Welcome to Paramiko's documentation!

This site covers Paramiko's usage & API documentation. For basic info on what Paramiko is, including its public changelog & how the project is maintained, please see the main project website.

API documentation

The high-level client API starts with creation of an `SSHClient` object. For more direct control, pass a socket (or socket-like object) to a `Transport`, and use `start_server` or `start_client` to negotiate with the remote host as either a server or client.

As a client, you are responsible for authenticating using a password or private key, and checking the server's host key. (Key signature and verification is done by paramiko, but you will need to provide private keys and check that the content of a public key matches what you expected to see.)

As a server, you are responsible for deciding which users, passwords, and keys to allow, and what kind of channels to allow.

Once you have finished, either side may request flow-controlled `channels` to the other side, which are Python objects that act like sockets, but send and receive data over the encrypted session.

For details, please see the following tables of contents (which are organized by area of interest.)

Core SSH protocol classes

- Channel
- Client
- Message
- Packetizer
- Transport

Authentication & keys

- SSH agents
- Host keys / `known_hosts` files
- Key handling
  - Parent key class
  - DSA (DSS)
  - RSA
  - ECDSA
  - Ed25519
- GSS-API authentication
- GSS-API key exchange

Other primary functions

- Configuration
  - Keywords currently supported
    - Expansion tokens
  - `config` module API documentation
- `ProxyCommand` support
- Server implementation
- SFTP

Miscellany

- Buffered pipes
- Buffered files