Static key configurations offer the simplest setup, and are ideal for point-to-point VPNs or proof-of-concept testing.

Static Key advantages

- Simple Setup
- No X509 PKI (Public Key Infrastructure) to maintain

Static Key disadvantages

- Limited scalability -- one client, one server
- Lack of *perfect forward secrecy* -- key compromise results in total disclosure of previous sessions
- Secret key must exist in plaintext form on each VPN peer
- Secret key must be exchanged using a pre-existing secure channel

Simple Example

This example demonstrates a bare-bones point-to-point OpenVPN configuration. A VPN tunnel will be created with a server endpoint of 10.8.0.1 and a client endpoint of 10.8.0.2. Encrypted communication between client and server will occur over UDP port 1194, the default OpenVPN port.

Generate a static key:

```
openvpn --genkey --secret static.key
```

Copy the static key to both client and server, over a pre-existing secure channel.

Server configuration file

```
dev tun
ifconfig 10.8.0.1 10.8.0.2
secret static.key
```

Client configuration file

```
remote myremote.mydomain
dev tun
ifconfig 10.8.0.2 10.8.0.1
secret static.key
```

Firewall configuration

Make sure that:

- UDP port 1194 is open on the server, and
- the virtual TUN interface used by OpenVPN is not blocked on either the client or server (on Linux, the TUN interface will probably be called **tun0** while on Windows it will probably be called something like **Local Area Connection *n*** unless you rename it in the Network Connections control panel).

Bear in mind that 90% of all connection problems encountered by new OpenVPN users are firewall-related.

Testing the VPN

Run OpenVPN using the respective configuration files on both server and client, changing **myremote.mydomain** in the client configuration to the domain name or public IP address of the server.

To verify that the VPN is running, you should be able to ping 10.8.0.2 from the server and 10.8.0.1 from the client.

Expanding on the Simple Example

Use compression on the VPN link

Add the following line to both client and server configuration files:

```
comp-lzo
```

Make the link more resistent to connection failures

Deal with:

- keeping a connection through a NAT router/firewall alive, and
- follow the DNS name of the server if it changes its IP address.

Add the following to both client and server configuration files:

```
keepalive 10 60
ping-timer-rem
persist-tun
persist-key
```

Run OpenVPN as a daemon (Linux/BSD/Solaris/MacOSX only)

Run OpenVPN as a daemon and drop privileges to user/group **nobody**.

Add to configuration file (client and/or server):

```
user nobody
group nobody
daemon
```

Allow client to reach entire server subnet

Suppose the OpenVPN server is on a subnet **192.168.4.0/24**. Add the following to client configuration:

```
route 192.168.4.0 255.255.255.0
```

Then on the server side, add a route to the server's LAN gateway that routes 10.8.0.2 to the OpenVPN server machine (only necessary if the OpenVPN server machine is not also the gateway for the server-side LAN). Also, don't forget to enable IP Forwarding on the OpenVPN server machine.