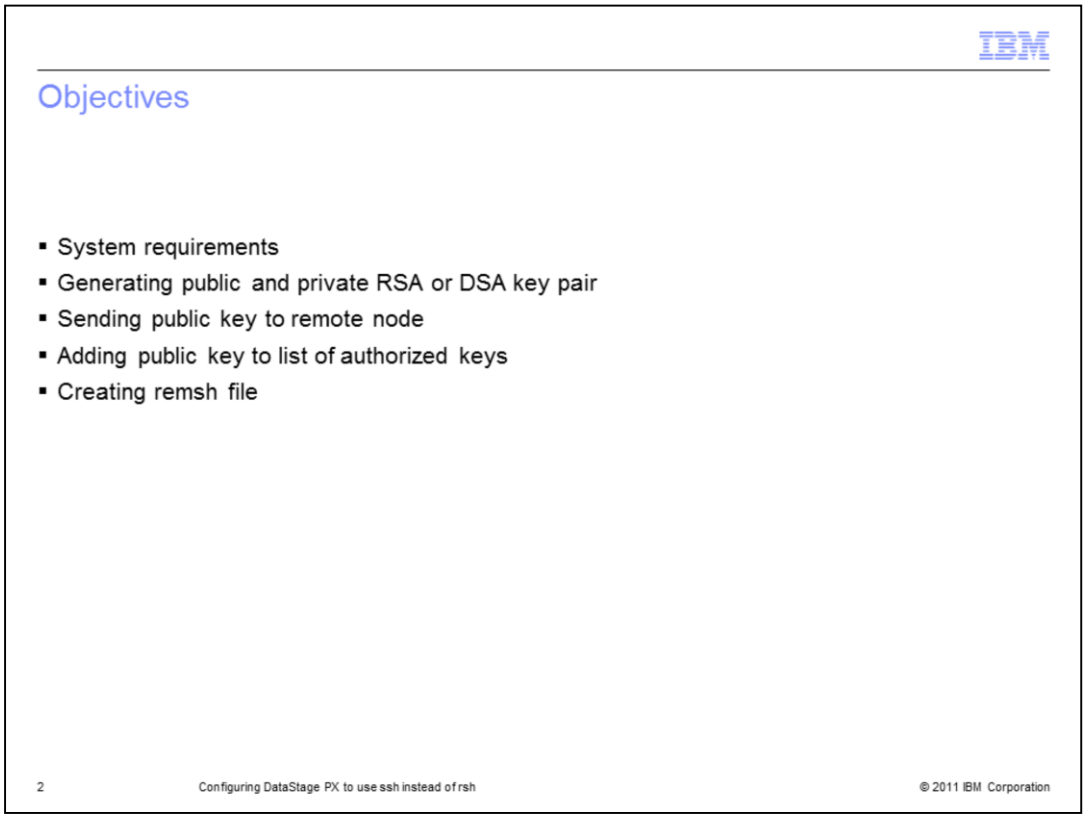
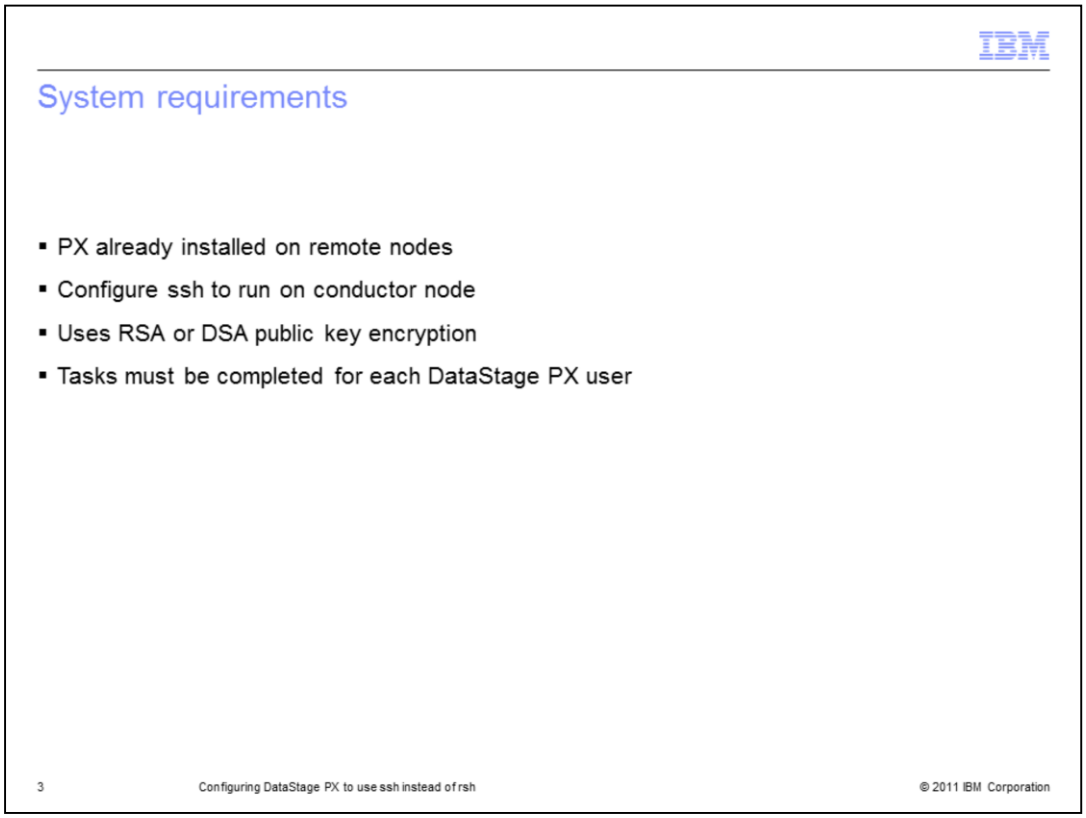
