TortoiseGitによる

SSH通信用秘密鍵・公開鍵生成手順

2013/6/20

* **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。

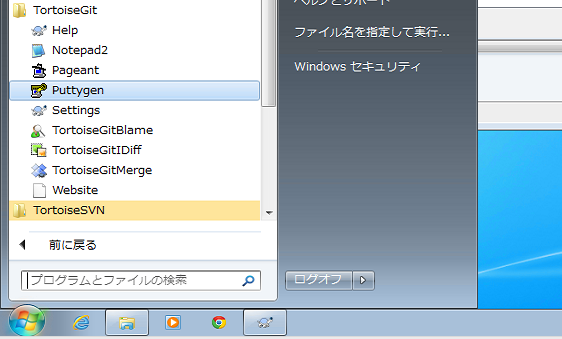
* **秘密鍵と公開鍵の生成方法①：puttygenを使用する方法**

前提①：Windows上で鍵を生成する。

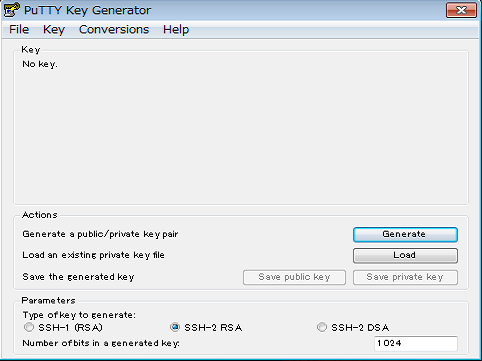
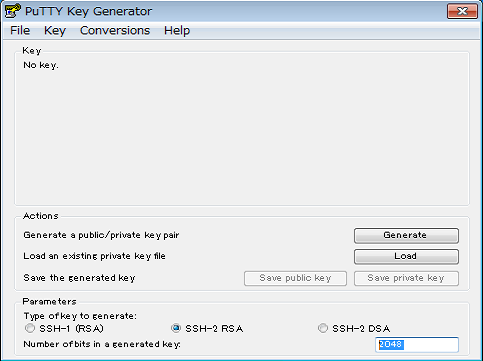
前提②：TortoiseGitをインストールしている。

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認証方式が使用さ可能）　※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。

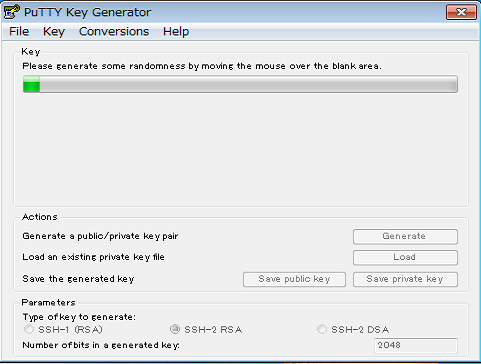
**手順①：** スタートメニューから「TortoiseGit」→「Puttygen」を実行。



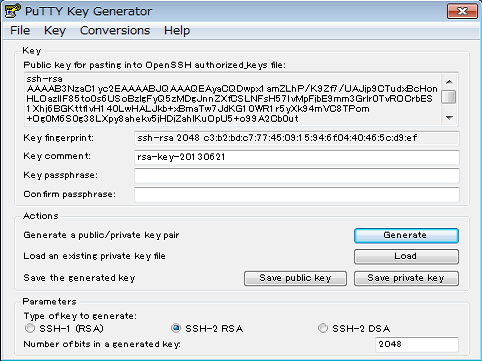
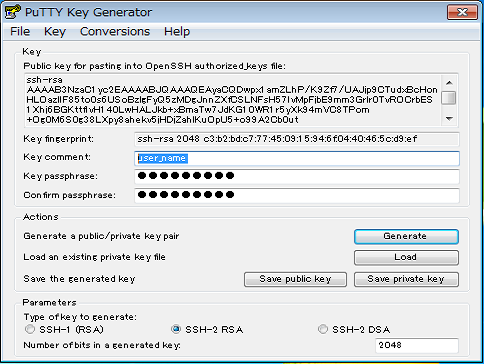
**手順②：** PuTTY Key Generator が起動したら、まず、「Parameters」にて、「SSH-2 RSA」を選択し、「Number of bits in a generated key」に「2048」を入力し、［Generate］ボタンを押す。

