

# TortoiseGit による SSH 通信用秘密鍵・公開鍵生成手順

2013 年 7 月 19 日 初稿



稿	改訂日	改訂者	改訂内容
初稿	2013 年 7 月 19 日	板垣 衛	(初稿)

## ■ 目次

■ 本書が扱うソフトウェアとバージョン .....	1
■ 秘密鍵と公開鍵について .....	1
■ 秘密鍵と公開鍵の生成方法①：puttygen を使用する方法 .....	1
■ 秘密鍵と公開鍵の生成方法②：ssh-keygen コマンドを使用する方法 .....	4
■ 公開鍵をサーバーに送る方法（GitHub の例） .....	6
■ 秘密鍵のパスフレーズを変更する方法：puttygen を使用する方法.....	7

## ■ 本書が扱うソフトウェアとバージョン

- ・ msysGit ... Ver.1.8.3.msysgit.0
- ・ TortoiseGit ... Ver.1.8.3.0 – 64 Bit

## ■ 秘密鍵と公開鍵について

秘密鍵と公開鍵は、通信データを暗号化・複合化する際に、一対で扱われるものである。

以下の特徴がある。

- ・ 秘密鍵と公開鍵とは、何桁かの数字の集まりであり、通常テキストファイルの形式で保存される。
- ・ 秘密鍵と公開鍵は、ユーザー自身が作成する。
- ・ 秘密鍵はそのままユーザーが自分の PC に保持する。(秘密鍵がそれを作成した PC の外に出る事は基本的にない)
- ・ 秘密鍵は、「パスフレーズ」と呼ばれる一種のパスワードで保護されている為、万が一秘密鍵が漏洩しても、パスフレーズを知らなければ扱う事ができない。
- ・ 秘密鍵と公開鍵は一緒に作成され、公開鍵は通信先のサーバーに保存する。
- ・ 公開鍵の受け渡し方法はサーバーによりさまざま。
- ・ 秘密鍵、公開鍵共に幾つかの規格がある為、サーバーに合わせた規格を使用する。

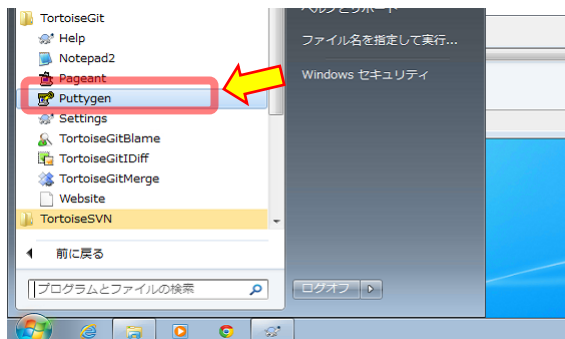
## ■ 秘密鍵と公開鍵の生成方法①：puttygen を使用する方法

前提①：Windows 上で鍵を生成する。

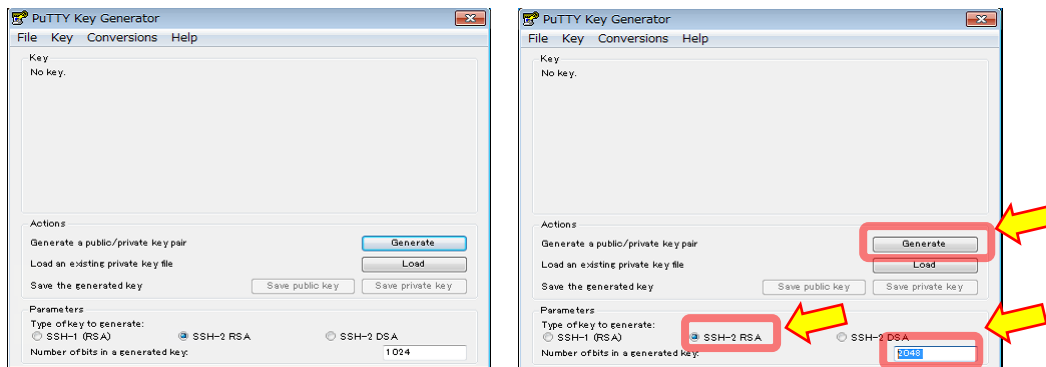
前提②：TortoiseGit をインストールしている。

前提③：サーバー側で SSH-2 という方式の通信サービスが稼働している。(RSA 認証方式が使用可) ※SSH 通信に対応したサーバーなら、大抵がこの状態と思ってよい。

