Preparing for the XML Transcript Release

Obtaining an Account for the Texas Exchange Services
Setting Up PGP
Generating Keys for File Exchange
Changing the Transcript Export and Import Processes

Office of Information and Instructional Technology
April 2008
This page left blank to facilitate front/back printing.
Table of Contents

Introduction to PGP Installation................................................................. 1
  Introduction and Instructions on Obtaining a Test Account...................... 1
  Purpose...................................................................................................... 2
  Support ..................................................................................................... 2

Preparing Your Environment for the PGP Transcript Solution .................... 3

  Step 1: Installation and/or Configuration of PGP Software ..................... 3
    1. Install PGP ......................................................................................... 3
    2. Check Executable ............................................................................. 3
    3. Set PGPPATH Variable .................................................................... 3
    4. Define TZ Variable .......................................................................... 3

  Step 2: Generating an RSA Key Pair .................................................... 3
    1. Check PGPPATH Variable................................................................. 3
    2. Enter pgp –kg Command .................................................................. 3
    3. Select Key Algorithm ....................................................................... 4
    4. Select Key Size ............................................................................... 4
    5. Enter User ID and E-Mail ................................................................. 4
    6. Enter Expiration Time ..................................................................... 4
    7. Enter Passphrase ............................................................................ 4
    8. Enter Keystrokes ............................................................................ 4
    9. Enter Default Preference .................................................................. 4

  Step 3: Exporting Your Public Key in ASCII-armor Format ..................... 5
    1. Check PGPPATH Variable................................................................. 5
    2. Change Directories .......................................................................... 5
    3. Enter pgp –kxa Command ................................................................. 5
    4. Enter File Name ............................................................................... 5
    5. Receive Confirmation ...................................................................... 5

  Step 4: E-Mailing Your Extracted Public Key to the Texas Server .......... 5
    1. Ensure Registration ........................................................................ 5
    2. E-Mail Public Key .......................................................................... 5
    3. Receive Confirmation ...................................................................... 5

  Step 5: Importing the Texas Server Public Key into Your Keyring .......... 6
    1. Save Public Key ............................................................................. 6
    2. Check PGPPATH Variable ................................................................. 6
    3. Import UT Public Key ..................................................................... 6
    4. Sign UT Austin Key ......................................................................... 6

Testing Your Encryption and Decryption...................................................... 7

  Testing Encryption ................................................................................. 7
    1. Check PGPPATH Variable ................................................................. 7
    2. Copy Text File ............................................................................... 7
    3. Test Encryption ............................................................................... 7
    4. View Encrypted File ...................................................................... 7
Testing Decryption ................................................................. 8

1. Check PGPPATH Variable .................................................. 8
2. Test Decryption ................................................................. 8

Additional Resources .......................................................... 8

Changing SHRPESE and SHRPESI to Use PGP Encryption .......... 9

Introduction ................................................................................. 9

Changing the SHRPESE pgp.properties File .......................... 10

1. Change Directories .......................................................... 10
2. Make Backup ..................................................................... 10
3. Extract pgp.properties ......................................................... 10
4. Edit pgp.properties ............................................................ 10
5. Save Changes ................................................................. 10
6. Update shrpese.jar ........................................................... 10
7. Delete pgp.properties File .................................................. 10

Changing the shrpese.shl Script .............................................. 11

Change shrpese.shl Script ......................................................... 11
Set JDBC connection information ........................................... 11

Changing the SHRPESI pgp.properties File ............................ 11

1. Change Directories .......................................................... 11
2. Make Backup ..................................................................... 11
3. Extract pgp.properties ......................................................... 11
4. Edit pgp.properties ............................................................ 12
5. Save Changes ................................................................. 12
6. Update shrpesi.jar ........................................................... 12
7. Delete pgp.properties File .................................................. 12

Changing the shrpesi.shl Script .............................................. 12

Change shrpesi.shl Script PGPPATH ..................................... 12
Set JDBC connection information ........................................... 12

Stopping and Restarting Job Submission ................................. 13

Stop and Restart Job Submission ............................................. 13

Sharing the Texas Server Address for Entry on the SHACTRL/SOASBGA .... 13
Share Texas Server Address .................................................... 13
Preparing for the XML Transcript Release

Introduction to PGP Installation

The XML Transcript release has been designed to use the transcript brokerage service provided by the University of Texas at Austin. The service provided by the University of Texas Server is available free of charge to any institution that wishes to participate and can meet the participation criteria.