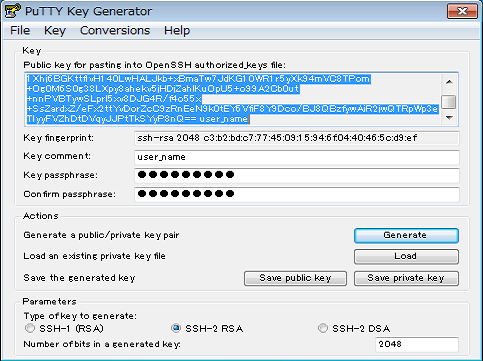
**手順③：** ウインドウの上でマウスカーソルをぐるぐる動かすとプログレスバーが進むので、完了するまで動かし続ける。（これによって不規則な鍵を生成する）

****

**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」欄に同じパスフレーズを入力する。

****　 

**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ペし、「（ユーザーID）.pub」といったファイル名で保存する。

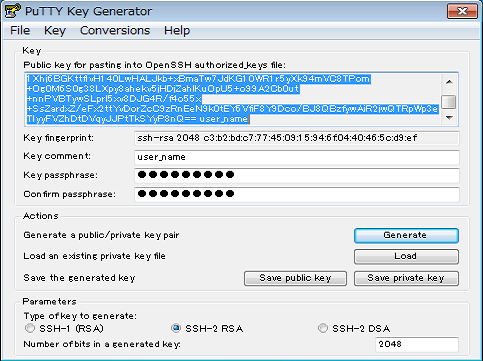
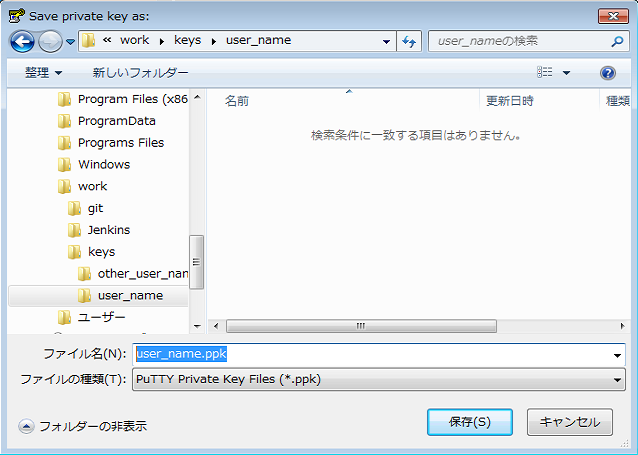
****

* ［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+hHDp4qLslSxkXV6tbK+dY2tF9tDihq2jqzwj9zWic8L5vOztL7bfaNobfOwTa8mWtPgN6b58kEZAtWJ4JH5XlB/USP6k7x01YXmDABsnDRm7RwfXfv8BM9/d9lJPXv6bGKxux4BmRZOn+IW8HSboATBAj2fPykFewTAQh9d0Ncmu5O0bttbiVkYL3vaRTj6ofSrYXFvGrdmK+UvUsoWun4DG7Z4LF43hBvUWPJ+K97fa6CnI+Rr1P40DiAfiiP+QtMdzecMkcj/LenouPMMKZsM/0yMhoZzxnJ6lwgXbZ51R2yxLJGAE00CVjYPvZCL1DOhw== 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

