[Generieren eines neuen SSH-Schlüssels und Hinzufügen des Schlüssels zum ssh-agent - GitHub-Dokumentation](https://docs.github.com/de/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent)

**Generieren eines neuen SSH-Schlüssels und Hinzufügen des Schlüssels zum ssh-agent**

Wenn du geprüft hast, ob SSH-Schlüssel vorhanden sind, kannst du einen neuen SSH-Schlüssel für die Authentifizierung generieren und zum SSH-Agent hinzufügen.

**Platform navigation**

* [Mac](https://docs.github.com/de/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent?platform=mac)
* [**Windows**](https://docs.github.com/de/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent?platform=windows)
* [Linux](https://docs.github.com/de/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent?platform=linux)

[**Informationen zu SSH-Schlüsselpassphrasen**](https://docs.github.com/de/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent#about-ssh-key-passphrases)

Über SSH (Secure Shell Protocol) kannst du auf Daten in Repositorys in GitHub.com Daten in diese Repositorys schreiben. Wenn du über SSH eine Verbindung herstellst, authentifizierst du dich mit einer privaten Schlüsseldatei auf deinem lokalen Computer. Weitere Informationen findest du unter [Informationen zur SSH](https://docs.github.com/de/authentication/connecting-to-github-with-ssh/about-ssh).

Wenn du einen SSH-Schlüssel generierst, kannst du eine Passphrase hinzufügen, um ihn noch besser zu schützen. Wenn du den Schlüssel verwendest, muss du die Passphrase eingeben. Wenn dein Schlüssel eine Passphrase umfasst und du diese nicht jedes Mal eingeben möchtest, wenn du den Schlüssel verwendest, kannst du diesen dem SSH-Agent hinzufügen. Der SSH-Agent verwaltet deine SSH-Schlüssel und speichert deine Passphrase.

Wenn du noch keinen SSH-Schlüssel hast, musst du einen neuen SSH-Schlüssel generieren, der für die Authentifizierung verwendet werden soll. Wenn du nicht sicher bist, ob du bereits über einen SSH-Schlüssel verfügst, kannst du nach vorhandenen Schlüsseln suchen. Weitere Informationen findest du unter [Auf vorhandene SSH-Schlüssel prüfen](https://docs.github.com/de/authentication/connecting-to-github-with-ssh/checking-for-existing-ssh-keys).

Wenn du einen Hardwaresicherheitsschlüssel zum Authentifizieren von GitHub verwenden möchtest, musst du einen neuen SSH-Schlüssel für deinen Hardwaresicherheitsschlüssel generieren. Du musst deinen Hardwaresicherheitsschlüssel mit deinem Computer verbinden, wenn du dich mit dem Schlüsselpaar authentifizierst. Weitere Informationen findest du unter [Versionshinweise zu OpenSSH 8.2](https://www.openssh.com/txt/release-8.2).

[**Einen neuen SSH-Schlüssel erzeugen**](https://docs.github.com/de/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent#generating-a-new-ssh-key)

Du kannst einen neuen SSH-Schlüssel auf deinem lokalen Computer generieren. Nachdem du den Schlüssel generiert hast, kannst du den öffentlichen Schlüssel deinem Konto auf GitHub.com hinzufügen, um die Authentifizierung für Git-Vorgänge über SSH zu aktivieren.

**Hinweis:** GitHub verbessert die Sicherheit, indem ältere, unsichere Schlüsseltypen am 15. März 2022 gelöscht werden.

Ab diesem Datum werden DSA-Schlüssel (ssh-dss) nicht mehr unterstützt. Du kannst deinem persönlichen Konto keine neuen DSA-Schlüssel für GitHub.com hinzufügen.

RSA-Schlüssel (ssh-rsa) mit einem valid\_after vor dem 2. November 2021 können weiterhin einen Signaturalgorithmus verwenden. RSA-Schlüssel, die nach diesem Datum generiert wurden, müssen einen SHA-2-Signaturalgorithmus verwenden. Einige ältere Clients müssen möglicherweise aktualisiert werden, um SHA-2-Signaturen zu verwenden.