The XML Transcript release has been designed to use Pretty Good Privacy (PGP) to encrypt the transcript files after they are extracted from Banner to the file system and prior to sending the files to the Texas Server for processing. Both the file encryption and the ftp transfer will be performed by the new 7.3.x SHRPES process from Banner job submission. A second new 7.3.x SHRPESI process will decrypt the transcript files received from other institutions via the Texas server and launch the process to load the data into Banner.

In order to use the XML Transcript release, campuses must first obtain a test account from the University of Texas. An account Registration Form must be completed in order to obtain your account on the University of Texas Server. This Registration Form and other information for establishing a test account may be found at:

http://www.utexas.edu/student/giac/speede/ediserv.html

This test account will be used in the exchange of encrypted transcript files between the Texas Server and the institution’s server.

Important: The Registration Form includes a setting for your school code. This code must not be more than six alphanumeric characters. Using FICE code is recommended.

When requesting your test account and later your production account, you should give serious consideration to whether you wish to allow network traffic through to write files on your database server. You will need to create a Unix account on some server to allow the transcript files destined for your campus to be sent back to your campus from the Texas server. During testing at OIIT, an account was created on a non-production application server to receive the encrypted files from the Texas Server. We then copied the encrypted files to the database server for decryption and use by the SHRPESI import process.

When completing your request for a test account, the Registration Form allows you to select from two different protocol methods (FTP and SMTP). The XML Transcript release has been developed to work with FTP using PGP, or secure FTP. You may need to have direct communications via email with the University of Texas technical staff to configure your Texas server account to allow use of SFTP to send
files to the Texas server. **Important:** *Neither unsecure, plain FTP without the use of PGP nor SMTP transfer protocols will be supported by the XML Transcript Release.*

The XML Transcript release will contain job submission processes that will export data from transcript requests in Banner into XML format files in the job submission output directory. The export process will then encrypt the XML transcript files, using the public key obtained from the Texas IT staff, and send the encrypted files via FTP to the Texas server. This process uses configuration information that will be established in the database or in shell scripts associated with the export process.

Please refer to the release documentation for further information regarding the setup and configuration needed within Banner to use the XML Transcript release.

### Purpose

This document explains how to prepare for the XML Transcript release, and includes the following topics:

- Introduction to PGP installation and configuration
- Preparing your environment to use PGP for the transcript solution
- Testing your encryption and decryption
- Changing SHRPESE and SHRPESI to use PGP encryption

### Support

For additional OIIT resources and support, contact the OIIT HELPDESK:

- **Web**
  - [http://www.usg.edu/customer_services](http://www.usg.edu/customer_services)
  - (self-service ticket submission)

- **E-mail**
  - helpdesk@usg.edu

For urgent or production down situations, call the OIIT HELPDESK:

- **Local**
  - 706-583-2001

- **Toll free within Georgia**
  - 1-888-875-3697
## Preparing Your Environment for the PGP Transcript Solution

### Step 1: Installation and/or Configuration of PGP Software

1. **Install PGP**
   
   The PGP software required to use the XML Transcript must be installed on the server where job submission will be started. This PGP software is already installed and ready for use in the OIIT HP-UX 11.0 release. Campuses that are not using the OIIT HP-UX 11.0 release must install the PGP Version 6.5.8 package that supports generating RSA keys. You can obtain PGP freeware for different platforms from the following Web site:

   http://www.pgpi.org/products/pgp/versions/freeware/

2. **Check Executable**

   Ensure that the path to the pgp command executable is included in your operating system PATH environment variable.

3. **Set PGPPATH Variable**

   Set the PGPPATH environment variable. PGP uses the value of the environment variable PGPPATH to store files needed to support file encryption. This variable must point to a directory to which the job submission shell account has write privileges. The default location is $HOME/.pgp of the specified Unix account. In this case that would be the Unix account executing the job submission. **Important:** It is highly recommended that the default location *not* be used to store keys for the job submission account. Use the normal method associated with the Unix shell being used when setting this variable. Be sure to use an absolute path name rather than a relative path name when specifying PGPPATH. Using a relative path could cause problems with the way PGP sets preferences.