AAAAB3NzaC1yc2EAAAABJQAAAQEAyaCQDwpx1amZLhP/K9Zf7/UAJjp9CTudxBcH

onHLOazlIF85to0s6USoBzlgFyQ5zMDgJnnZXfCSLNFsH57IvMpFjbE9mm3Grlr0

TvROCrbES1Xhj6BGKttfIvH140LwHALJkb+xBmaTw7JdKG10WR1r5yXk94mVC8TP

om+Og0M6S0g38LXpy8ahekv5jHDjZahlKuOpU5+o99A2Cb0ut+nnPVBTywSLprl5

xv8DJG4R/f4o55x+SsZardxZ/eFx2ttYvDorZcC9zRnEeN9k0tEY6VfiF8Y9Dco/

BJ8QBzfywAiR2jwQTRpWp3eTIyyFVZhDtDVqyJJPtTkSYyP8nQ==

Private-Lines: 14

ct6DvZmJ/JPd90suwAoI2VKZA8a1eKYRyt0Nv/9tvdGyRDPp9n+aUj+Yt/i6aZal

fuSMjik0WMvrDHW1KN1N46esti+EDLCUsLDpyl4XtrUC//sdZ2ZMj+8jV109fDwy

WhUB2aPa66RTlS0D/qO1Eo8MKCx/EHH8uvVUEshnVu5VeBThbe5D3TwjWWS6hmCs

QIIR/yR+m+HKP0MqDfgMJ+bLjhfn4Y4lpmBFqit2P0vF5ZS/Mq8laVQ/5MhXfEUC

Dvff9ZZ69p2l/sJbuJbG+EYACfPMJJJZO4220QG3X0O19SQgWuZf5x27+hy5tKos

e06TNMYZoji1hif66p8WBJ1WMh4RjQFXwUX0+nsjlZJ0Uw9zhrqws458ZIuq5pKM

3SqtyA1EgMM+jK3gEKvJqnic6zz3pWkBTDvORX34IlOxj2s/al6kgwVChA7H+akq

QPPenoDI0kuwlypIvHybH9T7uYqSlx9rGbTe23hxmqAhpCV0v2Bc5MVAD8JLUtZk

wQLTs24T+gr3nAwCdrXG9D58Fx1t1UQDvaLh8CDyXGlhmy0iue7nsiZI30TSV7XC

Yyi4CcXlDWPt6mR0fI0rRD2g5A8+5Pu3bFuCAClAJJUYF8xMnm2Ob76zul4wF+4W

V2ZpyFg8oda/HPvCZ7WApIFez/IWT9oRPQvozZcGRH+yiA4Ox5KD+oJkkLUn2Fmy

UxpAaB/9AyLuKh8rFGRJpkqpdIC+41QfbKTBRRzR7Tsr1wfAM1JHkB6X8dP+v/nr

NZCr+ut8Y0gTX7nPFw8ifOEM6LtkQGVmla3ykxOz+/EMt8gvYlAmo+uq0Tvv1a65

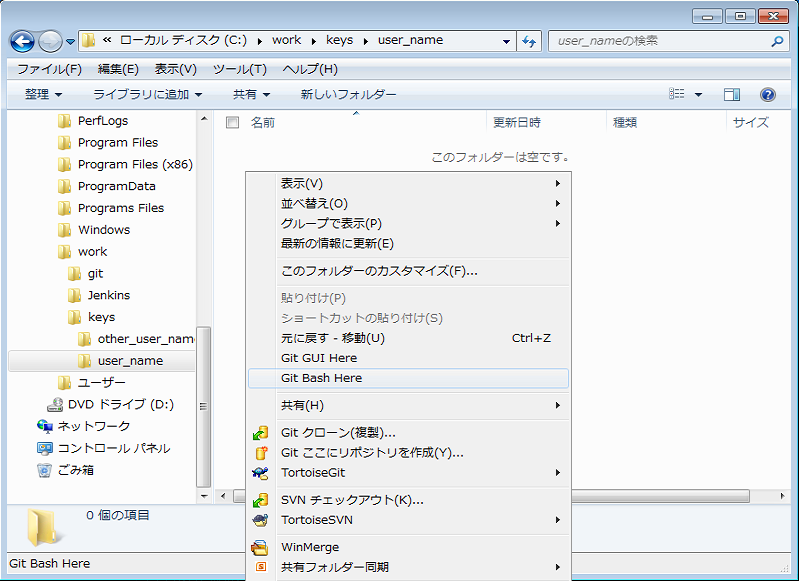
zztcS2zW2Cp7oBLUojYd0lVDKxQtZ/ZNOwoaB7uznsH+xzX5TUa7lID5dKr42zGK

Private-MAC: 0e39ab3db21a808adc8be806fa44bfcbd7db97d9

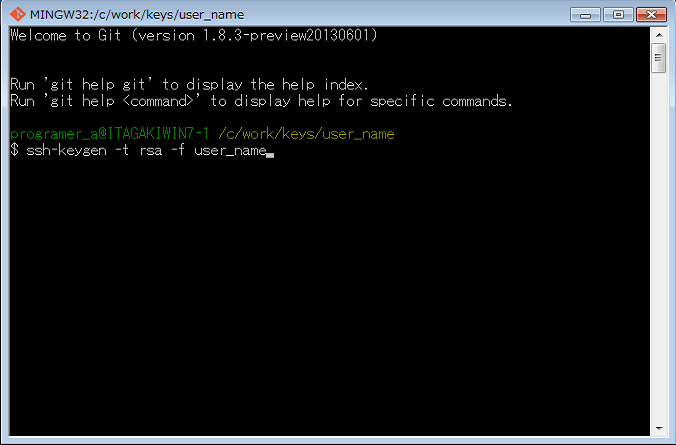
* **秘密鍵と公開鍵の生成方法②：ssh-keygenコマンドを使用する方法**