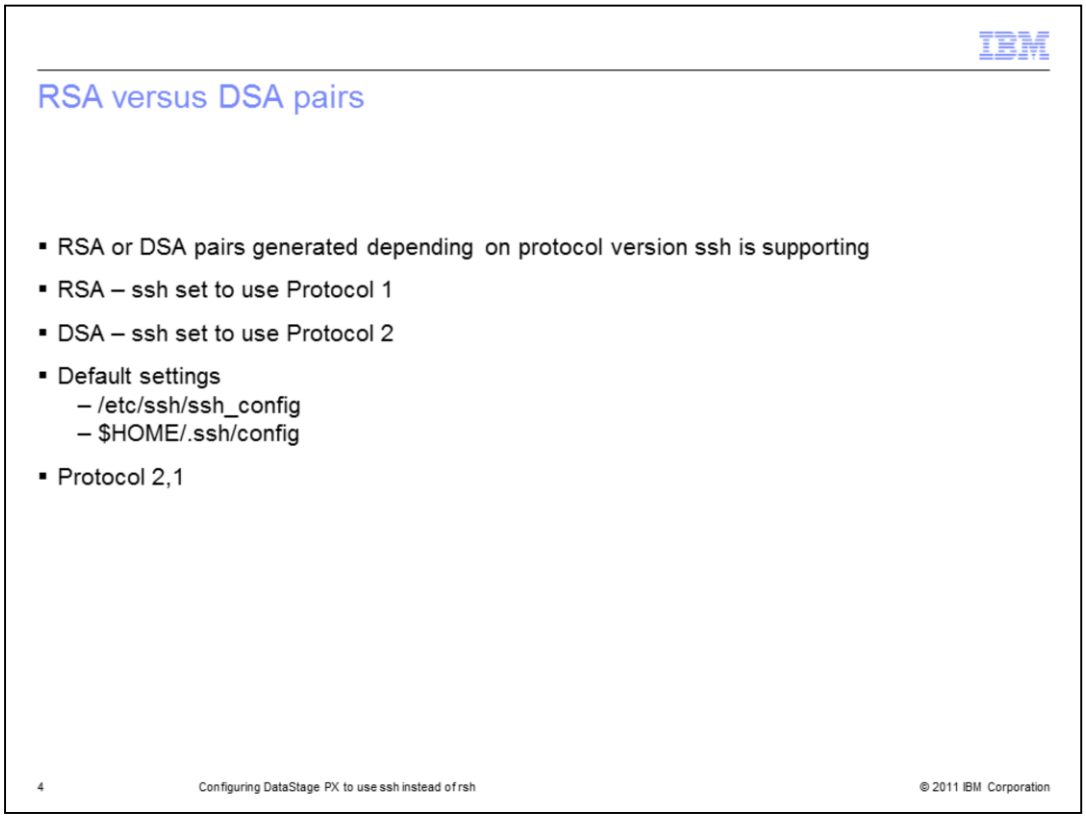