   For example: export PGPPATH=/home/banjobs/received/.pgp

4. **Define TZ Variable**

   Define the TZ environment variable. TZ specifies the time adjustment from Greenwich Mean Time (GMT). For Eastern time: SET TZ=EST5EDT. (This value may already be set in your environment.)

### Step 2: Generating an RSA Key Pair

1. **Check PGPPATH Variable**

   Transcripts will be sent from the server where job submission is started. Therefore, if you plan to use PGP, it is recommended that the job submission Unix user be used to generate and own your PGP keys. Ensure that your PGPPATH variable for the Banner job submission user is set to the directory where you wish to store your pgp key files.

2. **Enter pgp –kg Command**

   As the Banner job submission user, enter the command: pgp –kg
### Step 3: Select Key Algorithm
When prompted to select a key algorithm, choose Option 2 to select an RSA key algorithm.

### Step 4: Select Key Size
When prompted to select key size, select Option 2 for 2048 bit key size.

### Step 5: Enter User ID and E-Mail
When prompted for a user ID and e-mail address, enter a user ID and password that will be used to identify the key holder. **Important**: When entering your userid the Unix operating system will accept spaces and any special characters in the passphrase. However, testing has indicated that spaces and special characters (underscores may be used) should not be used in the passphrase as this causes problems when the export and import processes attempt to pass the passphrase as a variable to these processes.

### Step 6: Enter Expiration Time
When prompted to enter the key expiration time, enter the number, in days, that you would like this key to remain valid before it expires. If you enter 0, the key will never expire.

### Step 7: Enter Passphrase
When prompted to enter a passphrase, enter the passphrase you want to use to access your private key. For security purposes, do not use the same password that was used for your Texas Server account as your passphrase. **Important**: Use strong password guidelines when setting this passphrase and make sure you remember the passphrase. When creating your passphrase the Unix operating system will allow spaces and any special characters in the passphrase. But, testing has indicated that spaces and special characters (underscores may be used) should not be used in the passphrase as this causes problems when the export and import processes attempt to pass the passphrase as a variable to these processes. If you forget this passphrase there is no way to decrypt files that have been encrypted with your key. Keep this passphrase secure.

### Step 8: Enter Keystrokes
When prompted to enter random keystrokes, enter random key strokes until the process indicates that enough keystrokes have been entered.

### Step 9: Enter Default Preference
When prompted whether or not to make this your default signing key, enter Y for yes.

If you received no errors during the process, you have now successfully created a PGP key pair.

You may view your keys by executing the command: `pgp –kv`. 
### Step 3: Exporting Your Public Key in ASCII-armor Format

Once your key pairs have been generated, you must share your public key with those with whom you want to share encrypted files. For the XML Transcript release, this should be the Texas Server. Use the commands below to extract your public key into a file that can be shared with the Texas Server.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Check PGPPATH Variable</td>
<td>As the job submission user, ensure that your PGPPATH variable is set to the directory where the pgp files were created in “Step 2: Generating an RSA Key Pair” above.</td>
</tr>
<tr>
<td>2. Change Directories</td>
<td>Change the directory to the $PGPPATH location.</td>
</tr>
<tr>
<td>3. Enter pgp –kxa Command</td>
<td>Enter the following command: <code>pgp –kxa &quot;&lt;userid from Step 2&gt;&quot;</code>; For example, if your user ID was <code>gcsutrans</code>, you would enter the command as: <code>pgp –kxa “gcsutrans”</code>.</td>
</tr>
<tr>
<td>4. Enter File Name</td>
<td>When prompted to extract the above key(s) into which file, enter a file name to hold your public key. It is a good idea to name the key file something that would be applicable to the username. For example, “gcsutrans”</td>
</tr>
<tr>
<td>5. Receive Confirmation</td>
<td>When the process is successful, you should receive a message indicating that the key was extracted into the file name you entered followed by an “asc” extension.</td>
</tr>
</tbody>
</table>

