# Generate and use SSH keys

### Why keys?

SSH keys serve as a means of identifying yourself to an SSH server using public-key cryptography. One immediate advantage this method has over traditional password authentication is that you can be authenticated by the server without ever having to send your password over the network. In addition to this additional security, SSH key authentication can be more convenient than the more traditional password authentication. When used with a program known as an SSH agent, SSH keys can allow you to connect to a server, or multiple servers, without having to remember or enter your password for each system.



### A bit of theory

Wikipedia has a good article on this concept, but I will briefly cover the main points here. In public-key authentication, a pair of unique keys are generated, one of which is used to encrypt data, and the other is used to decrypt. As implemented, these are respectively known as public and private keys.

#### blocked URL

When you run ssh-keygen, this key pair is generated as two files called id\_rsa and id\_rsa.pub. id\_rsa.pub contains your public key, and the contents of this file are added to a list of authorized keys (usually a file called authorized\_keys) on the SSH server to grant access to the user with the matching private key. It is important to keep your private key secret, as anyone who possesses it can potentially log in as you without a password (we'll get to protecting your private key with a passphrase in a bit).

### Generating and using keys on Linux and OS X

#### (and any other Unix-like OS with OpenSSH)

Assuming that you are logged in as you, open up a terminal and run ssh-keygen -t rsa -b 2048

#### You'll see something like this:

```
$ ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
```

You can typically leave this as is. It will create the keys in your ~/.ssh directory by default.

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

You will definitely want to use a strong passphrase for your key. This is used to unlock the private key so that it can be used, and is the last line of defense if your private key is stolen.

```
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
e8:f5:96:98:d3:b3:09:c3:54:69:aa:aa:37:41:cc:80 user@somehost.example.com
```

The key fingerprint is a hash of your key that uniquely identifies it. You can use this to, for example, compare keys added in a github account to your keys without seeing the key itself. You can calculate the fingerprint for a key with ssh-keygen -lf <keyfile>, where <keyfile> is a public or private key.

And there you go, you have a shiny new key. Get your public key (the contents of id\_rsa.pub) added to some hosts, and you're in business.

# Agent Forwarding

This section was removed due to security concerns.

# Generating and using keys on Windows with PuTTY and Pageant

Windows is not a native speaker of SSH, but luckily there is an great third-party implementation called PuTTY. Go ahead and fetch the Windows install package here and run the installer. I'll wait.

Now that the PuTTY suite is installed, open up PuTTYgen and click Generate.

😴 PuTTY	Key Generator			? 🔀
<u>F</u> ile <u>K</u> ey	Con <u>v</u> ersions	<u>H</u> elp		
Key No key				
Actions				
Genera	te a public/private	key pair	$\rightarrow$	Generate
Load ar	n existing private ke	ey file		Load
Save th	e generated key	(	Save p <u>u</u> blic key	Save private key
Parame Type of © SSH	ters keytogenerate: I- <u>1</u> (RSA)	© SSH-2 <u>R</u> SA	SSF	1-2 <u>D</u> SA
Number	of <u>b</u> its in a genera	ted key:		1024

Windows doesn't have a truly random number generator like /dev/random, so you'll need to provide your own randomness.

😴 PuTTY Key Generator	? 🔀
<u>File K</u> ey Con <u>v</u> ersions <u>H</u> elp	
Key Please generate some randomness by moving the mouse over the bl	ank area.
Actions	
Generate a public/private key pair	Generate
Load an existing private key file	Load
Save the generated key Save public key	Save private key
Parameters	
Type of key to generate:           SSH-1 (RSA)         Image: SSH-2 RSA         Image: SSH-2 RSA	H-2 <u>D</u> SA
Number of <u>b</u> its in a generated key:	1024

After you've shaken your mouse frantically for a minute or so, you'll have your new key. Make sure to set a strong passphrase on it.

😴 PuTTY Key Genera	tor		? 💌	
<u>File Key Conversi</u>	ons <u>H</u> elp			
Key Public key for pasting	g into OpenSSH authorized	d_keys file:		
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAIEA1PunpkSUsNUf74f90+fcb/PolMIHLBJ9MX8Mm NHmjb4Npi6QGt3/P/FoRDw3qkvRbv6my6P8QFg91hUiYbazwCxuq16lFgB5QCJxPTtf mswl1EN634A3uUNf8iJxiNCcHFO5cVM56hDgmRu0xQriJFxKtQaYN2zWEea1kJkjy18 = rsa-key-20120308				
Key fingerprint:	erprint: ssh-rsa 1024 51:6b:e2:a1:8e:06:44:20:2e:bc:e3:59:0c:4b:ba:a8			
Key <u>c</u> omment:	rsa-key-20120308	rsa-key-20120308		
Key p <u>a</u> ssphrase:	•••••	•••••		
Confirm passphrase:	••••••			
Actions				
Generate a public/pr	ivate key pair		<u>G</u> enerate	
Load an existing priv	ate key file		Load	
Save the generated	key	Save p <u>u</u> blic key	Save private key	
Parameters				
Type of key to gener SSH- <u>1</u> (RSA)	ate:	© SSH	I-2 <u>D</u> SA	
Number of <u>b</u> its in a g	enerated key:		1024	

Next, click these buttons to export both parts of the key to files. Save them in a location where they can stay permanently but are easy to find. We'll need them in a bit.

Actions		
Generate a public/private key pair	•	<u>G</u> enerate
Load an existing private key file	<b>4</b>	
Save the generated key	Save p <u>u</u> blic key	Save private key

Once you have the keys, open Pageant and double-click on the computer wearing a fedora in the system tray. Hit "Add Key" to add the key we just created.

Pageant Key List			? <mark>×</mark>
Help	Add Key	<u>R</u> emove Key	Close

Find the private key (.ppk file) that you just created and add it. You'll be prompted to enter the passphrase for the key.

Pageant Key List		8 8
	Pageant: Enter Passphrase Enter passphrase for key rsa-key-20120308	
	Add Key <u>R</u> emove Key	
<u>H</u> elp		Close

And that's it. PuTTY will attempt to use the keys added in Pageant before prompting for a password. You can enable authentication forwarding in PuTTY as well by checking "Allow agent forwarding" in Connection -> SSH -> Auth.

Pageant Key List	? <b>-</b> ×
ssh-rsa 1024 51:6b:e2:a1:8e:06:44:20:2e:bc:e3:59:0c:4b:ba:a8 rsa-kev-20120	308
Add Key Remove Key	Qose