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）

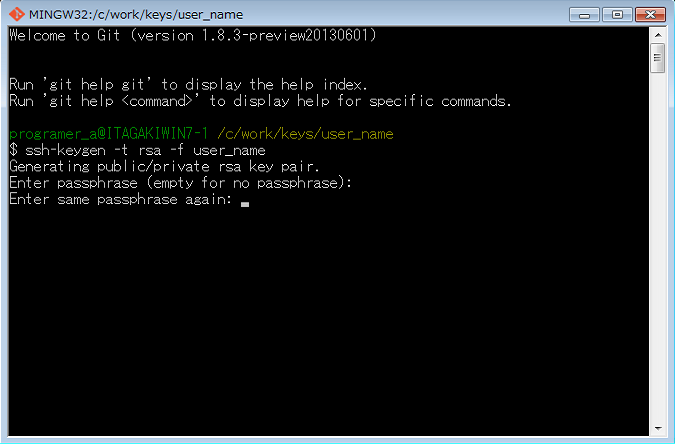
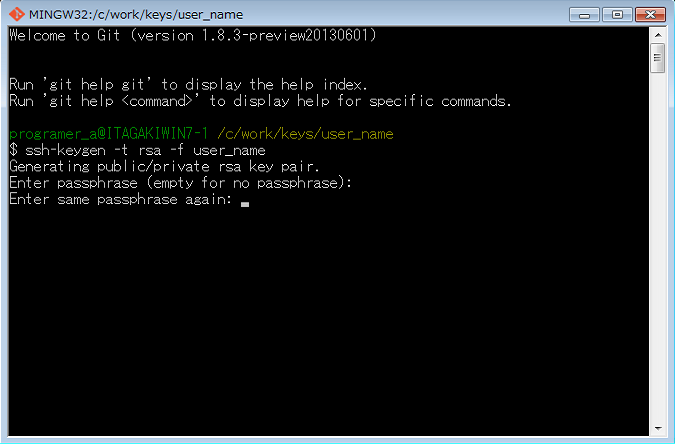
**手順①：** エクスプローラーにて、鍵を生成したいフォルダ上で右クリックし、コンテキストメニューから「Git Bash Here」を実行する。

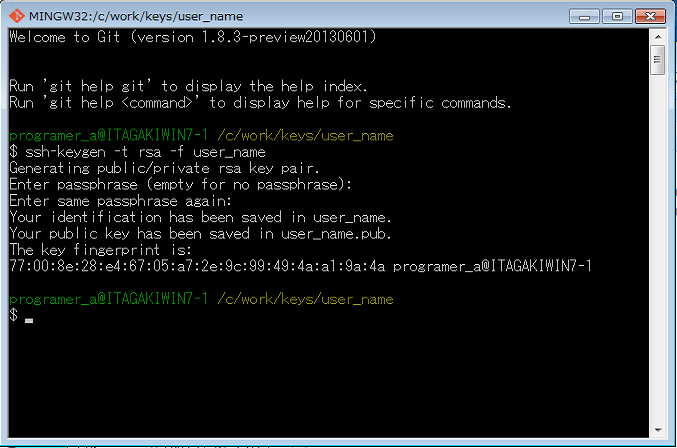
****

**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。

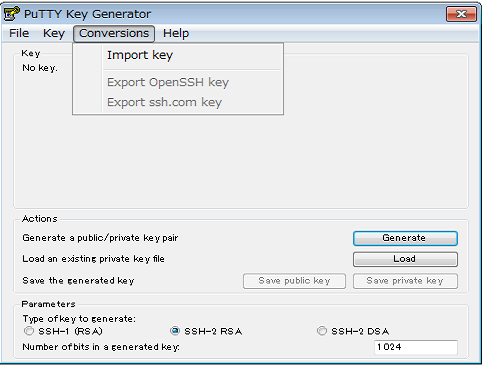
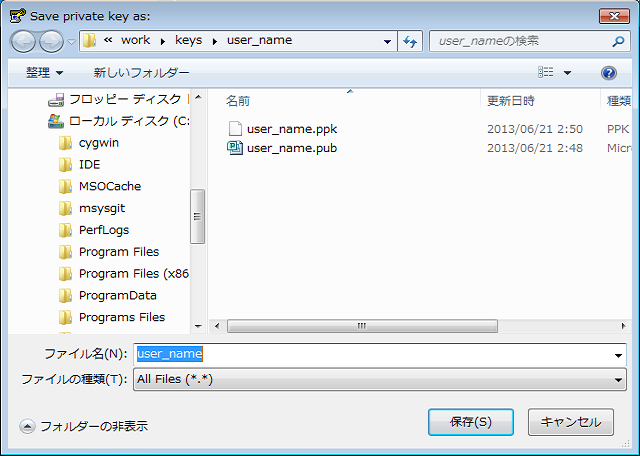


