has an explicit relationship with the SP. The ASP is authenticated by the SP and provides information about the User (e.g. including how the User was authenticated to the ASP). Note the contrast with pseudo SSO, where the SP does need not to be aware of the ASP.




## Local versus proxy-based

- The ASP host can be a remote 'proxy' or actually the same as the User host.
- For pseudo SSO, a local ASP is the 'obvious' case, where a process local to the user manages passwords for use with a variety of SPs.
- For true SSO, a proxy-based ASP is the 'obvious' case, since the ASP must be trusted by the SP.

26

The ASP host can be a remote 'proxy' or actually the same as the User host. For pseudo SSO, a local ASP is the 'obvious' case, where a process local to the user manages passwords for use with a variety of SPs. For true SSO, a proxy-based ASP is the 'obvious' case, since the ASP must be trusted by the SP.



Information Security Group

## Examples

- Microsoft Passport, Liberty Alliance and Kerberos are all **proxy-based true SSO** schemes.
- Programs which manage User passwords are **local pseudo SSO** schemes.
- **Proxy-based pseudo SSO** and **local true SSO** schemes can be devised which have practical advantages.

27

Microsoft Passport, Liberty Alliance and Kerberos are all **proxy-based true SSO** schemes. Programs which manage User passwords are **local pseudo SSO** schemes. **Proxy-based pseudo SSO** and **local true SSO** schemes can be devised which have practical advantages.




## c. Properties of SSO schemes

- SSO identity pseudonymity and unlinkability are potentially desirable.
- Pseudo-SSO schemes cannot guarantee unlinkability, since identifiers are SP-specific, e.g. email address, name.
- True SSO systems can be designed so that a different (and unlinkable) identifier is used with every SP. (However, depends on pseudonymous payment scheme!)

28

SSO identity pseudonymity and unlinkability are potentially desirable properties for an SSO scheme to possess. Pseudo-SSO schemes cannot guarantee unlinkability, since identifiers are SP-specific, e.g. email address, name. True SSO systems can be designed so that a different (and unlinkable) identifier is used with every SP. (However, unlinkability also depends on the existence of a pseudonymous payment scheme!)



Information Security Group

## Anonymous network access

- Even if true SSO scheme provides unlinkable pseudonymity, lower layer protocols may betray user identity via network addresses.
- This problem can be addressed by using anonymising proxies for network access.

29

Even if a true SSO scheme provides unlinkable pseudonymity, lower layer protocols may betray the user identity via the network addresses that are used. This problem can be addressed by using anonymising proxies for network access.




## User mobility

- Proxy-based SSO schemes inherently support user mobility (user just needs means to authenticate to the proxy).
- Local pseudo-SSO schemes can be made to support mobility by storing (encrypted) credential database at a third party – products exist.
- Local true SSO schemes more difficult because of trust requirements.

30

Proxy-based SSO schemes inherently support user mobility (the user just needs means to authenticate to the proxy). Local pseudo-SSO schemes can be made to support mobility by storing (encrypted) credential database at a third party – products of this type exist. Local true SSO schemes are more difficult to implement because of trust requirements.




Information Security Group

## Untrusted environments

- Users will sometimes want to access SPs from untrusted hosts (e.g. Internet cafés).
- Users will wish to avoid giving host access to long term secrets (e.g. passwords).
- In such cases, Proxy-based schemes advantageous, although initial authentication to proxy must be one-time.

31

Users will sometimes want to access SPs from untrusted hosts (e.g. Internet cafés). Users will wish to avoid giving an untrusted host access to long term secrets (e.g. passwords). In such cases, Proxy-based schemes are advantageous, although the initial authentication to the proxy must be one-time.



Information Security Group

## Costs

- In general, far less costly to deploy pseudo-SSO schemes, since no impact on SP.
- This may decrease over time, since pseudo-SSO ASP must reflect all changes in SP authentication method.
- Running costs of local SSO are likely to be less, since no server required.

