Technical white paper

# HPE NONSTOP SSL COMPARISON WITH NONSTOP CIP IPSEC

# CONTENTS

# OVERVIEW

## Introduction

HPE NonStop Socket Secure Layer (SSL) provides data encryption, which is sent or received by programs running in HPE NonStop systems over TCP/IP.

The HPE NonStop Cluster I/O Protocols (CIP) subsystem provides industry-standard networking connectivity to applications running on an HPE NonStop system. It includes support for the IPSec protocol suite, a collection of protocols and encryption libraries intended to secure the communication over an untrusted IP network.

This white paper provides a technical overview of HPE NonStop SSL and HPE NonStop CIP IPSec operation. It also compares and contrasts them.

## Audience

This white paper is intended for HPE employees and customers wanting to know the similarities and differences between HPE NonStop SSL and CIP IPSec. It is assumed that the reader has a basic understanding of how SSL and IPSec work.

## Definitions, acronyms, and abbreviations

Table 1 provides definitions for acronyms and abbreviations that are used throughout this document.

**TABLE 1.** Terminologies

| Word or phrase | Meaning in this document |
|---|---|
| Open Systems Interconnection (OSI) | OSI is a model that characterizes and standardizes the communication functions of a communicating system. |
| Transmission Control Protocol (TCP) | Transport layer of the OSI model responsible for the connection-oriented communication. |
| User Datagram Protocol (UDP) | Transport layer of the OSI model responsible for the connectionless communication. |
| Internet Protocol (IP) | Network layer of the OSI model responsible for the packet routing. |
| Internet Key Exchange (IKE) | The protocol is used to set up a Security Association (SA) in the IPSec protocol suite. |
| Security Association (SA) | A SA is the record of agreed upon security attributes between two network devices to support secure communication. |
| RDMA over Converged Ethernet (RoCE) | RoCE is a network protocol that allows remote direct memory access (RDMA) over an Ethernet network. |

# SSL

SSL is the cryptographic protocol that provides privacy and integrity of the data that is being sent between the two systems over the network. The two systems can be server and client (for example, website and browser) or server to server (for example, payroll data).

SSL is the predecessor to Transport Layer Security (TLS). SSL is used throughout this document to refer to TLS. SSL resides in the application layer of the OSI model. SSL is an independent application protocol, so higher-level protocols can layer on top of the SSL protocol transparently.

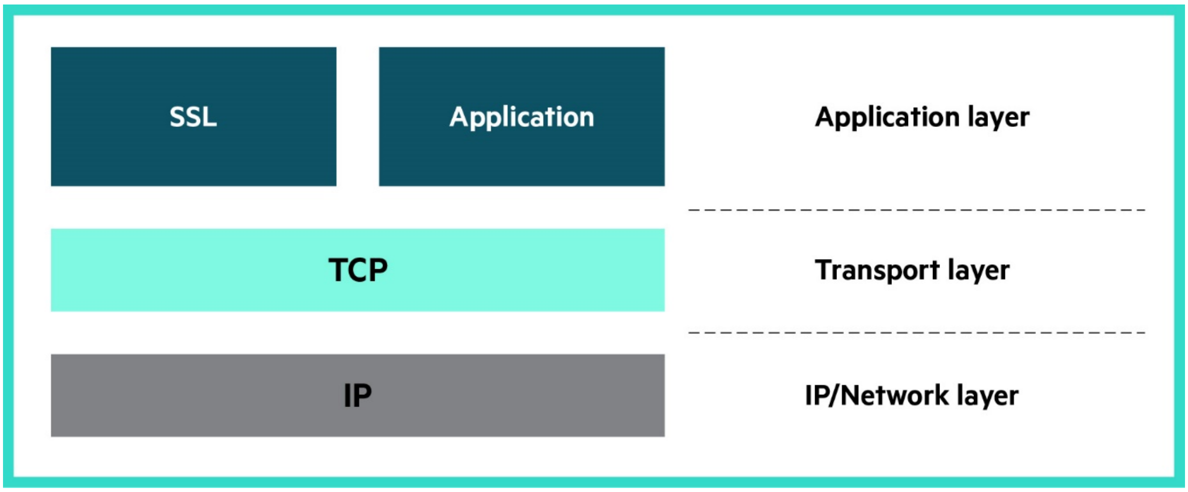


**FIGURE 1.** SSL in application layer

### Establishing SSL sessions

One of the communicating peers initiates a TCP connection. Upon successful establishment of a TCP connection, SSL initiates the SSL handshake. If the TCP connection fails, the SSL handshake is not initiated. The communicating peers must use the TCP protocol for communication to use the SSL security feature. The SSL handshake creates the SSL session between the communicating peers. As part of the SSL handshake, the peer authentication and the encryption algorithms and message integrity algorithms are negotiated.

