

8. Cryptographic Computations

In order to begin connection protection, the TLS Record Protocol requires specification of a suite of algorithms, a master secret, and the client and server random values. The authentication, encryption, and MAC algorithms are determined by the `cipher_suite` selected by the server and revealed in the `ServerHello` message. The compression algorithm is negotiated in the hello messages, and the random values are exchanged in the hello messages. All that remains is to calculate the master secret.

8.1. Computing the Master Secret

For all key exchange methods, the same algorithm is used to convert the `pre_master_secret` into the `master_secret`. The `pre_master_secret` should be deleted from memory once the `master_secret` has been computed.

```
master_secret = PRF(pre_master_secret, "master secret",
                    ClientHello.random + ServerHello.random)
                    [0..47];
```

The master secret is always exactly 48 bytes in length. The length of the premaster secret will vary depending on key exchange method.

8.1.1. RSA

When RSA is used for server authentication and key exchange, a 48- byte `pre_master_secret` is generated by the client, encrypted under the server's public key, and sent to the server. The server uses its private key to decrypt the `pre_master_secret`. Both parties then convert the `pre_master_secret` into the `master_secret`, as specified above.

8.1.2. Diffie-Hellman

A conventional Diffie-Hellman computation is performed. The negotiated key (`Z`) is used as the `pre_master_secret`, and is converted into the `master_secret`, as specified above. Leading bytes of `Z` that contain all zero bits are stripped before it is used as the `pre_master_secret`.

Note: Diffie-Hellman parameters are specified by the server and may be either ephemeral or contained within the server's certificate.