32

In general, it is likely to be far less costly to deploy pseudo-SSO schemes, since there is no impact on SP. This advantage may decrease over time, since the pseudo-SSO ASP must reflect all changes in the SP authentication method.

The running costs of local SSO are likely to be less, since no server is required.



## Trust relationships

- Users and SPs need to trust ASP in all schemes.
- However, differences exist between true and pseudo SSO schemes.
- In a true SSO scheme, the SP/ASP trust relationship is explicit.
- In pseudo SSO scheme less clear – the SP may not be aware of ASP.

33

Users and SPs need to trust the ASP in all schemes. However, differences exist between true and pseudo SSO schemes. In a true SSO scheme, the SP/ASP trust relationship is explicit. In pseudo SSO scheme the trust relationship is less clear – the SP may not be aware of ASP.




## Evidence generation

- For proxy-based schemes, the proxy operator may act as a trusted third party to provide evidence of events.
- Situation better for true SSO, where ASP/SP relationship well-defined.
- Local SSO schemes much less useful, since evidence unreliable (although if User host is a 'trusted system' then evidence it provides may be of value).

34

For proxy-based schemes, the proxy operator may act as a trusted third party to provide evidence of events. The situation (with regard to evidence generation) is better for true SSO, where the ASP/SP relationship is well-defined. Local SSO schemes are much less useful for evidence generation than proxy-based schemes, since in this case the evidence is unreliable (although if the User host is a 'trusted system' then the evidence it provides may be of value).



Information Security Group

## Open/closed environments

- Privacy protection likely to be less of an issue in a closed environment, where costs are likely to be the main issue.
- Since costs are generally less for pseudo SSO, such systems may be preferred in corporate environments.
- Greater privacy protection possibilities may make true SSO more suitable for open environments.

35

Privacy protection is likely to be less of an issue in a closed environment, where costs are likely to be the main issue. Since costs are generally less for pseudo SSO, such systems may be preferred in corporate environments. Greater privacy protection possibilities may make true SSO more suitable for open environments.



## Security

- It is necessary to (somehow) bind user authentication to the ASP to the user/SP session.
- User does not necessarily authenticate the ASP – hence a false ASP may obtain user authentication information.
- This argues in favour of ‘one-time’ authentication to the ASP.

36

It is necessary to (somehow) bind the (one-off) process of user authentication to the ASP to the potentially multiple user/SP sessions. The user does not necessarily authenticate the ASP – hence a false ASP may obtain user authentication information. This argues in favour of ‘one-time’ authentication to the ASP.



## User control

- In a local SSO scheme, the user retains control over the SSO process.
- With (most) proxy-based solutions, User must select from a small number of SSO service providers, and hence has much less control over the process.
- There exist significant privacy issues with use of third party SSO providers.

