INDIANA UNIVERSITY (HTTP://WWW.IU.EDU)

Set up SSH public-key authentication to connect to a remote system

On this page:

Before you begin

Using SSH (aelc) public-key authentication (alqk) to connect to a remote system is a robust, more secure alternative to logging in with an account password or passphrase (acpu). SSH public-key authentication relies on asymmetric cryptographic algorithms that generate a pair of separate keys (i.e., a key pair), one "private" and the other "public". You keep the private key a secret and store it on the computer you use to connect to the remote system. Conceivably, you can share the public key with anyone without compromising the private key; you store it on the remote system in a `.ssh/authorized_keys` directory.

To use SSH public-key authentication:

- The remote system must have a version of SSH installed. The information in this document assumes the remote system uses OpenSSH, which is generally the case for UITS (ahaw) central systems at Indiana University. If the remote system is using a different version of SSH (e.g., Tectia SSH (https://www.ssh.com/products/tectia-ssh)), the process outlined below may not be correct.
- The computer you use to connect to the remote server must have a version of SSH installed. This document includes instructions for generating a key pair with command-line SSH on a Linux or OS X computer, and with PuTTY on a Windows computer.
- You need to be able to transfer your public key to the remote system. Therefore, you must either be able to log into the remote system with an established account username and password/passphrase, or have an administrator on the remote system add the public key to the `~/.ssh/authorized_keys` file in your account.

Back to top

Set up public-key authentication using SSH on a Linux or OS X computer

To set up public-key authentication using SSH on a Linux or OS X computer:

1. Log into the computer you'll use to access the remote host, and then use command-line SSH to generate a key pair using either the DSA or RSA algorithm:

   - To generate DSA keys, on the command line, enter:

     ```
     ssh-keygen -t dsa
     ```

   - To generate RSA keys, on the command line, enter:

     ```
     ssh-keygen -t rsa
     ```

2. You will be prompted to supply a filename (for saving the key pair) and a password (for protecting your private key):

   - **Filename:** To accept the default filename (and location) for your key pair, press `Enter` or `Return` without entering a filename.

     Alternatively, you can enter a filename (e.g., `my_ssh_key`) at the prompt, and then press `Enter` or `Return` . However, many remote hosts (including IU's research computing systems) are configured to accept private keys with the default filename and path (`~/.ssh/id_rsa` for RSA keys; `~/.ssh/id_dsa` for DSA keys) by default. Consequently, to authenticate with a private key that has a different filename, or one that is not stored in the default location, you must explicitly invoke it either on the SSH command line or in an SSH client configuration file (`~/.ssh/config`); see below for instructions.

   - **Password:** Enter a password that contains at least five characters, and then press `Enter` or `Return` . If you press `Enter` or `Return` without entering a password, your private key will be generated without password-protection.

     > **Note:**
     >
     > UITS strongly recommends password-protecting your private key. If you don't password-protect your private key, anyone with access to your computer conceivably can SSH (without being prompted for a password) to your account on any remote system that has the corresponding public key.

   Your private key will be generated using the default filename (e.g., `id_rsa`) or the filename you specified (e.g., `my_ssh_key`), and stored on your computer in a `.ssh` directory off your home directory (e.g., `~/.ssh/id_rsa` or `~/.ssh/my_ssh_key`).

   The corresponding public key will be generated using the same filename (but with a `.pub` extension added) and stored in the same location (e.g., `~/.ssh/id_rsa.pub` or `~/.ssh/my_ssh_key.pub`).

3. Use SFTP (akqg) or SCP (agye) to copy the public key file (e.g., `~/.ssh/id_rsa.pub`) to your account on the remote system (e.g., `darvader@deathstar.empire.gov`); for example, using command-line SCP:

   ```
   scp ~/.ssh/id_rsa.pub darvader@deathstar.empire.gov:
   ```

Or, to use IU's Karst cluster as an example (replace `username` with your Network ID (beml) username):

```
scp ~/.ssh/id_rsa.pub username@karst.uits.iu.edu:
```

You'll be prompted for your account password (or, if you're copying to an IU system, your Network ID passphrase). Your public key will be copied to your home directory (and saved with the same filename) on the remote system.