### SSL security features

The SSL protocol provides the following security features to secure the communication between two systems across a network:

- **Authentication**
- **Confidentiality**
- **Compression**
- **Integrity**

### Authentication

SSL allows peer authentication. During the SSL handshake, peer authentication is performed. The peer's identity can be authenticated using public key cryptography. This authentication can be made optional. Clients can authenticate a server to ensure it is connecting to a trusted peer (SSL server authentication). The server can optionally request the client to present a certificate for authentication (SSL client authentication).

### Confidentiality

SSL secures the communication between the two systems by encrypting the data. As part of the SSL handshake, both the peers will agree on the encryption algorithms to be used.

### Compression

SSL optionally compresses data prior to encrypting the same. The encrypted data that is received is first decrypted at the receiver's end and then is uncompressed before handing over the data to the application.

Note: For most of the SSL/TLS deployments, the use of compression is discouraged because of the risk of Compression Ratio Info-leak Made Easy (CRIME)-style attacks. The compression option is removed from TLS from TLS 1.3 onward.

### Data integrity

SSL ensures the connection is reliable. SSL checks for the integrity and authenticity of each and every message that is transferred.

## HPE NONSTOP SSL

HPE NonStop SSL secures application data in the transport layer. HPE NonStop SSL uses SSL in the TLS variant as per the standard RFCs (2246, 4346, and 5246).

HPE NonStop SSL supports X.509 certificates for authentication. For encryption, HPE NonStop SSL supports multiple configurable cipher suits. For ensuring data integrity, HPE NonStop SSL supports all the standard MACs. For configuring HPE NonStop SSL and the list of supported encryption and MAC algorithms, refer to the HPE NonStop SSL Reference Manual.

HPE NonStop SSL can be configured to run as a SSL server or SSL client. It will run as a secure proxy supporting the following modes of operations, acting as:

- Secure proxy server for generic TCP/IP servers.
- Secure proxy client for generic TCP/IP clients.
- Secure proxy server for HPE services such as Telnet, FTP, EXPAND, and ODBC/MX.

This paper elaborates on NonStop SSL proxy server and proxy client mode. It only discusses the HPE NonStop SSL, which runs as a proxy process.

## HPE NonStop SSL proxy server

In a secure proxy server mode, the HPE NonStop SSL proxy accepts SSL connections from a client and tunnels them to the generic TCP/IP application server. The HPE NonStop SSL proxy communicates with the NonStop application server over the HPE NonStop TCP/IP stack.