37

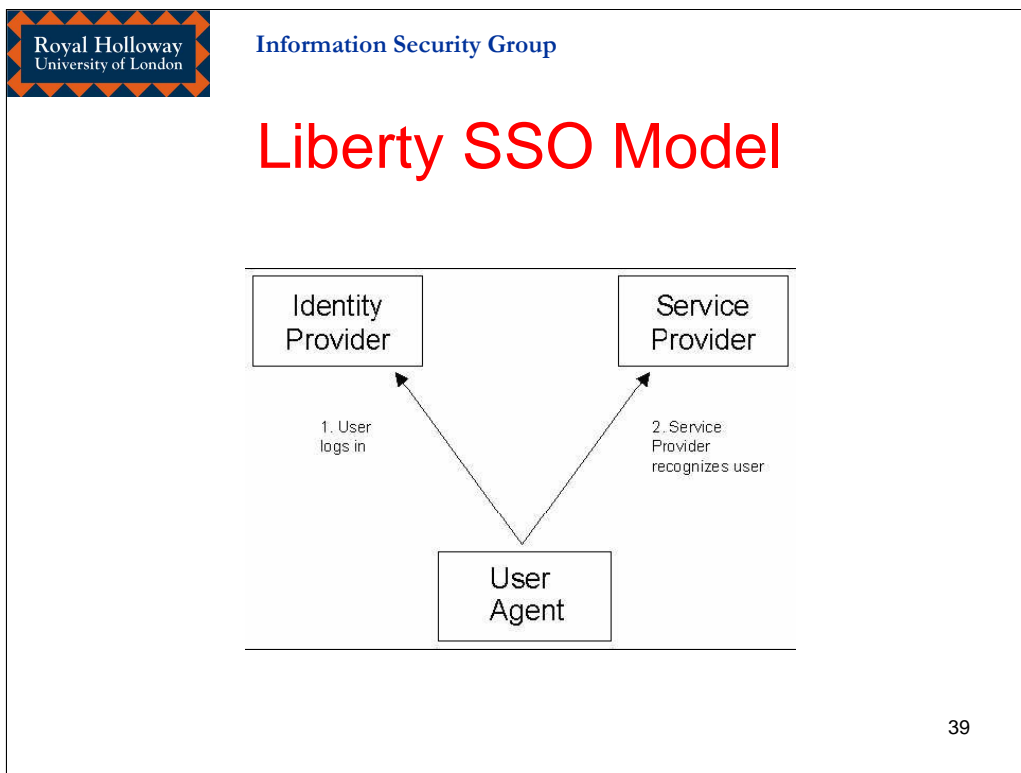
In a local SSO scheme, the user retains control over the SSO process. With (most) proxy-based solutions, the User must select from a small number of SSO service providers, and hence has much less control over the process. There exist significant privacy issues with the use of third party SSO providers since they will for example, know the identities of the ISPs with whom a user interacts.




## d. Liberty Alliance

- The Liberty Alliance is a consortium of companies interested in SSO and identity management.
- It has published a series of specifications for an 'open' XML-based SSO system.

The Liberty Alliance is a consortium of companies interested in SSO and identity management. It has published a series of specifications for an 'open' XML-based SSO system.



The slide shows the fundamental idea behind the Liberty ID Framework. A user logs in to an Identity Provider which subsequently helps the user be authenticated to multiple Service Providers.



Information Security Group

## Role of Identity Provider

- In Liberty, a User authenticates to a Liberty *Identity Provider* (IP), i.e. the ASP in the context of our taxonomy.
- The IP then automatically authenticates user to other SPs.
- User then needs only one password (or other means to authenticate to IP).
- Works using http redirection.

40

In Liberty, a User authenticates to a Liberty *Identity Provider* (IP), i.e. the ASP in the context of our taxonomy. The IP then automatically authenticates the user to other SPs. The User thus needs only one password (or other means to authenticate to IP). It works using http redirection.



## Liberty operation I

- Typical operational scenario is as follows.
- User visits web site of SP, and SSL connection established.
- SP then redirects user web browser to Liberty IP which establishes SSL connection and then authenticates the user (if necessary).
- Liberty IP then redirects user back to SP.

A typical operational scenario is as follows. The User visits the web site of SP, and an SSL connection established. The SP then redirects the user web browser to Liberty IP, which establishes SSL connection and then authenticates the user (if necessary). Liberty IP then redirects the user back to the SP.



## Liberty operation II

- Messages need to be passed between SP and IP.
- SP sends authentication request and IP responds with authentication response (containing 'security assertions').
- Messages passed either embedded in URLs or in http forms (using POST method).
- Syntax of messages based on SAML.

42


Messages need to be passed between the SP and the IP. The SP sends an authentication request, and the IP responds with an authentication response (containing 'security assertions'). Messages are passed either embedded in URLs or in http forms (using the POST method). The syntax of messages is based on SAML.



## Pseudonymity

- Liberty requires the IP to use a different pseudonym with each SP.
- Gives a level of unlinkability.
- However, may be compromised through network addresses.