1. Öffne Git Bash.
2. Füge den folgenden Text ein, und ergänze dabei deine GitHub-E-Mail-Adresse.
3. ssh-keygen -t ed25519 -C "your\_email@example.com"

**Hinweis:** Wenn du ein Legacy-System verwendest, das den Ed25519-Algorithmus nicht unterstützt, verwende:

ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"

Dadurch wird ein neuer SSH-Schlüssel erzeugt und die angegebene E-Mail-Adresse als Kennzeichnung verwendet.

> Generating public/private ALGORITHM key pair.

Wenn du aufgefordert wirst, eine Datei einzugeben, in der der Schlüssel gespeichert werden soll, kannst du die **EINGABETASTE** drücken, um den Standarddateispeicherort zu akzeptieren. Hinweis: Wenn du zuvor SSH-Schlüssel erstellt hast, wirst du möglicherweise von ssh-keygen aufgefordert, einen anderen Schlüssel neu zu schreiben. In diesem Fall wird empfohlen, einen SSH-Schlüssel mit benutzerdefiniertem Namen zu erstellen. Gib hierzu den Standarddateispeicherort ein, und ersetze id\_ssh\_keyname durch deinen benutzerdefinierten Schlüsselnamen.

> Enter a file in which to save the key (/c/Users/YOU/.ssh/id\_ALGORITHM):[Press enter]

1. Gib an der Eingabeaufforderung eine sichere Passphrase ein. Weitere Informationen findest du unter [SSH-Schlüssel-Passphrasen verwenden](https://docs.github.com/de/authentication/connecting-to-github-with-ssh/working-with-ssh-key-passphrases).
2. > Enter passphrase (empty for no passphrase): [Type a passphrase]
3. > Enter same passphrase again: [Type passphrase again]

[**Deinen SSH-Schlüssel zum SSH-Agenten hinzufügen**](https://docs.github.com/de/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent#adding-your-ssh-key-to-the-ssh-agent)

Bevor du dem SSH-Agent einen neuen SSH-Schlüssel hinzufügst, um deine Schlüssel zu verwalten, solltest du nach vorhandenen SSH-Schlüsseln überprüft und einen neuen SSH-Schlüssel generiert haben.

Wenn du [GitHub Desktop](https://desktop.github.com/) installiert hast, kannst du dies verwenden, um Repositorys zu klonen und nicht mit SSH-Schlüsseln arbeiten zu müssen.

1. Vergewissern Sie sich in einem neuen *Terminalfenster mit Administratorrechten (PowerShell oder CMD)*, dass der ssh-Agent ausgeführt wird. Sie können die Anleitung für „Automatisches Starten von ssh-agent“ unter „[SSH-Schlüssel-Passphrasen verwenden](https://docs.github.com/de/articles/working-with-ssh-key-passphrases)“ verwenden oder diese Komponente manuell starten:
2. # start the ssh-agent in the background
3. Get-Service -Name ssh-agent | Set-Service -StartupType Manual
4. Start-Service ssh-agent
5. Fügen Sie in einem Terminalfenster ohne höhere Rechte Ihren privaten SSH-Schlüssel zum ssh-agent hinzu. Wenn du deinen Schlüssel unter einem anderen Namen erstellt hast oder einen vorhandenen Schlüssel hinzufügst, der einen anderen Namen aufweist, ersetze *id\_ed25519* im Befehl durch den Namen der Datei deines privaten Schlüssels.
6. ssh-add C:\Users\YOU/.ssh/id\_ed25519
7. Füge deinem Konto auf GitHub den öffentlichen SSH-Schlüssel hinzu. Weitere Informationen findest du unter [Einen neuen SSH-Schlüssel zum GitHub-Konto hinzufügen](https://docs.github.com/de/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account).

[**Generieren eines neuen SSH-Schlüssels für einen Hardwaresicherheitsschlüssel**](https://docs.github.com/de/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent#generating-a-new-ssh-key-for-a-hardware-security-key)

Wenn du macOS oder Linux verwendest, musst du möglicherweise deinen SSH-Client aktualisieren oder einen neuen SSH-Client installieren, bevor du einen neuen SSH-Schlüssel generierst. Weitere Informationen findest du unter [Fehler: Unbekannter Schlüsseltyp](https://docs.github.com/de/authentication/troubleshooting-ssh/error-unknown-key-type).

