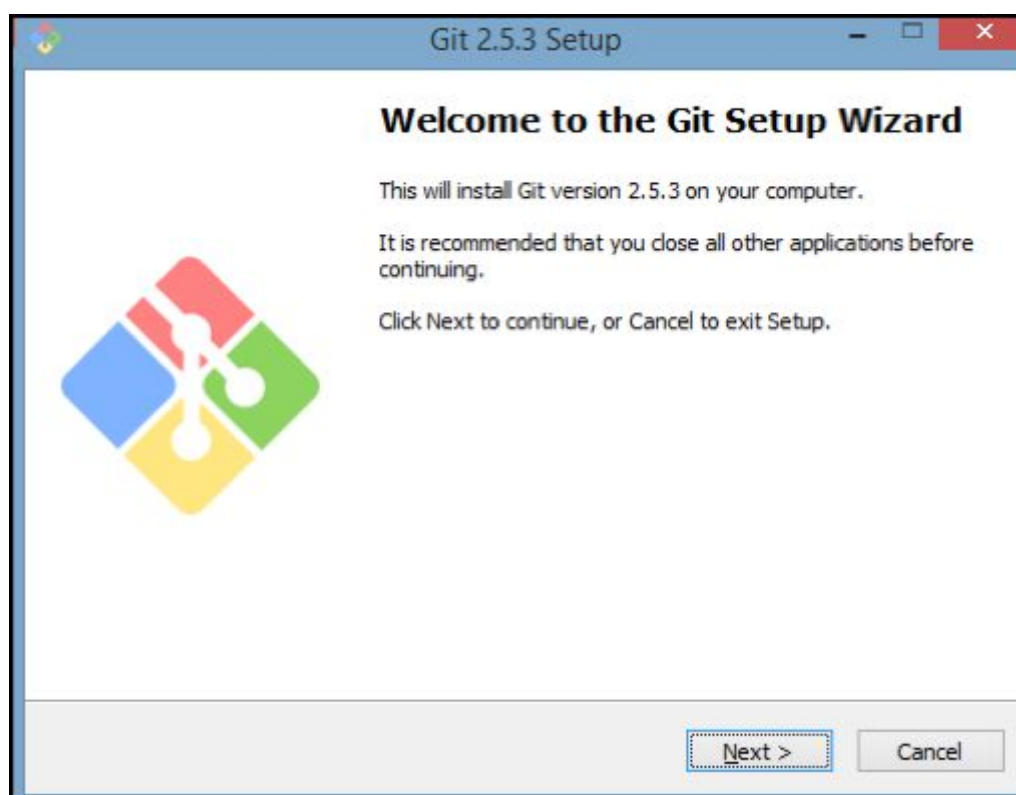


Установка Git

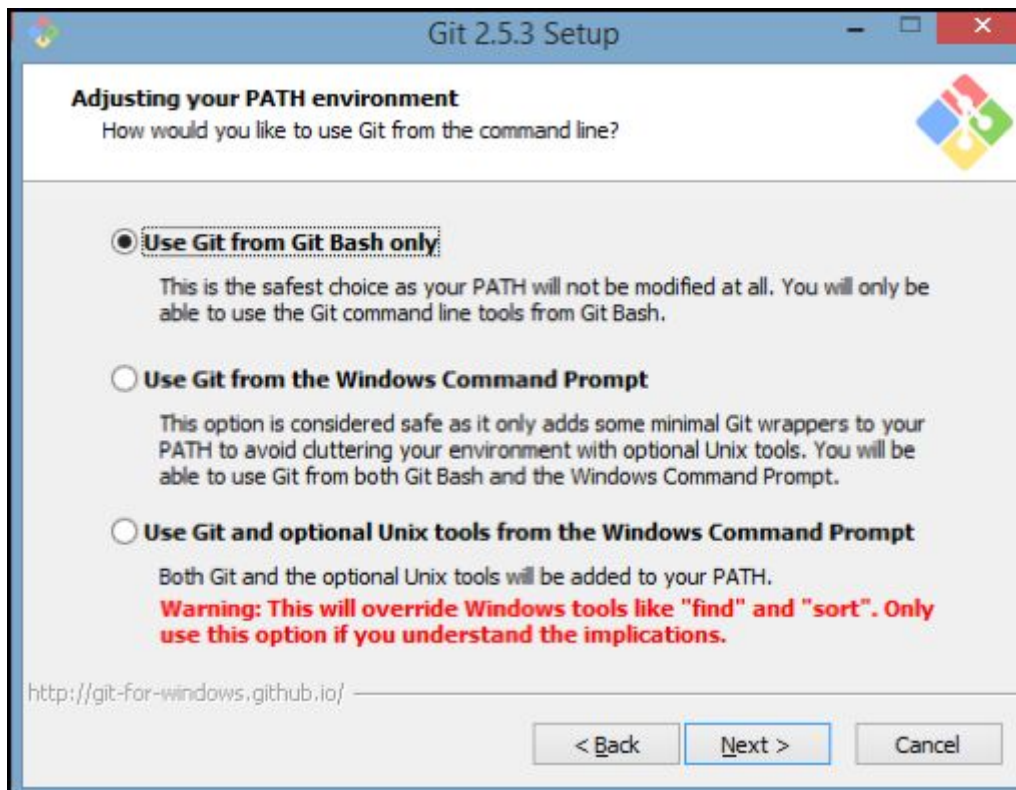
В UNIX-подобных операционных системах консольная утилита git может быть уже установлена. Тем не менее важно уметь устанавливать git в любой операционной системе, с которой вам доведется работать. Здесь мы рассмотрим три наиболее популярных: Windows, Ubuntu (дистрибутив Linux) и MacOS.

Установка в Windows

Для установки Git в операционной системе Windows лучше всего воспользоваться пакетом Git for Windows, который можно скачать по адресу <https://git-for-windows.github.io/>. Данный пакет включает в себя эмуляцию командного