Liberty requires the IP to use a different pseudonym with each SP. This gives a level of unlinkability. However, this unlinkability may be compromised through network addresses. It also requires complete trust in the IP.



Information Security Group

## Vulnerabilities

- Since the Liberty IP is potentially used by many web sites, compromising user authentication to the IP could be serious.
- Unfortunately, web spoofing, or presence of a single malicious web server, could achieve this compromise through false redirections.

44

Since the Liberty IP is potentially used by many web sites, compromising user authentication to the IP could be serious. Unfortunately, web spoofing, or the presence of a single malicious web server, could achieve this compromise through false redirections.




## e. Microsoft Passport

- Microsoft's passport server acts as an ASP.
- Users register with ASP by supplying email address and password.
- Every user given a unique 64-bit Passport User ID (PUID).
- SPs wishing to use Passport must register with Microsoft and pay a fee (and receive a secret key).

45

Microsoft's passport server acts as an ASP. Users register with the ASP by supplying an email address and a password. Every user is given a unique 64-bit Passport User ID (PUID). SPs wishing to use Passport must register with Microsoft and pay a fee (and receive a secret key).



Information Security Group

## Passport operation I

- SSL/TLS used to protect User host/Passport server and User host/SP channels (like Liberty).
- SP host redirects User browser to Passport server (ASP).
- ASP checks for Ticket Granting Cookie (TGC) in User host – if one found which checks correctly then OK.

46

SSL/TLS is used to protect the User host/Passport server and the User host/SP channels (just like Liberty). The SP host redirects the User browser to the Passport server (ASP). The ASP then checks for the presence of a Ticket Granting Cookie (TGC) in the User host – if one is found which checks correctly then all is OK.



## Passport operation II

- If not, then User authenticated and TGC created and stored on User host.
- The ASP now uses the TGC to create a set of cookies encrypted using the SP's secret key.
- User browser redirected back to SP, which reads the cookies.

47

If not, i.e. if no acceptable cookie is found, then the User is authenticated and a TGC is created and stored on the User host. The ASP now uses the TGC to create a set of cookies encrypted using the SP's secret key. Finally, the User browser is redirected back to the SP, which reads the cookies.



## Vulnerabilities

- Like Liberty, Passport is subject to redirection attacks where a malicious SP can redirect the User host to a fake ASP.
- This fake ASP can then capture user authentication information.
- Attack prevented if 'one time' user authentication method used.

Like Liberty, Passport is subject to redirection attacks, where a malicious SP can redirect the User host to a fake ASP. This fake ASP can then capture the user authentication information. This attack can be prevented if a 'one time' user authentication method used.



## f. A local true SSO scheme

- In a true SSO scheme the SP must trust the ASP to have correctly authenticated the User.
- This is tricky to achieve if the ASP is running on the User host!
- However, can be done if the SP trusts a hardware subsystem within the User host to act correctly.

49

In a true SSO scheme the SP must trust the ASP to have correctly authenticated the User. This is tricky to achieve if the ASP is running on the User host! However, this can be achieved if the SP trusts a hardware subsystem within the User host to act correctly.



## Trusted platforms I

- Such a hardware subsystem can be implemented using a 'Trusted Platform', e.g. using a PC adhering to TCPA/TCG or Microsoft NGSCB (formerly Palladium) specifications.
- A paper design exists using TCPA specifications, where SP can verify the software running on the User host.

50

Such a hardware subsystem can be implemented using a 'Trusted Platform', e.g. using a PC adhering to TCPA/TCG or Microsoft NGSCB (formerly Palladium) specifications. A paper design exists using the TCPA specifications, where an SP can verify the software running on the User host.



Information Security Group

## Trusted platforms II

- SP host can also check origin (manufacturer) of trusted subsystem within the User host.
- If SP trusts the software state and the manufacturer, then the SP can trust the ASP.

51

The SP host can also check the origin (manufacturer) of the trusted subsystem within the User host. If the SP trusts the software state and the manufacturer, then the SP can trust the ASP.



