



Sun™ Crypto Accelerator 4000 Board Version 1.1 Installation and User's Guide

Sun Microsystems, Inc.
www.sun.com

Part No. 817-3693-10
January 2004, Revision A

Submit comments about this document at: <http://www.sun.com/hwdocs/feedback>

Copyright 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

This product or document is distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, SunVTS, AnswerBook2, docs.sun.com, Sun ONE, Sun Enterprise, Sun Enterprise Volume Manager, Sun Fire, SunSolve, Netra, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc. Netscape is a trademark or registered trademark of Netscape Communications Corporation. This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>). This product includes cryptographic software written by Eric Young (ey@cryptsoft.com). This product includes software developed by Ralf S. Engelschall <rse@engelschall.com> for use in the mod_ssl project (<http://www.modssl.org/>).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, CA 95054 Etats-Unis. Tous droits réservés.

Ce produit ou document est distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, SunVTS, AnswerBook2, docs.sun.com, Sun ONE, Sun Enterprise, Sun Enterprise Volume Manager, Sun Fire, SunSolve, Netra, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. Netscape est une marque de Netscape Communications Corporation aux Etats-Unis et dans d'autres pays. Ce produit comprend le logiciel développé par le Project OpenSSL pour l'utilisation dans le Toolkit OpenSSL (<http://www.openssl.org/>). Ce produit comprend le logiciel cryptographique écrite par Eric Young (ey@cryptsoft.com). Ce produit comprend le logiciel développé par Ralf S. Engelschall <rse@engelschall.com> pour l'utilisation dans le projet mod_ssl (<http://www.modssl.org/>).

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.



Contents

1. Product Overview	1
Product Features	1
Key Protocols and Interfaces	2
Key Features	2
Supported Applications	2
Supported Cryptographic Protocols	3
Diagnostic Support	3
Cryptographic Algorithm Acceleration	3
Supported Cryptographic Algorithms	4
IPsec Acceleration	4
SSL Acceleration	5
Bulk Encryption	6
Hardware Overview	6
Sun Crypto Accelerator 4000 MMF Adapter	6
LED Displays	8
Sun Crypto Accelerator 4000 UTP Adapter	8
LED Displays	10
Dynamic Reconfiguration and High Availability	10
Load Sharing	11

Hardware and Software Requirements	11
Required Patches	12
Apache Web Server Patch	12
Solaris 8 Patches	12
Solaris 9 Patches	13
2. Installing the Sun Crypto Accelerator 4000 Board	15
Handling the Board	15
Installing the Board	16
▼ To Install the Hardware	16
Installing the Sun Crypto Accelerator 4000 Software	18
▼ To Install the Software	18
Choosing the Optional Packages to Install	21
Directories and Files	22
Removing the Sun Crypto Accelerator 4000 Software	24
▼ To Remove the Software With the <code>remove</code> Script	24
▼ To Remove the Software With the <code>/var/tmp/crypto_acc.remove</code> Script	24
3. Configuring Driver Parameters	25
Ethernet Device Driver (<code>vca</code>) Parameters	25
Driver Parameter Values and Definitions	26
Advertised Link Parameters	27
Flow Control Parameters	29
Gigabit Forced Mode Parameter	30
Interpacket Gap Parameters	30
Interrupt Parameters	31
Random Early Drop Parameters	32
PCI Bus Interface Parameters	33

Setting vca Driver Parameters	34
Setting Parameters Using the ndd Utility	34
▼ To Specify Device Instances for the ndd Utility	34
Noninteractive and Interactive Modes	35
Setting Autonegotiation or Forced Mode	38
▼ To Disable Autonegotiation Mode	38
Setting Parameters Using the vca.conf File	39
▼ To Set Driver Parameters Using a vca.conf File	39
Setting Parameters for All Sun Crypto Accelerator 4000 vca Devices With the vca.conf File	41
▼ To Set Parameters for All Sun Crypto Accelerator 4000 vca Devices With the vca.conf File	41
Example vca.conf File	41
Enabling Autonegotiation or Forced Mode for Link Parameters With the OpenBoot PROM	42
Cryptographic and Ethernet Driver Operating Statistics	44
Cryptographic Driver Statistics	44
Ethernet Driver Statistics	45
Reporting the Link Partner Capabilities	49
▼ To Check Link Partner Settings	52
IPsec In-Line Acceleration Statistics	53
Network Configuration	54
Configuring the Network Host Files	54
Configuring IPsec Hardware Acceleration	56
Enabling Out-of-Band IPsec Acceleration	57
Enabling In-Line IPsec Acceleration	57
▼ To Enable In-Line IPsec Hardware Acceleration	57
4. Administering the Sun Crypto Accelerator 4000 Board	59

Using the vcaadm Utility	59
Modes of Operation	61
Single-Command Mode	61
File Mode	62
Interactive Mode	62
Logging In and Out With vcaadm	62
Logging In to a Board With vcaadm	63
Logging Out of a Board With vcaadm	65
Entering Commands With vcaadm	66
Getting Help for Commands	67
Quitting the vcaadm Utility in Interactive Mode	68
Initializing the Board With vcaadm	68
▼ To Initialize the Board With a New Keystore	69
Initializing the Board to Use an Existing Keystore	70
▼ To Initialize the Board to Use an Existing Keystore	71
Managing Keystores With vcaadm	71
Naming Requirements	72
Password Requirements	72
Populating a Keystore With Security Officers	73
Populating a Keystore With Users	74
Listing Users and Security Officers	75
Changing Passwords	75
Enabling or Disabling Users	76
Deleting Users	77
Deleting Security Officers	77
Backing Up the Master Key	77
Locking the Keystore to Prevent Backups	78
Managing Boards With vcaadm	78