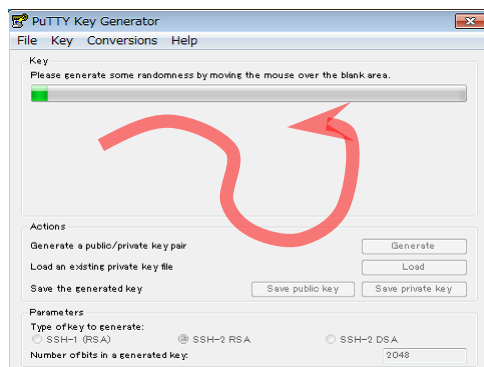
手順①： スタートメニューから「TortoiseGit」→「Puttygen」を実行。



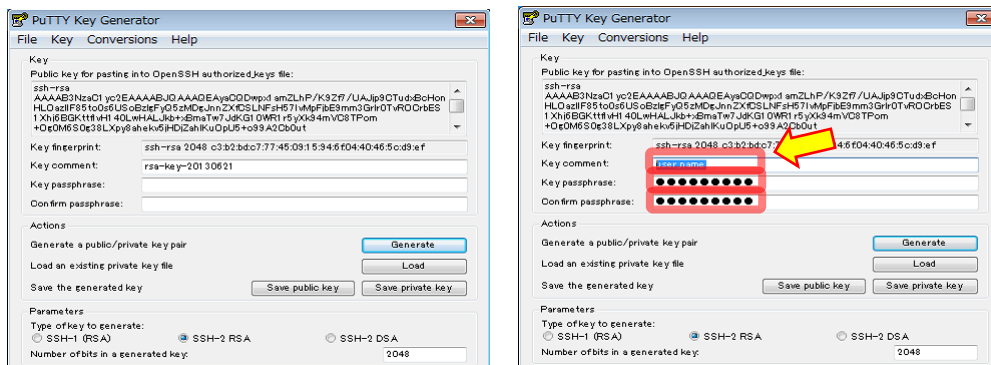
手順② : PuTTY Key Generator が起動したら、まず、「Parameters」にて、「SSH-2 RSA」を選択し、「Number of bits in a generated key」に「2048」を入力し、「Generate」ボタンを押す。



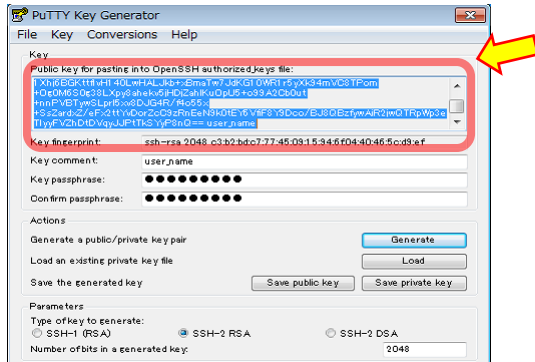
手順③ : ウィンドウの上でマウスカーソルをぐるぐる動かすとプログレスバーが進むので、完了するまで動かし続ける。(これによって不規則な鍵を生成する)



手順④ : 鍵の生成が完了したら、「key-comment」欄にユーザーIDを入力し、「Key passphrase」欄と「Confirm passphrase」欄に同じパスワードを入力する。



手順⑤ : 「Public key for pasting into OpenSSH authorized\_keys file」欄の文字列が、サーバー上に配置する公開鍵。これを全文選択してメモ帳などにコピーし、「(ユーザーID) .pub」といったファイル名で保存する。