**手順③：** 続いてパスフレーズを2回入力すると、同フォルダにユーザーIDをファイル名とした公開鍵ファイルと秘密鍵ファイルが生成される。（公開鍵ファイルは拡張子が .pub で、秘密鍵ファイルは拡張子なし）

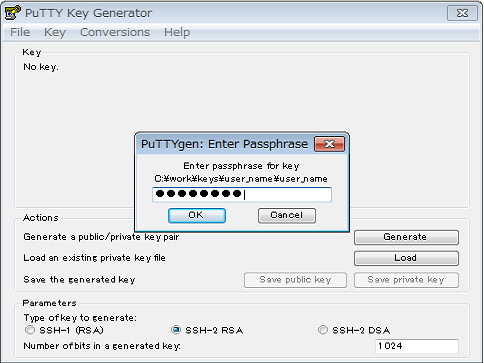
　



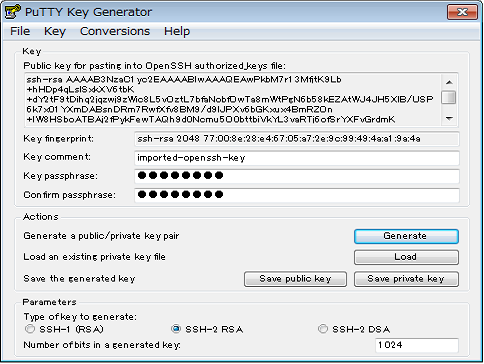
**手順④：** ここで生成された秘密鍵はOpenSSH形式の秘密鍵である為、TortoiseGitで使えるように、PuTTY形式のファイル（.ppkファイル）に変換する必要がある。その為に、まず、pputtygenを起動し、作成した秘密鍵を［Conversions］→［Import Key］メニューを実行してインポートする。