This presentation describes the steps needed to configure DataStage® PX to use ssh instead of rsh. This presentation assumes that ssh servers have been installed on all the machines where PX is to run.




The objective of this presentation is to describe how to generate the required public and private RSA or DSA key pair and how to send the generated keys to the remote node. You will also learn how to add the generated keys to the list of authorized keys, and how to create the remsh file.

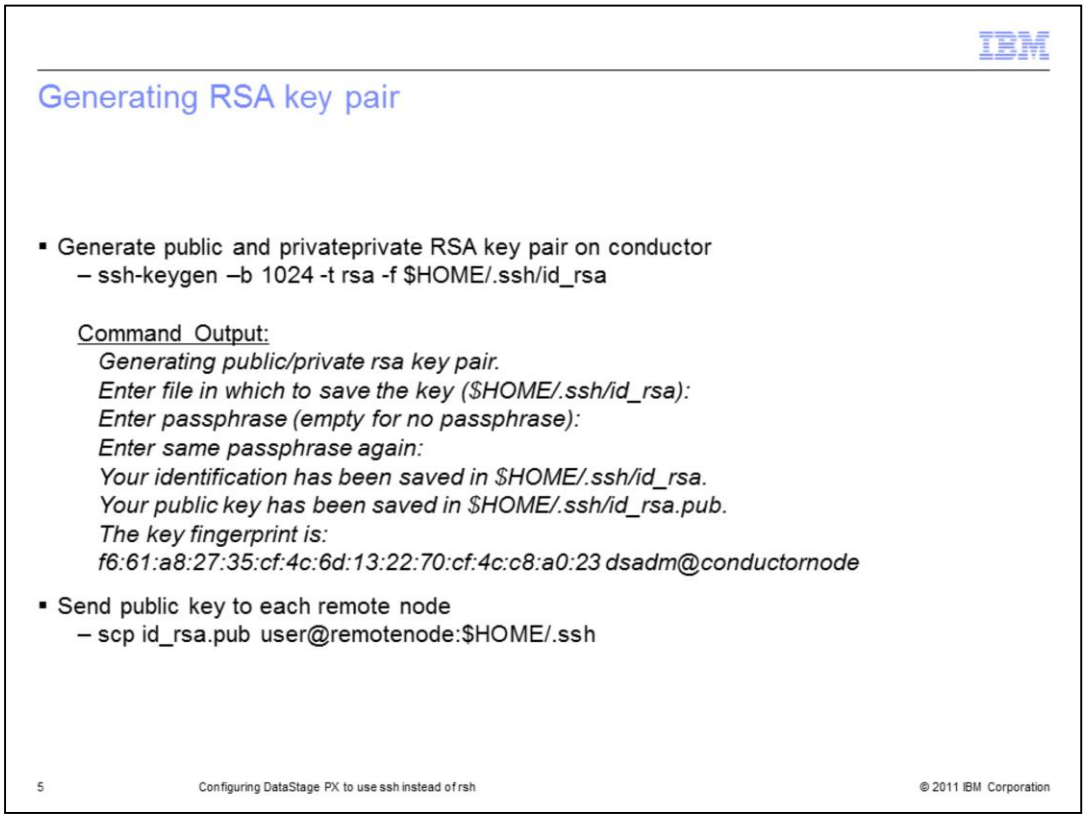



This presentation assumes that you have already successfully installed the PX engine on all of your remote nodes. In order for DataStage PX to use ssh, ssh must be configured so that a command can be launched from the conductor node to all other nodes without a password; authenticating only by way of public key encryption. When you configure ssh to work with the parallel engine, the engine connects from the primary computer to all of the other computers, using RSA or DSA public key encryption for authentication. This task must be completed for each user that runs parallel jobs. In the next steps, the primary computer, or the Conductor Node, is the computer that contains the IBM Information Server engine. The secondary computers, or Remote Nodes, are the other computers that contain parallel engines.




