

SharkSSL Embedded SSL/TLS Server

SharkSSL is a compact SSL/TLS stack designed from the ground up to enable secure communication and management of remote embedded devices.

Why use SSL in embedded devices

The ubiquitous Internet is becoming the norm for connecting and controlling devices, but the Internet is inherently insecure. It is, therefore, necessary to encrypt data that is interchanged with devices to prevent unauthorized users from intercepting and potentially gaining access to the data or the devices themselves. A remote server can also authenticate itself to an SSL enabled client such as, Internet Explorer, Firefox, etc...

Why use SharkSSL

Before developing our own Secure Sockets Layer (SSL), we searched for a small implementation. This proved very difficult to find. The companies providing such solutions were either providing expensive solutions or solutions too big for practical use in embedded devices. For example, the standard OpenSSL library is over 1 MB in size, and thus is not suitable for embedded systems. We designed SharkSSL to be very small. Thus far, it is the smallest SSL/TSL Server on the market.

Comparing SharkSSL to other SSL stacks

SharkSSL is tested with the Barracuda Embedded Web Server, which is much more than a web-server. Barracuda has helped us test SharkSSL with many types of non-browser clients such as C++ HTTP libraries, Python, Java, and WebDAV. The WebDAV protocol is an excellent robustness test as it puts much more strain on the SSL stack than a typical web-browser.

FEATURES

- Object-oriented library in ANSI C (with C++ wrapper code)
- Supports all Freescale® ColdFire® hardware-acceleration encryption engines and is being extended to ARM encryption engines
- Code size is less than 20KB total footprint on ColdFire® 547x/8x supporting the on chip hardware-acceleration encryption engine.
- Includes crypto software library for processors without hardware encryption support or with partial hardware encryption acceleration (AES, DES, 3DES, ARC4, SHA1, MD5)
- Includes proprietary RSA crypto library that can be retargeted to dedicated DSP engines
- Configurable session caching
- Advanced embedded buffer management with no coding required to handle the SSL buffers. Custom memory allocators can be specified.
- Transport agnostic, works with any transport type, including TCP/IP.
- Multithreading support for optimal performance when used with multitasking/multiprocess OS's
- Easily portable to any RTOS and any hardware-acceleration encryption engine.
- Off the shelf support for SMX.
- The Barracuda™ Embedded Web Server can take advantage of the optimized memory management supported by SharkSSL for persistent HTTP 1.1 connections.

Testing a SSL server with only browsers is a poor test since typically little data is sent from the client to the server. It is important to find out what browser and non-browser clients an SSL stack has been tested with.

SSLv3 & TLS 1.0 ciphers supported (in decreasing order of strength):

- AES_256_CBC_SHA
- AES_128_CBC_SHA
- 3DES_EDE_CBC_SHA
- RC4_128_SHA
- RC4_128_MD5
- DES_CBC_SHA
- NULL_SHA
- NULL_MD5

Footprint for ARM7/ Thumb

SharkSSL server sizes (KB) for ARM7TDMI, with EWARM v4.41A:

ARM mode	ROM	RAM
library excluding AES and DES encryption software	28	2
AES encryption software	21	4
DES/3DES encryption software	12	
Thumb mode		
library excluding AES and DES encryption software	20	2
AES encryption software	18	4
DES/3DES encryption software	10	

The AES, DES, 3DES algorithm can be replaced with hardware accelerator versions, with much better performance and reduced footprint (a few KB).

SharkSSL server sizes (KB) for MCF547x/548x with CodeWarrior 6.3:

ColdFire	ROM	RAM
library excluding crypto engine	17	0
software crypto library	50	6
hardware crypto engine	2.5	0