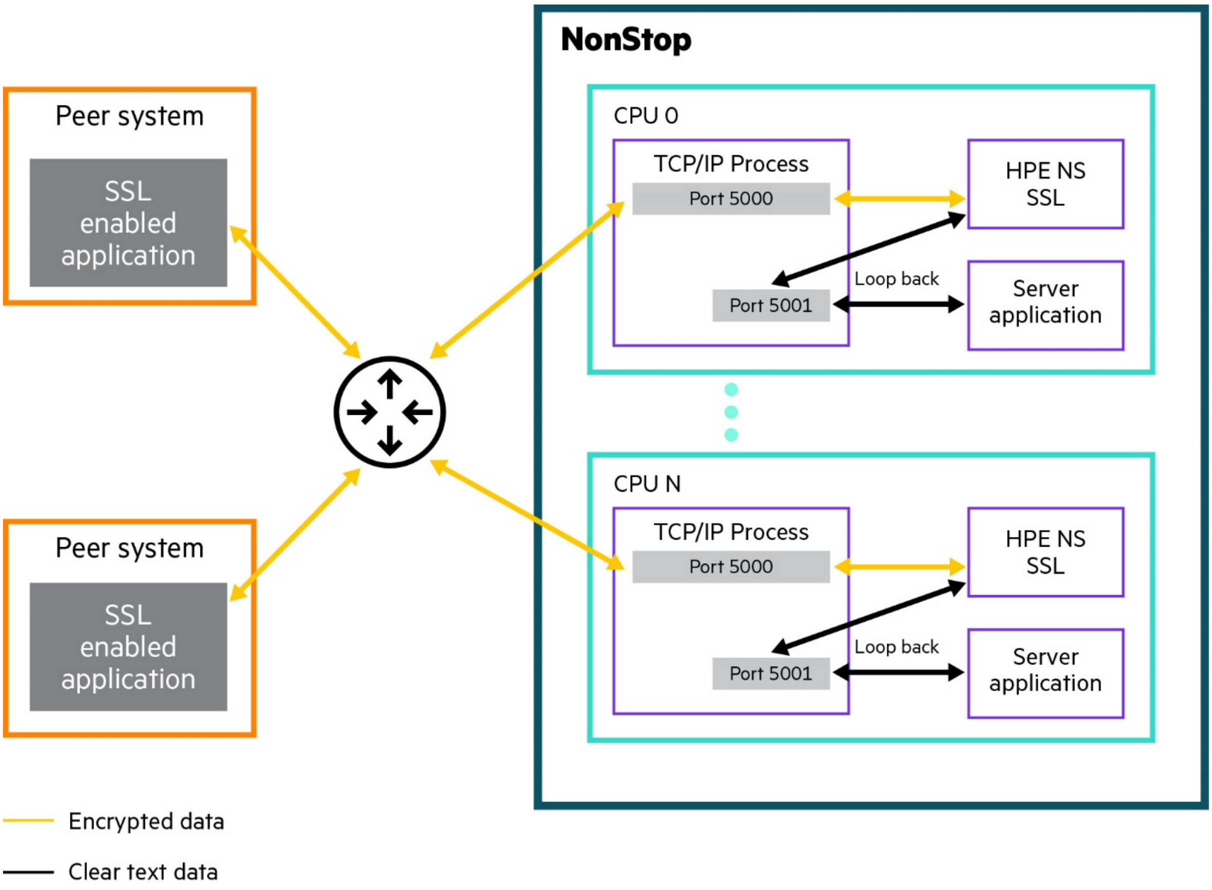


**FIGURE 2.** HPE NonStop SSL architecture

The sequence of events for a secure connection between the external client and the TCP server running in HPE NonStop are:

- The NonStop SSL proxy server listens on the port, which is advertised to the remote client as the one where the application server is also listening.
- The SSL client initiates the TCP/IP connection to the advertised port.
- The SSL proxy server accepts the TCP connection from the client.
- The client then initiates SSL handshake to establish SSL connection with the proxy server. If the SSL connection is not established for some reason, the TCP connection established in the preceding steps is terminated.
- After the successful establishment of the SSL connection, the SSL proxy server uses an unused local port to initiate the TCP/IP connection to the HPE NonStop application server port that is running in the NonStop system.
- The application server accepts the new incoming connection from the proxy server.
- The encrypted data received from the SSL client is decrypted by the HPE NonStop SSL and forwarded to the application server.
- Unencrypted data received from the server is encrypted by the NonStop SSL and sent to the SSL client.
- If another client wants another secure connection, these steps are repeated with the proxy server using a different unused local port for the connection with the application server.

These sequence of events is illustrated with a TCP dump:

- The HPE NonStop SSL proxy server is listening on 192.168.43.1:11000 (11000 is the port advertised to the remote client) and the remote client initiates the TCP/IP connection:

| Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|
| 192.168.44.2 | 192.168.43.1 | TCP | 76 | 41958 → 10000 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=287360… |
| 192.168.43.1 | 192.168.44.2 | TCP | 76 | 10000 → 41958 [SYN, ACK] Seq=0 Ack=1 Win=15928 Len=0 MSS=1460 SACK_PERM=1 T… |
| 192.168.44.2 | 192.168.43.1 | TCP | 68 | 41958 → 10000 [ACK] Seq=1 Ack=1 Win=14624 Len=0 TSval=2873605232 TSecr=4543… |

- The remote client initiates the SSL handshake:

| | | | | |
|---|---|---|---|---|
| 192.168.44.2 | 192.168.43.1 | TLSv1.2 | 358 | Client Hello |
| 192.168.43.1 | 192.168.44.2 | TCP | 68 | 10000 → 41958 [ACK] Seq=1 Ack=291 Win=15638 Len=0 TSval=454377406 TSecr=287… |
| 192.168.43.1 | 192.168.44.2 | TLSv1.2 | 1516 | Server Hello |
| 192.168.43.1 | 192.168.44.2 | TLSv1.2 | 416 | Certificate, Server Key Exchange, Server Hello Done |
| 192.168.44.2 | 192.168.43.1 | TCP | 68 | 41958 → 10000 [ACK] Seq=291 Ack=1449 Win=17504 Len=0 TSval=2873605239 TSecr… |
| 192.168.44.2 | 192.168.43.1 | TCP | 68 | 41958 → 10000 [ACK] Seq=291 Ack=1797 Win=20416 Len=0 TSval=2873605239 TSecr… |
| 192.168.44.2 | 192.168.43.1 | TLSv1.2 | 262 | Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message |
| 192.168.43.1 | 192.168.44.2 | TLSv1.2 | 326 | New Session Ticket, Change Cipher Spec, Encrypted Handshake Message |

- The SSL proxy server initiates the TCP/IP connection to the NonStop server application listening on 127.0.0.1:12000, a loop back address:

| | | | | |
|---|---|---|---|---|
| 127.0.0.1 | 127.0.0.1 | TCP | 76 | 61005 → 12000 [SYN] Seq=0 Win=16384 Len=0 MSS=65495 SACK_PERM=1 T… |
| 127.0.0.1 | 127.0.0.1 | TCP | 76 | 12000 → 61005 [SYN, ACK] Seq=0 Ack=1 Win=44000 Len=0 MSS=65495 SA… |
| 127.0.0.1 | 127.0.0.1 | TCP | 68 | 61005 → 12000 [ACK] Seq=1 Ack=1 Win=16384 Len=0 TSval=454377407 T… |

