SSH Key Generation and Login (MobaXterm version)

Last update: Jul 9, 2025, tested with MobaXterm 25.2

Introduction

The aim of this page is to explain how to login to RCSS supercomputer using MobaXterm.

Installation

You can download MobaXterm from https://mobaxterm.mobatek.net/.

SSH Key Generation using MobaKeyGen

Invoke Key Generator of MobaXterm



Launch MobaXterm and select MobaKeyGen (SSH key generator) from "Tools" menu. If you already have PuTTY session settings (in registry?), they will be automatically loaded on left pane of the window. This PuTTY setting may be usable.

Key Generation (1) - choose key type

🦹 MobaXterm SSH Key Generator			×
File Key Conversions Help			
Key No key.			
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
Parameters Type of key to generate: ORSA Curve to use for generating this key:	1 _{() ECDSA}	(€) EdDSA	

You may see window like above when you invoke MobaKeyGen (SSH Key Generator).

1. Choose Key Type

EdDSA (Ed25519), ECDSA (256, 384 bits), and RSA 4096 bits^{*} of keys are recommended in RCCS. Please choose EdDSA (Ed25519) if you have no preference.

Please don't choose Ed448. This type is not yet available on RCCS login server.

- Old MobaXterm Personal versions do not support SHA2 algorithms of RSA (rsa-sha2-256/512). Please use newer version of MobaXterm.
- ECDSA-521 is disabled due to the issue on PuTTY 0.68-0.80.

2. Choose Key Type/Length

In case of EdDSA, ECDSA or RSA type, you may find optional item at the position specified by "2". Please choose/input value there.

Please don't choose Ed448. This is not available on RCCS login server.

3. Start Generation of Key

Once you click the "Generate" button, the key generation will begin. You need to move mouse cursor around to proceed the key generation after clicking the button.

Key Generation (2) - set passphrase and save keys

MobaXterm SSH Key Generator X File Key Conversions Help Key Public key for pasting into OpenSSH server (7/ssh/authorized keys file): 1 Ssh-ed25519 eddsa-key- 20220308 eddsa-key- Key fingerprint: ssh-ed25519 255 Key comment: eddsa-key-20220308 Key passphrase: Confirm passphrase: Confirm passphrase: Confirm passphrase: Confirm passphrase: Confirm passphrase: Actions Generate Generate a public/private key pair Generate Load an existing private key file Load Save the generated key Save public key Parameters 3 Type of key to generate: OBA O BSA EdDSA OSH-1 (RSA) Curve to use for generating this key: Ed25519 (255 bits) V						
Key 1 Public key for pasting into OpenSSH server (7/ssh/authorized keys file). 1 ssh=ed25519 eddsa-key- 20220308 eddsa-key- Key fingerprint: ssh=ed25519.255 Key comment: eddsa-key-20220308 Confirm passphrase: Image: Confirm passphrase: Confirm passphrase: Image: Confirm passphrase: Generate a public/private key pair Generate Load Save public key Save the generated key Save public key Parameters 3 Type of ley to generate: U DSA Image: EdDSA W EdDSA U SSH=1 (RSA)	🦹 MobaXterm SSH I	Key Generator				\times
Public key for pasting into OpenSSH server (7/ssh/authorized keys file) 1 ssh=ed25519 eddsa-key= 20220308 eddsa-key= Key fingerprint: ssh=ed25519 255 Key comment: eddsa-key=20220308 Confirm passphrase: Image: Confirm passphrase: Confirm passphrase: Image: Confirm passphrase: Generate a public/private key pair Generate Load Save public key Save the generated key Save public key Parameters 3 Type of ley to generate: U DSA Image: EdDSA W EdDSA U SSH=1 (RSA)	File Key Conversion	ns Help				
Key comment: eddsa-key-20220308 Key passphrase: Image: Confirm passphrase: Confirm passphrase: Image: Confirm passphrase: Actions Image: Confirm passphrase: Generate a public/private key pair Image: Confirm passphrase: Load an existing private key file Image: Confirm passphrase: Save the generated key Save public key Save the generated key Save public key Parameters 3 Type of key to generate: Image: Confirm passphrase: In RSA Image: Confirm passphrase:	Public key for pasting ssh-ed25519	t into OpenSSH server	- (~/.ssh/authorized	kevs file):	1 eddsa-key-	^ ~
Key passphrase: Image: Confirm passphrase: Image: Con	Key fingerprint:	ssh-ed25519 255				
Confirm passphrase: •••••••••••••••••••••••••••••	Key comment:	eddsa-key-2022030	8			
Actions Generate a public/private key pair Load an existing private key file Save the generated key Parameters Type of key to generate: ORSA ODSA OECDSA OECDSA OECDSA OSH-1 (RSA)	Key passphrase:	•••••				
Generate a public/private key pair Generate Load an existing private key file Load Save the generated key Save public key Parameters 3 Type of key to generate: U DSA U RSA U DSA	Confirm passphrase:	••••••			2	
Load an existing private key file Save the generated key Parameters Type of key to generate: ORSA OECDSA OECDSA OECDSA OECDSA OSH-1 (RSA)	Actions					
Save the generated key Save public key Save private key Parameters Type of key to generate: URSA UDSA UECDSA (•) EdDSA USH-1 (RSA)	Generate a public/pr	ivate key pair			Generate	
Parameters 3 Type of key to generate: U RSA U DSA U ECDSA (*) EdDSA U SSH-1 (RSA)	Load an existing priva	ate key file			Load	
Type of key to generate: U RSA U DSA U ECDSA (*) EdDSA U SSH-1 (RSA)	Save the generated	ey		Save public key	Save private key	
Û RSA Û DSA Û ECDSA (●) EdDSA Û SSH-1 (RSA)	Parameters				3	
Curve to use for generating this key: Ed25519 (255 bits) V			⊖ eodsa	() EdDSA	⊖ ssh-1 (rsa)	
	Curve to use for gen	erating this key:			Ed25519 (255 bits)	~