4. Log into the remote system using your account username and password. For an IU system, log in with your Network ID username and passphrase.

> **Note:**
>
> If the remote system is not configured to support password-based authentication, you will need to ask system administrators to add your public key to the `~/.ssh/authorized_keys` file in your account (if your account doesn't have `~/.ssh/authorized_keys` file, system administrators can create one for you). Once your public key is added to your `~/.ssh/authorized_keys` file on the remote system, the setup process is complete, and you should now be able to SSH to your account from the computer that has your private key.

5. If your account on the remote system doesn't already contain a `~/.ssh/authorized_keys` file, create one; on the command line, enter the following commands:

```
mkdir -p ~/.ssh touch ~/.ssh/authorized_keys
```

> **Note:**
>
> If your account on the remote system already has a `~/.ssh/authorized_keys` file, executing these commands will not damage the existing directory or file.

6. On the remote system, add the contents of your public key file (e.g., `~/id_rsa.pub`) to a new line in your `~/.ssh/authorized_keys` file; on the command line, enter:

```
cat ~/id_rsa.pub >> ~/.ssh/authorized_keys
```

You may want to check the contents of `~/.ssh/authorized_keys` to make sure your public key was added properly; on the command line, enter:

```
more ~/.ssh/authorized_keys
```

7. You may now safely delete the public key file (e.g., `~/id_rsa.pub`) from your account on the remote system; on the command line, enter:

```
rm ~/id_rsa.pub
```

Alternatively, if you prefer to keep a copy of your public key on the remote system, move it to your `.ssh` directory; on the command line, enter:

```
mv ~/id_rsa.pub ~/.ssh/
```

8. Optionally, repeat steps 3-7 to add your public key to other remote systems that you want to access from the computer that has your private key using SSH public-key authentication.

9. You now should be able to SSH to your account on the remote system (e.g., `username@host2.somewhere.edu`) from the computer (e.g., `host1`) that has your private key (e.g., `~/.ssh/id_dsa`):

   - If your private key is password-protected, the remote system will prompt you for the password or passphrase (your private key password/passphrase is not transmitted to the remote system):

     ```
     [username@host1 ~]$ ssh username@host2.somewhere.edu Enter passphrase for key
     '/username/Host1/.ssh/id_dsa': Last login: Mon Oct 20 09:23:17 2014 from
     host1.somewhere_else.edu
     ```

   - If your private key is not password-protected, the remote system will place you on the command line in your home directory without prompting you for a password or passphrase:

     ```
     [username@host1 ~]$ ssh username@host2.somewhere.edu Last login: Mon Oct 20 09:23:17
     2014 from host1.somewhere_else.edu
     ```

   If the private key you're using does not have the default name, or is not stored in the default path (i.e., not `~/.ssh/id_rsa` or `~/.ssh/id_dsa`), you must explicitly invoke it in one of two ways:

   - **On the SSH command line:** Add the `-i` flag and the path to your private key.

     For example, to invoke the private key `host2_key`, stored in the `~/.ssh/old_keys` directory, when connecting to your account on a remote host (e.g., `username@host2.somewhere.edu`), enter:

     ```
     ssh -i ~/.ssh/old_keys/host2_key username@host2.somewhere.edu
     ```

   - **In an SSH client configuration file:** SSH gets configuration data from the following sources (in this order):

     1. From command-line options
     2. From the user's client configuration file (`~/.ssh/config`), if it exists
     3. From the system-wide client configuration file (`/etc/ssh/ssh_config`)

     The SSH client configuration file is a text file containing keywords and arguments. To specify which private key should be used for connections to a particular remote host, use a text editor to create a `~/.ssh/config` that includes the `Host` and `IdentityFile` keywords.