The next step is to generate an RSA or DSA key pair. Which one you will generate depends on the protocol version ssh is using on your nodes. If you are using protocol 1 then you will want to generate an RSA key pair. If you are using protocol 2, then generate a DSA key pair. Your system will have the default settings in /etc/ssh/ssh_config. The user can also have this set in $HOME/.ssh/config. Look in these files for the line with “Protocol” to see what the default setting is. If the default is "2,1‘, for example, ssh will try version 2 and falls back to version 1 if version 2 is not available.

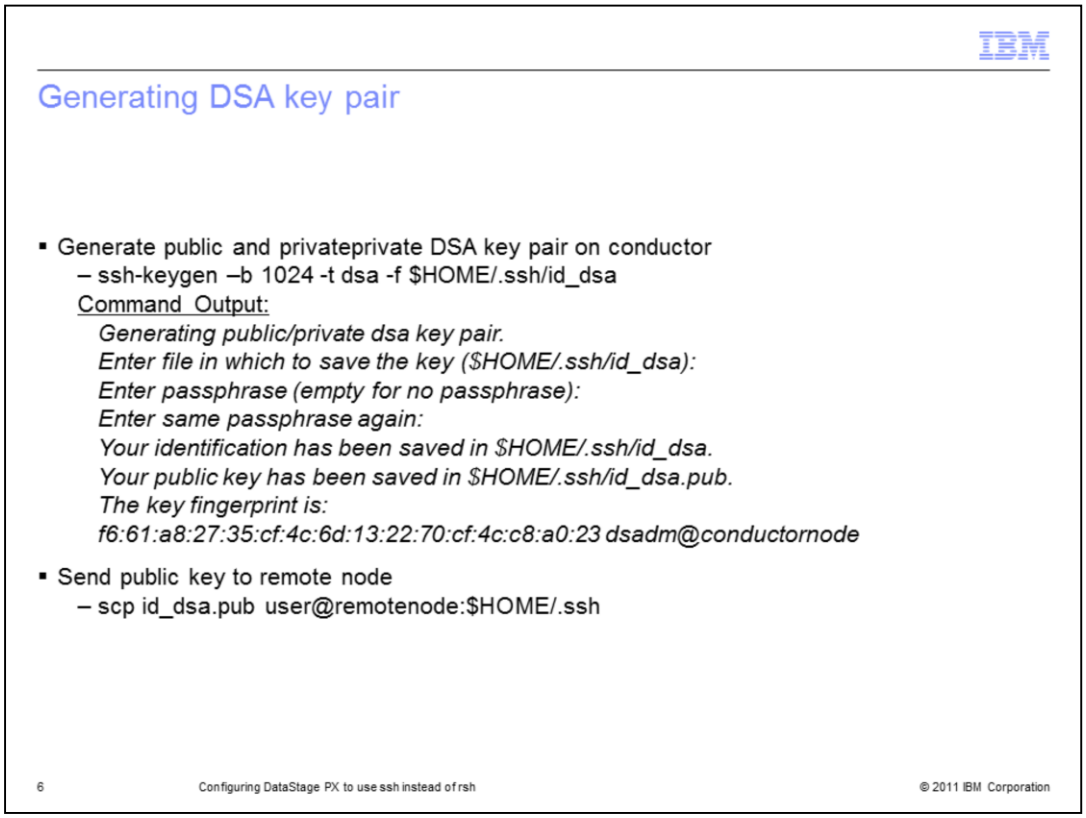
If you are unsure which protocol you are using, you can generate both the RSA and DSA keys for each remote machine.