Once the key generation completed, the appearance of the window will change like above.

1. OpenSSH type public key

The public key shown as a string in this field is what we need. Extract all the contents in this filed into notepad or others, and then save it! (Do not miss ssh-/ecdsa- part in the beginning!) Note: you don't need public key from "Save public key" button; we need only OpenSSH format one.

You can rebuild public keys via "Load" button or "Conversion" menu if you still have private key. (If you lost the private key, you need to generate a new key.)

2. Set passphrase for private key

You can set passphrase for private key here. We, RCCS, recommend passphrase of 10 or more characters containing 4 types of characters - "lower-case", "upper-case", "number", and "symbol".

3. Save the private key

After setting passphrase, click "Save private key" button to save the key. Easy-to-understand name such as "rccs.ppk" or "ccfep.ppk" may be a good choice. (NOTE: THE PRIVATE KEY FILE MUST BE KEPT SECRET!)

Register Public Key (Common)

You need to register the public key before login. Please register your public key according to the instructions in https://ccportal.ims.ac.jp/en/account.

Please note that the public key here is an OpenSSH type one not the one from "Save public key" button.

Login

Create a new session



Quit MobaKeyGen (or restart MobaXterm) to go back to the MobaXterm window. Click"Session" button on top-left part of the window to create a new session.



Theh, Click "SSH" to create a new SSH session.

				×
SH Telnet Rsh Xdmcp	RDP VNC FTP		Shell Browser Mosh	💖 🔳 Aws S3 WSL
Basic SSH settings	pecify	username	~ 2 7	Port 22 🛟
Advanced SSH settings 🛛 Ter	2 rminal settings 🔆 Network	< settings 👌 📩 Bookmark	settings	
3				
X11-Forwarding	Compression Rem	note environment: Interac	tive shell \sim	
_				
Execute command:		Do not exit after	command ends	•
_	• protocol	 Do not exit after Follow SSH path 		S
Execute command:	? protocol id_rsa		(experimental)	٩,

You need to complete following settings in this page.

- 1. input login server name ccfep.ims.ac.jp into the box.
- 2. check "Specify username", then input RCCS user ID (three-letters ID) in the textbox.
- 3. click "Advanced SSH settings" tab to expand this.

4. check "Use private key" and specify private key file location (both of OpenSSH style and PuTTY style private keys are accepted).

Input Passphrase



After the connection, you need to input passphrase of private key.

Note: MobaXterm skips verification of the connecting host in default. This verification can be enabled by checking "Validate host identity at first connection" item in SSH tab of "Settings" -> "Configuration" menu. Valid fingerprints of the login server are listed below.

- ad:de:79:30:81:b0:b1:6a:17:f5:6f:ea:f4:b4:3b:de (MD5)
- e3:a9:bb:6f:e4:a9:37:fd:98:14:54:16:3c:81:2d:aa (MD5)
- 07:7e:df:7b:16:47:a8:f5:7c:48:b0:a3:d3:86:71:6a (MD5)
- wnEM30z4AxyDJ9XI/DdGr2PINeoivFRR8v5krXHEmdU (SHA256)
- 0KL38Yn/kBee1pAuxyKwenEwXjtPxr9ZElolfVqXvbI (SHA256)
- Nhg+9Lgj3XeuW//A/j7jqgUJllxWehryCtStlp1Dir (SHA256)

💐 ccfep. Terminal	ims.ac.jp Sessions	View	×	Table Ca		14	and the				- 0	×
5ession	Servers	Tools	Games	★ Sessions	View	igs ivi III Split	Acros Help	eling Packages	Settings	? Help	X server	U Exit
ssions	home/users Name		-	Las V	≻ SS • •	SSH o SSH-b X11-f DISPL or mor	sion to ompression rowser orwarding AY e info, ctr Jul 12	• Moba) nt, X-serv @ccfep.cer : - : - : - : - : - (remo : - (auto	(term 11 ver and nter.ims ote disp omatical n <u>help</u> c	1 • networking tools)	ough SSH)	

You may see window like above if you successfully logged in.

Note 1: SFTP

Login Completed



MobaXterm has SFTP functionality. In the left pane of the window (marked with red square), you can download/upload files via SFTP.

Note 2: X11



MobaXterm has internal X server. Therefore, you can use X window application without special settings. ("X11-forwarding" item must be checked in the message displayed upon successful login.)