Setting the Auto-Logout Time	79
Displaying Board Status	79
Loading New Firmware	80
Resetting the Board	80
Rekeying the Board	81
Performing a Software Zeroize on the Board	82
Using the <code>vcaadm diagnostics</code> Command	83
Using the <code>vcad</code> Command	83
<code>vcad</code> Configuration File	85
<code>vcad</code> Daemon Security	87
▼ To Configure the <code>vcad</code> Daemon to Run as a Different Username	87
Using the <code>vcadiag</code> Utility	89
Using the <code>pk11export</code> Utility	92
Using the <code>iplsslcfg</code> Script	93
▼ To Use Option 1 of the <code>iplsslcfg</code> Script for Sun ONE Web Server 4.1	93
▼ To Use Option 1 of the <code>iplsslcfg</code> Script for Sun ONE Web Server 6.0	93
▼ To Use Option 2 of the <code>iplsslcfg</code> Script	94
▼ To Use Option 3 of the <code>iplsslcfg</code> Script	95
▼ To Use Option 4 of the <code>iplsslcfg</code> Script	96
Using the <code>apsslcfg</code> Script	98
▼ To Use Option 1 of the <code>apsslcfg</code> Script	98
Using Option 2 of the <code>apsslcfg</code> Script	98
▼ To Generate a Keypair and Request a Certificate for Apache	98
▼ To Export Apache (PEM Encoded X.509) Keys to PKCS#12 Format	100
▼ To Import Keys From PKCS#12 Format to Apache (PEM encoded X.509)	101
Assigning Different MAC Addresses to Multiple Boards Installed in the Same Server	103
▼ To Assign Different MAC Addresses From a Terminal Window	103

▼	To Assign Different MAC Addresses From the OpenBoot PROM Level	103
5.	Installing and Configuring Sun ONE Server Software	105
	Administering Security for Sun ONE Web Servers	105
	Concepts and Terminology	106
	Tokens and Token Files	108
	Token Files	108
	Enabling and Disabling Bulk Encryption	109
	Configuring Sun ONE Web Servers	110
	Passwords	110
	Populating a Keystore	111
	▼ To Populate a Keystore	111
	Overview of Enabling Sun ONE Web Servers	112
	Configuring Sun ONE Web Servers to Start Up Without User Interaction on Reboot	113
	▼ To Create an Encrypted Key for Automatic Startup of Sun ONE Web Servers on Reboot	113
	Installing and Configuring Sun ONE Web Server 4.1	113
	▼ To Install Sun ONE Web Server 4.1	114
	Configuring Sun ONE Web Server 4.1	114
	▼ To Create a Trust Database	115
	▼ To Register the Board With the Web Server	116
	▼ To Generate a Server Certificate	117
	▼ To Install the Server Certificate	120
	▼ To Enable the Web Server for SSL	121
	Installing and Configuring Sun ONE Web Server 6.0	123
	▼ To Install Sun ONE Web Server 6.0	123
	Configuring Sun ONE Web Server 6.0	124

- ▼ To Create a Trust Database 124
- ▼ To Register the Board With the Web Server 125
- ▼ To Generate a Server Certificate 127
- ▼ To Install the Server Certificate 130
- ▼ To Enable the Web Server for SSL 131

Installing and Configuring Sun ONE Application Server 7 133

- ▼ To Install Sun ONE Application Server 7 133
- ▼ To Install the Sun ONE Application Server Add-Ons Software 135

Configuring Sun ONE Application Server 7 135

- ▼ To Create a Trust Database 136
- ▼ To Register the Board With the Application Server 137
- ▼ To Generate a Server Certificate 139
- ▼ To Install the Server Certificate 141
- ▼ To Enable the Application Server for SSL 142

Installing and Configuring Sun ONE Directory Server 5.2 146

Installing Sun ONE Directory Server 5.2 146

- ▼ To Install Sun ONE Directory Server 5.2 146

Configuring Sun ONE Directory Server 5.2 147

- ▼ To Create a Trust Database 147
- ▼ To Register the Board With the Directory Server (32-Bit) 149
- ▼ To Register the Board With the Directory Server (64-Bit) 150

Generating and Installing a Server Certificate 151

- ▼ To Generate a Server Certificate 151
- ▼ To Install the Server Certificate 152

Viewing and Installing Root CA Certificates 152

- ▼ To View Root CA Certificates Known to the Directory Server 152
- ▼ To Install Root CA Certificates 153