- The SSL proxy server receives the encrypted data from the remote client:

```
192.168.44.2   192.168.43.1   TLSv1.2  106   Application Data

› Frame 17: 106 bytes on wire (848 bits), 106 bytes captured (848 bits)
› Linux cooked capture
› Internet Protocol Version 4, Src: 192.168.44.2, Dst: 192.168.43.1
› Transmission Control Protocol, Src Port: 41958, Dst Port: 10000, Seq: 485, Ack: 2055, Len: 38
˅ Transport Layer Security
  ˅ TLSv1.2 Record Layer: Application Data Protocol: Application Data
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 33
      Encrypted Application Data: f40af0617c1c54fc4d235d85496f2dc2799f9e3582dfa7f7…

0000  00 00 00 01 00 06 d8 9d  67 18 99 32 00 00 08 00   ········ g··2····
0010  45 00 00 5a 81 ff 40 00  40 06 e0 4a c0 a8 2c 02   E··Z··@· @··J··,·
0020  c0 a8 2b 01 a3 e6 27 10  6f 96 3f 6e 5a 3a 95 6a   ··+···'· o·?nZ:·j
0030  80 18 02 d8 2a fa 00 00  01 01 08 0a ab 47 bc 81   ····*··· ·····G··
0040  1b 15 3f cf 17 03 03 00  21 f4 0a f0 61 7c 1c 54   ··?····· !···a|·T
0050  fc 4d 23 5d 85 49 6f 2d  c2 79 9f 9e 35 82 df a7   ·M#]·Io- ·y··5···
0060  f7 09 f3 cc b8 ee 5a 62  f5 d6                     ······Zb ··
```

- The SSL proxy server decrypts the traffic before forwarding it to the HPE NonStop server application (over the loop back address):

```
127.0.0.1     127.0.0.1     TCP     77    61005 → 12000 [PSH, ACK] Seq=1 Ack=1 Win=16384 Len=9 TSval=454377423 TSecr=…

› Frame 18: 77 bytes on wire (616 bits), 77 bytes captured (616 bits)
› Linux cooked capture
› Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
› Transmission Control Protocol, Src Port: 61005, Dst Port: 12000, Seq: 1, Ack: 1, Len: 9
˅ Data (9 bytes)
    Data: 486920536572766572
    [Length: 9]

0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ········ ········
0010  45 00 00 3d 78 f5 40 00  40 06 c3 c3 7f 00 00 01   E··=x·@· @·······
0020  7f 00 00 01 ee 4d 2e e0  c3 8f 47 4e 7c c5 15 99   ·····M.· ··GN|···
0030  80 18 40 00 fe 31 00 00  01 01 08 0a 1b 15 3f cf   ··@··1·· ······?·
0040  1b 15 3f bf 48 69 20 53  65 72 76 65 72            ··?·Hi S erver
```

- For outgoing traffic, the HPE NonStop server application will send the unencrypted data to the SSL proxy server:

```
127.0.0.1      127.0.0.1      TCP      77     12000 → 61005 [PSH, ACK] Seq=1 Ack=10 Win=44000 Len=9 TSval=454386877 TSecr…
› Linux cooked capture
› Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
› Transmission Control Protocol, Src Port: 12000, Dst Port: 61005, Seq: 1, Ack: 10, Len: 9
˅ Data (9 bytes)
    Data: 486920436c69656e74
    [Length: 9]

0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ········ ········
0010  45 00 00 3d 2a 28 40 00  40 06 12 91 7f 00 00 01   E··=*(@· @·······
0020  7f 00 00 01 2e e0 ee 4d  7c c5 15 99 c3 8f 47 57   ····.··M |·····GW
0030  80 18 05 5f fe 31 00 00  01 01 08 0a 1b 15 64 bd   ···_·1·· ······d·
0040  1b 15 3f cf 48 69 20 43  6c 69 65 6e 74            ··?·Hi C lient
```

- The SSL proxy server will encrypt the data from the NonStop server application before sending it out to the remote client:

```
192.168.43.1  192.168.44.2  TLSv1.2 106   Application Data
› Frame 25: 106 bytes on wire (848 bits), 106 bytes captured (848 bits)
› Linux cooked capture
› Internet Protocol Version 4, Src: 192.168.43.1, Dst: 192.168.44.2
› Transmission Control Protocol, Src Port: 10000, Dst Port: 41958, Seq: 2055, Ack: 523, Len: 38
˅ Transport Layer Security
  ˅ TLSv1.2 Record Layer: Application Data Protocol: Application Data
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 33
      Encrypted Application Data: c589f1791b3ad49afeaef26ca5e2a8ac0620bb6754c03067…

0000  00 04 00 01 00 06 1c 98  ec 2a ef 15 00 00 08 00   ········ ·*······
0010  45 00 00 5a b2 e0 40 00  40 06 af 69 c0 a8 2b 01   E··Z··@· @··i··+·
0020  c0 a8 2c 02 27 10 a3 e6  5a 3a 95 6a 6f 96 3f 94   ··,·'··· Z:·jo·?·
0030  80 18 3c 2e ee 66 00 00  01 01 08 0a 1b 15 64 bd   ··<.·f·· ······d·
0040  ab 47 bc 81 17 03 03 00  21 c5 89 f1 79 1b 3a d4   ·G······ !···y·:·
0050  9a fe ae f2 6c a5 e2 a8  ac 06 20 bb 67 54 c0 30   ····l··· ·· ·gT·0
0060  67 a4 16 0d 82 6b b0 8a  6a 6c                     g····k·· jl
```