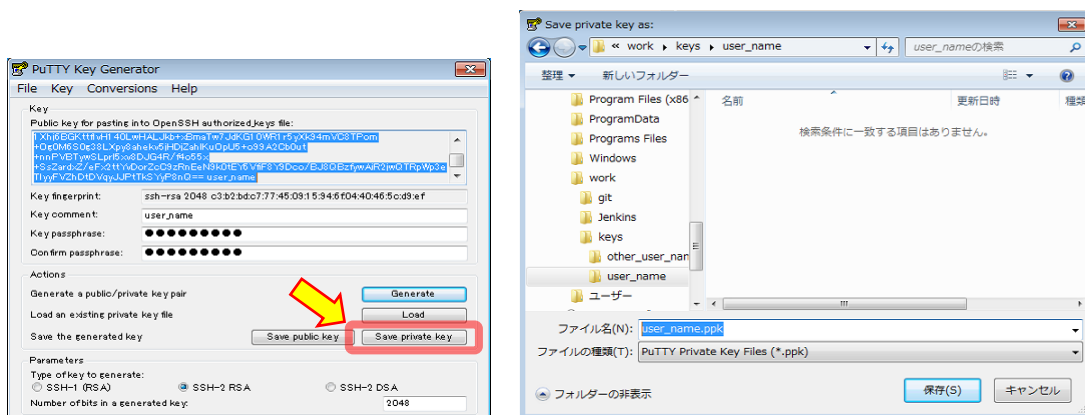


※ [Save public key] ボタンで保存されるファイルとは形式が違うので注意。多くの SSH サーバー (Git でよく使用されるサーバー) は、上記の形式 (OpenSSH 形式) で扱われる。途中改行を一切含まない、長い 1 行の文字列である点に注意。

公開鍵 (OpenSSH 形式) の内容例 :

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAwPkbM7r13MfjtK9Lb+hHDp4qLsISxkXV6tbK+dY2tF9tDihq2jqzwj9zWic8L5vOztL7bfaNobf0wTa8mWtPgN6b
58kEZAtWJ4JH5XIB/USP6k7x01YXmDABsnDRm7RwfXfv8BM9/d9lJPXv6bGKxux4BmRZOn+IW8HSboATBAj2fPykFewTAQh9dONcmu500bttbiVkyL3v
aRTj6ofSrYXfVgrdmK+UvUsoWun4DG7Z4LF43hBvUWPJ+K97fa6CnI+Rr1P40DiAfiiP+QtMdzecMkcJ/LenouPMMKZsM/OyMhoZzxJ6lWgXBZ51R2y
xLJGAEOOCvjYPvZCL1D0hw== user_name
```

手順⑥ : [Save private key] ボタンを押し、秘密鍵をローカル PC 上に保存し、以後ファイルをむやみに他者やサーバーにコピーしたり紛失したりしないように各自気を付ける。



※ファイルの拡張子は「.ppk」。秘密鍵にも幾つかの形式があるが、TortoiseGit では、この .ppk の形式 (PuTTY 形式) が使用される。

秘密鍵 (PuTTY 形式) の内容例 :

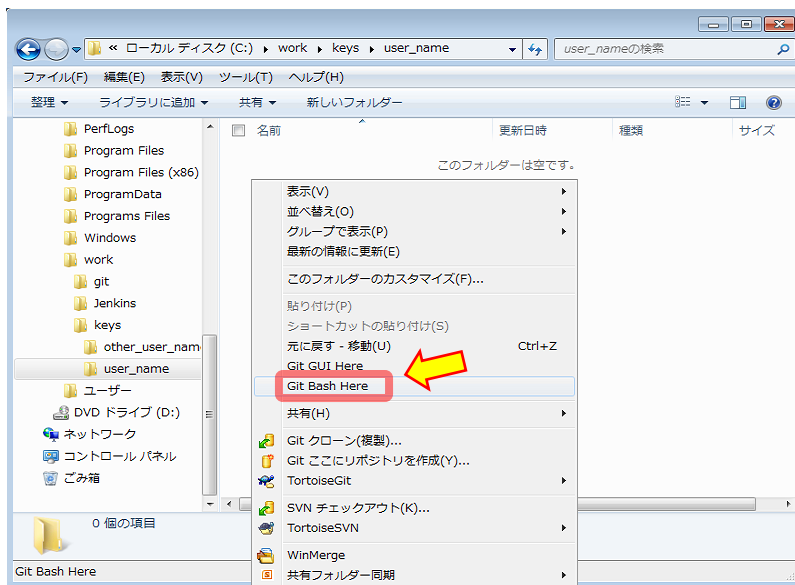