интерпретатора bash, а также графическую утилиту для управления Git. Следует выбрать 64- или 32-битный дистрибутив в зависимости от используемой версии Windows. После загрузки и запуска дистрибутива откроется окно приветствия мастера установки.



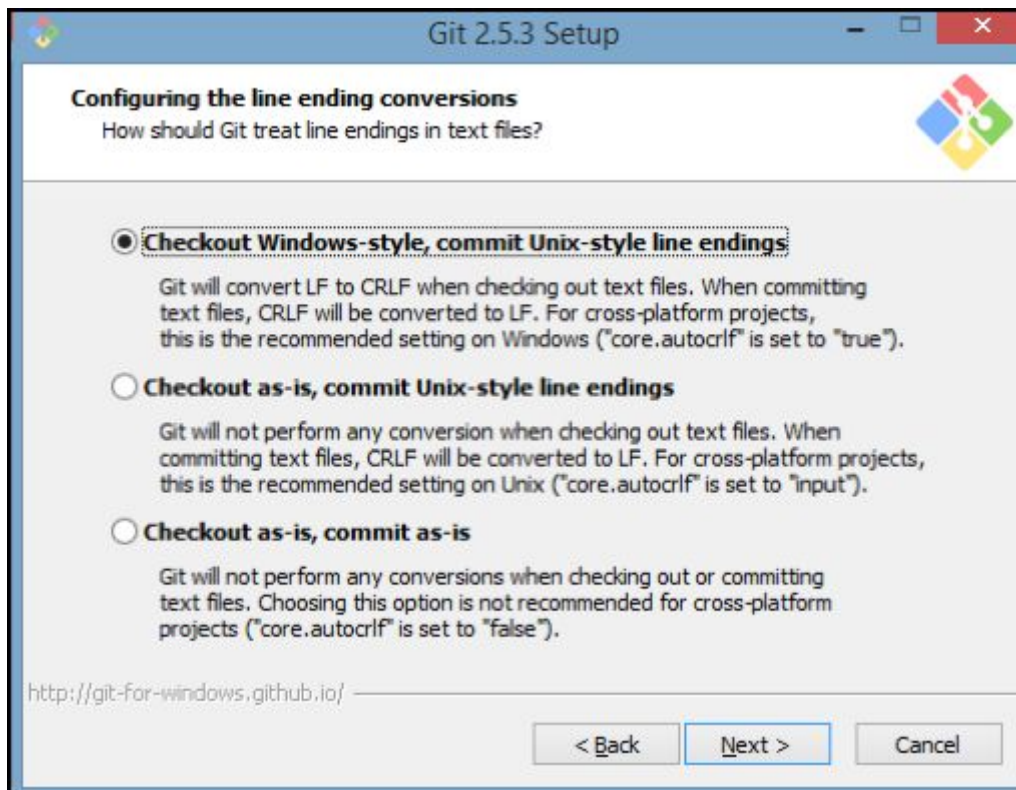
Мастер установки предлагает довольно длительную серию диалоговых окон, в которых настраивается работа UNIX-окружения командной строки. После согласия с лицензионным соглашением, расположением Git в файловой системе, составом устанавливаемого программного обеспечения откроется окно выбора видимости команд, которое предоставляет три варианта.