## Properties

- User can retain control over SSO process.
- User does not need to choose between a small number of major SSO providers.
- Both SP and User gain through lack of need to pay fees for third party service provision.

In this scenario the User can retain control over the SSO process. The User does not need to choose between a small number of major SSO providers. Both the SP and the User gain through the lack of any need to pay fees for third party service provision.



## g. A proxy-based pseudo-SSO scheme


- We next briefly outline an example of a proxy-based pseudo-SSO scheme.
- This scheme, known as Impostor, has been developed at RHUL by Andreas Pashalidis as part of his PhD research project.

53

We next briefly outline an example of a proxy-based pseudo-SSO scheme. This scheme, known as Impostor, has been developed at RHUL by Andreas Pashalidis as part of a PhD research project.

Andreas's PhD thesis is available at:

<http://www.ma.rhul.ac.uk/techreports/2005/RHUL-MA-2005-13.pdf>



Information Security Group

## *Impostor*

- *Impostor* is a proxy-based pseudo SSO scheme, a prototype of which is available as open source.
- The User sets his web browser to use the SSO proxy (*Impostor*) host as a web proxy.
- The *Impostor* proxy monitors all User host/SP host http traffic.

54

*Impostor* is a proxy-based pseudo SSO scheme, a prototype of which is available as open source. The User sets his web browser to use the SSO proxy (*Impostor*) host as a web proxy. The *Impostor* proxy monitors all User host/SP host http traffic.

For further details see

<http://www.isg.rhul.ac.uk/~cjm/iassos2.pdf>

The open source project is available at:


<http://impostor.sourceforge.net/>



## *Impostor* operation

- *Impostor* is configured to recognise each SP's individual user authentication procedure.
- Requests for username/password are intercepted and 'completed' by *Impostor*, after *Impostor* has authenticated the user (preferably using a one-time method).

*Impostor* is configured to recognise each SP's individual user authentication procedure. Requests for username/password are intercepted and 'completed' by *Impostor*, after *Impostor* has authenticated the user (preferably using a one-time method).



Information Security Group

## *Impostor* properties

- Despite it being a proxy-based system, User retains control since User could run *Impostor* from an Internet-connected home PC.
- Transparent to SP, and hence immediate deployment possible.
- Ideal for untrusted environments.

56

Despite it being a proxy-based system, the User retains control since the User could run *Impostor* from an Internet-connected home PC. The scheme is transparent to the SP, and hence immediate deployment is possible. Such a scheme is ideal for use from untrusted environments.



Information Security Group

## h. Concluding remarks

- We conclude this introduction to SSO by summarising the properties of the various categories of SSO scheme.


57

We conclude this introduction to SSO by summarising the properties of the various categories of SSO scheme.



## Comparisons

<b>Local</b>	<b>Proxy-based</b>	<b>True SSO</b>	<b>Pseudo SSO</b>
<ul style="list-style-type: none"> <li>• Low running costs.</li> <li>• Retention of user control.</li> </ul>	<ul style="list-style-type: none"> <li>• Supports anonymous network access.</li> <li>• User mobility support.</li> <li>• Supports use in untrusted environment.</li> </ul>	<ul style="list-style-type: none"> <li>• Pseudonymity support.</li> <li>• Lower maintenance costs?</li> <li>• Well-defined trust relationships.</li> </ul>	<ul style="list-style-type: none"> <li>• Low deployment costs.</li> <li>• Unchanged SP authentication.</li> <li>• Better for closed systems.</li> </ul>



Information Security Group

## Structure of this part of the course

1. Introduction
2. Single sign-on
3. Kerberos
4. Liberty Alliance
5. Case study: Microsoft InfoCard
6. Further resources

59

We next consider the operation of the Kerberos authentication system in greater detail. Kerberos is probably the longest-standing example of a distributed SSO system.



