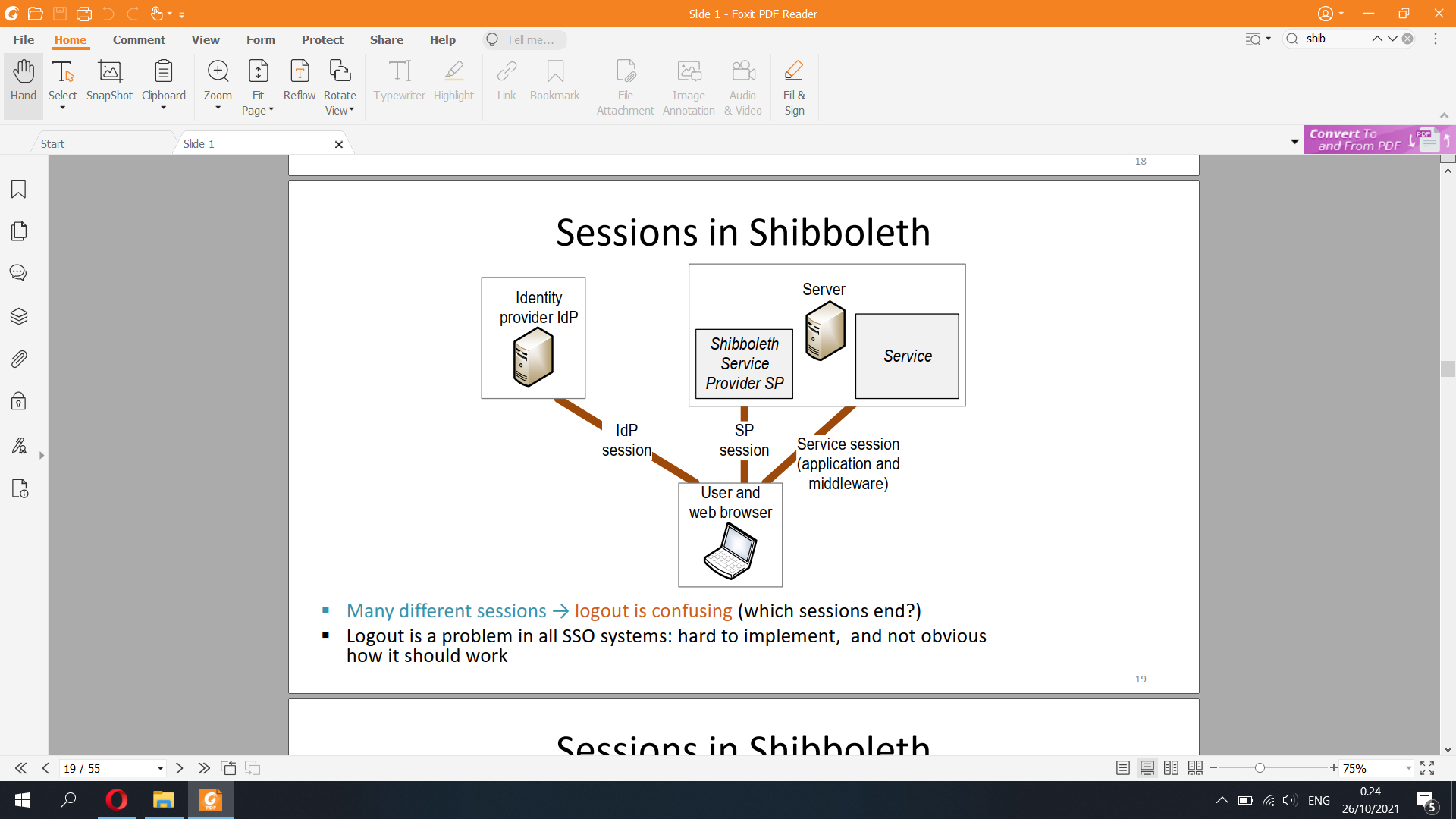


Sessions in Shibboleth



Sessions in Shibboleth

▪ Shibboleth implements two kinds of sessions:

– IdP session between browser and the IdP implemented with IdP cookies

→ user only needs to type in password once

– SP session between browser and the SP component with SP cookies

▪ Additional application sessions:

– Applications and web-application frameworks implement sessions using cookies

or with fields in URLs or web forms

▪ No single logout

– Logging out of SP does not usually log the user out of the IdP

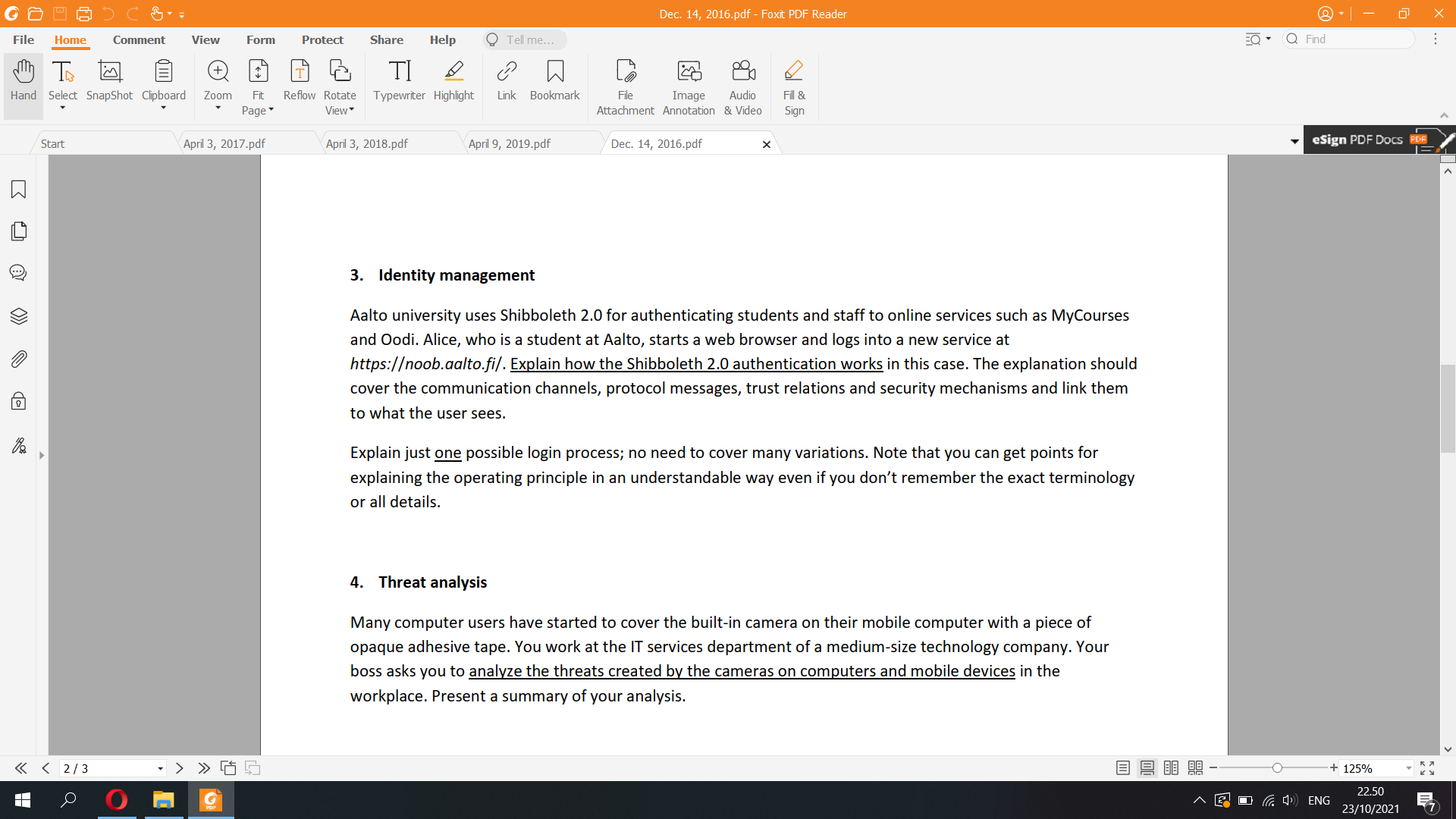
→ can log back to SP without password

– Logging out of IdP does not log the user out of SPs

– Logging out of one SP does not log the user out of other SPs

– Application sessions complicate the situation further

→ Shibboleth logout behavior is difficult to understand



Shibboleth is a web SSO within an organization (Aalto univ) based on SAML.

**Communication channels:**

**Protocol messages:**

**Trust relations**: In SAML single sign-on, a trust relationship is established between two parties: the identity provider and the service provider. The identity provider has the user identity information stored on a directory server. The service provider uses the identity provider's user identities for its own authentication and account creation.

**The identity provider and the service provider establish trust by exchanging a SAML metadata document with one another.**

**Security mechanisms:**

In the canonical use case:

A user first accesses a resource hosted by a web server (the service provider) that has Shibboleth content protection enabled.

The SP crafts an authentication request that is passed through the browser using URL query parameters to supply the requester's SAML entityID, the assertion consumption location, and optionally the end page to return the user to.

The user is redirected to either their home IdP or a WAYF (Where Are You From) service, where they select their home IdP for further redirection.