▼	To Enable the Directory Server for SSL	154
	Installing and Configuring Sun ONE Messaging Server 5.2	158
	Installing Sun ONE Messaging Server 5.2	158
▼	To Install Sun ONE Messaging Server 5.2	158
	Configuring Sun ONE Messaging Server 5.2	158
▼	To Create a Trust Database	159
▼	To Register the Board With the Messaging Server	160
▼	To Generate a Server Certificate	160
▼	To Install the Server Certificate	165
▼	To Enable the Messaging Server for SSL	168
	Installing and Configuring Sun ONE Portal Server 6.2	169
	Installing Sun ONE Portal Server 6.2	170
▼	To Install Sun ONE Portal Server 6.2	170
	Configuring Sun ONE Portal Server 6.2	171
▼	To Register the Board With the Portal Server	171
	Generating and Installing a Server Certificate	172
▼	To Generate a Server Certificate	172
▼	To Install the Server Certificate	173
	Viewing and Installing Root CA Certificates	173
▼	To View Root CA Certificates Known to the Portal Server	173
▼	To Install Root CA Certificates	173
▼	To Enable the Portal Server for SSL	174
6.	Installing and Configuring Apache Web Server Software	175
	Configuring Apache Web Server 1.3x	176
▼	To Configure Apache Web Server	176
▼	To Generate a Server Certificate	178
▼	To Install the Server Certificate	182

Building and Configuring Apache Web Server 2.x	182
Building Apache 2.x Web Server	183
▼ To Build Apache 2.x	183
Configuring Apache Web Server 2.x	184
▼ To Generate a Server Certificate	184
▼ To Install the Server Certificate	185
▼ To Enable SSL	185
Configuring the Apache Web Server to Start Up Without User Interaction on Reboot	186
▼ To Create an Encrypted Key for Automatic Startup of Apache Web Server on Reboot	186
Configuring the Sun Crypto Accelerator 1000 for Use With Apache After the Sun Crypto Accelerator 4000 Software is Installed	187
7. Diagnostics and Troubleshooting	189
SunVTS Diagnostic Software	189
Installing SunVTS <code>netlbtst</code> and <code>nettest</code> Support for the <code>vca</code> Driver	190
Using SunVTS Software to Perform <code>vcatest</code> , <code>nettest</code> , and <code>netlbtst</code>	191
▼ To Perform <code>vcatest</code>	191
Test Parameter Options for <code>vcatest</code>	193
<code>vcatest</code> Command-Line Syntax	193
▼ To Perform <code>netlbtst</code>	194
▼ To Perform <code>nettest</code>	195
Using <code>kstat</code> to Determine Cryptographic Activity	198
Using the OpenBoot PROM FCode Self-Test	199
▼ Performing the Ethernet FCode Self-Test Diagnostic	199
Troubleshooting the Sun Crypto Accelerator 4000 Board	202
<code>show-devs</code>	202

.properties 203

watch-net 204

8. PKCS#11 Interface 205

General Issues 205

Administering the Board to Use PKCS#11 206

Installing and Administering Applications That Use Cryptographic Services 207

PKCS#11 and FIPS Mode 208

Hardware Acceleration and Sensitive Keys 209

Developing Applications to Use PKCS#11 211

A. Specifications 219

Sun Crypto Accelerator 4000 MMF Adapter 219

Connectors 219

Physical Dimensions 221

Performance Specifications 221

Power Requirements 221

Interface Specifications 222

Environmental Specifications 222

Sun Crypto Accelerator 4000 UTP Adapter 222

Connectors 222

Physical Dimensions 224

Performance Specifications 224

Power Requirements 224

Interface Specifications 225

Environmental Specifications 225

B. Installing the Software Without the Installation Script 227

Installing the Software Manually 227

- ▼ To Install the Software Manually 227
 - Installing the Optional Packages 229
 - Directories and Files 230
 - Removing the Software Manually 231
- ▼ To Remove the Software Manually 232
- C. SSL Configuration Directives for Apache Web Servers 233**
- D. Configuring Custom Applications to Use the Board 241**
 - Configuring Custom Applications to Use the Board 241
 - ▼ To Configure Custom Applications to Use the Board 241
- E. Software Licenses 243**
 - Third Party License Terms 246
- F. Manual Pages 251**
- G. Zeroizing the Hardware 253**
 - Zeroizing the Sun Crypto Accelerator 4000 Hardware to the Factory State 253
 - ▼ To Zeroize the Sun Crypto Accelerator 4000 Board With a Hardware Jumper 254

Declaration of Conformity (Fiber MMF)

Compliance Model Number: Venus-FI
Product Family Name: Sun Crypto Accelerator 4000 - Fiber (X4012A)

EMC

USA - FCC Class B

This equipment complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

- 1) This equipment may not cause harmful interference.
- 2) This equipment must accept any interference that may cause undesired operation.

European Union

This equipment complies with the following requirements of the EMC Directive 89/336/EEC:

As Telecommunication Network Equipment (TNE) in both Telecom Centers and Other Than Telecom Centers per (as applicable):

EN300-386 V.1.3.1 (09-2001) Required Limits:

EN55022/CISPR22	Class B
EN61000-3-2	Pass
EN61000-3-3	Pass
EN61000-4-2	6 kV (Direct), 8 kV (Air)
EN61000-4-3	3 V/m 80-1000MHz, 10 V/m 800-960 MHz and 1400-2000 MHz
EN61000-4-4	1 kV AC and DC Power Lines, 0.5 kV Signal Lines,
EN61000-4-5	2 kV AC Line-Gnd, 1 kV AC Line-Line and Outdoor Signal Lines, 0.5 kV Indoor Signal Lines > 10m.
EN61000-4-6	3 V
EN61000-4-11	Pass

As information Technology Equipment (ITE) Class B per (as applicable):

EN55022:1998/CISPR22:1997 Class B

EN55024:1998 Required Limits:

EN61000-4-2	4 kV (Direct), 8 kV (Air)
EN61000-4-3	3 V/m
EN61000-4-4	1 kV AC Power Lines, 0.5 kV Signal and DC Power Lines
EN61000-4-5	1 kV AC Line-Line and Outdoor Signal Lines, 2 kV AC Line-Gnd, 0.5 kV DC Power Lines
EN61000-4-6	3 V
EN61000-4-8	1 A/m
EN61000-4-11	Pass

EN61000-3-2:1995 + A1, A2, A14 Pass

EN61000-3-3:1995 Pass

Safety

This equipment complies with the following requirements of the Low Voltage Directive 73/23/EEC:

EC Type Examination Certificates:

EN 60950:2000, 3rd Edition

IEC 60950:2000, 3rd Edition

Evaluated to all CB Countries

UL 60950, 3rd Edition, CSA C22.2 No. 60950-00

Supplementary Information

This product was tested and complies with all the requirements for the CE Mark.

/S/

Dennis P. Symanski
Manager, Compliance Engineering
Sun Microsystems, Inc.
4150 Network Circle, MPK15-102
Santa Clara, CA 95054, USA
Tel: 650-786-3255
Fax: 650-786-3723

/S/

Pamela J Dullaghan
Quality Program Manager
Sun Microsystems Scotland, Limited
Springfield, Linlithgow
West Lothian, EH49 7LR
Scotland, United Kingdom
Tel: +44 1 506 672 395
Fax: +44 1 506 672 855

Declaration of Conformity (Copper UTP)

Compliance Model Number: Venus-CU

Product Family Name: Sun Crypto Accelerator 4000 - Copper (X4011A)

EMC

USA - FCC Class B

This equipment complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

- 1) This equipment may not cause harmful interference.
- 2) This equipment must accept any interference that may cause undesired operation.

European Union

This equipment complies with the following requirements of the EMC Directive 89/336/EEC:

As Telecommunication Network Equipment (TNE) in both Telecom Centers and Other Than Telecom Centers per (as applicable):

EN300-386 V.1.3.1 (09-2001) Required Limits:

EN55022/CISPR22	Class B
EN61000-3-2	Pass
EN61000-3-3	Pass

EN61000-4-2	6 kV (Direct), 8 kV (Air)
EN61000-4-3	3 V/m 80-1000MHz, 10 V/m 800-960 MHz and 1400-2000 MHz
EN61000-4-4	1 kV AC and DC Power Lines, 0.5 kV Signal Lines,
EN61000-4-5	2 kV AC Line-Gnd, 1 kV AC Line-Line and Outdoor Signal Lines, 0.5 kV Indoor Signal Lines > 10m.
EN61000-4-6	3 V
EN61000-4-11	Pass

As information Technology Equipment (ITE) Class B per (as applicable):

EN55022:1998/CISPR22:1997 Class B

EN55024:1998 Required Limits:

EN61000-4-2	4 kV (Direct), 8 kV (Air)
EN61000-4-3	3 V/m
EN61000-4-4	1 kV AC Power Lines, 0.5 kV Signal and DC Power Lines
EN61000-4-5	1 kV AC Line-Line and Outdoor Signal Lines, 2 kV AC Line-Gnd, 0.5 kV DC Power Lines
EN61000-4-6	3 V
EN61000-4-8	1 A/m
EN61000-4-11	Pass

EN61000-3-2:1995 + A1, A2, A14 Pass

EN61000-3-3:1995 Pass

Safety

This equipment complies with the following requirements of the Low Voltage Directive 73/23/EEC:

EC Type Examination Certificates:

EN 60950:2000, 3rd Edition

IEC 60950:2000, 3rd Edition

Evaluated to all CB Countries

UL 60950, 3rd Edition, CSA C22.2 No. 60950-00

Supplementary Information

This product was tested and complies with all the requirements for the CE Mark.

/S/

Dennis P. Symanski
 Manager, Compliance Engineering
 Sun Microsystems, Inc.
 4150 Network Circle, MPK15-102
 Santa Clara, CA 95054, USA
 Tel: 650-786-3255
 Fax: 650-786-3723

/S/

Pamela J Dullaghan
 Quality Program Manager
 Sun Microsystems Scotland, Limited
 Springfield, Linlithgow
 West Lothian, EH49 7LR
 Scotland, United Kingdom
 Tel: +44 1 506 672 395
 Fax: +44 1 506 672 855

Regulatory Compliance Statements

Your Sun product is marked to indicate its compliance class:

- Federal Communications Commission (FCC) — USA
- Industry Canada Equipment Standard for Digital Equipment (ICES-003) — Canada
- Voluntary Control Council for Interference (VCCI) — Japan
- Bureau of Standards Metrology and Inspection (BSMI) — Taiwan

Please read the appropriate section that corresponds to the marking on your Sun product before attempting to install the product.

FCC Class A Notice

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

1. This device may not cause harmful interference.
2. This device must accept any interference received, including interference that may cause undesired operation.

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy, and if it is not installed and used in accordance with the instruction manual, it may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his own expense.

Shielded Cables: Connections between the workstation and peripherals must be made using shielded cables to comply with FCC radio frequency emission limits. Networking connections can be made using unshielded twisted-pair (UTP) cables.

Modifications: Any modifications made to this device that are not approved by Sun Microsystems, Inc. may void the authority granted to the user by the FCC to operate this equipment.

FCC Class B Notice

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

1. This device may not cause harmful interference.
2. This device must accept any interference received, including interference that may cause undesired operation.