## HPE NonStop SSL proxy client

HPE NonStop SSL acts as a client proxy for plain TCP/IP client application, to secure the communication between the NonStop client application and remote SSL-enabled server programs. The HPE NonStop SSL proxy will communicate with the client application using an unencrypted TCP/IP connection.

The sequence of events for a secure connection between a server and a client application running in HPE NonStop are:

- The client application running in NonStop initiates a TCP/IP connection to the SSL process on the port the SSL process is listening on.
- The SSL process accepts the TCP/IP connection from the HPE NonStop client application.
- After the successful establishment of the TCP/IP connection, the SSL process uses an unused local port to initiate a TCP/IP connection to the external server.
- The external server accepts the new incoming TCP/IP connection from the SSL proxy process.
- The HPE NonStop SSL process then initiates the SSL handshake to establish an SSL connection with the external server. If the SSL connection is not established for any reason, the TCP/IP connection established in the preceding steps is terminated.
- The unencrypted data received from the HPE NonStop client application is encrypted by the NonStop SSL proxy and then it is sent to the external server.
- The encrypted data received from the external server is decrypted by the HPE NonStop SSL process and forwarded to the client application.
- If another NonStop client application wants a secure connection, these steps are repeated with the SSL proxy using a different unused local port for the connection with the external server.

### HPE NonStop CPU for SSL process

The HPE NonStop SSL proxy and the server/client application on NonStop can either execute on the same NonStop CPU or different CPUs. If they execute on different CPUs, failure of the CPU executing the HPE NonStop SSL process will result in loss of the SSL session. Upon restarting the HPE NonStop SSL, connection to the user application needs to be re-established.

It is recommended to run both the SSL process and the NonStop server/client application on the same HPE NonStop CPU. It is also recommended to run the HPE NonStop SSL process on each of the NonStop CPUs that run the NonStop server/client applications.

### Single HPE NonStop SSL process for different NonStop applications

A single HPE NonStop SSL process can listen on multiple TCP/IP connections at once and forward data transferred through them to different NonStop client/server applications, each of which has its own TCP/IP connection. A single instance of the SSL process is sufficient to cater to multiple NonStop server/client applications.

### Securing communication between HPE NonStop SSL process and NonStop applications

The HPE NonStop SSL process and the NonStop server/client application can use the same or different HPE NonStop TCP/IP processes to communicate unencrypted data. If they use different TCP/IP processes to communicate, then there is a possibility of unencrypted data traversing the network where it could be observed by outside attackers. So for communication between the HPE NonStop SSL process and the NonStop application, use of the local loopback (127.0.0.1) address rather than any other address configured in the same HPE NonStop TCP/IP process is recommended.

Also, if the NonStop server application listens on a non-loopback address, then the port it is listening on is potentially open for security attacks from outside the system. Therefore, the NonStop server application should not listen on INADDR_ANY and instead should listen only on the local loopback address if it’s running on the same CPU as the HPE NonStop SSL process.

### Fault-Tolerance

The HPE NonStop SSL services can be configured as persistent processes under control of the kernel subsystem, enabling automatic recovery from failures, such as CPU outages. HPE NonStop SSL cannot be run as a nonstop process (process-pair). Running as a process pair would not add value, as TCP sessions are reset upon CPU takeover. Protection against CPU failure is achieved with HPE NonStop SSL configured as a persistent process that is automatically restarted upon failures.

The restart of the HPE NonStop SSL process results in termination of the connection with the remote (server/client) application. If the remote application is a client application, then it needs to re-establish the SSL connection. If the remote application is a server application, then the client application running on the HPE NonStop system will re-establish the SSL connection with the server.

### Load balancing

For load balancing the incoming TCP/IP connections, the underlying HPE NonStop TCP/IP stack allows multiple instances of the same application to be run on different CPUs and the stack will uniformly distribute the incoming connections to all the CPUs. This feature is called Round Robin filtering. The Round Robin filtering can be used along with HPE NonStop SSL to uniformly distribute the incoming SSL sessions across multiple CPUs.

## IPSec

Internet Protocol Security (IPSec) is a protocol suite for securing IP communications by encrypting and optionally authenticating each IP packet of a communication session. IPSec can be used in protecting data flows between a pair of connected devices. The pair of devices can be two systems, two gateways, or a system and a gateway.