Login to the conductor node as the user that you will be running DataStage PX jobs as. Generate the public and private RSA key pair on the conductor node using the ssh-keygen command where $HOME is your user's home directory: ssh-keygen –b 1024 -t rsa -f $HOME/.ssh/id_rsa.

Press enter twice to set a null pass-phrase. The identification keys will have been saved in $HOME/.ssh/id_rsa. Next, ensure that you are in $HOME/.ssh. Send the public key to each of the remote nodes using the scp command, scp id_rsa.pub user@remotenode:$HOME/.ssh

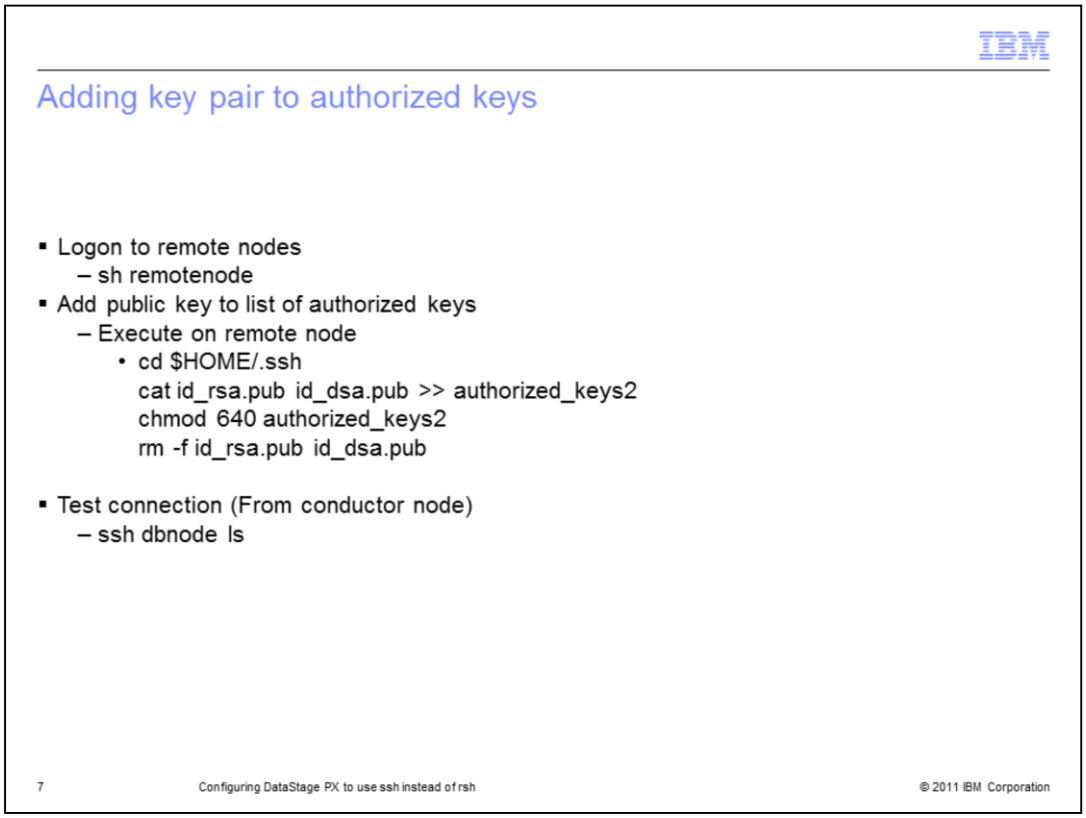
where remotenode is the name of your remote PX node. You must do this step for every remote node that you are configuring.




