# File centric explanation of the openVPN connexion

**Scenario:**

- Client (using client1.ovpn) connects to an OpenVPN server.
- The server is running inside a Docker container, as set up in our previous examples.
- TLS with ephemeral Diffie-Hellman (DHE) is used for key exchange.
- RSA is used for server authentication.

**Files on the Server (Inside the openvpn-data/conf directory):**

1. **ca.crt (CA Certificate):**
   - **Content:** The X.509 certificate of the Certificate Authority (CA). This is a plain text file that contains the CA's public key and information about the CA (e.g., name, organization, validity period). It's used to verify the authenticity of other certificates.
   - **Example Snippet:**
     ```
     -----BEGIN CERTIFICATE-----
     MIIDdTCCAl ... (Base64-encoded certificate data) ...
     -----END CERTIFICATE-----
     ```

2. **pki/private/ca.key (CA Private Key):**
   - **Content:** The CA's private key. This file is **highly sensitive** and should be protected with strict permissions. It's used to sign other certificates (server and client certificates).
   - **In our OpenVPN setup, this file is not directly accessible within the container.** It's used by the easyrsa scripts during the PKI initialization, but OpenVPN itself doesn't need to read it directly.
   - **Example Snippet (if it were accessible):**
     ```
     -----BEGIN RSA PRIVATE KEY-----
     Proc-Type: 4,ENCRYPTED
     DEK-Info: AES-256-CBC, ... (Hexadecimal salt) ...

     ... (Base64-encoded encrypted private key) ...
     -----END RSA PRIVATE KEY-----
     ```

3. **pki/issued/server.crt (Server Certificate):**
   - **Content:** The OpenVPN server's X.509 certificate. It contains the server's public key and information about the server. It's signed by the CA using ca.key.
   - **Example Snippet:**
     ```
     -----BEGIN CERTIFICATE-----
     MIIDdTCCAl ... (Base64-encoded certificate data) ...
     -----END CERTIFICATE-----
     ```

4. **pki/private/server.key (Server Private Key):**
   - **Content:** The OpenVPN server's private key. This file is used for decryption and signing during the TLS handshake. It should be kept secure.
   - **Example Snippet:**
     ```
     -----BEGIN RSA PRIVATE KEY-----
     ... (Base64-encoded private key, potentially encrypted) ...
     ```

```
-----END RSA PRIVATE KEY-----
```

5. **pki/dh.pem (Diffie-Hellman Parameters):**
    - **Content:** Contains the pre-calculated Diffie-Hellman parameters (prime modulus p and generator g).
    - **This file is not a key, but it's essential for the DH key exchange.**
    - **Example Snippet:**
```
-----BEGIN DH PARAMETERS-----
... (Base64-encoded DH parameters) ...
-----END DH PARAMETERS-----
```

6. **server.conf (OpenVPN Server Configuration):**
    - **Content:** The main configuration file for the OpenVPN server. It references the certificate and key files, sets network parameters, and defines other options.
    - **Example Snippets:**
```
port 40000
proto udp
dev tun
ca ca.crt
cert pki/issued/server.crt
key pki/private/server.key
dh pki/dh.pem
server 172.20.0.0 255.255.0.0
... other directives ...
```

**Files on the Client:**

1. **client1.ovpn (OpenVPN Client Configuration):**
    - **Content:** This file contains the configuration for the client to connect to the server. It includes directives and embedded certificate/key data.
    - **Example Snippets:**
```
client
dev tun
proto udp
remote YOUR_PUBLIC_IP 40000
... other directives ...
<ca>
-----BEGIN CERTIFICATE-----
... (Base64-encoded CA certificate - same as ca.crt on
server) ...
-----END CERTIFICATE-----
</ca>
<cert>
-----BEGIN CERTIFICATE-----
... (Base64-encoded client certificate - client1.crt) ...
-----END CERTIFICATE-----
</cert>
<key>
-----BEGIN RSA PRIVATE KEY-----
```

```
... (Base64-encoded client private key - client1.key) ...
-----END RSA PRIVATE KEY-----
</key>
```

2. **pki/issued/client1.crt (Client Certificate):**
    - **Content:** The client's X.509 certificate, signed by the CA.
    - **Note:** This file is usually embedded within client1.ovpn.
3. **pki/private/client1.key (Client Private Key):**
    - **Content:** The client's private key.
    - **Note:** This file is usually embedded within client1.ovpn.

**TLS Handshake in Action (Simplified with File References):**

1. **Client Hello:**
    - Client reads client1.ovpn and initiates a connection to YOUR_PUBLIC_IP:40000.
    - Client sends a list of supported cipher suites (including those with DHE) and a random value ("client random").
2. **Server Hello:**
    - Server receives the connection request.
    - Server selects a cipher suite (e.g., one with DHE).
    - Server reads server.crt and sends its certificate to the client.
    - Server sends its own random value ("server random").
3. **Server Authentication:**
    - Client receives server.crt.
    - Client extracts the CA certificate from within client1.ovpn (the <ca> block).
    - Client verifies server.crt using the CA certificate:
        - Checks if server.crt is signed by the CA.
        - Checks for expiration and other validity conditions.
        - Checks that the hostname in server.crt matches YOUR_PUBLIC_IP.
4. **Key Exchange (DHE):**
    - Server reads dh.pem to get the DH parameters.
    - Server generates its DH private key (b) and calculates its DH public key (B).
    - Server sends B to the client.
    - Client generates its DH private key (a) and calculates its DH public key (A).
    - Client sends A to the server.
    - Client and server perform DH calculations using their private keys and the other party's public key to arrive at the same shared secret.
5. **Key Derivation:**
    - Client and server use the shared secret, "client random," and "server random" as input to a KDF (Key Derivation Function) to generate the master secret and then the session keys.
6. **Change Cipher Spec/Finished:**
    - Client and server signal that they will start using encryption and send encrypted "Finished" messages to verify the key exchange.
7. **Tunnel Establishment:**
    - OpenVPN creates the tun0 interface on both client and server.
    - The server assigns an IP address to the client from the 172.20.0.0/16 subnet.
8. **Data Transfer:**
    - Data is encrypted using the session keys and transmitted through the tun0 interface.

**Important Notes:**

- **File Access:** During the handshake, the OpenVPN server process needs to read server.crt, server.key, and dh.pem. The client process needs to read the embedded certificates and keys from client1.ovpn.
- **Security:** The private keys (ca.key, server.key, client1.key) are extremely sensitive. They should be protected with strong file permissions and never shared or transmitted insecurely.
- **Ephemeral DH:** In this example, we're assuming ephemeral Diffie-Hellman (DHE). This means that new DH parameters are generated for each connection, providing forward secrecy.
- **RSA vs. DH:** RSA is used here for server authentication (signing the server's certificate), while Diffie-Hellman is used for key exchange.

This detailed, file-centric explanation should give you a very clear picture of how OpenVPN uses these files during a real-world connection.

*Thanks to Gemini*