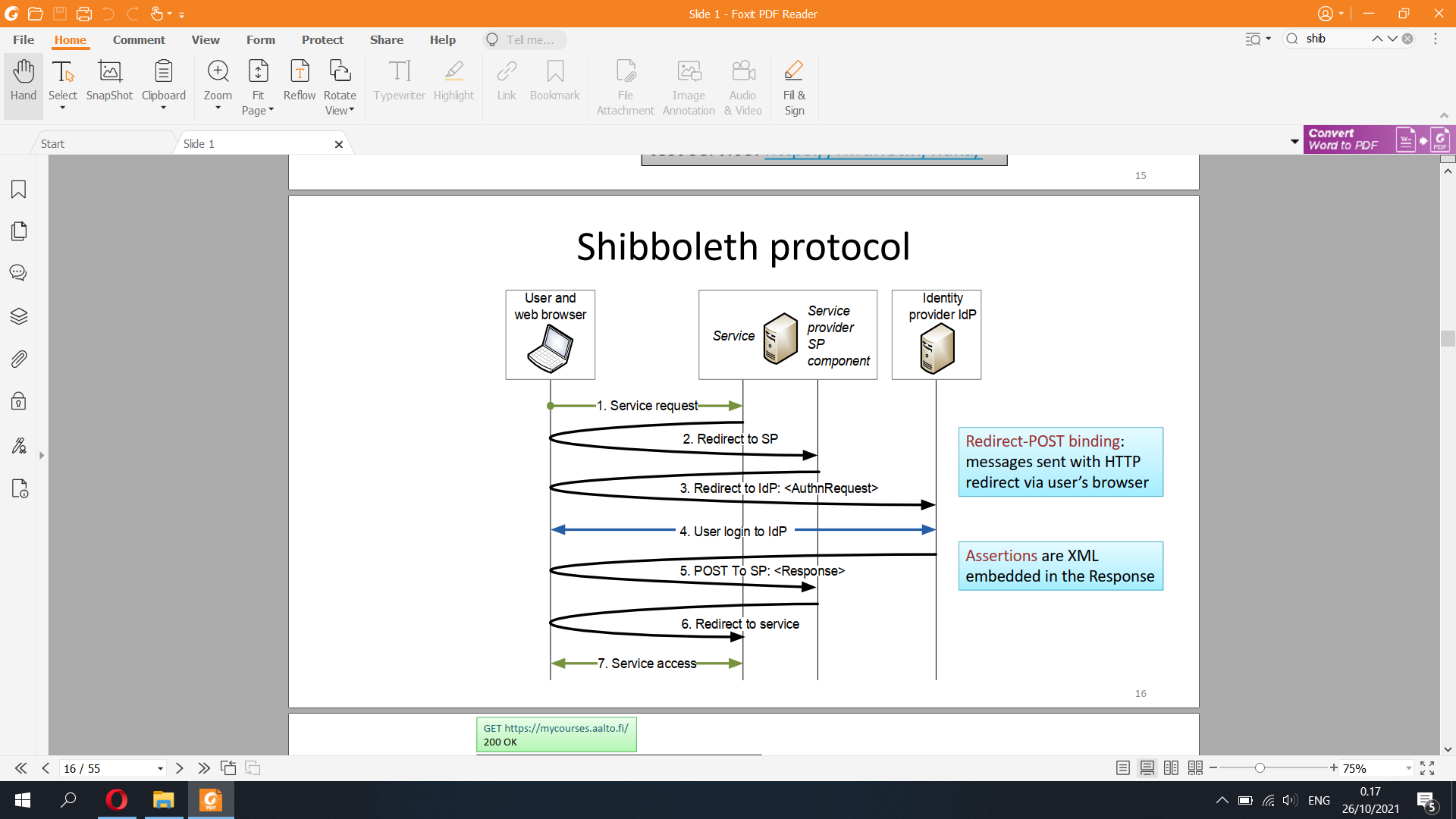
The user authenticates to an access control mechanism external to Shibboleth.

Shibboleth generates an authentication assertion with a temporary "handle" contained within it. This handle allows the IdP to recognize a request about a particular browser user as corresponding to the principal that authenticated earlier.

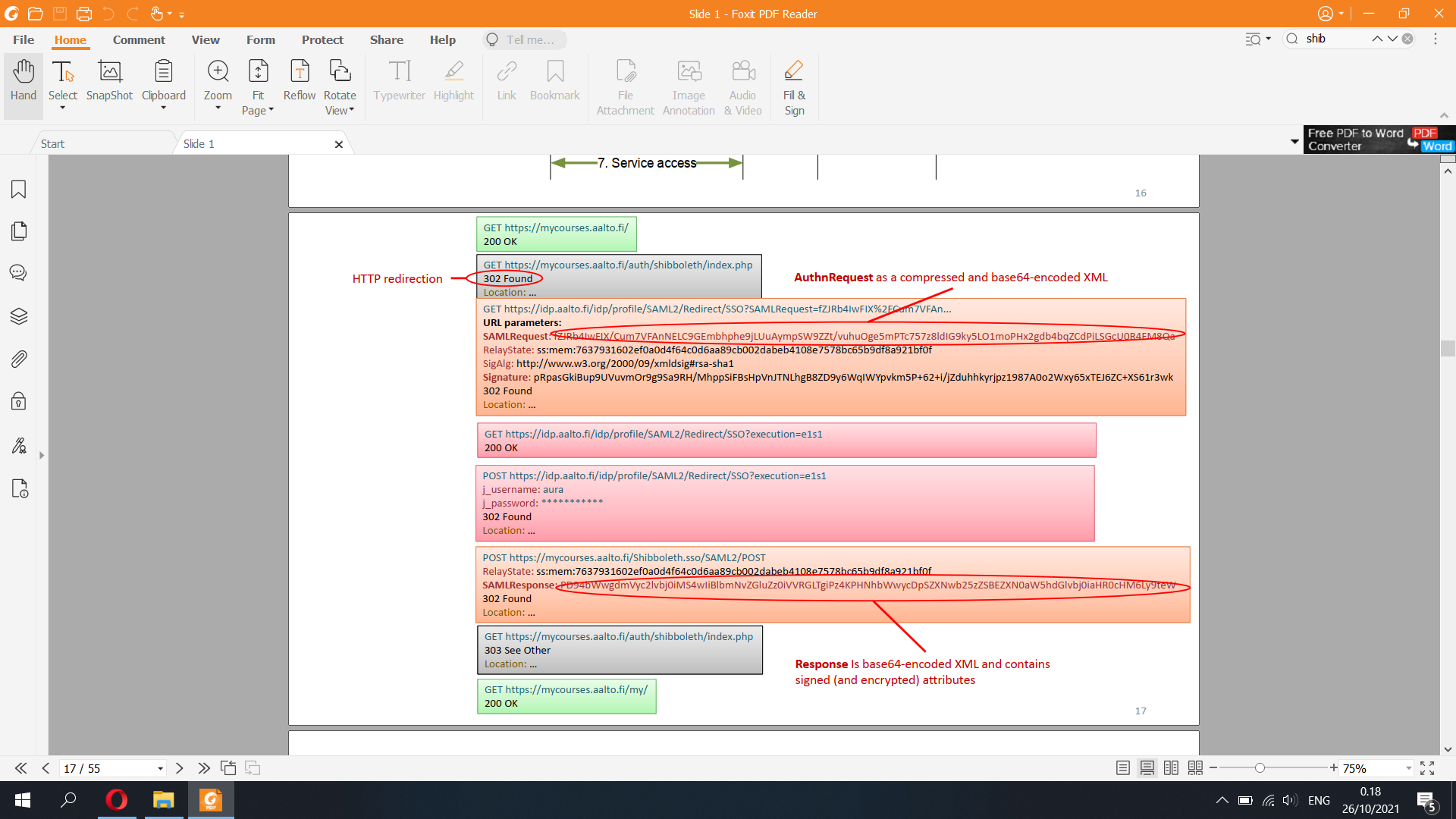
The user is POSTed to the assertion consumer service of the SP. The SP consumes the assertion and issues an AttributeQuery to the IdP's attribute service for attributes about that user, which may or may not include the user's identity.