1. Füge deinem Computer deinen Hardwaresicherheitsschlüssel hinzu.
2. Öffne Git Bash.
3. Füge den folgenden Text ein und ersetze ihn in der E-Mail-Adresse deines Kontos auf GitHub.
4. ssh-keygen -t ed25519-sk -C "YOUR\_EMAIL"

**Hinweis:** Wenn der Befehl fehlschlägt und du den Fehler invalid format oder feature not supported, erhältst, verwende möglicherweise einen Hardwaresicherheitsschlüssel, der den Ed25519-Algorithmus nicht unterstützt. Gib stattdessen den folgenden Befehl ein.

ssh-keygen -t ecdsa-sk -C "your\_email@example.com"

1. Wenn du dazu aufgefordert wirst, drücke die Schaltfläche auf deinem Hardwaresicherheitsschlüssel. 1. Wenn du aufgefordert wirst, „Eine Datei einzugeben, in der die Taste gespeichert werden soll", drücke EINGABE, um den Standarddateispeicherort zu akzeptieren.

> Enter a file in which to save the key (/c/Users/YOU/.ssh/id\_ed25519\_sk):[Press enter]

1. Wenn du aufgefordert wirst, eine Passphrase einzugeben, drücke **Eingeben**.
2. > Enter passphrase (empty for no passphrase): [Type a passphrase]
3. > Enter same passphrase again: [Type passphrase again]
4. Füge deinem Konto auf GitHub den öffentlichen SSH-Schlüssel hinzu. Weitere Informationen findest du unter [Einen neuen SSH-Schlüssel zum GitHub-Konto hinzufügen](https://docs.github.com/de/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Key mit benutzerdefiniertem namen

Wenn du einen SSH-Schlüssel mit einem benutzerdefinierten Namen erstellen möchtest, kannst du die folgenden Schritte ausführen. In den Schritten wird angenommen, dass du die Befehlszeile verwendest, um dies zu tun:

1. \*\*Öffnen eines Terminals oder einer Befehlszeile:\*\* Öffne ein Terminal oder eine Befehlszeile auf deinem Computer.

2. \*\*Führe den `ssh-keygen`-Befehl aus:\*\* Gib den folgenden Befehl ein und drücke die "Enter"-Taste:

```bash

ssh-keygen -t rsa -f ~/.ssh/id\_rsa\_custom

```

Dieser Befehl erstellt einen SSH-Schlüssel des Typs "rsa" mit einem benutzerdefinierten Dateinamen (`id\_rsa\_custom`) und speichert ihn im Standardverzeichnis `~/.ssh`. Du kannst den Dateinamen an deine Bedürfnisse anpassen, indem du `id\_rsa\_custom` durch deinen gewünschten Namen ersetzt. Zum Beispiel:

```bash

ssh-keygen -t rsa -f ~/.ssh/my\_custom\_key

```

3. \*\*Optionale Passphrase hinzufügen:\*\* Nach dem Ausführen des Befehls wirst du aufgefordert, eine optionale Passphrase für den Schlüssel einzugeben. Die Passphrase dient als zusätzliche Sicherheitsebene. Du kannst eine Passphrase eingeben oder einfach die "Enter"-Taste drücken, um keine Passphrase zu verwenden.

4. \*\*SSH-Schlüssel erstellen:\*\* Der Befehl wird den SSH-Schlüssel mit dem von dir angegebenen Namen erstellen. Du findest die Dateien im `~/.ssh`-Verzeichnis, wobei die öffentliche Schlüsseldatei den gleichen Namen wie die private Schlüsseldatei hat, aber die Endung `.pub` trägt.

Nachdem du diese Schritte ausgeführt hast, hast du einen benutzerdefinierten SSH-Schlüssel erstellt, den du zum Authentifizieren bei SSH-Diensten verwenden kannst. Stelle sicher, dass du die private Schlüsseldatei (z.B., `~/.ssh/my\_custom\_key`) sicher aufbewahrst und die öffentliche Schlüsseldatei (z.B., `~/.ssh/my\_custom\_key.pub`) bei den Diensten, bei denen du dich authentifizieren möchtest, hinzufügst.

Denke daran, dass du, wenn du bereits SSH-Schlüssel erstellt hast und `ssh-keygen` dich auffordert, einen anderen Schlüssel neu zu schreiben, den bestehenden Schlüssel beibehalten und einen neuen mit deinem benutzerdefinierten Namen erstellen kannst. Dies gibt dir die Flexibilität, verschiedene Schlüssel für unterschiedliche Zwecke zu verwenden.