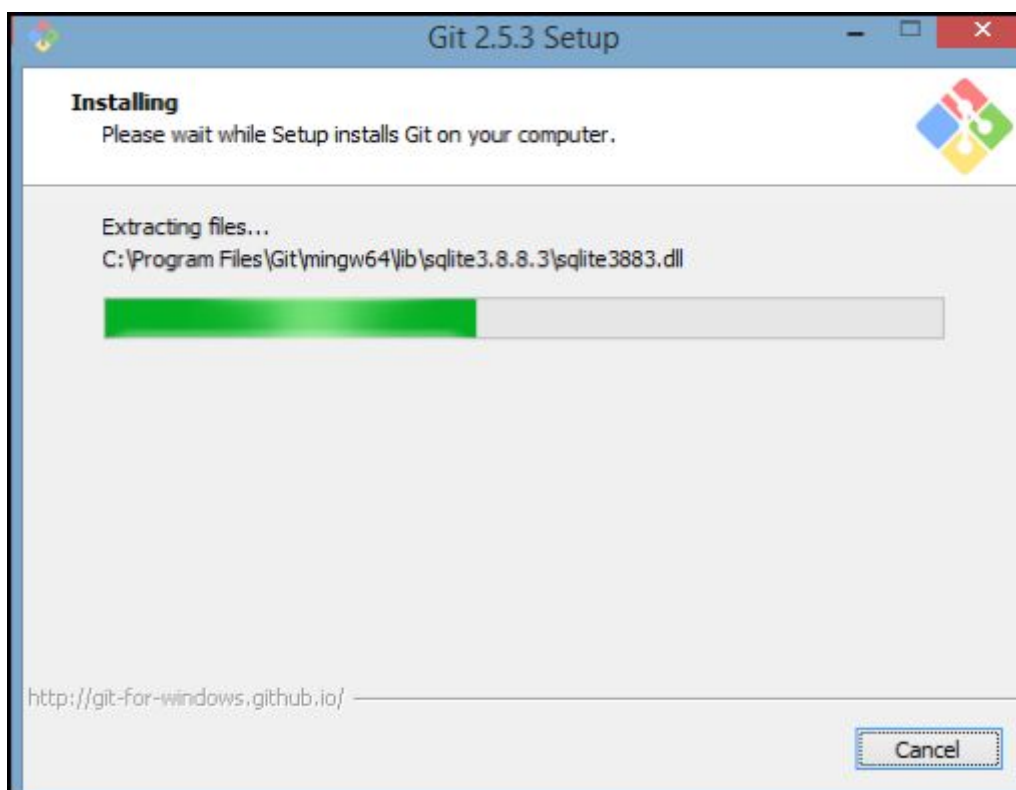
Первый вариант предлагает использование команд Git и UNIX-команд только из bash-консоли, которая запускается отдельным приложением. Второй вариант предлагает минимальный набор команд для использования в командной строке Windows. Третий вариант предлагает полноценное UNIX-окружение для командной строки Windows. В последнем варианте могут изменяться свойства некоторых консольных команд Windows, таких как find или sort.