The IdP sends an attribute assertion containing trusted information about the user to the SP.

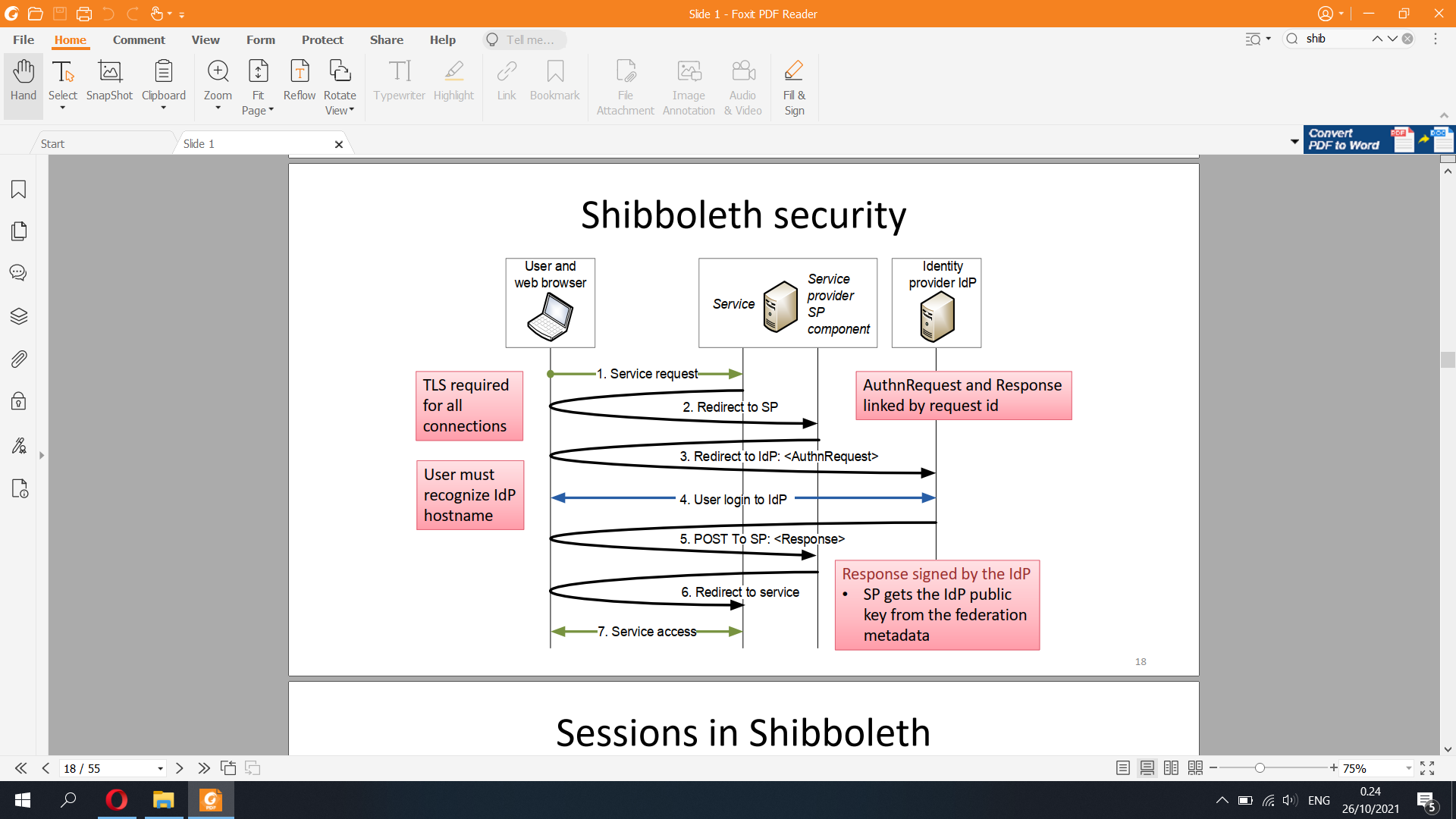
The SP either makes an access control decision based on the attributes or supplies information to applications to make decisions themselves.