```
PuTTY-User-Key-File-2: ssh-rsa
Encryption: aes256-cbc
Comment: user_name
Public-Lines: 6
AAAAB3NzaC1yc2EAAAABIwAAAQEAwPkbM7r13MfjtK9Lb+hHDp4qLsISxkXV6tbK+dY2tF9tDihq2jqzwj9zWic8L5vOztL7bfaNobf0wTa8mWtPgN6b
58kEZAtWJ4JH5XIB/USP6k7x01YXmDABsnDRm7RwfXfv8BM9/d9lJPXv6bGKxux4BmRZOn+IW8HSboATBAj2fPykFewTAQh9dONcmu500bttbiVkyL3v
aRTj6ofSrYXfVgrdmK+UvUsoWun4DG7Z4LF43hBvUWPJ+K97fa6CnI+Rr1P40DiAfiiP+QtMdzecMkcJ/LenouPMMKZsM/OyMhoZzxJ6lWgXBZ51R2y
xLJGAEOOCvjYPvZCL1D0hw== user_name
Private-Lines: 14
ct6DvZmJ/JPD90suwAoI2VKZA8a1eKYRytONv/9tvdGyRDPp9n+auJ+Yt/i6aZaI
fuSMj iK0WmvrDHW1KN1N46est i+EDLCuSLDpyl4XtrUC//sdZ2ZMj+8jV109fDwy
WhUB2aPa66RTISOD/q01Eo8MKCx/EHH8uvVUEshnVu5VeBThbe5D3TwjWWS6hmCs
Q1IR/yR+m+HKPOMQDfgMj+bLjhfn4Y4lpmBfqi t2P0vF5ZS/Mq8 laVQ/5MhXFEUC
Dvff9Z269p2l/jSbuJbG+EYACfPMJJJZ04220QG3X0019SQgWuZf5x27+hy5tKos
```

