Secure data transmission with OTT ecoLog 1000

Why using HTTPS:

The motivation for HTTPS is authentication of the accessed server and protection of the privacy and integrity of the exchanged data while in transit. It protects against man-in-the-middle attacks. The bidirectional encryption of communications between a client and server protects the communication against attacks.

OTT ecoLog 1000:	HTTPS Client TLS 1.2*
Data Server:	HTTPS Server TLS 1.2 e.g. OTT Hydras 3 Web Server, Apache Server with Open SSL, …

Technology: HTTPS (Hypertext Transfer Protocol Secure) also called HTTP over TLS
TLS 1.2 (Transport Layer Security) – for security reason no TLS <1.2 or SSL
TLS 1.2 is described in RFC 5246
Port: 443
Max key length = 4096 bit
Support of ECDHE – Elliptic Curve Diffie Hellman
Support of ECDSA - Elliptic Curve Digital Signature Algorithm
Support of AES 256 - Advanced Encryption Standard
Support of SHA 384 – Secure Hash Algorithm

Supported Cipher Suites:

A cipher suite is a set of algorithms that help secure a network connection that uses Transport Layer Security (TLS). The set of algorithms that cipher suites usually contain include: a key exchange algorithm, a bulk encryption algorithm and a message authentication code algorithm.

Cipher suites are not classified directly as strong and weak, because this depends on both the algorithm used and the key length. Thus a strong encryption algorithm with a very short key is actually offering a weak protection.

* ecoLog 1000 is enabled to support TLS 1.3 in future as well

Cipher Suites supported by OTT ecoLog 1000

Cipher suites are ordered in terms of compatibility into three groups:

1. "Modern" group that offers high security and will be supported also in near future
2. "In use" group which is still in frequent use today although some of the ciphers are not as secure as those in the first group and certain key lengths may be considered insecure in the future
3. "Compatibility" group that offers least secure options but can still be used by some older systems

ID	Cipher Name	Security	Supported
0xC02C	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	MODERN	Y
0XC02B	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	MODERN	Y
0xC02F	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	MODERN	Y
0x009E	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	MODERN	Y
0xC028	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	MODERN	Y
0xC02F	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	MODERN	Y
0xC024	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	MODERN	Y
0xC023	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	MODERN	Y
0xC027	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	MODERN	Y
0x009F	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	IN USE	Y
0x0067	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	IN USE	Y
0x006B	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256	IN USE	Y
0x009C	TLS_RSA_WITH_AES_128_GCM_SHA256	IN USE	Y
0x009D	TLS_RSA_WITH_AES_256_GCM_SHA384	IN USE	Y
0x003C	TLS_RSA_WITH_AES_128_CBC_SHA256	IN USE	Y
0xC0AC	TLS_ECDHE_ECDSA_WITH_AES_128_CCM	MODERN	Y
0xC0AE	TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8	MODERN	Y
0xC09E	TLS_DHE_RSA_WITH_AES_128_CCM	MODERN	Y
0xC0A2	TLS_DHE_RSA_WITH_AES_128_CCM_8	MODERN	Y
0xC0A3	TLS_DHE_RSA_WITH_AES_256_CCM_8	MODERN	Y
0xC0AD	TLS_ECDHE_ECDSA_WITH_AES_256_CCM	MODERN	Y
0xC0AF	TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8	MODERN	Y
0xC09F	TLS_DHE_RSA_WITH_AES_256_CCM	MODERN	Y
0xC087	TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_GCM_SHA384	MODERN	Y
0xC08B	TLS_ECDHE_RSA_WITH_CAMELLIA_256_GCM_SHA384	MODERN	Y
0xC07D	TLS_DHE_RSA_WITH_CAMELLIA_256_GCM_SHA384	MODERN	Y
0x00C4	TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA256	IN USE	Y
0xC086	TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_GCM_SHA256	MODERN	Y
0xC08A	TLS_ECDHE_RSA_WITH_CAMELLIA_128_GCM_SHA256	MODERN	Y
0xC08D	TLS_ECDH_RSA_WITH_CAMELLIA_256_GCM_SHA384	MODERN	Y
0xC089	TLS_ECDH_ECDSA_WITH_CAMELLIA_256_GCM_SHA384	MODERN	Y
0xC07A	TLS_RSA_WITH_CAMELLIA_128_GCM_SHA256	MODERN	Y
0x003D	TLS_RSA_WITH_AES_256_CBC_SHA256	COMPAT	Y

Note:

These are the most common cipher suites; the device supports more and would support more which are not included in default application to optimize space utilization.