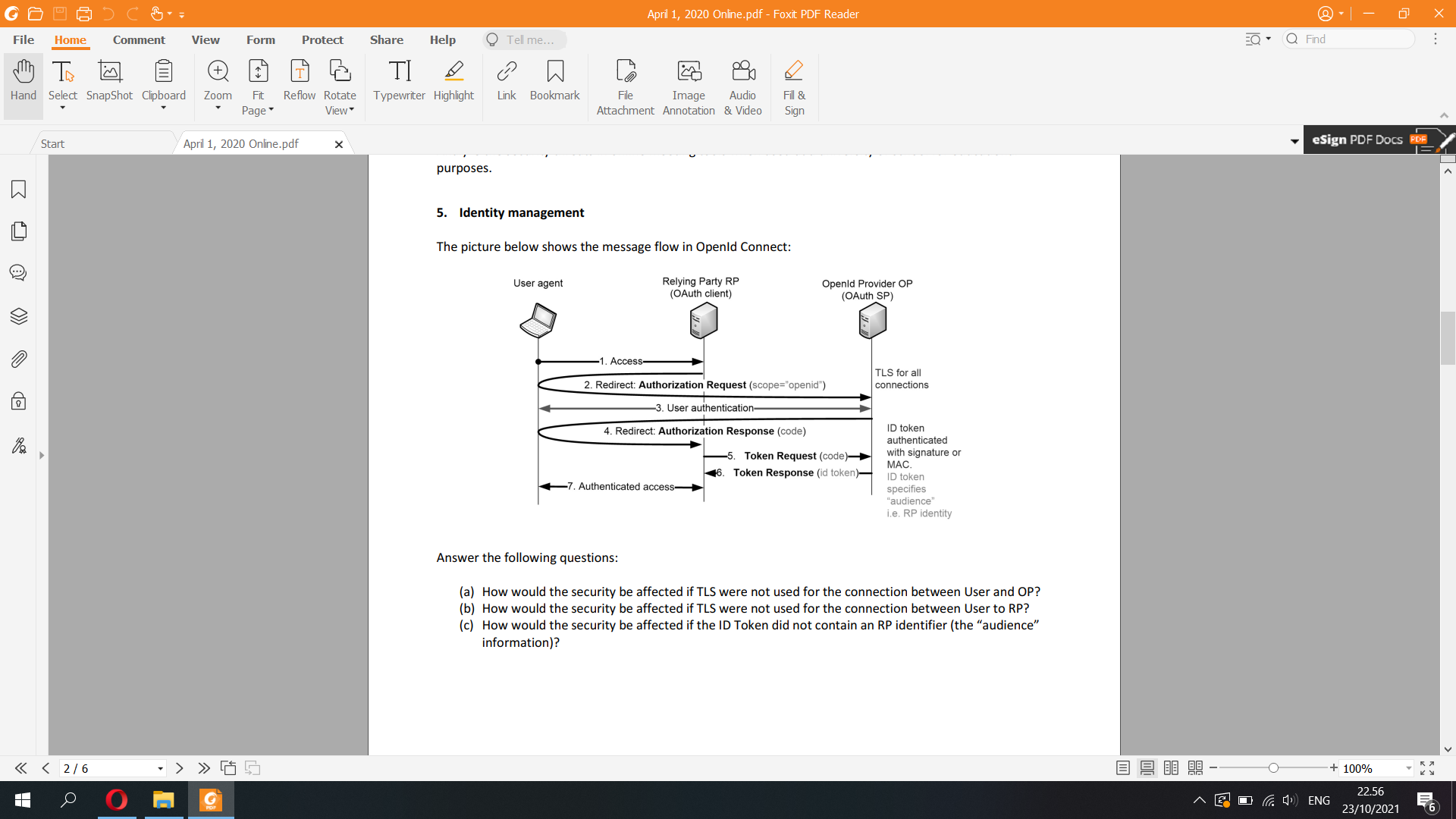


Get, Get, Get, Get, Post, Post, Get Get

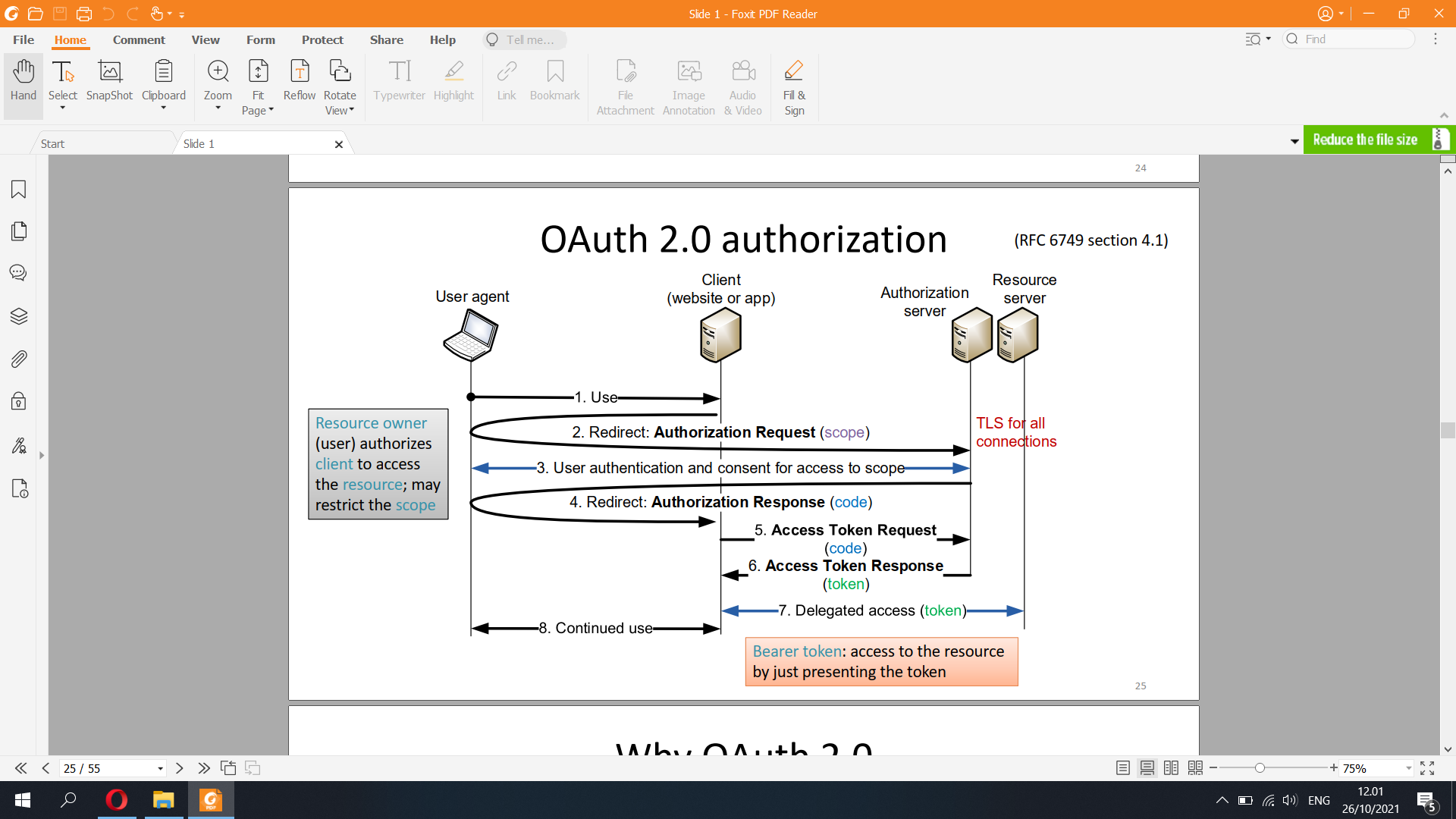


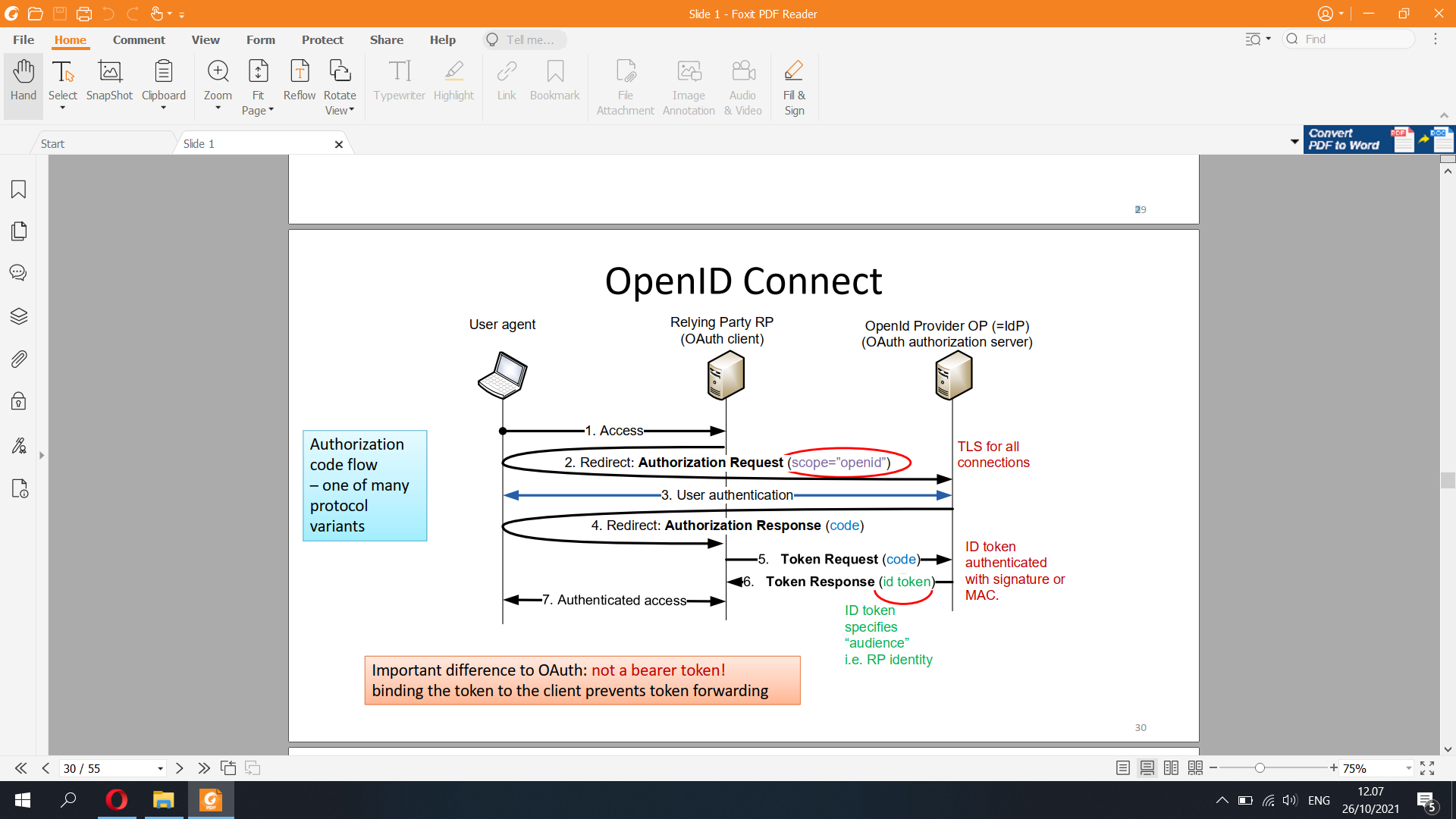
Shibboleth security





Oath 2 authentication





a) Without TLS between the User and OP,, sensitive authentication information such as username and passwords are easily stolen and hijacked by sniffing attackers in the network. The attacker can even spoof as the OP and extract information from the user.