Следующее диалоговое окно предлагает варианты управления символами перевода строки. В UNIX-подобных операционных системах переводом строки служит символ `\n`, в то время как в Windows для перевода строки используются два символа `\r\n`. В результате, в UNIX-редакторах в конце строки может появляться лишний символ `\r` при получении файлов из Windows, а в Windows-редакторах, наоборот, строки могут сливаться в одну длинную строку при попытке прочтения файлов из UNIX-подобных операционных систем.

- Первый вариант предлагает конвертировать UNIX-переводы строк `\n` в Windows-переводы `\r\n` при формировании рабочего каталога. При создании коммита осуществляется обратное преобразование.
- Второй вариант предлагает конвертировать все переводы строк в UNIX-формат `\n`.
- При выборе третьего варианта переводы строк остаются как есть без какого-либо дополнительного преобразования.



После этого продолжится установка программного обеспечения.



По завершении установки Git- и UNIX-команды станут доступными из командной строки Windows.

Активация Linux-подсистемы в Windows

Начиная с версии Windows 10 можно активировать Linux-подсистему. В результате у вас появится специальная командная строка, в которой будут работать все UNIX-команды выбранного linux-дистрибутива (ubuntu, suse, debian).

Активация Linux-подсистемы подробно описывается в официальной документации по ссылке <https://docs.microsoft.com/ru-ru/windows/wsl/install-win10>. Рекомендуется выбрать дистрибутив Ubuntu, это позволит воспользоваться командами из следующего раздела.

Установка в Ubuntu

Проверить, установлена ли система контроля версий Git на рабочей станции, можно, выполнив команду git с параметром --version:

```
$ git --version
git version 1.7.9.5
```

Если в результате выполнения команды будет выведена версия Git, то система контроля версий установлена и готова к работе. Если вместо версии будет выведено сообщение о том, что команда git не найдена, это будет означать, что система Git не устанавливалась на рабочей станции или сервере.

```
$ git --version
-bash: git: команда не найдена
```

Для того чтобы установить Git в Ubuntu, следует выполнить команду:

```
$ sudo apt-get install git
-bash: git: команда не найдена
```

Установка в Mac OS X

Для установки Git достаточно загрузить и установить пакет Command Line Tools for XCode из AppStore. XCode — это интегрированная среда разработки приложений для Mac OS X и iOS. Полная загрузка XCode не обязательна, однако если XCode уже установлен в вашей системе, Git будет доступен для использования. Убедиться в том, установлен ли XCode, можно при помощи команды:

```
$ xcode-select -p
/Applications/Xcode.app/Contents/Developer
```

Если вместо указанного выше пути выводится предложение установить Command Line Tools, следует установить этот пакет, выполнив команду

```
$ xcode-select --install
```

После установки Git будет доступен для использования, в чем можно убедиться, воспользовавшись командой:

```
$ git --version  
git version 2.3.8 (Apple Git-58)
```

Git, который входит в состав Commands Line Tools, зачастую является не самой последней версией. Для того чтобы установить последнюю версию Git, можно воспользоваться менеджером пакетов Homebrew.

