**Lesson 5 SSH generate**

SSH-ключ **Secure Shell (безопасная оболочка)**— это специальный код, который позволяет удалённому компьютеру понять кто вы есть и какими правами на этом компьютере обладаете.

## **Как работает SSH-ключ**

SSH-ключ разделён на две части. Одна часть называется приватной и должна всегда храниться только на вашем компьютере. Вторая часть ключа называется публичной и эту часть нужно копировать на другие компьютеры. При подключении к удалённому компьютеру он сравнивает публичную часть, которую вы ему дали с приватной частью, которая хранится у вас. Если части ключа совпадают, то вы получаете доступ к удалённому компьютеру (на самом деле всё гораздо сложнее, но принцип работы именно такой). На вашем компьютере может быть создано сколько угодно SSH-ключей. То есть вы можете использовать один SSH-ключ для доступа к сотне компьютеров или для каждого удалённого компьютера создавать отдельный ключ.

## **Защита SSH ключа**

Вы никогда и ни при каких обстоятельствах не должны никому передавать приватную часть своего ключа. Чтобы эту приватную часть нельзя было украсть можно её зашифровать, тогда даже если она попадёт в чужие руки её не смогут использовать. Если вы зашифруете приватную часть SSH-ключа, то каждый раз при его использовании вам нужно будет вводить ключ шифрования, называемый passphrase. Есть общее правило безопасности: если ключ используется не скриптом, а человеком, то он должен быть зашифрован.

## **Как создать SSH-ключ под Windows**

Windows не имеет встроенной поддержки SSH протокола. Поэтому, для создания SSH-ключа нужна специальная программа. Подойдёт Git Bash, которая [устанавливается вместе с Git под Windows](https://blog.vistro.ru/ide/git/how-to-install-git-on-windows/).

Инструкция по созданию SSH-ключа:

1. запустите Git Bash (используется только в ОС Windows, на Linux & MacOS есть свои терминалы);
2. введите команду ssh-keygen;
3. программа предложит выбрать место, куда сохранить ключ. По умолчанию создаётся каталог .ssh в папке текущего пользователя Windows. Например, у меня это C:\Users\Denis\.ssh. Этот вариант подходит, поэтому нажимайте Enter;
4. программа запросит passphrase для шифрования ключа. Если просто нажать Enter, то приватная часть ключа не будет зашифрована. Помните, что если SSH-ключ используется не скриптом, то он должен быть зашифрован. Относитесь к passphrase как к паролю, его нужно запомнить;
5. после нажатия Enter, программа попросит ввести passphrase повторно. Если вы не шифруете ключ, то ничего не вводите. Следующее нажатие Enter завершит создание ключа.

Теперь в каталоге .ssh созданы два файла:

* id\_rsa — приватный ключ (никому не передавать);
* id\_rsa.pub — публичный ключ (нужно копировать на другие компьютеры).

Git — система управления версиями с распределенной архитектурой. В отличие от некогда популярных систем вроде CVS и Subversion (SVN), где полная история версий проекта доступна лишь в одном месте, в Git каждая рабочая копия кода сама по себе является репозиторием. Это позволяет всем разработчикам хранить историю изменений в полном объеме.

There are many color codes, but you will often see only the 7 colors listed below:

* **White (No color code):** Regular File or Normal File
* **Blue:** Directory
* **Bright Green:** Executable File
* **Bright Red:** Archive file or Compressed File
* **Magenta:** Image File
* **Cyan:** Audio File
* **Sky Blue:** Symbolic Link File

Подсказки по уроку:

**Статья про ГИТ:**

<https://habr.com/ru/post/541258/>

**генерируем ключ SSH:**

ssh-keygen

**Посмотреть доступ к файлам с ключами:**

ls -la .ssh

**Имя сервера:**

ssh -i ~/.ssh/id\_rsa [ec2-user@linux.telran-edu.de](mailto:ec2-user@linux.telran-edu.de)

**Добавление ключа редактором nano:**

nano .ssh/authorized\_keys

**показать ключ:**

cat .ssh/id\_rsa.pub

**Выйти из nano:**

ctrl+x