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/television technician for help.

Shielded Cables: Connections between the workstation and peripherals must be made using shielded cables in order to maintain compliance with FCC radio frequency emission limits. Networking connections can be made using unshielded twisted pair (UTP) cables.

Modifications: Any modifications made to this device that are not approved by Sun Microsystems, Inc. may void the authority granted to the user by the FCC to operate this equipment.

ICES-003 Class A Notice - Avis NMB-003, Classe A

This Class A digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la classe A est conforme à la norme NMB-003 du Canada.

ICES-003 Class B Notice - Avis NMB-003, Classe B

This Class B digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la classe B est conforme à la norme NMB-003 du Canada.

VCCI 基準について

クラス A VCCI 基準について

クラス A VCCI の表示があるワークステーションおよびオプション製品は、クラス A 情報技術装置です。これらの製品には、下記の項目が該当します。

この装置は、情報処理装置等電波障害自主規制協議会 (VCCI) の基準に基づくクラス A 情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

クラス B VCCI 基準について

クラス B VCCI の表示  があるワークステーションおよびオプション製品は、クラス B 情報技術装置です。これらの製品には、下記の項目が該当します。

この装置は、情報処理装置等電波障害自主規制協議会 (VCCI) の基準に基づくクラス B 情報技術装置です。この装置は、家庭環境で使用することを目的としていますが、この装置がラジオやテレビジョン受信機に近接して使用されると、受信障害を引き起こすことがあります。取扱説明書に従って正しい取り扱いをしてください。

BSMI Class A Notice

The following statement is applicable to products shipped to Taiwan and marked as Class A on the product compliance label.

警告使用者：
這是甲類的資訊產品，在居住的環境中使用時，可能會造成射頻干擾，在這種情況下，使用者會被要求採取某些適當的對策。

Tables

TABLE 1-1	IPsec Cryptographic Algorithms	4
TABLE 1-2	SSL Cryptographic Algorithms	4
TABLE 1-3	Accelerated IPsec Algorithms	4
TABLE 1-4	Supported SSL Algorithms	5
TABLE 1-5	Front Panel Display LEDs for the MMF Adapter	8
TABLE 1-6	Front Panel Display LEDs for the UTP Adapter	10
TABLE 1-7	Hardware and Software Requirements	11
TABLE 1-8	Required Solaris 8 Patches	12
TABLE 1-9	Required Solaris 9 Patches	13
TABLE 2-1	Files in the <code>/cdrom/cdrom0</code> Directory	19
TABLE 2-2	Sun Crypto Accelerator 4000 Directories	22
TABLE 3-1	<code>vca</code> Driver Parameter, Status, and Descriptions	26
TABLE 3-2	Operational Mode Parameters	28
TABLE 3-3	Read-Write Flow Control Keyword Descriptions	29
TABLE 3-4	Gigabit Forced Mode Parameter	30
TABLE 3-5	Parameters Defining <code>enable-ipg0</code> and <code>ipg0</code>	30
TABLE 3-6	Read-Write Interpacket Gap Parameter Values and Descriptions	31
TABLE 3-7	RX Blanking Register for Alias Read	31
TABLE 3-8	RX Random Early Detecting 8-Bit Vectors	32
TABLE 3-9	PCI Bus Interface Parameters	33

TABLE 3-10	Device Path Name	40
TABLE 3-11	Local Link Network Device Parameters	42
TABLE 3-12	Cryptographic Driver Statistics	44
TABLE 3-13	Ethernet Driver Statistics	45
TABLE 3-14	TX and RX MAC Counters	46
TABLE 3-15	Current Ethernet Link Properties	48
TABLE 3-16	Read-Only <code>vca</code> Device Capabilities	48
TABLE 3-17	Read-Only Link Partner Capabilities	49
TABLE 3-18	Driver-Specific Parameters	50
TABLE 3-19	Cryptographic Driver Statistics for In-Line IPsec Acceleration	53
TABLE 3-20	Solaris Release Requirements for IPsec Acceleration	56
TABLE 4-1	<code>vcaadm</code> Options	60
TABLE 4-2	<code>vcaadm</code> Prompt Variable Definitions	65
TABLE 4-3	<code>connect</code> Command Optional Parameters	66
TABLE 4-4	Security Officer Name, User Name, and Keystore Name Requirements	72
TABLE 4-5	Password Requirement Settings	73
TABLE 4-6	Key Types	81
TABLE 4-7	<code>vcad</code> Command Options	84
TABLE 4-8	Command-Line Directives For the <code>vcad</code> Command	86
TABLE 4-9	<code>vcadiag</code> Options	90
TABLE 4-10	<code>pk11export</code> Options	92
TABLE 5-1	Passwords Required for Sun ONE Web Servers	110
TABLE 5-2	Requestor Information Fields	119
TABLE 5-3	Fields for the Certificate to Install	121
TABLE 5-4	Requestor Information Fields	129
TABLE 5-5	Fields for the Certificate to Install	131
TABLE 5-6	Requestor Information Fields	140
TABLE 5-7	Fields for the Certificate to Install	142
TABLE 5-8	32- and 64-Bit Path Variable Differences	151
TABLE 5-9	<code>certutil</code> Variable Descriptions	151

