Fixing the “too open” permission error

Situation	Your Laptop	Lab Systems
Who owns the .pem file	You (single user)	Shared system, different users per batch
File location	Inside your user folder (safe by default)	Often in Downloads of another user, or in Public/shared folder
Windows permissions	Automatically restricted to your account	Other users might have access to the same folder
Result	Works fine	SSH refuses: “Permission denied (publickey)” or “Unprotected private key file”

Give permission only to a particular user (lab system with multiple users)

On Windows, these commands **remove everyone else’s access and give read permission only to one user.** That’s why it fixes the “too open” permission error.

Run these commands in PowerShell as Administrator:

```
# Go to folder where pem file is stored
cd "C:\Users\studentX\Downloads"

# Remove inherited permissions (others lose access)
icacls .\keyfile.pem /inheritance:r

# Grant permission only to the current logged-in user (replace studentX)
icacls .\keyfile.pem /grant:r "studentX:(R)"
```

icacls → Integrity Control Access Control Lists

- **“I” → Integrity**
  Refers to *Windows integrity levels* (used for protecting system files and preventing lower-privilege processes from modifying higher-level ones).
- **“Cacls” → Change Access Control Lists**
  The old command in Windows was cacls (Change ACLs).
  Microsoft later improved it and introduced **icacls**, which adds support for:
  - Viewing and modifying file/folder permissions
  - Handling NTFS access control lists (ACLs)
  - Managing ownership and inheritance
  - Supporting integrity levels

After running these commands, your .pem file becomes:

- **Readable only** by the current student user.
- **Protected** from access by any other batch login.
- **Fully compliant** with AWS's SSH key security rule (same as chmod 400).

These commands are **Windows equivalents** of the Linux command:

chmod 400 key.pem

Linux command means: "Make the key file readable **only** by the file's owner."

In Windows, we achieve the same effect using **icacls** — the tool for managing file access control lists (ACLs).

Go to the folder where the .pem file is stored

cd "C:\Users\studentX\Downloads"

cd = *Change Directory*
It moves PowerShell's current working location into the folder that contains your key file.
If your key file is in another folder (say Desktop), you'd use:

cd "C:\Users\studentX\Desktop"

Remove inherited permissions (others lose access)

icacls .\keyfile.pem /inheritance:r

- icacls = the Windows command for changing file permissions.
- . \keyfile.pem = means "this file named keyfile.pem in the current folder."
- /inheritance:r = "remove inherited permissions."

What it does:
By default, Windows files *inherit* permissions from their parent folder (like "Downloads").
That means "Administrators" or "Users" groups may have access.
This command stops that — so the .pem file no longer automatically allows anyone else.

Give permission only to the current logged-in user

icacls .\keyfile.pem /grant:r "studentX:(R)"

- /grant:r = grant rights, replacing any existing ones.
- "studentX:(R)" = give **Read-only** access to the user studentX.

After this command:
Only studentX can read the .pem file.
No other users (like student2, admin, or system) can access it.