### Step 4: E-Mailing Your Extracted Public Key to the Texas Server

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Ensure Registration</td>
<td>Make sure you have registered an account with the Texas Server. <a href="http://www.utexas.edu/student/giac/speede/ediserv.html">http://www.utexas.edu/student/giac/speede/ediserv.html</a></td>
</tr>
<tr>
<td>2. E-Mail Public Key</td>
<td>E-mail your public key to <a href="mailto:regist@ediserver.reg.utexas.edu">regist@ediserver.reg.utexas.edu</a> with an appropriate message indicating the username on the Texas Server to which the public key applies.</td>
</tr>
<tr>
<td>3. Receive Confirmation</td>
<td>Within a full business day, you should receive an acknowledgement from the Texas Server personnel indicating that your public key has been received and applied.</td>
</tr>
</tbody>
</table>
## Step 5: Importing the Texas Server Public Key into Your Keyring

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. <strong>Save Public Key</strong></td>
<td>You will receive an e-mail message from the Texas Server providing the Texas Server’s UT Austin public key (utkey.asc) as an attachment. Save the utkey.asc file attachment to the PGPPATH location where the job submission account key files were stored in “Step 2: Generating an RSA Key Pair.” If you use a Windows-based e-mail tool to save the file to your desktop, you will need to transfer the file from your desktop to your Unix server. Use an ASCII transfer method to transfer the files.</td>
</tr>
<tr>
<td>2. <strong>Check PGPPATH Variable</strong></td>
<td>Ensure that the PGPPATH variable is set to the directory where the pgp files were created earlier in “Step 2: Generating an RSA Key Pair.”</td>
</tr>
<tr>
<td>3. <strong>Import UT Public Key</strong></td>
<td>Import the Texas Server UT Austin public key into your keyring by entering the command: <code>pgp –ka utkey.asc</code></td>
</tr>
</tbody>
</table>
| 4. **Sign UT Austin Key** | Sign the Texas Server UT Austin key so it is a trusted key by entering the following command: `pgp –ks “UT Austin”`  
If this command returns a prompt asking whether you are prepared to solemnly certify that the public key actually belongs to the user specified by the specified user ID, answer Y for yes. |
# Testing Your Encryption and Decryption

You are now ready to encrypt and sign files.

## Testing Encryption

1. **Check PGPPATH Variable**
   
   As the job submission user, ensure that the PGPPATH variable is set to the directory where the pgp files were created earlier in “Step 2: Generating an RSA Key Pair.”

2. **Copy Text File**
   
   Copy a text file, such as a .lis or .log file from the job submission output directory into your PGPPATH directory.

3. **Test Encryption**
   
   Use the following command to test encryption of the text file you just copied:

   ```
   pgp -sea cleartext_filename PublicKeyUserName -u privateKeyUserName -z yourkeyPassphrase -o encryptedfilename
   ```

   Where…

   - `cleartext_filename` = the text file that you want to encrypt
   - `PublicKeyUserName` = the userid associated with your pgp key pair
   - `privateKeyUserName` = the userid associated with your private key generated earlier in “Step 2: Generating an RSA Key Pair”
   - `yourkeyPassphrase` = the passphrase associated with your private key generated earlier in “Step 2: Generating an RSA Key Pair”
   - `encryptedfilename` = the desired name of the encrypted version of the file to be created.

   For example:

   ```
   pgp -sea <textfile> -o <output filename> -u "gcsutrans"
   ```

   If prompted for userid, enter the userid you used when you generated your pgp keys in “Step 2: Generating an RSA Key Pair.”

   When prompted for passphrase, enter the passphrase for the userid you used when you generated your pgp keys in “Step 2: Generating an RSA Key Pair.”

4. **View Encrypted File**
   
   Once the file has been encrypted, you can use your preferred file viewer command, such as “more” to view the encrypted file to verify that the encrypted file is no longer clear text.

   Using the above example, the command **more transcript.asc** should display the encrypted version of the file.
**Testing Decryption**

1. **Check PGPPATH Variable**

   As the job submission user, ensure that the PGPPATH variable is set to the directory where the pgp files were created earlier in “Step 2: Generating an RSA Key Pair.”

2. **Test Decryption**

   Use the following command to test decryption of the text file you just encrypted:

   ```
   pgp encryptedfilename –o cleartext_filename –z yourkeyPassphrase
   ```

   Where…

   - `encryptedfilename` = the encrypted file
   - `cleartext_filename` = the desired output filename to use for the clear text version of the encrypted file
   - `yourkeyPassphrase` = the passphrase associated with your private key generated earlier in “Step 2: Generating an RSA Key Pair.”

