

MatrixSSL 3.2.0 Open Source Release Notes

Overview

Who Is This Document For? 2

MatrixSSL 3.2.0 Release Notes

Enhancements to Features and Functionality

Added TLS 1.1 protocol support 3

Added PKCS#8 private key parsing 3

IN and OUT default buffer sizes 3

Zero length SSL records now returned to user 3

Public API Changes

Added matrixSslEncodeToOutdata 4

Support and Bug Reporting

Contacting Support or Reporting Bugs 4

Overview

Thank you for choosing MatrixSSL. The 3.2 version is a minor revision to the 3.1 releases and adds TLS 1.1 protocol support in addition to a handful of other changes. The previous version is MatrixSSL 3.1.4

If you are migrating from a 2.x version of MatrixSSL please contact PeerSec Networks for documentation that can help with the upgrade process.

Who Is This Document For?

- Software developers that are securing applications with MatrixSSL
- Software developers upgrading to MatrixSSL 3.2.0 from a previous version
- Anyone wanting to learn more about MatrixSSL

MatrixSSL 3.2.0 Release Notes

This section highlights the differences between version 3.1.4 and 3.2.0

Enhancements to Features and Functionality

Added TLS 1.1 protocol support

TLS 1.1 protocol support is now available in the open source version of MatrixSSL. The protocol is enabled/disabled through the compile time define `USE_TLS_1_1` in *matrixsslConfig.h*. If enabled, the protocol negotiation will default to TLS 1.1 for any communicating SSL peer that also supports it. It is also now possible to disable SSL 3.0 using the `DISABLE_SSLV3` define if only TLS version protocols are desired

Added PKCS#8 private key parsing

The PKCS#8 standard is becoming more widespread for newly issued private keys. PKCS#8 parsing is now included by default in the open source version of MatrixSSL. This support is built into the existing `matrixSslLoadRsaKeys` API.

IN and OUT default buffer sizes

Previous versions of MatrixSSL used a single compile time setting for the default internal input and output buffers. This define has now been split into `SSL_DEFAULT_OUT_BUF_SIZE` and `SSL_DEFAULT_IN_BUF_SIZE` defines to give the user more memory control for the specific use case. For example, if the integrator knows that incoming data will be short requests and the outgoing reply data will be large files, the `SSL_DEFAULT_IN_BUF_SIZE` may be set smaller than `SSL_DEFAULT_OUT_BUF_SIZE` to help streamline this implementation.

Zero length SSL records now returned to user

Callers of `matrixSslReceivedData` will now be informed of zero length SSL records with the standard return code of `MATRIXSSL_APP_DATA` and length values of 0. Previous versions of MatrixSSL would silently discard empty records.

Public API Changes

Added `matrixSslEncodeToOutdata`

An SSL record encoding alternative to the existing `matrixSslGetWritebuf/`
`matrixSslEncodeWritebuf` combination has been introduced. The new
`matrixSslEncodeToOutdata` enables integrators to encode plaintext from where it exists in an
external memory location.

This differs from the `matrixSslEncodeWritebuf` API that requires the plaintext has been
written or copied into the internal library buffer. The new `matrixSslEncodeToOutdata` function
will leave the plaintext buffer untouched while encoding to the internal library buffer. The
encoded data is still retrieved for sending using `matrixSslGetOutdata`. Please see the API
documentation for more information on this new function.

Support and Bug Reporting

Contacting Support or Reporting Bugs

Email support@peersec.com