

SSH Key Pair Generation

Managed File Transfer (MFT)



The PuTTY Key Generator application (putygen.exe) is a free downloadable application you can use to create a new SSH key pair consisting of a private and public key. These keys function as the password when signing on to the BOKF MFT environment using the SFTP protocol. Follow these simple instructions on how to use PuTTY Key Generator to create a new SSH key pair for use with your BOKF MFT account.

You can download the **putygen.exe** by copying and pasting this URL into your browser https://www.chiark.greenend.org.uk/%7Esgtatham/putty/latest.html.

- 1. Once you have downloaded the **putygen.exe** executable, navigate to the folder where you saved the downloaded file.
- 2. Double-click the **putygen.exe** executable to launch the application. Once open, it should appear as it does here.

PuTTY Key Generator			? ×
ile Key Conversions Help Key No key.			
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:	⊖ ECDSA	⊖ EdDSA	O SSH-1 (RSA) 2048

- Click "Key" from the menu to view the current configured settings. If different than what is show here, match your settings as these are the most common used.
- You can also set the "Number of bits in a generated key." The default is 2048 bits. You could go to a stronger 4096 bits, but it is not necessary.

le Key	Conversions Help		
Ke	Generate key pair	1	
Nc	Add certificate to key		
	Remove certificate from key		
	SSH-1 key (RSA)		
•	SSH-2 RSA key		
	SSH-2 DSA key		
	SSH-2 ECDSA key		
	SSH-2 EdDSA key		
•	Use probable primes (fast)		
Ac	Use proven primes (slower)		22
Ge	Use proven primes with even distribution (slowest)		Generate
			Load
Lo	Use "strong" primes as RSA key factors		
Lo Sa	Use "strong" primes as RSA key factors Parameters for saving key files	key	Save private key
Lo Sa Pa •	Use "strong" primes as RSA key factors Parameters for saving key files Show fingerprint as SHA256	key	Save private key



 Now that your Key has been set and you have picked the number of bits to be generated in the key (2048 or 4096), click the "Generate" button to begin the RSA key generation process.

			?
ile Key Conversions Help			
Key			
No key.			
Actions			
Actions Generate a public/private key pair			Generate
Actions Generate a public/private key pair Load an existing private key file			Generate
Actions Generate a public/private key pair Load an existing private key file Save the generated key		Save public key	Generate Load Save private key
Actions Generate a public/private key pair Load an existing private key file Save the generated key Parameters		Save public key	Generate Load Save private key
Actions Generate a public/private key pair Load an existing private key file Save the generated key Parameters Type of key to generate: © RSA ODSA	() ECDSA	Save public key	Generate Load Save private key O SSH-1 (RSA)

 When the key generation process starts, you will be prompted to "generate some randomness." This is done by moving your mouse cursor around on your screen in random patterns. As you do this, you will see the green progress bar grow.

🚰 PuTTY Key Generator		?	×
File Key Conversions Help			
Key			
Please generate some randomness by moving the mouse over t	he blank 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: ● RSA ○ DSA ○ ECDSA	⊖ EdDSA	🔿 SSH-1 (RSA)	
Number of bits in a generated key:		2048	



- 7. Once the key generation is complete, you will see information like what is shown here. Enter a "Key passphrase" and "Confirm passphrase". This secures your private key and ensures that no one else can use it. You will want to record the passphrase (password) in a password vault or wherever passwords are stored.
- 8. Next, save both your "private" and "public" keys somewhere safe.

	erator			? >	
e Key Convers	sions Help				
Key Public key for pastir	ng into Open SSH aut	thorized_keys file:			
ssh-rsa AAAAB3NzaC1yc2 UHhkqzASYLkcjrC +zDn7yYcVyxdCD(+nZ7yQ9DBkkfb4	2EAAAADAQABAAA R81uil6d195x1e74G GjOCJgCKjtufG9L89 CSMPccLkkQu9x7	BAQCYT3pgdrjyBZBrl 3h5+z3UgAP6NgitsXI/ gJPXgm6Bxftfa1fmkF4 3astUkBshocKlyBBxG	SZrO5ViHfO2Esibp5NGGztl p1sD9XLwsRfFswX/Fv1H /3q /omMKC9bT0Va3ET040t5d	bEjkc0eKo09TgOeNQ2v	
Kev fingerprint:	ssh-rsa 2048 SHA	256:w39lir5F6HCYkL	YHR7L6SJEpE3rPtcha3D	2PkEAO5U	
Key comment:	mment: rsa-key-20230727				
Kev passphrase:					
Confirm passphrase	•••••	•••••			
Confirm passphrase Actions	•••••	•••••			
Confirm passphrase Actions Generate a public/p	private key pair	••••		Generate	
Confirm passphrase Actions Generate a public/p Load an existing priv	orivate key pair vate key file	•••••		Generate	
Confirm passphrase Actions Generate a public/p Load an existing priv Save the generated	private key pair vate key file I key		Save public key	Generate Load Save private key	
Confirm passphrase Actions Generate a public/p Load an existing pri Save the generated Parameters	private key pair vate key file I key	[Save public key	Generate Load Save private key	
Confirm passphrase Actions Generate a public/p Load an existing priv Save the generated Parameters Type of key to gene ● RSA	orivate key pair vate key file I key erate: O DSA	⊖ ecdsa	Save public key	Generate Load Save private key O SSH-1 (RSA)	

 When you save your private key, it should have a .ppk file extension. Including some details in the filename will help you in the future.

😴 Save private key a	15:	×
← → ~ ↑	> This PC > Windows (C:) > SSH-Keys v ひ Search SSH-	
Or Back (Alt + Left)	Arrow) r	III 👻 (
Name	Date modified Type Size	
	No items match your search.	
File name:	<companyname>_SSH_RSA_2048.ppk</companyname>	~
Save as type:	PuTTY Private Key Files (*.ppk)	~
∧ Hide Folders	Save	Cancel



10. When saving your public key, you will need to specify the file extension **.pub** at the end of the filename.



11. You should now have an SSH key pair consisting of one private key and one public key. You will configure your client to use the private key with your BOKF MFT user ID. You will need to provide the public key to BOKF so it can be imported into the MFT environment and tied to your account for authentication.