## Kerberos – introduction

- Kerberos is a proxy-based true SSO scheme.
- Devised as part of Project Athena at MIT.
- Designed to provide means to authenticate workstation users (clients) to servers (and vice versa).
- Uses symmetric encryption and a Manipulation Detection Code (MDC).

60

Kerberos is a proxy-based true SSO scheme. It was originally devised as part of Project Athena at MIT in the 1980s. It has been designed to provide a means for workstation users (clients) and servers (and vice versa) to authenticate one another. MIT still maintains a website for Kerberos, and many of the original documents describing Kerberos can be found here:

<http://web.mit.edu/kerberos/www/>

I find Briang Tung's "Moron's Guide to Kerberos" very helpful – see:

<http://www.isi.edu/gost/brian/security/kerberos.html>

A number of versions of Kerberos have been produced; we concentrate on the latest version, known as Version 5. It is important to note that the previous version of Kerberos (Version 4) has been widely used, and is *significantly different* from Version 5. A complete specification of Kerberos Version 5 is given in Internet RFC 1510 (which is readily available from the Internet).

Kerberos Version 5 is based on the use of symmetric encryption and a Manipulation Detection Code (MDC) (for integrity and origin checking) and time-stamps (for freshness checking).

Kerberos is widely integrated into versions of the Unix operating system, and code implementing the protocol is available on the Internet.



## Kerberos – TTPs

- Kerberos makes use of two types of TTP:
  - an authentication server (AS), and
  - a ticket-granting server (TGS).
- User has a long-term shared secret key with the AS, which then sets up a short term shared secret key with the TGS.
- The TGS is then involved in setting up shared session keys between entities.

61

Kerberos makes use of two different types of TTP:

- an *authentication server (AS)*, and
- a *ticket-granting server (TGS)*.

The client has a long-term shared secret key with the AS, which is the used to set up a short term shared secret key with the TGS.

The TGS is then involved in setting up shared session keys between the client and server.

The AS and TGS jointly fill the role of the Identity Provider.

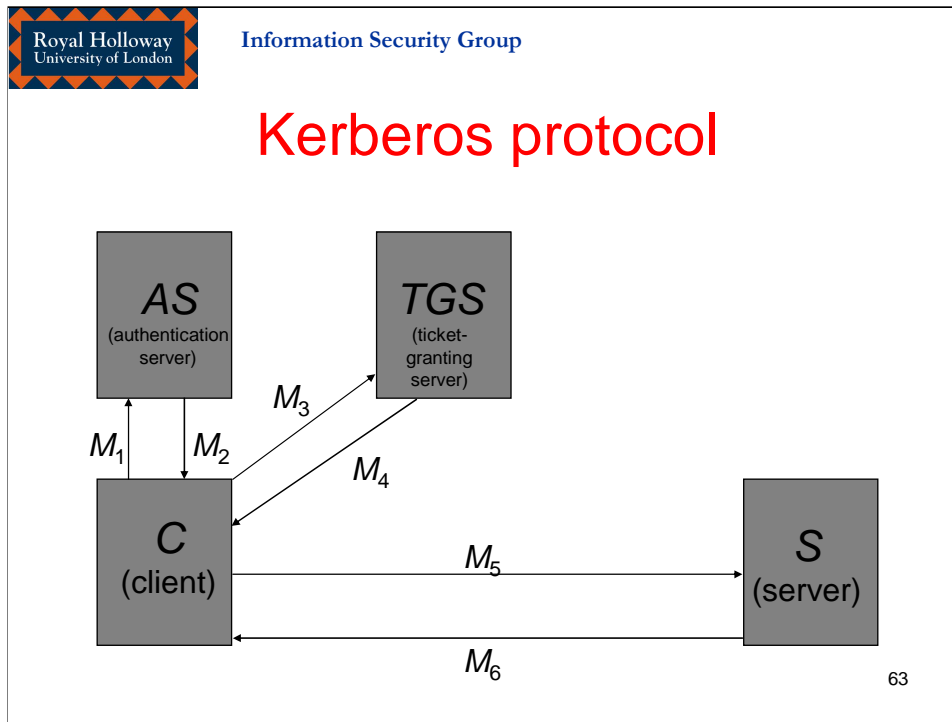


## Kerberos – motivation

- Idea of having two TTPs is that a user only needs load his/her long-term secret key into the work-station for the minimum time.
- Once the short-term secret key is established (with TGS) the long-term secret key can be erased from the workstation.
- This minimises the risk of exposure of the long-term secret key.