```
e06TNMYZoji1hi f66p8WBj1WMh4RjQFXwUX0+nsj1ZJ0Uw9zhrqws458Z1uq5pKM
3SqtyA1EgMM+JK3gEKvJqnic6zz3pWkBDvORX341IOxj2s/al6kgwVChA7H+akq
QPPenoDIOkuwlyp1vHybH9T7uYqSlx9rGbTe23hxmqaHpCV0v2Bc5MVAD8JLUtZk
wQLTs24T+gr3nAwCdrXG9D58Fx1t1UQDvaLh8CDyXG1hmy0iue7nsiZ130TSV7XC
Yyi4CcXIDWPt6mR0f1OrRD2g5A8+5Pu3bFuCACIAJjUYF8xMnm20b76zu14wF+4W
V2ZpyFg8oda/HPvCZ7WAp1Fez/IWT9oRPQvozZcGRH+yiA40x5KD+oJkkLUn2Fmy
UxpAaB/9AyLuKh8rFGRJpkpd1C+41QfbKTBRrZr7Tsr1wfAM1JHk6X8dP+v/nr
NZCr+ut8Y0gTX7nPFw8i f0EM6LtkQGVmla3yKx0z+/EMt8gvYlAmo+uq0Tvv1a65
zztcS2zW2Cp7oBLUojYd01VDKxQtZ/ZNOwoaB7uznsH+xzX5TUa7lID5dKr42zGK
Private-MAC: 0e39ab3db21a808adc8be806fa44bfcdb7db97d9
```