For example, for connections to `host2.somewhere.edu`, to make SSH automatically invoke the private key `host2_key`, stored in the `~/.ssh/old_keys` directory, create a `~/.ssh/config` file with these lines included:

```
Host host2.somewhere.edu IdentityFile ~/.ssh/old_keys/host2_key
```

Once you save the file, SSH will use the specified private key for future connections to that host.

You can add multiple `Host` and `IdentityFile` directives to specify a different private key for each host listed; for example:

```
Host host2.somewhere.edu IdentityFile ~/.ssh/old_keys/host2_key Host
host4.somewhere.edu IdentityFile ~/.ssh/old_keys/host4_key Host host6.somewhere.edu
IdentityFile ~/.ssh/old_keys/host6_key
```

Alternatively, you can use a single asterisk (`*`) to provide global defaults for all hosts (i.e., specify one private key for several hosts); for example:

```
Host *.somewhere.edu IdentityFile ~/.ssh/old_keys/all_hosts_key
```

For more about the SSH client configuration file, see the manual page (afjm) for the OpenSSH SSH client configuration file on the web (http://man7.org/linux/man-pages/man5/ssh_config.5.html) or from the command line (`man ssh_config`).

Back to top

Set up public-key authentication using PuTTY on a Windows computer

The PuTTY command-line SSH client, the PuTTYgen key generation utility, the Pageant SSH authentication agent, and the PuTTY SCP and SFTP utilities are packaged together in a Windows installer available under The MIT License (https://opensource.org/licenses/MIT) for free download (http://the.earth.li/~sgtatham/putty/0.63/x86/putty-0.63-installer.exe) from the PuTTY development team.

To set up public-key authentication using PuTTY on a Windows computer:

1. Log into your computer and open the PuTTYgen key generation utility.
2. Under "Parameters", select either `SSH-2 RSA` or `SSH-2 DSA`; next to "Number of bits in a generated key", leave the default value (`1024`).
3. Under "Actions", click `Generate`, and then, when prompted use your mouse (or trackpad) to move your cursor around the blank area under "Key" (this generates randomness the utility uses to create your key pair).

   When the utility has generated your key pair, it will display the public key in the area under "Key".

4. In the "Key passphrase" and "Confirm passphrase" text boxes, enter a passphrase to passphrase-protect your private key.

   > **Note:**
   >
   > UITS strongly recommends passphrase-protecting your private key. If you don't passphrase-protect your private key, anyone with access to your computer will be able to SSH (without being prompted for a passphrase) to your account on any remote system that has the corresponding public key.

5. Save your public key:

   a. Under "Actions", next to "Save the generated key", click `Save public key`.

   b. Give the file a name (e.g., `putty_key`), select a location on your computer to store it, and then click `Save`.

6. Save your private key:

   a. Under "Actions", next to "Save the generated key", click `Save private key`.

   > **Note:**
   >
   > If you didn't passphrase-protect your private key, the utility will ask whether you're sure you want to save it without a passphrase. Click `Yes` to proceed or `No` to go back and create a passphrase for your private key.

   b. Keep "Save as type" set to `PuTTY Private Key Files (*.ppk)`, give the file a name (e.g., `putty_private_key`), select a location on your computer to store it, and then click `Save`.

7. Log into the remote system using your account username and password. (On IU systems, use your Network ID username and passphrase.)

   If the remote system does not support password-based authentication, you will need to ask system administrators to add your public key to the `~/.ssh/authorized_keys` file in your account (if your account doesn't have `~/.ssh/authorized_keys` file, system administrators can create one for you). Once your public key is added to your account's `~/.ssh/authorized_keys` file on the remote system...

8. If your account on the remote system doesn't already contain a `~/.ssh/authorized_keys` file, create one; on the command line, enter the following commands:

   ```
   mkdir -p ~/.ssh touch ~/.ssh/authorized_keys
   ```

   If your account on the remote system already has `~/.ssh/authorized_keys`, executing these commands will not damage the existing directory or file.