**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。

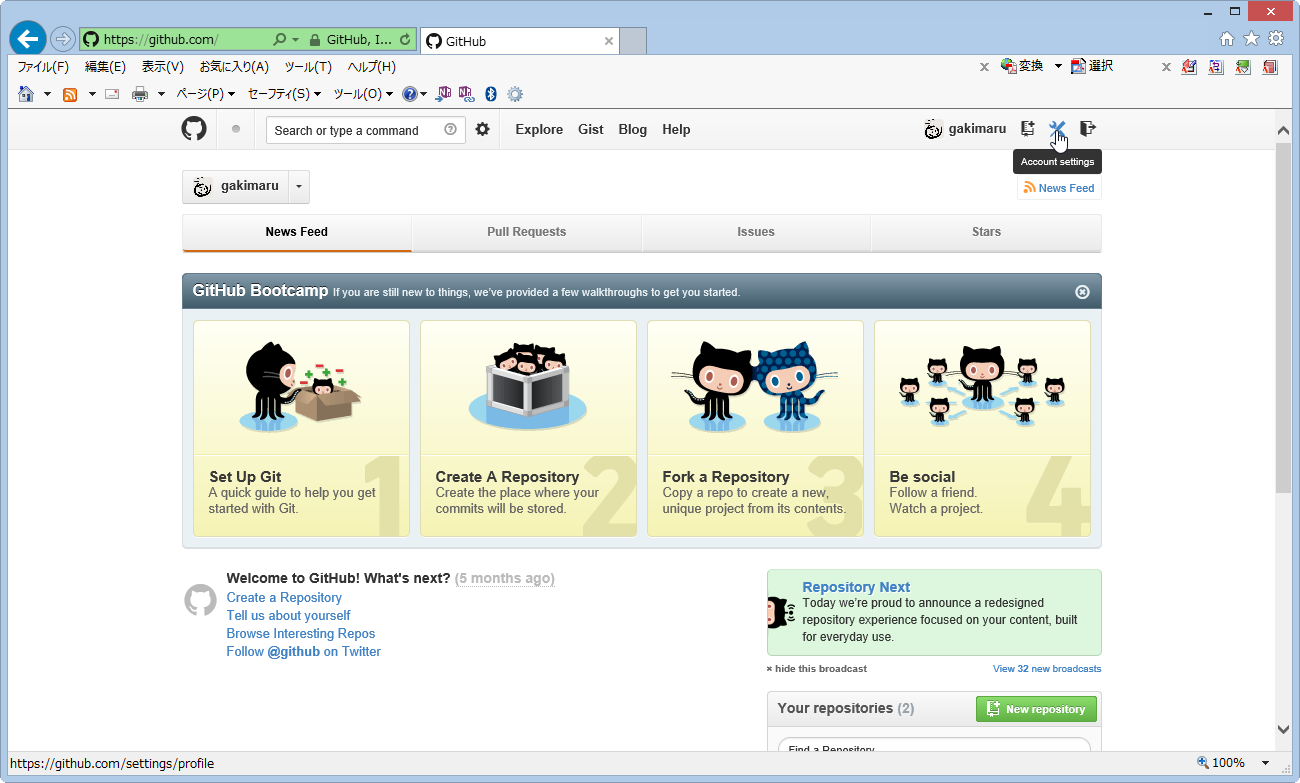


* **公開鍵をサーバーに送る方法（GitHubの例）**

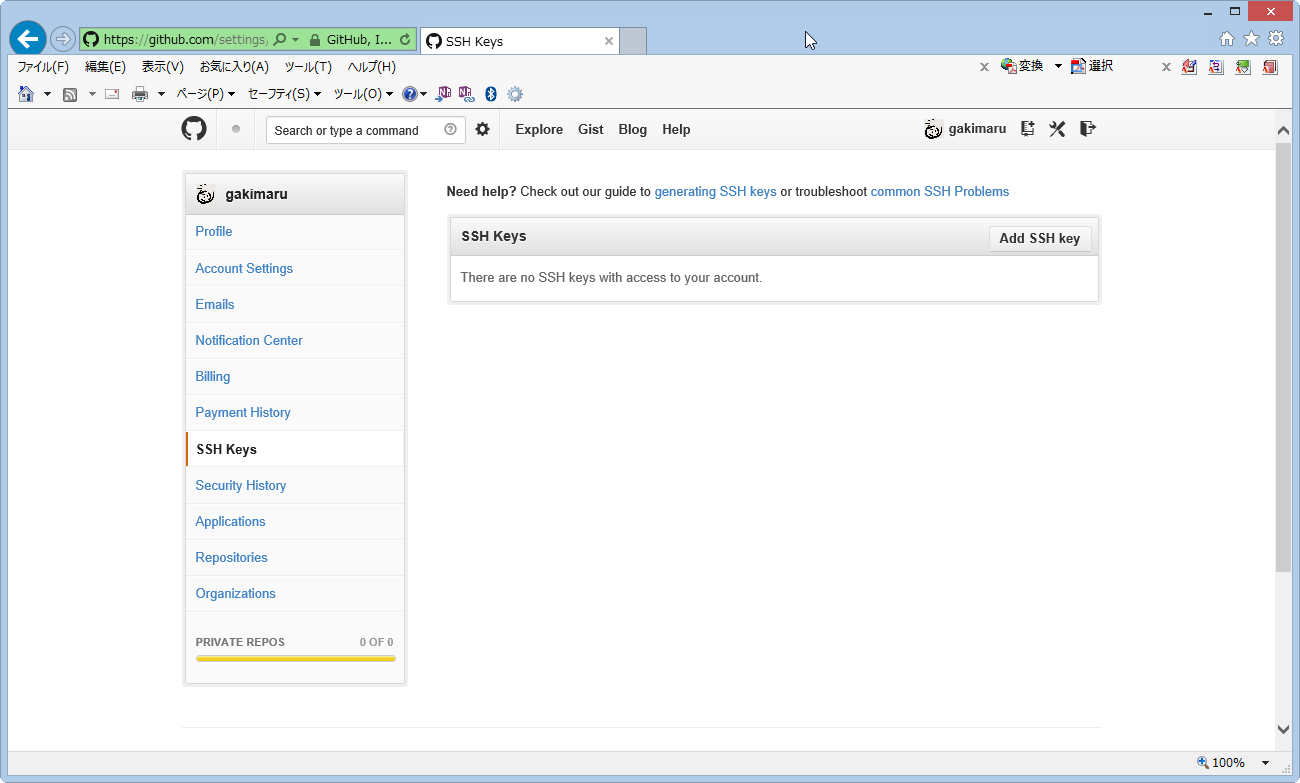
前提①： 既にGitHubのアカウントを持っている。

前提②： 事前にGitHub上でリポジトリを作成している。

