



MatrixSSL Pre-Shared Key Cipher Suites

Electronic versions are uncontrolled unless directly accessed from the QA Document Control system.

Printed version are uncontrolled except when stamped with 'VALID COPY' in red.

External release of this document may require a NDA.

© INSIDE Secure - 2013 - All rights reserved



TABLE OF CONTENTS

- 1 MATRIXSSL PSK CIPHER SUITES3**
 - 1.1 Basic PSK Cipher Suites.....3
 - 1.1.1 Minimal Build..... 3
 - 1.2 DHE_PSK Cipher Suites.....3

- 2 API.....5**
 - 2.1 matrixSslLoadPsk5

1 MATRIXSSL PSK CIPHER SUITES

MatrixSSL includes support for two modes of Pre-shared Key (PSK) cipher suites. These cipher suites offer an alternative authentication mechanism from the more standard RSA or ECC based public key encryption used by most TLS handshakes. In both PSK modes, authentication is performed based on each peer having access to the pre-shared keys and these keys must always be treated with the same level of secrecy as RSA private keys.

1.1 Basic PSK Cipher Suites

The first mode of support is basic PSK in which the pre-shared symmetric keys are used as the sole method of authentication between the peers. This mode is not generally recommended and should only be used in tightly constrained environments in which other ciphers cannot be used.

Cipher Suite	Cipher ID
TLS_PSK_WITH_AES_128_CBC_SHA	0x008C (140)
TLS_PSK_WITH_AES_256_CBC_SHA	0x008D (141)
TLS_PSK_WITH_AES_128_CBC_SHA256	0x00AE (174)

Table 1 - Supported Basic PSK Cipher Suites

1.1.1 Minimal Build

The smallest possible version of the MatrixSSL library can be built if your platform wishes to use only these basic PSK suites. If only these above suites are enabled in *matrixsslConfig.h* there is a set of defines that may be disabled in the other modules. The table below lists the #defines that should be enabled and disabled to create this small PSK-only library.

Code Define	Location	Comments
MATRIX_USE_FILE_SYSTEM	Build environment	Disable this define
USE_X509	cryptoConfig.h	Disable this define as there are no X.509 certificates involved
USE_RSA	cryptoConfig.h	Disable this define as there is no RSA public key crypto
USE_PRIVATE_KEY_PARSING	cryptoConfig.h	Disable this define
USE_DH	cryptoConfig.h	Disable this define
USE_3DES, USE_ARC4	cryptoConfig.h	Disable the unused symmetric ciphers
USE_PKCS5	cryptoConfig.h	Disable this define as no RSA private keys are used
DISABLE_PSTM	cryptoConfig.h	Enable this define to exclude the big math code components

Table 2 - Define Configuration for Minimal Build

1.2 DHE_PSK Cipher Suites

The second mode of support is DHE_PSK, which uses the pre-shared symmetric keys to authenticate a Diffie-Hellman handshake. The added advantages of the DHE_PSK suites over the basic PSK suites are additional protection against dictionary attacks by passive eavesdroppers and also provide Perfect Forward Secrecy.

Cipher Suite	Cipher ID
TLS_DHE_PSK_WITH_AES_128_CBC_SHA	0x0090 (144)
TLS_DHE_PSK_WITH_AES_256_CBC_SHA	0x0091 (145)

Table 3 - Supported DHE_PSK Cipher Suites

2 API

The only integration step necessary to use a PSK cipher suite is to nominate the key material during session initialization using the following API.

2.1 matrixSslLoadPsk

```
int32 matrixSslLoadPsk(sslKeys_t *keys, unsigned char *key,  
    uint32 keyLen, unsigned char *id, uint32 idLen);
```

Parameter	Input/Output	Description
keys	input	Key structure created from a previous call to matrixSslNewKeys
key	input	Pointer to a byte array the contains the secret Pre-Shared Key to be used for this session
keyLen	input	Length in bytes of key. Must be ≥ 1 and \leq SSL_PSK_MAX_KEY_SIZE (128 byte default)
id	input	Pre-Shared Key identity
idLen	input	Length in bytes of id. Muyst be ≥ 1 and \leq SSL_PSK_MAX_ID_SIZE (256 byte default)

Return Value	Description
PS_SUCCESS	Successful key load
PS_MEM_FAIL	Failure. Platform unable to allocate memory
PS_ARG_FAIL	Failure. NULL pointer for key or id parameters. Length tests of keyLen or idLen outside limits

Severs and Clients

This API is called to register a Pre Shared Key (PSK) and PSK Identity with a key structure that will be used when new SSL sessions are created. The PSK and Identity are both arbitrary byte values. The length of the PSK should be sufficiently long and random to provide adequate security. Typically a length of 16 bytes of true random data is viewed as “strong” for this purpose. It is **not recommended** to use a typical login type password for the PSK. If a password is used, it should only be used to produce a derived key via a Password Based Key Derivation Function such as `pkcs5pbkdf2()` in `crypto/cryptoApi.h`. The Identity is a string, which uniquely identifies the key. For example, a client which connects to several different host names may have one PSK per host, each with the Identity of the given host name. It is the Identity that is exchanged between the peers during the SSL handshake.

The keys parameter must have been previously allocated by a call to `matrixSslNewKeys`. Once loaded with the key material, the parameter is passed to `matrixSslNewClientSession` or `matrixSslNewServerSession`.

Servers

Servers may call this routine multiple times to register several Identities and Keys that are acceptable for authentication. The API should be called before accepting client connections, so that the server is able to authenticate the client during the SSL handshake.

Clients

Clients should only call this function once to register the key that identifies itself.

Memory Profile

The PSK material will be freed when `matrixSslDeleteKeys` is called on the keys