TABLE 5-10	Requestor Information Fields	162
TABLE 5-11	<code>configutil</code> Variable Descriptions	168
TABLE 5-12	<code>certutil</code> Variable Descriptions	172
TABLE 6-1	Requestor Information Fields	179
TABLE 6-2	Distinguished Name Fields	185
TABLE 7-1	SunVTS <code>netlbttest</code> and <code>nettest</code> Required Software for the <code>vca</code> Driver	190
TABLE 7-2	<code>vcatest</code> Subtests	193
TABLE 7-3	<code>vcatest</code> Command-Line Syntax	194
TABLE 8-1	Processing for Most Crypto Operations Involving Keys	210
TABLE 8-2	Failure Condition for <code>C_WrapKey</code> and <code>C_UnwrapKey</code>	211
TABLE 8-3	Maximum Key Sizes	216
TABLE A-1	SC Connector Link Characteristics (IEEE P802.3z)	220
TABLE A-2	Physical Dimensions	221
TABLE A-3	Performance Specifications	221
TABLE A-4	Power Requirements	221
TABLE A-5	Interface Specifications	222
TABLE A-6	Environmental Specifications	222
TABLE A-7	Cat-5 Connector Link Characteristics	223
TABLE A-8	Physical Dimensions	224
TABLE A-9	Performance Specifications	224
TABLE A-10	Power Requirements	224
TABLE A-11	Interface Specifications	225
TABLE A-12	Environmental Specifications	225
TABLE B-1	Files in the <code>/cdrom/cdrom0</code> Directory	228
TABLE B-2	Sun Crypto Accelerator 4000 Directories	230
TABLE C-1	SSL Protocols	234
TABLE C-2	Available SSL Ciphers	235
TABLE C-3	SSL Aliases	236
TABLE C-4	Special Characters to Configure Cipher Preference	237
TABLE C-5	SSL Verify Client Levels	238

TABLE C-6	SSL Log Level Values	239
TABLE C-7	Available SSL Options	240
TABLE F-1	Sun Crypto Accelerator 4000 Online Manual Pages	251

Preface

The *Sun Crypto Accelerator 4000 Board Version 1.1 Installation and User's Guide* lists the features, protocols, and interfaces of the Sun Crypto Accelerator 4000 board and describes how to install, configure, and manage the board in your system.

This book assumes that you are a network administrator with experience configuring one or more of the following: Solaris operating environment, Sun platforms with PCI I/O cards, Sun ONE and Apache Web Servers, IPsec, SunVTS™ software, and certification authority acquisitions.

How This Book Is Organized

This book is organized as follows:

- Chapter 1 lists the product features, protocols, and interfaces of the Sun Crypto Accelerator 4000 board, and describes the hardware and software requirements.
- Chapter 2 describes how to install and remove the Sun Crypto Accelerator 4000 hardware and software.
- Chapter 3 defines the Sun Crypto Accelerator 4000 tunable driver parameters, and describes how to configure them with the `ndd` utility and the `vca.conf` file. This chapter also describes how to enable autonegotiation or forced mode for link parameters at the OpenBoot™ PROM interface and how to configure the network `hosts` file.
- Chapter 4 describes how to configure the Sun Crypto Accelerator 4000 board and manage keystores with the `vcaadm` and `vcadiag` utilities.
- Chapter 5 explains how to configure the Sun Crypto Accelerator 4000 board for use with Sun ONE Web Servers.
- Chapter 6 explains how to configure the Sun Crypto Accelerator 4000 board for use with Apache Web Servers.

- Chapter 7 describes how to test the Sun Crypto Accelerator 4000 board with the SunVTS diagnostic application and the on board FCode self-test. This chapter also provides troubleshooting techniques with OpenBoot PROM commands.
- Chapter 8 describes how different configurations of the board work with the PKCS#11 interface.
- Appendix A lists the specifications for the Sun Crypto Accelerator 4000 board.
- Appendix B describes how to install the Sun Crypto Accelerator 4000 software manually without the installation script.
- Appendix C lists directives for using Sun Crypto Accelerator 4000 software to configure SSL support for Apache Web Servers.
- Appendix D describes the software supplied with the Sun Crypto Accelerator 4000 board and how to build OpenSSL-compatible applications to take advantage of the cryptographic acceleration features of the board.
- Appendix E provides software notices and licenses from other software organizations that govern the use of third-party software used with the Sun Crypto Accelerator 4000 board.
- Appendix F provides a description of the Sun Crypto Accelerator 4000 commands and lists the online manual pages for each command.
- Appendix G describes how to zeroize the Sun Crypto Accelerator 4000 board to the factory state which is the Failsafe mode for the board.

Using UNIX Commands

This document does not contain information on basic UNIX[®] commands and procedures such as shutting down the system, booting the system, and configuring devices.

See one or more of the following for this information:

- *Solaris Hardware Platform Guide*
- Online documentation for the Solaris operating environment, available at:
<http://docs.sun.com>
- Other software documentation that you received with your system

Shell Prompts

Shell	Prompt
C shell	<i>machine-name%</i>
C shell superuser	<i>machine-name#</i>
Bourne shell and Korn shell	\$
Bourne shell and Korn shell superuser	#

Typographic Conventions

Typeface	Meaning	Examples
<i>AaBbCc123</i>	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. % You have mail.
AaBbCc123	What you type, when contrasted with on-screen computer output	% su Password:
<i>AaBbCc123</i>	Book titles, new words or terms, words to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be superuser to do this.
	Command-line variable; replace with a real name or value	To delete a file, type <code>rm filename</code> .