b) Without TLS between the User and RP, credit card details and personal details can easily be gleaned by others, but also browsing habits, e-mail correspondence, online chats and conferencing calls can be monitored. By enabling client and server applications to support TLS, it ensures that data transmitted between them is encrypted with secure algorithms and not viewable by third parties.

c) If the ID token doesnt have RP identifier, the token wouldnt bind the token to that RP client. The RP can forward the token to other unintended fourth parties that collect datas from the id token.

The ID token resembles the concept of an identity card, in a standard JWT format, signed by the OpenID Provider (OP). To obtain one the client needs to send the user to their OP with an authentication request.

Features of the ID token:

Asserts the identity of the user, called subject in OpenID (sub).

Specifies the issuing authority (iss).

Is generated for a particular audience, i.e. client (aud).

May contain a nonce (nonce).

May specify when (auth\_time) and how, in terms of strength (acr), the user was authenticated.

Has an issue (iat) and expiration time (exp).

May include additional requested details about the subject, such as name and email address.

Is digitally signed, so it can be verified by the intended recipients.

May optionally be encrypted for confidentiality.

The ID token statements, or claims, are packaged in a simple JSON object:

{

"sub" : "alice",

"iss" : "https://openid.c2id.com",

"aud" : "client-12345",

"nonce" : "n-0S6\_WzA2Mj",

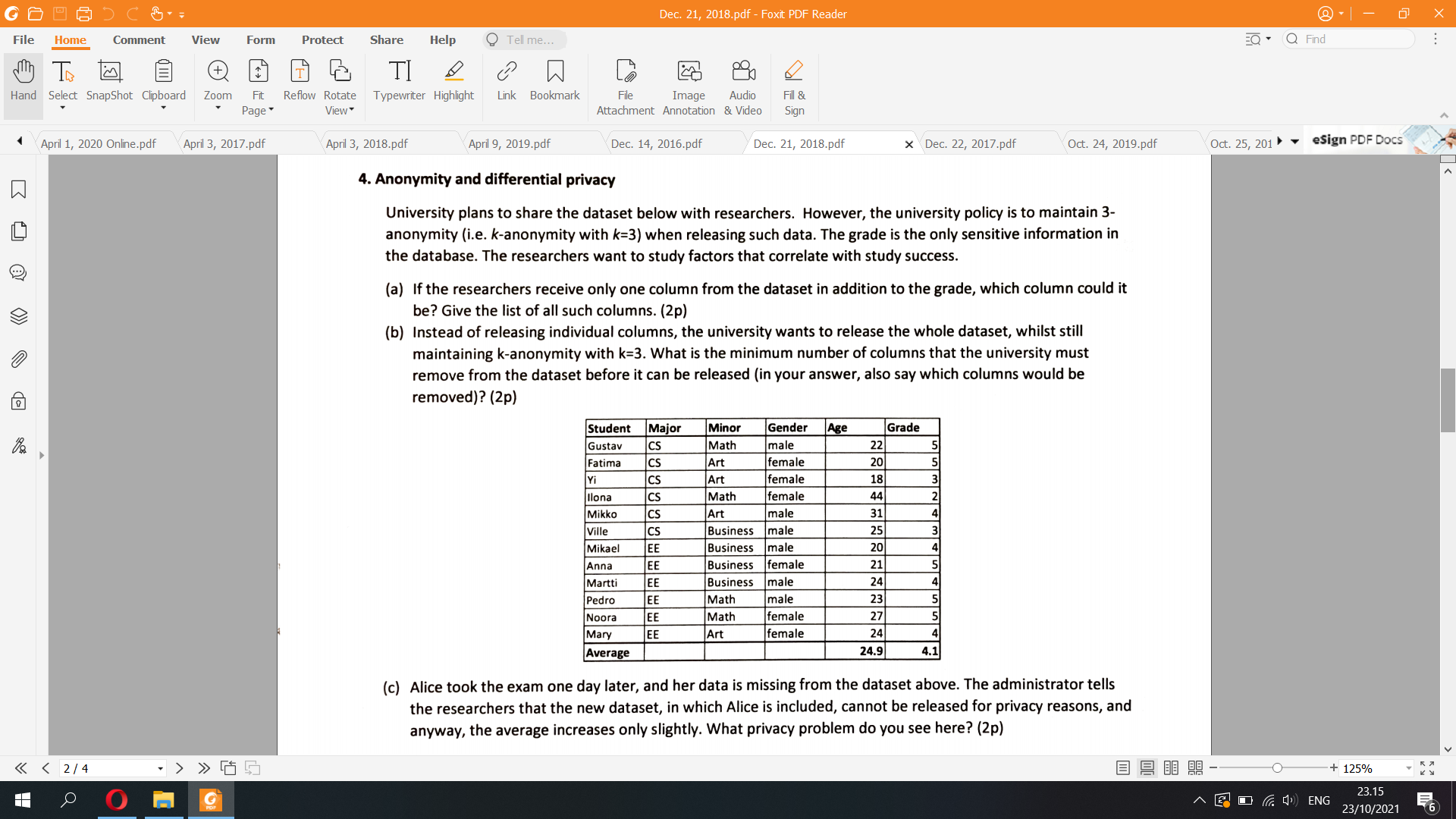
"auth\_time" : 1311280969,

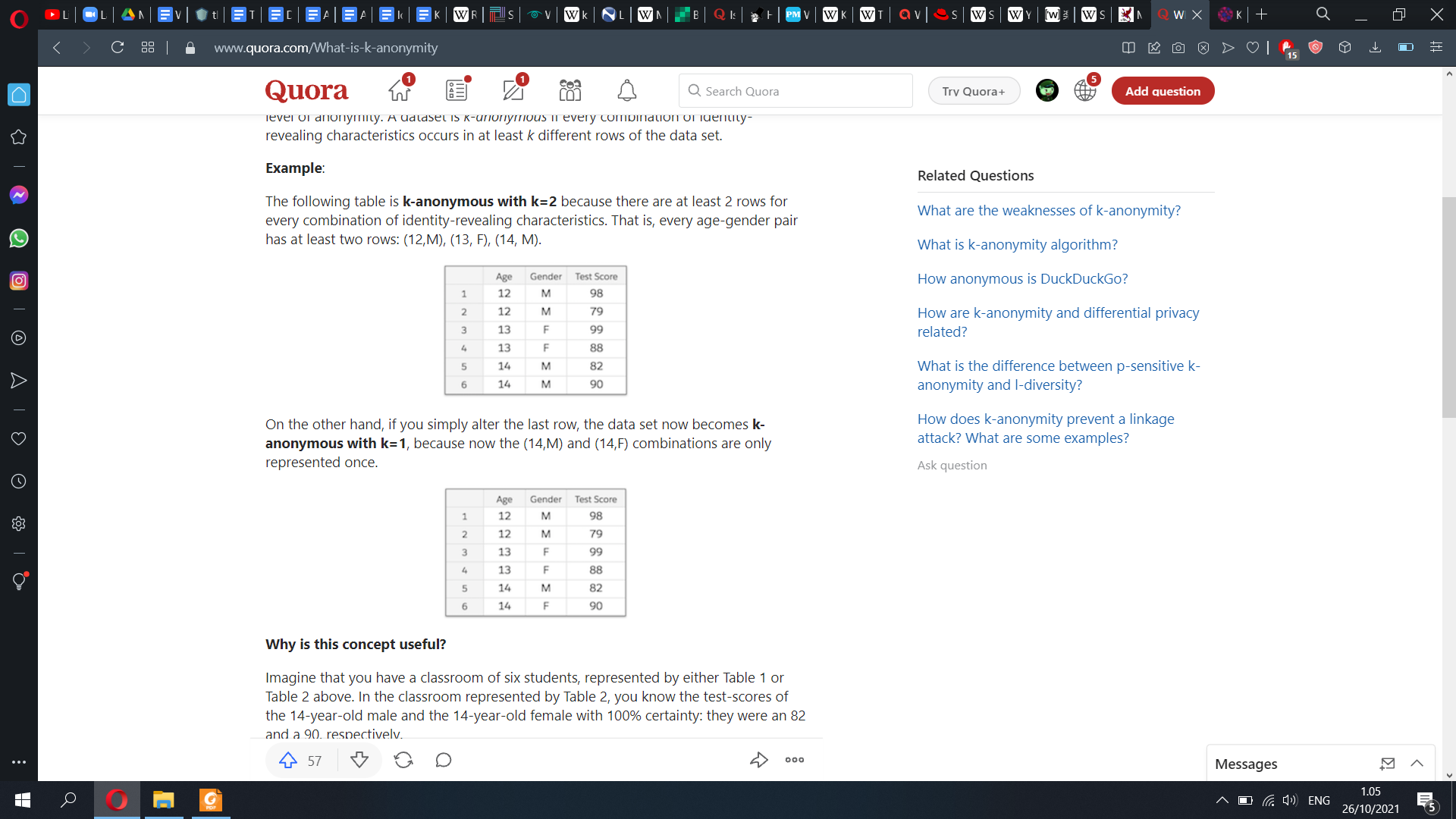
"acr" : "c2id.loa.hisec",

"iat" : 1311280970,

"exp" : 1311281970

}





a) k-anonymity is a property of a data set, usually used in order to describe the data set’s level of anonymity. A dataset is k-anonymous if every combination of identity-revealing characteristics occurs in at least k different rows of the data set.  
=> The researchers can receive the column Major, Minor or Gender

b) There’s only 5 columns, Student, Major, Minor, Gender, Age. Instead of thinking which columns to drop, think about adding which column will make the table less than 3-anonymity. Now drop all columns and start working up  
Adding just column name => 1-anonymity (violate). We must include this column  
Adding just column age => 1-anonymity (violate). We must include this column  
=> We must drop name and age column.

Now let’s keep 3 columns at the same time, Major, Minor, Gender => 1-anonymity (violate). Now we need to drop one column from these.

Drop major => 1-anonymity

Drop minor => 3-anonymity  
Drop gender => 1-anonymity  
=> minimum 3 number of columns dropped, which are name, age, minor

c)