# SSH RSA key authentication (Linux)

Main Page > General Information > Configuration Examples > Router control and monitoring > **SSH RSA key authentication (Linux)**
☐

# Contents

# Introduction

**Secure Shell** (**SSH**) is a cryptographic network protocol for operating network services securely over an unsecured network. The best known example application is for remote login to computer systems by users.

Typically an SSH login involves specifying a user name, IP address or hostname and the password for the user. When you login to a certain IP address/hostname, the system generates a public/private RSA key pair between the two parties. There is a possibility to login to via SSH using only that type of **key** with the help of the **dropbear** service, thus, eliminating the password step. This article provides instructions on how to set up public key authentication for logging in to a RUTxxx router via SSH using a Linux OS. For the Windows guide, click **here**.

# Prerequisites

To achieve the configuration described in this article you will need the following:

- A computer running on Linux
- A RUTxxx router of any type

# Configuration

- First, we'll need to generate the **rsa key**. This can be achieved with the help of the *ssh-keygen* command. Simply open a Terminal and enter the command:

  ```
  ssh-keygen
  ```

  When prompted, enter the file name, for example **rsa_key**. Two file will then be generated: **rsa_key** (private) and **rsa_key.pub** (public).

---

- Next, we'll have to add the key to your RUTxxx router. Simply execute this command via a Terminal (replace the IP address with your router's IP):

  ```
  ssh-copy-id -i rsa_key.pub root@192.168.1.1
  ```

- The key will be added to the *root/.ssh/authorized_keys* file. You must then copy it to *etc/dropbear/*. To do so, login to the router via SSH and execute this command:

```
cp /root/.ssh/authorized_keys /etc/dropbear/
```

- Next, while still connected to the router, add *read*, *write* and *execute* permissions for the *etc/dropbear* directory and *read*, *write* permissions for the *etc/dropbear/authorized_keys* file:

```
chmod 700 /etc/dropbear
chmod 600 /etc/dropbear/authorized_keys
```

- At this point, the configuration is complete. To test it, terminate your current SSH connection (you can do so by executing the *exit* command) and try logging in again - if everything is in order, the router should no longer require a password when connecting via SSH when using the command:

```
ssh -i rsa_key root@192.168.1.1
```

- **Additional notes**:
  - Other devices will not be able to connect using your key, but keep in mind that if someone gains physical access to your computer, they will be able to connect to the router without a password.
  - If you're using SSH remotely, don't forget to use the router's public IP address when logging in and enable remote SSH access on the router. You can do that by issuing the following commands:

    ```
    uci set firewall.@rule[5].enabled=1
    uci commit
    /etc/init.d/firewall restart
    ```

# (Optional) Adding additional security

- You can also add additional security for your router's SSH connections by disabling the password login entirely. To do so, SSH to your router and execute the following commands:

```
uci set dropbear.@dropbear[0].PasswordAuth=off
uci commit
/etc/init.d/dropbear restart
```

- If you have configured other users besides *root* and want to leave access with password ON for them, you can disable password login only for the user *root* by executing these commands instead:

```
uci set dropbear.@dropbear[0].RootPasswordAuth=off
uci commit
/etc/init.d/dropbear restart
```

- **Additional notes**:
  - When you disable SSH password authentication, only users with keys will be able to login via SSH. If another user needs access via SSH, you can temporarily enable SSH password authentication until the user in question sets up their authentication. Or the user can generate the key, send it to you and you can add it to the */etc/dropbear/authorized_keys* file.
  - When you disable SSH password authentication, make sure you don't accidentally delete your key as you will not be able to connect to your router via SSH. However, if this does happen, you can still login via the Command Line Interface (CLI) from the router's WebUI (**Services → CLI**) or other forms of CLI described **here**. When you login, simply enable SSH password authentication with these commands:

    ```
    uci set dropbear.@dropbear[0].PasswordAuth=on #### use uci set
    dropbear.@dropbear[0].RootPasswordAuth=on instead if you had only
    disabled password authentication for root
    uci commit
    /etc/init.d/dropbear restart
    ```

# See also

- SSH RSA key authentication (Windows) - the same guide but aimed at Windows users
- Command line interfaces - descriptions and instruction for all types of command line interfaces supported by RUTxxx devices