Accessing Sun Documentation Online

You can view, print, or purchase a broad selection of Sun documentation, including localized versions, at:

<http://www.sun.com/documentation>

Contacting Sun Technical Support

If you have technical questions about this product that are not answered in this document, go to:

<http://www.sun.com/service/contacting>

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can submit your comments by going to:

<http://www.sun.com/hwdocs/feedback>

Please include the title and part number of your document with your feedback:

Sun Crypto Accelerator 4000 Board Version 1.1 Installation and User's Guide,
part number 817-3693-10

Product Overview

This chapter provides an overview of the Sun Crypto Accelerator 4000 board, and contains the following sections:

- “Product Features” on page 1
- “Hardware Overview” on page 6
- “Hardware and Software Requirements” on page 11

Product Features

The Sun Crypto Accelerator 4000 board is a Gigabit Ethernet-based network interface card that supports cryptographic hardware acceleration for IPsec and SSL (both symmetric and asymmetric) on Sun servers. In addition to operating as a standard Gigabit Ethernet network interface card for unencrypted network traffic, the board contains cryptographic hardware to support a higher throughput for encrypted IPsec traffic than the standard software solution.

Once installed, the board is initialized and configured with the `vcaadm` utility which manages the keystore and user information and determines the level of security in which the board operates. Once a keystore and security officer account are configured, the Sun ONE Web and Application Servers, or the Apache Web Server can be configured to use the board for SSL acceleration with the `iplsslcfg` and `apsslcfg` scripts. The Sun ONE Directory, Messaging, and Portal Servers can also be configured to use the board for SSL acceleration with the Sun ONE administration console and the `modutil` and `certutil` utilities. Additionally, most applications that require a PKCS#11 interface for keystore and cryptographic services are compatible to use the board.

Key Protocols and Interfaces

The Sun Crypto Accelerator 4000 board is interoperable with existing Ethernet equipment assuming standard Ethernet minimum and maximum frame size (64 to 1518 bytes), frame format, and compliance with the following standards and protocols:

- Full-size PCI 33/66 Mhz, 32/64-bit
- IEEE 802.3 CSMA/CD (Ethernet)
- IEEE 802.2 Logical Link Control
- SNMP (limited MIB)
- Full- and half-duplex Gigabit Ethernet interface (IEEE 802.z)
- Universal dual voltage signaling (3.3V and 5V)

Key Features

- Gigabit Ethernet with either copper or fiber interface
- Accelerates IPsec and SSL cryptographic functions
- Session establishment rate: up to 4300 operations per second
- Bulk encryption rate: up to 800 Mbps
- Provides up to 2048-bit RSA encryption
- Delivers up to 10 times faster 3DES bulk data encryption
- Provides tamper-proof, centralized security key and certificate administration for Sun ONE Web Server for increased security and simplified key management
- Designed for FIPS 140-2 Level 3 certification
- Low CPU utilization—frees up server system resource and bandwidth
- Secure private key storage and management
- Dynamic reconfiguration (DR) and redundancy/failover support on Sun's midframe and high-end servers
- Load balancing for RX packets among multiple CPUs
- Full flow control support (IEEE 802.3x)

The Sun Crypto Accelerator 4000 boards are designed to comply with the security requirements for cryptographic modules as documented in the Federal Information Processing Standard (FIPS) 140-2, Level 3.

Supported Applications

- Solaris 8 and 9 operating environments (IPsec VPN)
- Sun ONE Web Server 4.1 and 6.0

- Sun ONE Application Server 7.0
- Sun ONE Directory Server 5.2
- Sun ONE Messaging Server 5.2
- Sun ONE Portal Server 6.2
- Apache Web Server 1.3.x and 2.x

Supported Cryptographic Protocols

The board supports the following protocols:

- IPsec for IPv4 and IPv6, including IKE
- SSLv2, SSLv3, TLSv1 (transmission layer security)

The board accelerates the following IPsec functions:

- ESP (DES, 3DES) encryption
- ESP (SHA1, MD5) authentication *
- AH (SHA1, MD5) authentication *

* When configured for in-line IPsec acceleration (See “In-Line IPsec Hardware Acceleration” on page 5)

The board accelerates the following SSL functions:

- Secure establishment of a set of cryptographic parameters and secret keys between a client and a server
- Secure key storage on the board—keys are encrypted if they leave the board

Diagnostic Support

- User-executable self-test using OpenBoot PROM
- SunVTS diagnostic tests

Cryptographic Algorithm Acceleration

The board accelerates cryptographic algorithms in both hardware and software. The reason for this complexity is that the cost of accelerating cryptographic algorithms is not uniform across all algorithms. Some cryptographic algorithms were designed specifically to be implemented in hardware, others were designed to be implemented in software. For hardware acceleration, there is the additional cost of moving data from the user application to the hardware acceleration device, and moving the results back to the user application. Note that a few cryptographic algorithms can be performed by highly tuned software as quickly as they can be performed in dedicated hardware.

Supported Cryptographic Algorithms

The Sun Crypto Accelerator 4000 driver (`vca`) examines each cryptographic request and determines the best location for the acceleration (host processor or Sun Crypto Accelerator 4000), to achieve maximum throughput. Load distribution is based on the cryptographic algorithm, the current job load, and the data size.

The board accelerates the following IPsec algorithms.