IPSec is an end-to-end security scheme operating in the internet layer of the IP suite. IPSec protects any application traffic over an IP network, so most application data can be automatically secured by IPSec at the IP layer.

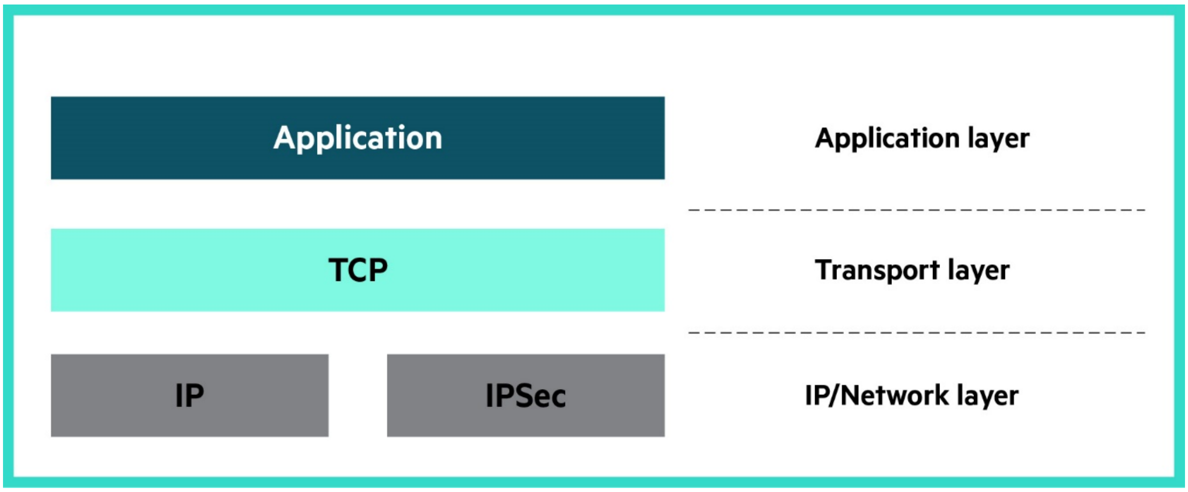


**FIGURE 3.** IPSec in OSI

## Establishing an IPSec session

The first packet from an application running on a host triggers the IPSec process to establish an IPSec session called a SA with its peer. The first packet might end up being dropped if the IPSec session is not already present. The IPSec session is established by the IPSec process using UDP (Port 500). The actual TCP/UDP port to which the data packet is communicated and the IPSec session establishment UDP port are different. So unlike SSL (which secures only TCP data packets), both TCP and UDP data packets can be secured through IPSec.

## IPSec features

IPSec provides the following security features when transferring packets across networks:

- **Authentication:** Verifies that the packet received is actually from the claimed sender.
- **Integrity:** Ensures that the contents of the packet did not change in transit.
- **Confidentiality:** Conceals the message content through encryption.
- **Compression:** Compresses IP datagrams to improve performance.

IPSec provides a high level of security by using cryptography-based services. Cryptography allows information to be transmitted securely by hashing (to provide integrity and authentication) and encrypting (to provide confidentiality) the data. IPSec can be used in two modes, transport mode and tunnel mode.

## IPSec in HPE NonStop CIP

HPE NonStop CIP supports IPSec, IPSec secures the data at the IP layer. The IPSec process runs in the HPE NonStop CLIM, which is part of the CIP subsystem. The IPSec process provides the IKE key management protocol. Figure 4 shows the encrypted and unencrypted data flow in HPE NonStop CIP IPSec.