62

The idea of having two TTPs is that a user only needs load his/her long-term secret key (shared with the AS) into the work-station for the minimum amount of time. Once the short-term secret key is established (with the TGS), the long-term secret key can be erased from the workstation. This minimises the risk of exposure of the long-term secret key.



Messages 1 and 2 are exchanged between the client and the AS.

Messages 3 and 4 are exchanged between the client and the TGS (using a key provided by the AS). Message 3 and 4 can be repeated a number of times without repeating messages 1 and 2, during the lifetime of the key set up between the client and the TGS.

Messages 5 and 6 are exchanged between the client and server (using a key provided by the TGS). Message 5 and 6 can be repeated a number of times without repeating messages 3 and 4, during the lifetime of the key set up between the client and the server.

## Kerberos – message formats

$$M_1 (C \rightarrow AS) = C || TG || \text{times} || N_C$$

$$M_2 (AS \rightarrow C) = C || eK_{AST}(K_{CT} || C || \text{times}) || eK_{ASC}(K_{CT} || \text{times} || N_C || TG)$$

$$M_3 (C \rightarrow TGS) = S || \text{times} || N'_C || eK_{AST}(K_{CT} || C || \text{times}) || eK_{CT}(C || T_1)$$

$$M_4 (TGS \rightarrow C) = C || eK_{TS}(K_{CS} || C || \text{times}) || eK_{CT}(K_{CS} || \text{times} || N'_C || S)$$

$$M_5 (C \rightarrow S) = eK_{TS}(K_{CS} || C || \text{times}) || eK_{CS}(C || T_2)$$

$$M_6 (S \rightarrow C) = eK_{CS}(T_2)$$

64

$C$  denotes the identifier of the client.

$TG$  denotes the identifier of the TGS.

$S$  denotes the identifier of the server.

$N_C$  and  $N'_C$  are nonces generated by the client  $C$ .

$K_{AST}$  is a secret key shared by the AS and the TGS.

$K_{ASC}$  is a secret key shared by the AS and the client  $C$  (the long term user key).


$K_{TS}$  is a secret key shared by the TGS and the server  $S$ .

$K_{CT}$  is a secret key shared by the client  $C$  and the TGS (established by messages  $M_1$  and  $M_2$ ).

$K_{CS}$  is a secret key shared by the client  $C$  and the server  $S$  (established by messages  $M_3$  and  $M_4$ ).

$T_1$  and  $T_2$  are time-stamps.

'times' denotes a specified time interval (start time and end time) - it is used to limit the validity of a key.



Information Security Group

## Kerberos – issues

- In practice, the ‘top level’ key shared by the user and the AS ( $K_{ASC}$ ) may be derived from a password.
- In such a case, given access to message  $M_2$ , password-guessing attacks are possible.
- Also, the use of cryptography (i.e. MDC + encryption) by Kerberos is not in accordance with modern best practice.

65

In practice, the ‘top level’ key shared by the user and the AS ( $K_{ASC}$ ) may be derived from a password. In such a case, given access to message  $M_2$ , password-guessing attacks are possible (i.e. attacks where the interceptor tries every possible password from a list, until one is found which gives a correctly formatted plaintext). Also, the use of cryptography (i.e. the combination of an MDC and encryption) by Kerberos is not in accordance with modern best practice.

For further information on the limitations of Kerberos, see:

<http://www.cs.columbia.edu/~smb/papers/kerblimit.usenix.pdf>