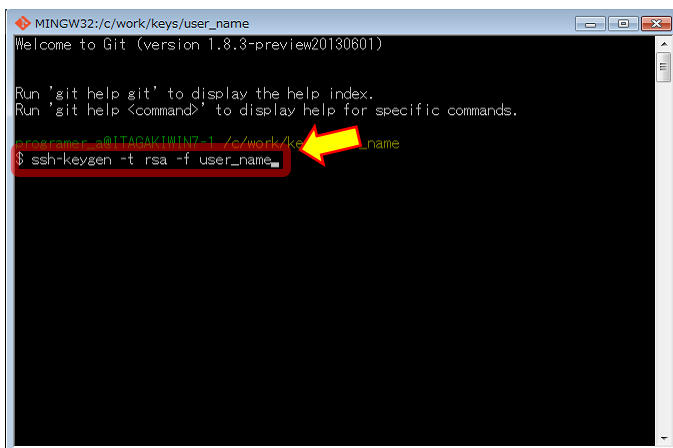
## ■ 秘密鍵と公開鍵の生成方法② : ssh-keygen コマンドを使用する方法

前提 : msysGit がインストールされている。(Linux 上で鍵を生成する場合の手順もほぼ同様)

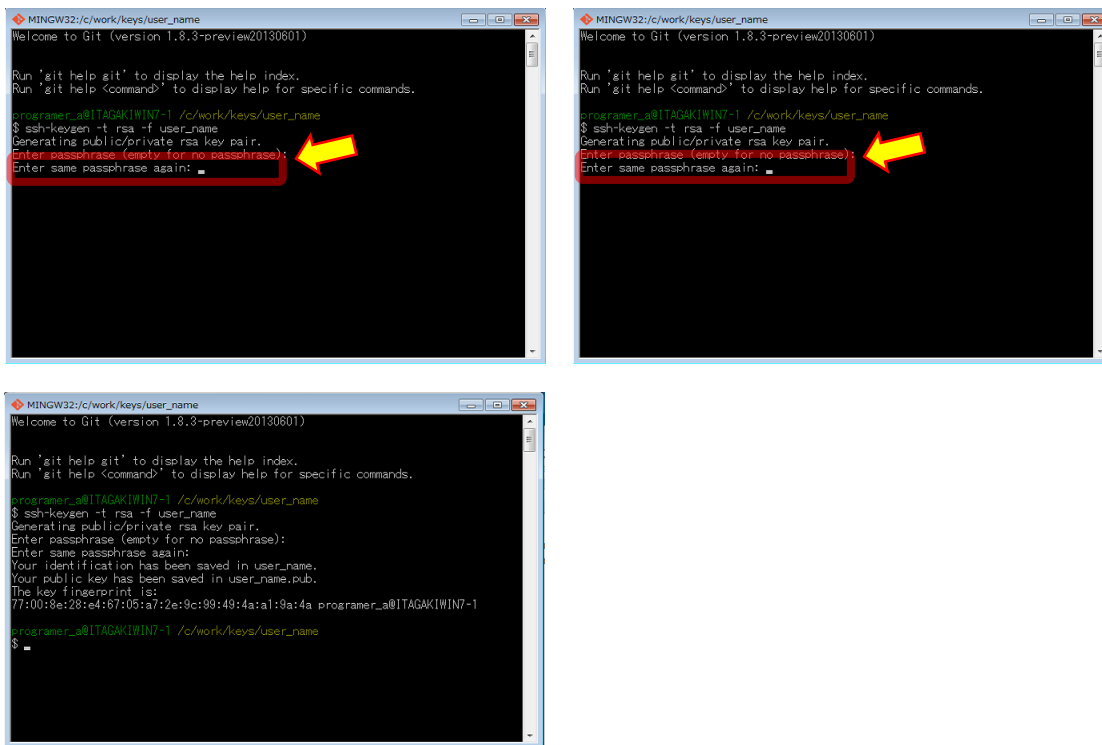
手順① : エクスプローラーにて、鍵を生成したいフォルダ上で右クリックし、コンテキストメニューから「Git Bash Here」を実行する。



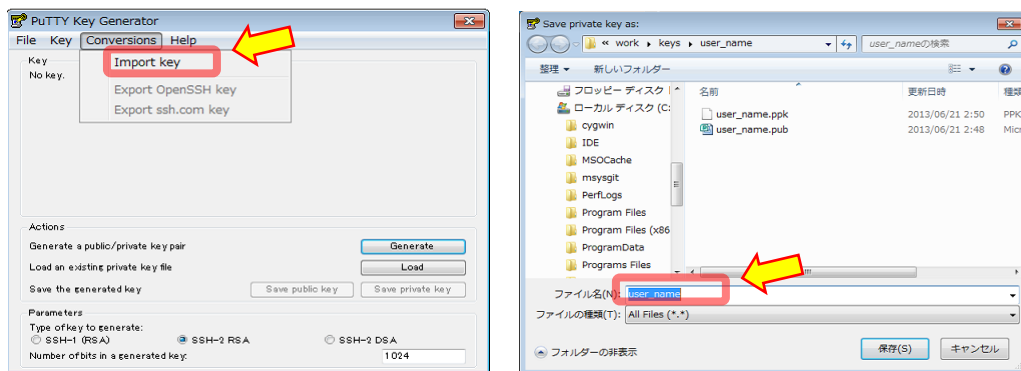
手順② : シェル (コマンドプロンプト) が起動したら、`ssh-keygen -t rsa -f (ユーザーID)` というコマンドを実行する。



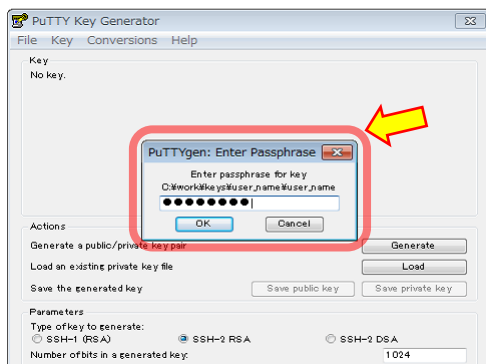
手順③： 続いてパスフレーズを 2 回入力すると、同フォルダにユーザーID をファイル名とした公開鍵ファイルと秘密鍵ファイルが生成される。(公開鍵ファイルは拡張子が .pub で、秘密鍵ファイルは拡張子なし)



手順④： ここで生成された秘密鍵は OpenSSH 形式の秘密鍵である為、TortoiseGit で使えるように、PuTTY 形式のファイル (.ppk ファイル) に変換する必要がある。その為に、まず、pputtygen を起動し、作成した秘密鍵を [Conversions] → [Import Key] メニューを実行してインポートする。