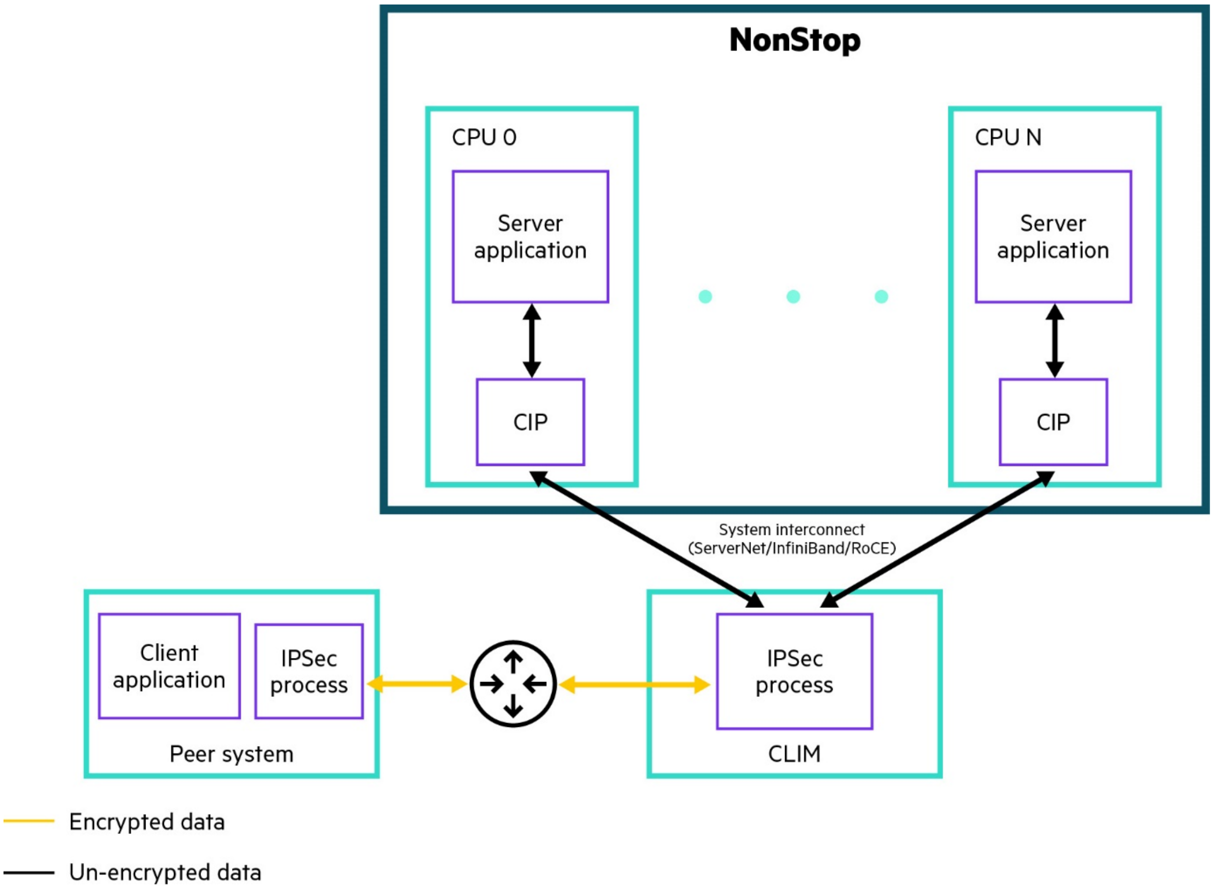


**FIGURE 4.** HPE NonStop CIP IPSec architecture

The sequence of events for secure communication between the NonStop application and the remote peer is stated as follows:

- Data from a NonStop application is forwarded as unencrypted data from the NonStop CPU to the CLIM.
- The first data packet will trigger the IPSec process running in the CLIM to establish the IPSec SA with the peer system.
- The IPSec SA negotiation is initiated on UDP Port 500.
- Once the IPSec SA is established, all further data from the NonStop application is encrypted by the IPSec process at the CLIM and the encrypted data is sent to the peer system.
- The encrypted data from the peer end is decrypted at the CLIM (under the direction of IPSec process) and the decrypted data is forwarded via the system interconnect (ServerNet interconnect in J-series system and InfiniBand/RoCE interconnect in the L-series systems) to the NonStop application running on the NonStop CPU.

For configuration of HPE NonStop CIP IPSec and more details on the modes in which IPSec can be used, refer to the HPE NonStop IPSec overview technical white paper.

### HPE NonStop CPU for IPSec process

The CIP IPSec process executes on the CLIM and does not use the HPE NonStop CPU cycles. The cryptography is carried out on the CLIM.

### Fault-Tolerance

IPSec configurations are not migrated during an interface failover. It is expected that the IPSec configuration on the home and failover CLIM is identical for addresses that can fail over between the CLIMs. Any packet originating from the HPE NonStop CPU after the interface failover results in the IPSec SA re-negotiation. This is because, the failover CLIM does not have any IPSec SA for the failed over IP address so, the packets transmitted after the failover triggers the IPSec SA re-negotiation (the packet that initiated the IPSec SA re-negotiation is dropped). This leads to automatically re-negotiating the SAs from the CLIM end upon interface failover.

However, if there are packets originating from the peer system after the interface failover, the IPSec SA is not re-negotiated (this is because the SAs still exist at the peer system). To overcome this issue, IPSec provides a feature called Dead Peer Detection (DPD) for detecting the existence and availability of the remote end. Through the DPD, the peer system detects the non-existence of the IPSec SA at the remote end after failover. The non-existence of the remote end IPSec SA results in the peer system removing the IPSec SAs at its end. Once the existing IPSec SAs are purged, new packets from the peer system after the interface failover will result in automatic re-negotiation of the new IPSec SA.

## Load balancing

Round Robin filters for load balancing work without modification with IPSec.

## Communication between IPSec process and NonStop applications

The unencrypted data is communicated between the IPSec process and the NonStop applications over the system interconnect between the CLIM and the HPE NonStop CPU. This data does not traverse the network, and, therefore, there is no possible exposure of unencrypted customer data to outside attackers listening on the network when using IPSec.