На момент написания данного текста, установить Homebrew можно было при помощи команды:

```
$ /usr/bin/ruby -e "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Впрочем, точную команду всегда можно выяснить на официальном сайте <https://brew.sh>. После установки в командной строке будет доступна команда **brew**, при помощи которой можно загружать, удалять и обновлять пакеты с программным обеспечением.

После того, как менеджер Homebrew установлен, приступаем к установке Git. Сделать это можно при помощи команды:

```
$ brew install git  
==> Downloading  
https://homebrew.bintray.com/bottles/git-2.5.3.yosemite.bottle.tar.gz  
##### 100,0%  
==> Pouring git-2.5.3.yosemite.bottle.tar.gz  
==> Caveats  
The OS X keychain credential helper has been installed to:  
  /usr/local/bin/git-credential-osxkeychain  
  
The "contrib" directory has been installed to:  
  /usr/local/share/git-core/contrib  
  
Bash completion has been installed to:  
  /usr/local/etc/bash_completion.d  
  
zsh completion has been installed to:  
  /usr/local/share/zsh/site-functions  
==> Summary  
  /usr/local/Cellar/git/2.5.3: 1390 files, 32M  
$ git --version  
git version 2.5.3
```

Протокол SSH

Для работы на удаленном сервере, установки на нем программного обеспечения, настройки конфигурационных файлов, мониторинга запущенных процессов требуется выполнение команд на сервере. Для связи с сервером, выполнения команд, обмена файлами администраторы и Web-разработчики используют протокол SSH (Secure Shell), обеспечивающий полное шифрование трафика.

При работе с удаленными Git-репозиториями (например `github.com`), связь тоже осуществляется через SSH.

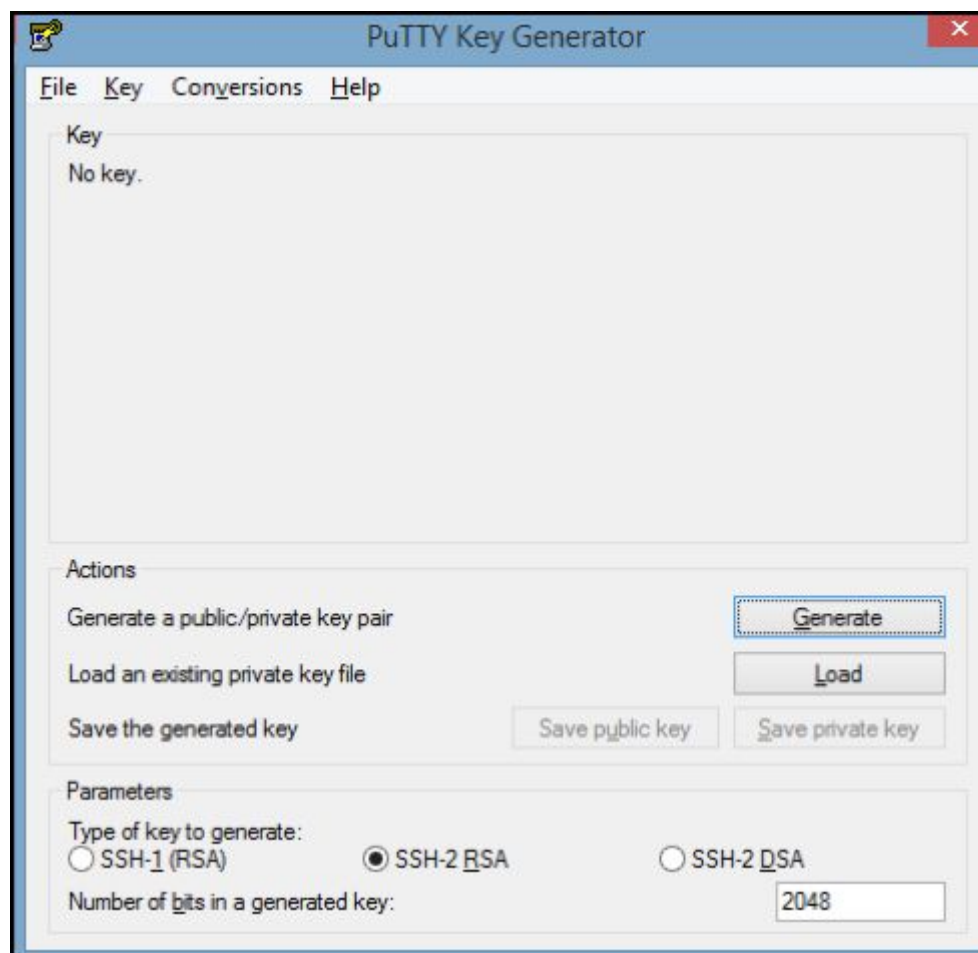
Доступ по SSH в Windows

Если вы воспользовались советом установить Git for Windows, ssh-клиент у вас уже установлен и им можно воспользоваться. Так как устанавливаем соединение мы будем с `github.com`, нам потребуются открытый и закрытый ключи.

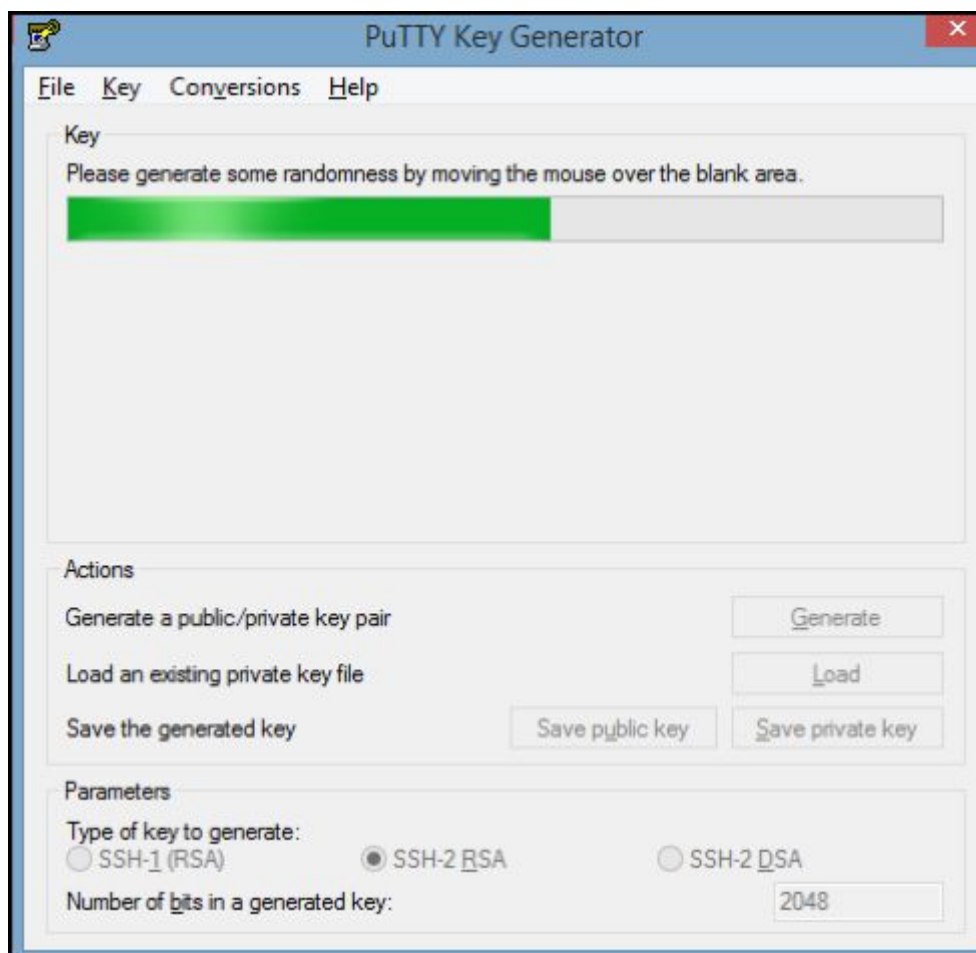
Закрытый ключ помещается в домашнем каталоге вашего пользователя в папку `.ssh`, а открытый ключ добавляем в github-аккаунт: настройки (Settings), раздел SSH and GPG keys, добавить новый SSH-ключ.

Для создания пары открытого и закрытого ключей, вам придется скачать утилиту `puttygen.exe` по ссылке <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>.

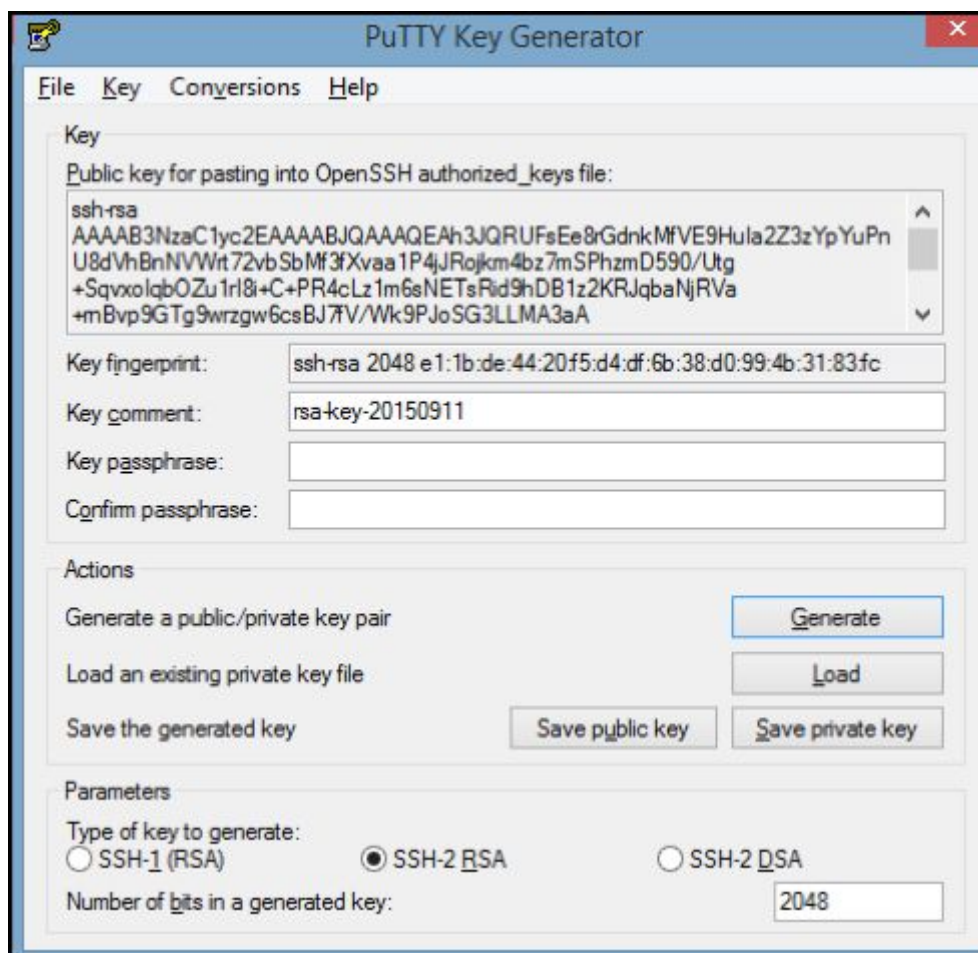
Утилита запускается без установки и выглядит так, как это представлено ниже.



Для создания ключа следует нажать кнопку **Generate**. Процедура генерации секретного ключа нуждается в последовательности случайных цифр, для этого утилита предлагает осуществлять случайные действия с мышью и клавиатурой. По мере накопления данных можно наблюдать за полосой прогресса.