   For example:

   ```
   pgp transcript.asc –o new_transcript.txt
   ```

   When prompted for passphrase, enter the passphrase for the userid you used when you generated your pgp keys in “Step 2: Generating an RSA Key Pair.”

**Additional Resources**

For more information on exchanging transcripts via the UT Austin EDI Server, visit:

[http://www.utexas.edu/student/giac/speede/ediserv.html](http://www.utexas.edu/student/giac/speede/ediserv.html)

For information specifically related to PGP and the UT Austin EDI Server, go to the PGP Beginner Guide document on the UT Austin Texas Server:

[http://www.utexas.edu/student/registrar/pgp/](http://www.utexas.edu/student/registrar/pgp/)
Changing SHRPESE and SHRPESI to Use PGP Encryption

Introduction

After the XML Transcript release has been applied to your environment and you have completed the PGP installation and configuration, you will need to perform some setup work prior to allowing your users to begin testing and using the new functionality.

To support encrypting your XML transcripts generated using the SHRPESE java program, you must first set up the PGP Software as outlined in the section “Preparing for the XML Transcript.”

You will also need to configure the SHRPESE process to do the encryption and use the proper public key users, private key user, and private key password (passphrase).

The PGP command line program is used within the SHRPESE java program. The Host Name defined on the SHACTRL/SOASBG form is used to find the corresponding PublicKeyUserName for the server to which the PGP encrypted file is being sent.

Setting up the SHRPESE and SHRPESI processes involves several steps:

- Changing the SHRPESE pgp.properties file
- Changing the shrpese.shl script
- Changing the SHRPESI pgp.properties file
- Changing the shrpesi.shl script
- Stopping and Restarting job submission
- Sharing the Texas Server address for entry on SHACTRL/SOASBG with your campus users

You will be changing several property entries in the shrpese.jar and the shrpesi.jar file as detailed below:

- `pgp.usePGP` — This indicates whether PGP should be used to encrypt the transcript files prior to transmission. This is the pgp.usePGP property in the pgp.properties file.
- `pgpUserName` — This is the userid associated with your private key. This is the pgp.privateKeyUserName property in the pgp.properties file.
- `pgpPassword` — This is the passphrase associated with your private key. This is the pgp.privateKeyPassword property in the pgp.properties file.
- `pgpFileName` — This is the encrypted version of the xml filename generated by the SHRPESE java program. It has the same file name as the xml file but the extension is changed from xml to pgp.
### Changing the SHRPESE pgp.properties File

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Change Directories</strong></td>
<td>As the Banner software owner user ID, change directories (cd) to the $BANNER_HOME/student/java directory for the database to which the Georgia XML Transcript release was applied. The shrpese.jar file was copied to this directory.</td>
</tr>
<tr>
<td><strong>2. Make Backup</strong></td>
<td>Make a backup copy of the shrpese.jar file.</td>
</tr>
<tr>
<td><strong>3. Extract pgp.properties</strong></td>
<td>Run the following command to extract pgp.properties from shrpese.jar: <code>jar xvf shrpese.jar pgp.properties</code></td>
</tr>
<tr>
<td><strong>4. Edit pgp.properties</strong></td>
<td>Use your preferred editor (such as vi) to edit pgp.properties. You will notice that an entry (UT Austin EDI server) has already been provided to indicate the host name to which the transcript is being sent: Changing the N in … pgp.usePGP to Y. Changing the xxxx in … pgp.privateKeyUserName=xxxxx …to the userid associated with your private key (which has the same user ID as the public key that you sent to the Texas server staff). This is the userID you used when you generated your pgp keys. In previous examples this was gcsutrans. Your entry should reflect your local user ID information. Changing the yyyy in … pgp.privateKeyPassword=yyyyyy …to the passphrase associated with the private key. In previous examples, this was gcsutrans. Your entry should reflect your local user ID information. If the default command of pgp is not working and you need to define an absolute path to the pgp command, Changing the pgp command … pgp.Command=pgp …to the full path of the pgp command.</td>
</tr>
<tr>
<td><strong>5. Save Changes</strong></td>
<td>Save your changes.</td>
</tr>
<tr>
<td><strong>6. Update shrpese.jar</strong></td>
<td>Run the following command to update shrpese.jar with the updated pgp.properties file: <code>jar uvf shrpese.jar pgp.properties</code></td>
</tr>
</tbody>
</table>
### Changing the shrpese.shl Script