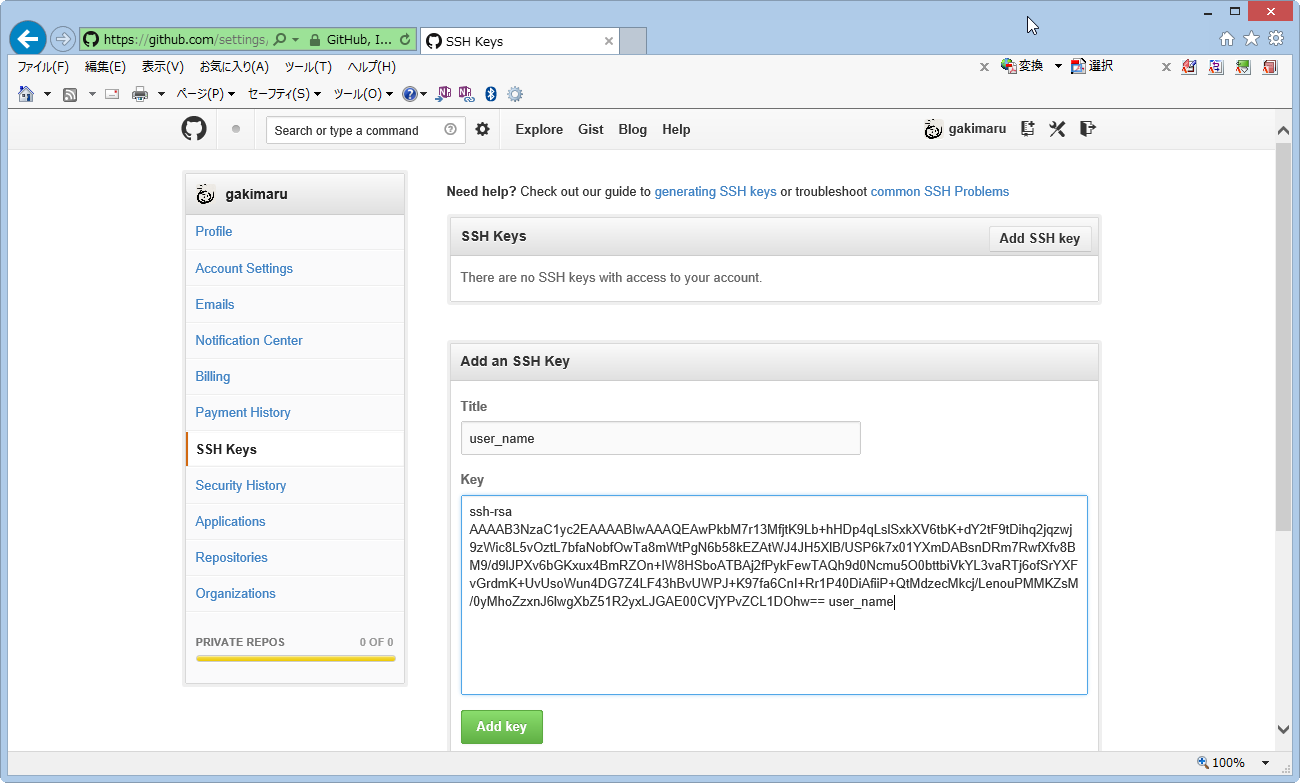
**手順①：** GitHubのWebサイト <https://github.com/> にアクセスし、サインインしたら、「Account Settings」を実行する。



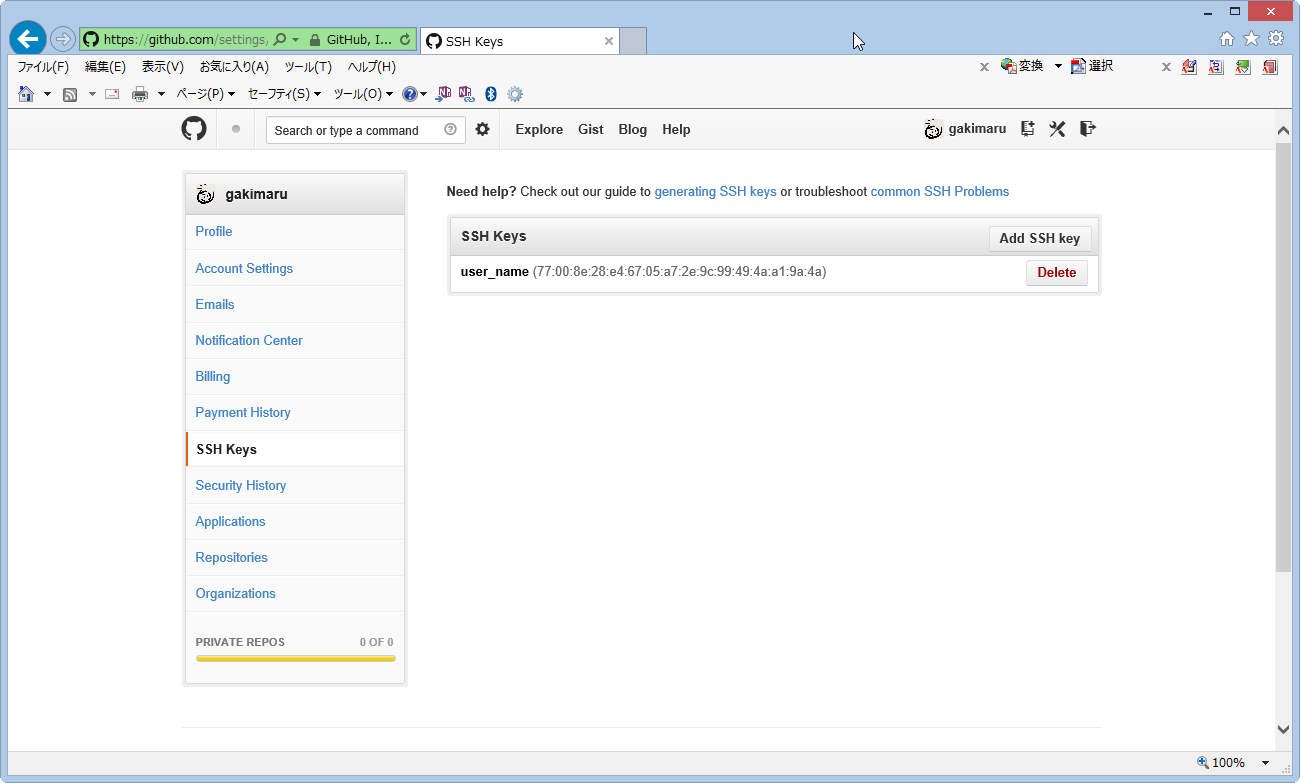
**手順②：** アカウント設定画面を開いたら、左側のメニューから「SSH Keys」を選択し、右側の「SSH Keys」欄右上の［Add SSH Key］ボタンを押す。



