

Technology Brief

iChipSec™ SSL3 Client

August 2005



International:
Connect One Ltd.
2 Hanagar Street
Kfar Saba 44425, Israel
Tel: +972-9-766-0456
Fax: +972-9-766-0461
E-mail: info@connectone.com
<http://www.connectone.com>

USA:
Connect One Semiconductors, Inc.
15818 North 9th Ave.
Phoenix, AZ 85023
Tel: 408-986-9602
Fax: 602-485-3715
E-mail: info@connectone.com
<http://www.connectone.com>

Introduction

Secure communication is an ever-increasing concern for enterprises that send sensitive, secret or confidential information across the Internet. Many products, such as POS terminals, medical devices, and utility meters, today require the transmission of encrypted data. The common method today for enabling encryption inside an embedded device is to develop or purchase a library supporting the SSL3/TLS1 protocol suite and to run this on the host processor. Often a more powerful host processor and more memory are required, and a new PCB must be designed and debugged. The result is that this development effort requires a lot of time, effort, and expense.

As more and more devices use the Internet for transmission, there is a growing need for a practical, low-cost, simple and quick solution for embedding encryption capabilities inside everyday devices. Connect One's iChipSec™ is such a solution. It is a new family within the iChip™ Internet Controller™ family that offloads encryption tasks from the host processor and adds them to the TCP/IP offload functionality of Connect One's iChip. The result is a chip designed specifically to simplify the development of, and speed the time-to-market for, encryption-enabled devices.

The security capabilities are implemented as new commands added to Connect One's AT+i™ command set, a high-level Internet extension to the industry-standard AT command set. AT+i commands enable the embedded designer to easily add Internet Protocol connectivity to any device. The new commands make it extremely simple to select and invoke the security protocols that run on Connect One's iChipSec family and to define the trusted certificate authority.

iChipSec supports the RFC2246-based SSL3/TLS secure socket protocol and uses the following cipher suites:

- * SSL_RSA_WITH_RC4_128_MD5
- * SSL_RSA_WITH_RC4_128_SHA
- * SSL_RSA_WITH_3DES_EDE_CBC_SHA

iChipSec also supports Secure FTP via the FTPS protocol. Support is planned for Secure SMTP (SMTPS) and Secure HTTP (HTTPS) during 2006.

Establishing an SSL3/TLS Socket Connection

iChipSec supports a single SSL3 connection over a TCP/IP socket. Opening the SSL3 socket on iChipSec involves two simple steps:

1. Opening a standard TCP/IP socket to an SSL3 server; and
2. Initiating an SSL3 handshake over the open socket to establish an SSL3 session. SSL3 handshake negotiations are initiated using the new [AT+iSSL command](#).

Command Syntax: `AT+iSSL:<sock>`
where `<sock>` is a handle to an open TCP/IP socket previously opened with an AT+iSTCP command.

iChipSec negotiates the SSL3 connection based on several SSL3 related parameters. Following a successful SSL3 handshake, iChipSec encrypts all data sent across the socket according to the cipher suite and keys agreed upon during the SSL3 connection. Data received on the socket is decrypted by iChipSec prior to making it available to the host processor.

Sending and Receiving Data over an SSL3/TLS Socket

The existing AT+iSSND command is used to send data through an SSL3/TLS socket, using the same syntax as for non-SSL3 sockets:

`AT+iSSND[%]:<hn>,<size>:<data>`

However, the `<size>` parameter is interpreted as the size of the data packet to encrypt. It is limited to 2KB. Receiving data on an SSL3/TLS socket is carried out with the existing AT+iSRCV command. iChipSec automatically decrypts data that arrives on the SSL3/TLS socket. The data transferred to the host always is decrypted (original) data.

SSL3/TLS Handshake and Session Example

An SSL3/TLS server located at secure.sslserver.com is running a secure application on port 1503. Using iChipSec, the following sequence opens a secure SSL3/TLS socket to that application and exchange data securely. For clarity, commands sent to iChipSec are **bold** while iChipSec replies are *italic*.

AT+iSTCP:secure.sslserver.com,1503

I/000

AT+iSSL:0

I/OK

AT+iSSND%:0,323:< ... 323 bytes of plaintext data ... >

I/OK

AT+iRP4

I/(1267,-200,-200,-200,-200,-200,-200,-200,-200)

AT+iSRCV:0

I/1267:< ... 1267 bytes of plaintext data ... >

AT+iscls:0

I/OK

I/DONE

Open socket to secure application.

iChipSec opens socket and returns handle 0.

iChipSec is instructed to negotiate an SSL3/TLS connection on socket handle 0.

SSL3/TLS handshake was successful. SSL3/TLS connection is established on socket handle 0.

Host sends 323 bytes of plaintext data via SSL3/TLS socket.

iChipSec will encrypt data and send the ciphertext over the Internet. The ‘%’ attribute indicates immediate flush.

iChipSec encrypted and sent the data.

Request Socket status.

Socket 0 has 1267 plaintext bytes buffered. The data was originally sent encrypted by the server.

iChipSec decrypted the ciphertext in the background.

Command to retrieve buffered plaintext.

iChipSec transmits buffered data to host.

Close socket handle 0.

SSL3/TLS socket is closed.

iChipSec goes offline.