## Comparison of HPE NonStop SSL and HPE NonStop CIP IPSec

| | **SSL** | **IPSec** |
|---|---|---|
| **Configuration** | The configuration can be supplied through TACL PARAMs, the configuration file and start up line parameters. | Can be configured through CLIMCONFIG invoked via CLIMCMD. |
| **Secures communication** | • Two systems | • Two systems<br>• Two gateways<br>• System and gateway |
| **Session establishment** | • TCP connection is established first<br>• SSL session is established<br>• The data is encrypted before transmitting | • IPSec session is established<br>• The data is encrypted before transmitting |
| **Security dimension** | • Authentication of peer<br>• Confidentiality of data<br>• Integrity and authenticity of data | • Authentication of peer<br>• Confidentiality of data<br>• Integrity and authenticity of data |
| **Authentication** | Supports:<br>• Public key certificate based authentication (X.509 certificate); dynamic refresh of certificates supported<br>• Optionally supports client authentication also | Supports:<br>• Public key certificate based authentication (X.509 certificate); dynamic refresh of certificates supported<br>• Manual keying<br>• Pre-shared key authentication |
| **Encryption of data** | Supported | Supported |
| **Compression of data** | Not supported | Supported |
| **Integrity and authenticity of every message** | Supported | Supported |
| **CPU cycles used** | HPE NonStop CPU cycles are used | CLIM CPU cycles are used |
| **OSI layer in which it operates** | Application layer | IP layer |
| **Survival of security sessions on interface failover** | The SSL session survives the interface failover provided, the TCP sessions survive the interface failover | No, the IPSec sessions need to be re-negotiated upon interface failover |
| **Survival of security sessions on HPE NonStop CPU failure** | No, that is why the HPE NonStop SSL process and the NonStop client/server application must run on the same CPU | Yes. As the IPSec process runs in the CLIM, an HPE NonStop CPU failure does not affect the IPSec sessions |
| **Can the security process be configured as process pair?** | No | No |
| **Can the security process be configured as persistent process?** | Yes | No; since the IPSec process runs in CLIM, it cannot be configured to execute as a persistent process. |
| **Supported protocol version** | SSL 3.0, TLS 1.0 (SSL 3.1), TLS 1.1 and TLS 1.2 are supported<br>**Note:** Usage of SSL 3.0 or TLS 1.0 is not recommended, as they have known design flaws that can be exploited to breach security. | IKEv1 |
| **Needs change in the application code to support** | There is usually no need for the application to go through the change to have security for the data it communicates with the outside world. HPE NonStop SSL runs as a proxy process providing the SSL security features for the NonStop application.<br>If the application manages distribution of sockets in an application-specific fashion rather than just acting as a simple client or server then it may require some changes. | No, there is no need for the application to go through any change to have security for the data it communicates with the outside world. HPE NonStop IPSec runs in the CLIM and provides the IPSec security features for the NonStop application. |

| | SSL | IPSec |
|---|---|---|
| **Scalability** | Both the HPE NonStop SSL proxy process and the user application needs to run in the same HPE NonStop CPU. As one adds more SSL sessions, the CPU consumption will rise due to both the SSL process and the user processes using additional CPU cycles.<br>The CLIM CPU cycles used do not increase as additional SSL sessions are added.<br>The HPE NonStop SSL proxy process and the user application need to be run on the same CPU to avoid security issues. | The IPSec process runs in the IP CLIM and the user processes runs in the HPE NonStop CPU. As one adds more IPSec sessions, the CLIM CPU consumption will rise only due to the user processes using additional CPU cycles.<br>The CLIM CPU cycles used increase as additional IPSec sessions are added.<br>User applications can be run on any CPU without any concerns for security. |
| **TCP/IP stacks supporting the security solution** | • HPE NonStop TCP/IP<br>• HPE Parallel TCP/IP<br>• HPE NonStop TCP/IPv6<br>• HPE NonStop CIP | • HPE NonStop CIP |
| **Support for IPv4 and IPv6 versions** | • Supports both IPv4 and IPv6 | • Supports both IPv4 and IPv6 |
| **Support for TCP** | Yes | Yes |
| **Support for UDP** | No | Yes |
| **Support for Round Robin filtering** | Yes | Yes |
| **Logging** | Different log levels are provided, which provide various details of the log. | Different log levels are provided, which provide various details of the log. |

## References


- HPE NonStop IPSec Overview (Technical white paper)
- HPE NonStop SSL reference manual
- HPE NonStop Cluster I/O Protocols (CIP) Configuration and Management Manual
  (detailed configuration information about the CIP subsystem, which supports the IP CLIM)
- HPE NonStop Cluster I/O Protocols Configuration and Management Manual Technical Update
- The following RFCs (the RFC details can be obtained from the internet)
  - RFC 2246 The TLS Protocol Version 1.0
  - RFC 4346 The Transport Layer Security (TLS) Protocol Version 1.1
  - RFC 5246 The Transport Layer Security (TLS) Protocol Version 1.2


## LEARN MORE AT

hpe.com/info/nonstop



a50000347ENW, November 2019