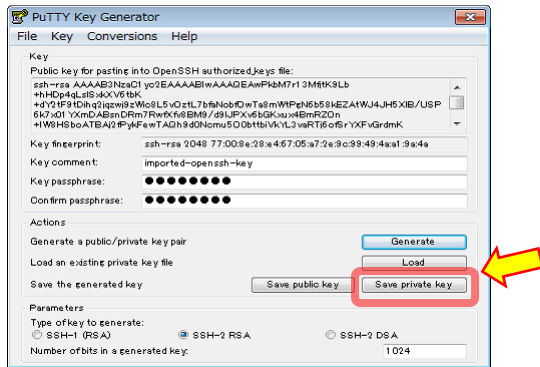


手順⑤： インポート時にパスフレーズの確認が求められるので、パスフレーズを入力する。





手順⑥： インポートが完了したら、[Save private key] ボタンを押し、PuTTY 形式の秘密鍵ファイルを保存する。

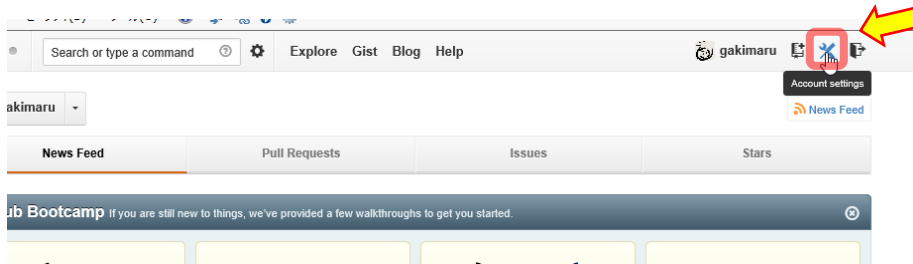


## ■ 公開鍵をサーバーに送る方法（GitHub の例）

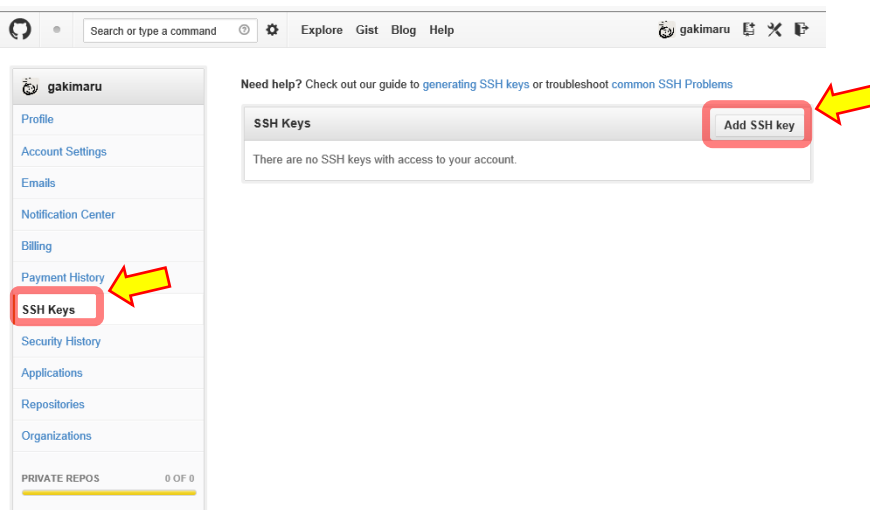
前提①： 既に GitHub のアカウントを持っている。

前提②： 事前に GitHub 上でリポジトリを作成している。

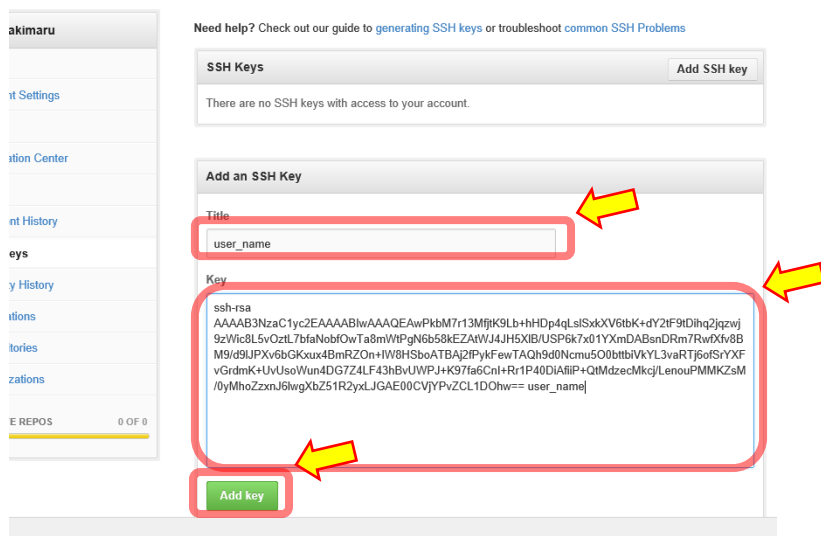
手順①： GitHub の Web サイト <https://github.com/> にアクセスし、サインインしたら、「Account Settings」を実行する。



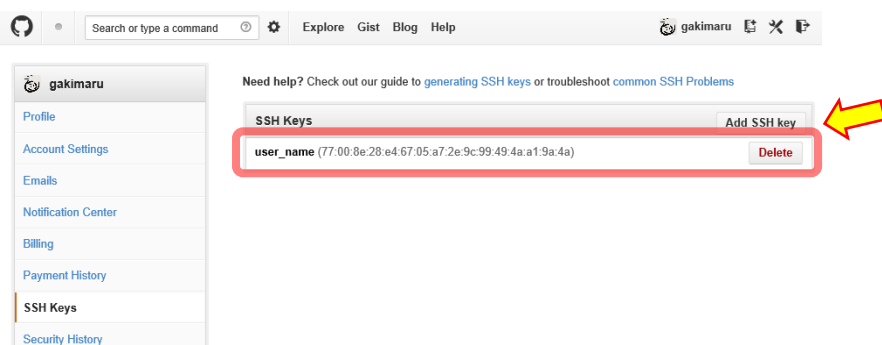
手順②： アカウント設定画面を開いたら、左側のメニューから「SSH Keys」を選択し、右側の「SSH Keys」欄右上の [Add SSH Key] ボタンを押す。