**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）の内容をコピペし（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 yey］ボタンを押す。

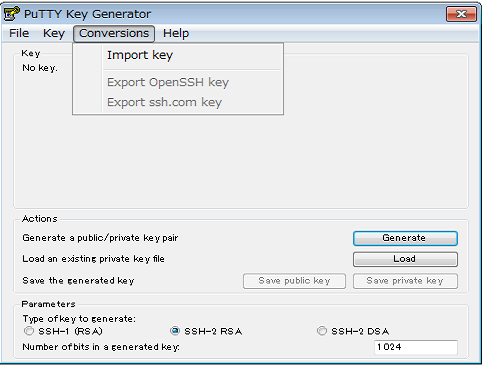
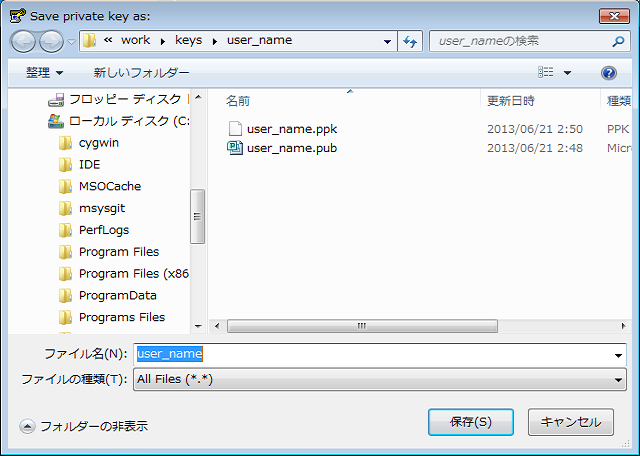


**手順④：** 完了すると、追加された鍵の情報がリストに表示されるようになる。

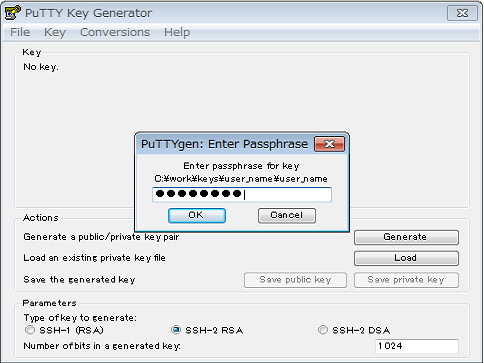


* **秘密鍵のパスフレーズを変更する方法：puttygenを使用する方法**

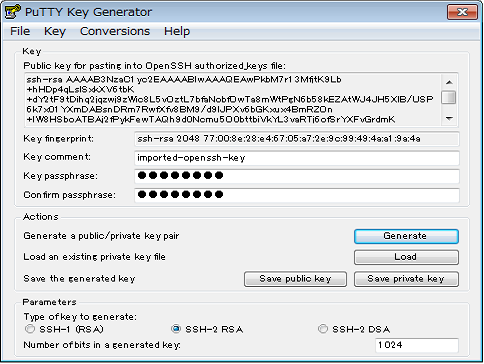
**手順①：** まず、pputtygenを起動し、既存の秘密鍵を［Conversions］→［Import Key］メニューを実行してインポートする。（PuTTY形式でもOpenSSH形式でも良い。）

**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。）



以上