SSL3-Related Parameters

+iCS	Cipher Suite Code	Defines iChipSec's preferred cipher suite. The default value (0) is the all-cipher selection. With this value, iChipSec sends the server its full list of supported ciphers. The server will select the most appropriate cipher to use. When a specific value is specified, iChipSec requires the server to use that specific cipher.
+iPKEY	Private Key	Contains the iChipSec's private key, used for SSL3 session negotiations. The private key is provided in PEM format. As with other passwords stored in iChipSec, it is only possible to store a private key. Once stored it may only be overwritten or erased. The AT+iPKEY? only reports the number of bits in the key.
+iCA	Certificate Authority	Contains the certificate of the Trusted Certificate Authority. This is the authority eligible to sign a server's certificate. iChipSec will accept a server's identity only if its certificate is signed by this authority.
+iCERT	Certificate	Contains the Client's (iChipSec's) certificate. The certificate contains iChipSec's Public Key used during the SSL3 handshake. Some server's request the client to identify itself during SSL3 negotiations. This is the certificate that iChipSec shall use in this case.

+iSSL - Establish an SSL3/TLS Socket Connection

Syntax:	AT+iSSL:<hn>	Negotiate an SSL3/TLS connection over an open TCP/IP socket.
Parameters:		<hn> = A previously open TCP/IP socket handle.
Scope:		iChipSec.
Command Options:	<hn>	Must have been obtained by a previous execution of an AT+iSTCP command during the current Internet mode session. Or a socket <i>accepted</i> by a listen socket.
Result code:	I/OK	If the SSL3/TLS negotiation was successful.
	I/ERROR	If some error occurred.

+iCS - Define the SSL3/TLS Cipher Suite

Syntax: AT+iCS=*n* Permanently sets the preferred cipher suite to use in the next SSL3/TLS negotiations.

Parameters: *n* = A supported cipher suite code, as defined in RFC2246

Command Options:

n = 0 set cipher suite to 'Propose All'.

n = 4 set cipher suite to SSL_RSA_WITH_RC4_128_MD5

n = 5 set cipher suite to SSL_RSA_WITH_RC4_128_SHA

n = 10 set cipher suite to SSL_RSA_WITH_3DES_EDE_CBC_SHA

When CS is set to 'Propose All', iChipSec will offer all supported cipher suites during SSL3/TLS negotiations. The server will select the most appropriate cipher suite.

Default: 0 (Propose All)

Scope: iChipSec.

Result code:

I/OK If *n* is 0 or a supported cipher suite code

I/ERROR Otherwise.

AT+iCS? Returns the current cipher suite value.
The reply is followed by I/OK.

AT+iCS=? Returns the value of the CS parameter.
The reply is followed by I/OK.

+iPKEY - Define SSL3/TLS Private Key

Syntax:	AT+iPKEY= <i>pkey</i>	Permanently set iChipSec's SSL3/TLS Private Key.
Parameters:	<i>pkey</i> = PEM format Private Key	
Command Options:	<i>pkey</i> ="	Empty: No Private Key defined.
	<i>pkey</i> =< <i>priv-key</i> >	<i>priv-key</i> will be used by iChipSec to encode data during SSL3/TLS negotiations.
Default:		Empty. No Private Key defined.
Scope:		iChipSec.
Result code:	I/OK	If <i>pkey</i> is an empty or legal Private Key.
	I/ERROR	Otherwise.
AT+iPKEY?		Report the current Private Key contents. The reported value will consist of the '*' character followed by the number of bits used in the key. If the Private Key value is empty, only <CRLF> will be returned. The reply is followed by I/OK.
AT+iPKEY=?		Returns the message 'String'. The reply is followed by I/OK.

+iCA - Define SSL3/TLS Trusted Certificate Authority

Syntax:	AT+iCA= <i>tca</i>	Permanently set iChipSec's SSL3/TLS Trusted Certificate Authority.
Parameters:	<i>tca</i> = PEM format Certificate	
Command Options:	<i>tca</i> =" <i>tca</i> =< <i>cert</i> >	Empty: No Trusted Certificate Authority. <i>cert</i> shall be referenced as the Trusted Certificate Authority's certificate during SSL3/TLS socket connection establishment (handshake). iChipSec shall establish an SSL3/TLS socket connection only to servers with a certificate authenticated by this Certificate Authority.
Default:		Empty. No Trusted Certificate Authority defined.
Scope:		iChipSec.
Result code:	I/OK I/ERROR	If <i>tca</i> is an empty or legal certificate. Otherwise.
AT+iCA?		Report the current trusted certificate contents. The reported value will display the Certificate Authority name, Certificate validity date range and the entire PEM contents. If the Trusted Certificate is empty, only <CRLF> will be returned. The reply is followed by I/OK.
AT+iCA=?		Returns the message 'String'. The reply is followed by I/OK.

+iCERT - Define SSL3/TLS Client Certificate

Syntax:	<i>AT+iCERT=cert</i>	Permanently set iChipSec's SSL3/TLS Client Certificate.
Parameters:	<i>cert</i>	= PEM format Certificate .
Command Options:	<i>tca ="</i>	Empty: No Client Certificate.
	<i>tca =<cli_cert></i>	<i>cli_cert</i> shall be used as the client side Certificate during SSL3/TLS socket connection establishment (handshake). iChipSec stores its Public key in this certificate. Furthermore, iChipSec shall provide this certificate to servers that require client side certificate authentication.
Default:		Empty. No client Certificate defined.
Scope:		iChipSec.
Result code:	<i>I/OK</i>	If <i>cert</i> is an empty or legal certificate.
	<i>I/ERROR</i>	Otherwise.
<i>AT+iCERT?</i>		Report the current client certificate contents. The reported value will display the client name, certificate validity date range and the entire PEM contents. If the certificate is empty, only <CRLF> will be returned. The reply is followed by I/OK.
<i>AT+iCERT=?</i>		Returns the message 'String'. The reply is followed by I/OK.