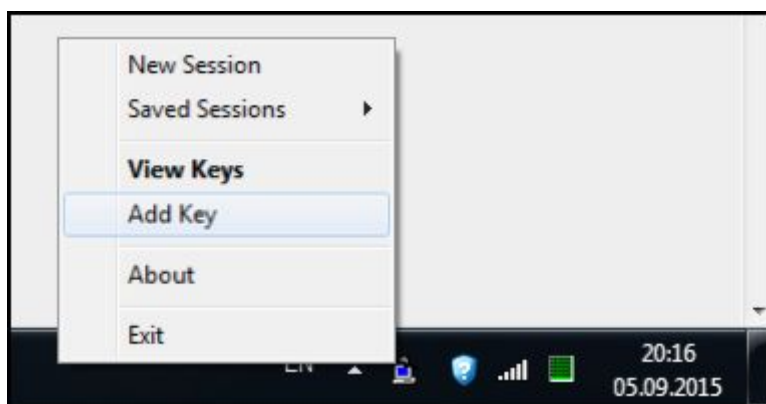


Для преобразования уже существующего ключа следует нажать кнопку **Load** и загрузить закрытый OpenSSH-ключ, созданный ранее в Ubuntu. Если закрытый ключ защищен паролем, утилита потребует ввести его. После генерации или преобразования ключей их следует сохранить: открытый ключ при помощи кнопки **Save public key**, закрытый — **Save private key**.



Получив ключи, необходимо запустить агента pageant, который также можно загрузить со страницы <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>.

После запуска утилита попадет в область уведомлений (системный трей).



Щелчком правой кнопкой мыши можно вызвать контекстное меню, в котором имеется пункт **Add Key**, вызывающий диалоговое окно для загрузки закрытого ключа. Если ключ защищен паролем, его придется ввести.

Агент будет перехватывать любые SSH-соединения и обеспечивать доступ по загруженному ключ (т. е. без логина и пароля).

Доступ по SSH в UNIX-подобных операционных системах

Как правило, SSH-клиент в UNIX-подобных операционных системах уже установлен. Если это не так в MacOS его можно установить в составе пакет Command Line Tools for XCode (как это сделать описано выше), а в Ubuntu можно воспользоваться командой:

```
$ sudo apt-get install ssh
```

Для генерации ключей служит команда `ssh-keygen`, которую следует выполнить на клиентской машине:

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/igor/.ssh/id_rsa):
Created directory '/home/igor/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/igor/.ssh/id_rsa.
Your public key has been saved in /home/igor/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:1eVTqU42ZVtns6DPMN2S1y3EDdK4K+C1I8XYApzevEU igor@cruiser
The key's randomart image is:
+---[RSA 2048]-----+
|  . .      .+ooo|
|    +   E  ..=+*=|
|  . + =   . +oX.O|
|    . = *..+.O.*o|
|    *S+  O.+   |
|    o + . .+   |
|    . . .     |
|              |
|              |
+----[SHA256]-----+
```

Во время выполнения команда задаст несколько вопросов, в частности она попросит указать путь, куда будут сохранены ключи (по умолчанию папка `.ssh` домашнего каталога пользователя), далее будет предложено ввести пароль для закрытого ключа. Настоятельно рекомендуется его указать, т. к. в случае если ключ будет похищен злоумышленниками, им не смогут воспользоваться без пароля.

В результате выполнения команда создаст в домашнем каталоге скрытый подкаталог `.ssh`, в котором размещается закрытый `id_rsa` и открытый `id_rsa.pub` ключи.

Если при создании пары открытого и закрытого ключей был задан пароль, то при каждой попытке воспользоваться ключом потребуются его ввод. Это может быть утомительным. Кроме того, некоторые операции, особенно в пакетном режиме, не позволяют вводить пароль. Для того чтобы не вводить парольную фразу каждый раз,

ее можно ввести единожды в начале сеанса, передав на хранение SSH-агенту. В Ubuntu агент запущен по умолчанию, поэтому им сразу можно воспользоваться, а для добавления ключа текущего пользователя можно выполнить команду `ssh-add`:

```
$ ssh-add
```

Команда запросит пароль ключа и передаст его агенту. После этого в течение текущего сеанса вводить парольную фразу не потребуется.