**Change shrpese.shl Script**

The shrpese.shl script must be changed to properly reflect the appropriate PGPPATH where your pgp keys are stored. An example has been provided in the shrpese.shl script that was distributed with the XML Transcript release. You must locate and change the value for PGPPATH to the appropriate value for your environment.

**Set JDBC connection information**

Additionally, the shrpese.shl script must be changed to reflect the proper connection information to allow a JDBC thin client connection to your Banner database.

In the shrpese.shl script, locate the line:

```
    #  CONNECTION="jdbc:oracle:thin:@HOST:PORT:SID"
```

**Note:** There may be two such lines in your shell script, so be sure to select the one that refers to “thin” and not the line that refers to “oci.”

Change this line so that “HOST” reflects the IP name of your Banner database server, “PORT” reflects the Sql*Net listener port for the database on which you are working, and “SID” reflects the ORACLE_SID of the database on which you are working. Be sure to uncomment this line if it is commented out.

### Changing the SHRPESI pgp.properties File

**1. Change Directories**

As the Banner software owner user ID, change directories (cd) to the $BANNER_HOME/student/java directory for the database to which the Georgia XML Transcript release was applied. The shrpesi.jar file was copied to this directory.

**2. Make Backup**

Make a backup copy of the shrpesi.jar file.

**3. Extract pgp.properties**

Run the following command to extract pgp.properties from shrpesi.jar:

```
jar xvf shrpesi.jar pgp.properties
```
4. **Edit pgp.properties**

   Use your preferred editor (such as vi) to edit pgp.properties.
   Change the N in …
   pgp.usePGP to Y.
   Change the yyyy in…
   pgp.privateKeyPassword=yyyyyy
   …to the passphrase associated with the private key. In previous examples, this was gcsutrans. Your entry should reflect your local user ID information.

   If the default command of pgp is not working and you need to define an absolute path to the pgp command,
   Change the pgp command…
   pgp.Command=pgp
   …to the full path of the pgp command.

5. **Save Changes**

   Save your changes.

6. **Update shrpesi.jar**

   Run the following command to update shrpesi.jar with the updated pgp.properties file:
   jar uvf shrpesi.jar pgp.properties

7. **Delete pgp.properties File**

   Delete the pgp.properties file from the directory.

---

**Changing the shrpesi.shl Script**

**Change shrpesi.shl Script PGPPATH**

The shrpesi.shl script must be changed to properly reflect the appropriate PGPPATH where your private key files are stored. An example has been provided in the shrpesi.shl script that was distributed with the XML Transcript release. You must locate and change the value for PGPPATH to the appropriate value for your environment.
Set JDBC connection information

Additionally, the shrpesi.shl script must be changed to reflect the proper connection information to allow a JDBC thin client connection to your Banner database.

In the shrpesi.shl script, locate the line:

```
"# CONNECTION="jdbc:oracle:thin:@HOST:PORT:SID"
```

Change this line so that “HOST” reflects the IP name of your Banner database server, “PORT” reflects the Sql*Net listener port for the database on which you are working, and “SID” reflects the ORACLE_SID of the database on which you are working.

Stopping and Restarting Job Submission

Stop and re-start job submission.

After changing the jar file or shell scripts associated with this process, you must use your normal procedures to stop and then restart the job submission processes for this database before your users can successfully run the SHRPSE or SHRPESI process.

Sharing the Texas Server Address for Entry on the SHACTRL/SOASBG

Share the Texas Server address for entry on SHACTRL/SOASBG with your campus users.

Once you have completed the technical setup steps for using PGP and have changed the shrpese.jar and shrpese.shl files, you will need to provide the username and password provided to you by the Texas Server staff to the personnel on your campus who have responsibility for receiving transcripts from other institutions. These users will need this information to complete certain setup items on the SHACTRL form in Banner before they can begin the transcript exchange process.