Login to the conductor node as the user that you will be running DataStage PX jobs as. Generate the public and private DSA key pair on the conductor node using the ssh-keygen command where $HOME is your user's home directory:

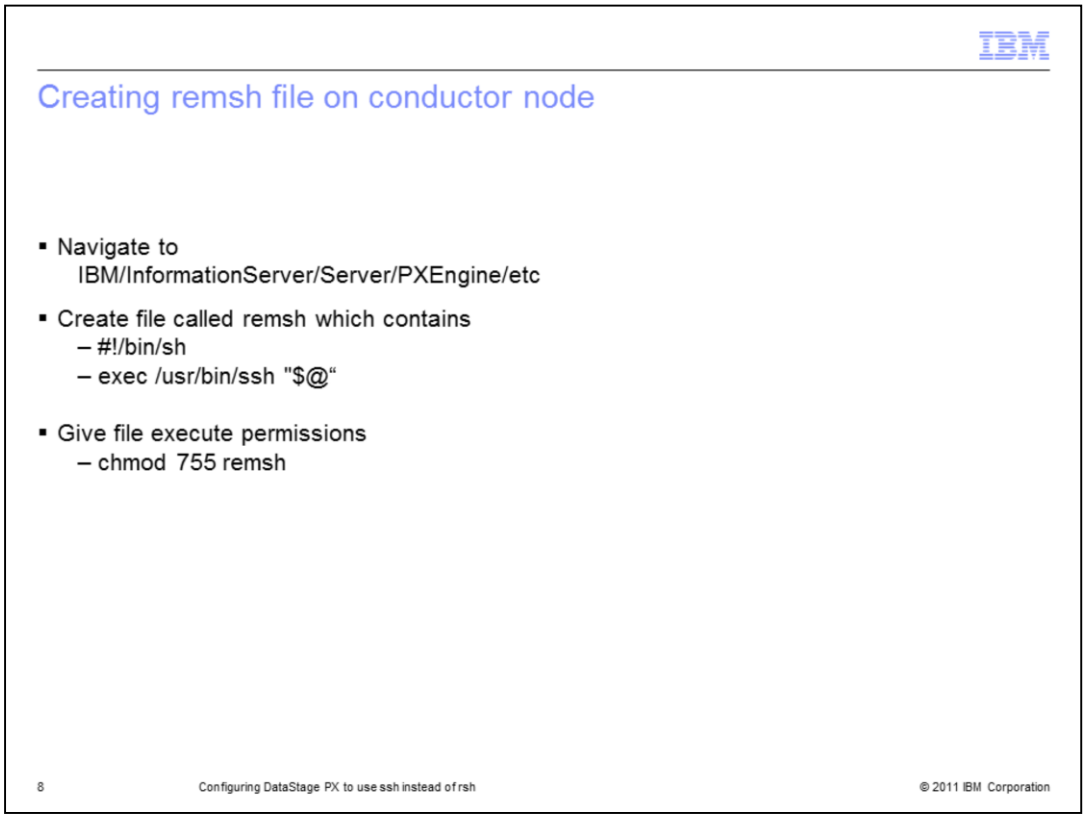
ssh-keygen –b 1024 -t dsa -f $HOME/.ssh/id_dsa.

Press enter twice to set a null pass-phrase. The identification keys will have been saved in $HOME/.ssh/id_dsa. Next, ensure that you are in $HOME/.ssh. Send the public key to the remote node using the scp command c where remotenode is the name of your remote PX node. You must do this step for every remote node that you are configuring.




Next, log in to the remote node and add the public key to the list of authorized keys. Change directories into the .ssh directory under the user's home directory. In there you should find the key pair you created in the last step. You want to append the key pair to the authorized_keys file. If you created both the RSA and DSA key pairs, you want to append both to the file. In the example displayed on this slide, you used both the RSA and DSA key pairs. Be aware the file containing the authorized keys file can either be named authorized_keys or authorized_keys2, depending on the version of ssh you have installed on your systems.

Test the connection by trying to ssh from the conductor node to the remote node. If successful, you should be able to log in to the remote shell without providing a password. An example command to use as a test is ssh dbnode ls where dbnode is the name of the remote node you are testing.




Next, on the conductor node, navigate to the IBM/InformationServer/Server/PXEngine/etc directory. Create a file called remsh which contains:

#!/bin/sh

exec /usr/bin/ssh "$@“

An example remsh file which uses rsh can be found at PXEngine/etc/remsh.example. Give the file execute permissions by running:  chmod 755 remsh.

# Trademarks, disclaimer, and copyright information