手順③ : 「Title」欄に鍵の見出しを入力し、「Key」欄に公開鍵 (.pub ファイル) の内容をコピー (この公開鍵の形式は、先述の、途中に一切の改行がない一行のテキストで表される形式)、入力が完了したら、画面下側の [Add key] ボタンを押す。

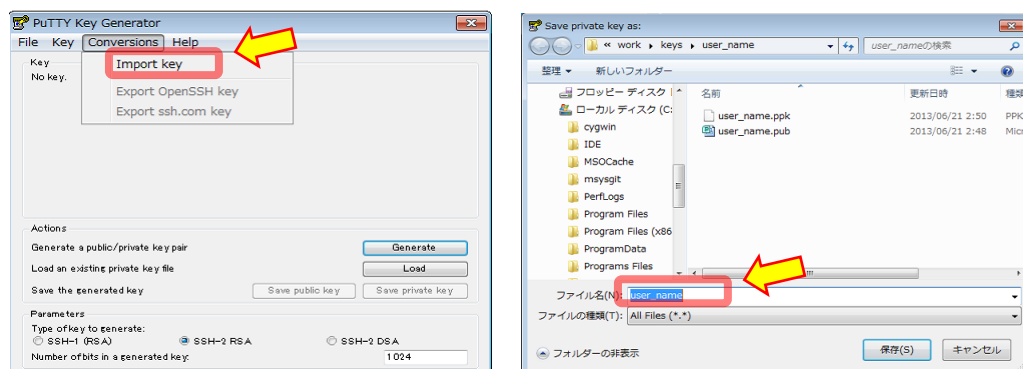


手順④ : 完了すると、追加された鍵の情報がリストに表示されるようになる。

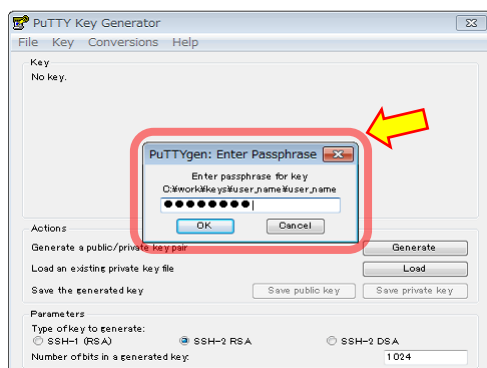


## ■ 秘密鍵のパスフレーズを変更する方法 : puttygen を使用する方法

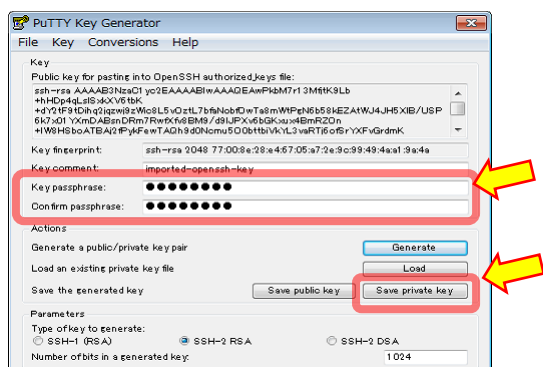
手順① : まず、pputtygen を起動し、既存の秘密鍵を [Conversions] → [Import Key] メニューを実行してインポートする。(PuTTY 形式でも OpenSSH 形式でも良い。)



手順②： インポート時にパスフレーズの確認が求められるので、パスフレーズを入力する。



手順③： インポートが完了したら、改めてパスフレーズを入力し直した上で、[Save private key] ボタンを押し、PuTTY 形式の秘密鍵ファイルを保存する。(公開鍵は変わっていないので、保存し直す必要はない。)



---

## TortoiseGit による SSH 通信用秘密鍵・公開鍵生成手順

---

以 上