9. On your computer, in the PuTTYgen utility, copy the contents of the public key (displayed in the area under "Key") onto your Clipboard. Then, on the remote system, use your favorite text editor to paste it onto a new line in your `~/.ssh/authorized_keys` file, and then save and close the file.

10. On your computer, open the Pageant SSH authentication agent. This utility runs in the background, so when it opens, you should see its icon displayed in the Windows notification area.

11. In the Windows notification area, right-click on the `Pageant` icon, select `Add Key`, navigate to the location where you saved your private key (e.g., `putty_private_key.ppk`), select the file, and then click `Open`.

12. If your private key is passphrase-protected, Pageant will prompt you to enter the passphrase; enter the passphrase for your private key, and then click `OK`.

    If your private key is not passphrase-protected, Pageant will add your private key without prompting you for a passphrase.

    Either way, Pageant stores the unencrypted private key in memory for use by PuTTY when you initiate an SSH session to the remote system that has your public key.

13. On your computer, open the PuTTY SSH client:

    a. On the `Session` screen:

    - Under "Host Name (or IP address)", enter your username coupled with the host name of the remote server that has your public key; for example:

      ```
      dsidious@deathstar.empire.gov
      ```

      Or, to use an account on IU's Karst research cluster as an example (replace `username` with your Network ID username):

      ```
      username@karst.uits.iu.edu
      ```

    - Under "Connection type", make sure `SSH` is selected.

    b. In the "Category" list on the left, navigate to the `Auth` screen (`Connection > SSH > Auth`). On the `Auth` screen, under "Authentication methods", select `Attempt authentication using Pageant`.

    c. Return to the `Session` screen, and under "Saved Sessions", enter a name (e.g., `Deathstar`), and then click `Save`.

    d. Click `Open` to connect to your account on the remote system. With Pageant running in the background, PuTTY will retrieve the unencrypted private key automatically from Pageant and use it to authenticate. Because Pageant has your private key's passphrase saved (if applicable), the remote system will place you on the command line in your account without prompting you for the passphrase.

Note:

Technically, at this point, the setup is complete. In the future, whenever you log into your Windows desktop, you can run Pageant, add the private key, and then use PuTTY to SSH to any remote resource that has your public key. Alternatively, you can create a shortcut in your Windows `Startup` folder to launch Pageant and load your private key automatically whenever you log into your desktop. For instructions, finish the rest of the following steps.

14. Open your `Startup` folder:
    - In Windows 8 (bclp), press `Win-r`, and in the "Open" field, type `shell:startup`, and then press `Enter`.
    - In Windows 7 (ayrx), from the `Start` menu, click `All Programs`, scroll to find `Startup`, right-click on it, and then select `Open`.

    Alternatively, in either version, navigate to your `Startup` folder (replace `user_profile` with the name of your Windows user profile):

    ```
    C:\Users\user_profile\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
    ```

15. Right-click inside the `Startup` folder, and then select `New` and `Shortcut`.
16. In the "Type the location of the item" text box, enter the path to the Pageant executable (`pageant.exe`) followed by the path to your private key file (e.g., `putty_private_key.ppk`); enclose both paths in double quotes; for example:

    ```
    "C:\Program Files (x86)\PuTTY\pageant.exe" "C:\Users\user_profile\ssh_key\putty_private.ppk"
    ```

17. Click `Next`, and then, in the "Type a name for this shortcut" text box, enter a name for the shortcut (e.g., `PAGEANT`).
18. Click `Finish`.

The next time you log into your Windows desktop, Pageant will start automatically, load your private key, and (if applicable) prompt you for the passphrase.

Back to top

Related documents
About ssh-agent and ssh-add in Unix (aeww)

This is document aews *in the Knowledge Base.*
Last modified on 2018-01-22 14:50:58.