TABLE 1-1 IPsec Cryptographic Algorithms

Type	Algorithm
Symmetric	DES, 3DES
Hash*	MD5, SHA1

* When configured for in-line IPsec hardware acceleration.

The board accelerates the following SSL algorithms.

TABLE 1-2 SSL Cryptographic Algorithms

Type	Algorithm
Symmetric	DES, 3DES, ARCFOUR
Asymmetric	Diffie-Hellman (Apache only) and RSA (up to 2048 bit key), DSA
Hash	MD5, SHA1

IPsec Acceleration

The board supports two forms of IPsec acceleration: out-of-band and in-line. Both configurations offload high-overhead cryptographic operations from the SPARC® processor to the board. See “Configuring IPsec Hardware Acceleration” on page 56.

TABLE 1-3 Accelerated IPsec Algorithms

Algorithm	Out-of-Band	In-Line
DES	X	X
3DES	X	X
MD5		X
SHA1		X

Out-of-Band IPsec Hardware Acceleration

When the board is configured for out-of-band IPsec acceleration, supported encryption and decryption operations are accelerated in hardware when installed on a Solaris 9 (or later) system. All IPsec specific packet processing is performed by the host Solaris IPsec software. See “Enabling Out-of-Band IPsec Acceleration” on page 57.

Note – No IPsec configuration or tuning is required to use the board for out-of-band IPsec acceleration in Solaris 9. You simply install the Sun Crypto Accelerator 4000 packages and reboot.

In-Line IPsec Hardware Acceleration

When configured for in-line IPsec acceleration, supported encryption, decryption, and authentication operations are accelerated in hardware when installed on a Solaris 9 12/03 (or later) system. Portions of the IPsec specific packet processing are performed directly by the board. See “Enabling In-Line IPsec Acceleration” on page 57 for instructions on how to configure the board for in-line IPsec acceleration.

SSL Acceleration

TABLE 1-4 shows which SSL accelerated algorithms may be off-loaded to hardware and which software algorithms are provided for Sun ONE and Apache Web Servers.

TABLE 1-4 Supported SSL Algorithms

Algorithm	Sun ONE Web Servers		Apache Web Servers	
	Hardware	Software	Hardware	Software
RSA	X	X	X	X
DSA	X	X	X	X
ARCFOUR		X		X
Diffie-Hellman			X	X
DES	X	X	X	X
3DES	X	X	X	X
MD5	X	X		
SHA1	X	X		

Bulk Encryption

The Sun Crypto Accelerator 4000 bulk encryption feature for Sun ONE server software is disabled by default. You must manually enable this feature by creating a file and restarting the Sun ONE server software.

To enable Sun ONE server software to use bulk encryption on the board, you simply create an empty file in the `/etc/opt/SUNWconn/cryptov2/` directory named `sslreg`, and restart the server software.

```
# touch /etc/opt/SUNWconn/cryptov2/sslreg
```

To disable the bulk encryption feature, you must delete the `sslreg` file and restart the server software.

```
# rm /etc/opt/SUNWconn/cryptov2/sslreg
```

The bulk encryption feature for Apache Web Server software is enabled by default and cannot be disabled.

Hardware Overview

The Sun Crypto Accelerator 4000 hardware is a full-size (4.2 inches x 12.283 inches) cryptographic accelerator PCI Gigabit Ethernet adapter that enhances the performance of IPsec and SSL on Sun servers.

Sun Crypto Accelerator 4000 MMF Adapter

The Sun Crypto Accelerator 4000 MMF adapter is a single-port Gigabit Ethernet fiber optics PCI bus card. It operates in 1000 Mbps Ethernet networks only.

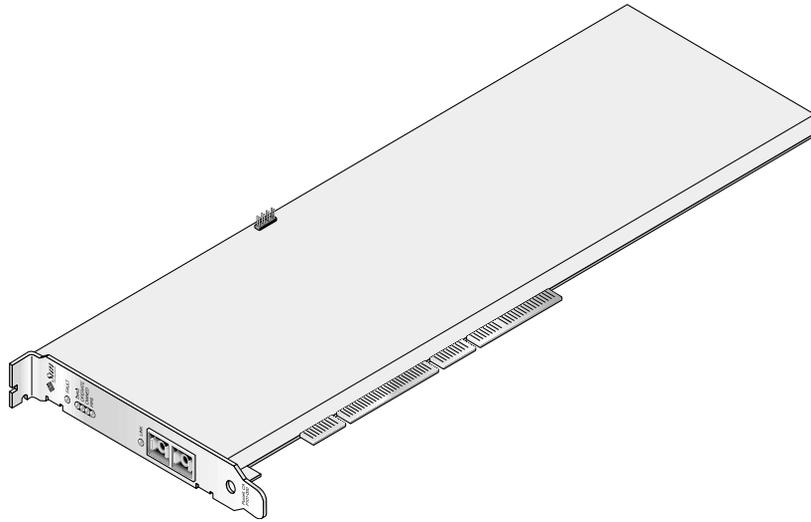


FIGURE 1-1 Sun Crypto Accelerator 4000 MMF Adapter

LED Displays

TABLE 1-5 Front Panel Display LEDs for the MMF Adapter

Label	Meaning if Lit	Color
FAULT	On when the board is HALTED (fatal error) state or low-level hardware initialization failed. Flashing if an error occurred during the boot process.	Red
DIAG	On in POST, DIAGNOSTICS, and FAILSAFE (firmware not upgraded) state. Flashing when running DIAGNOSTICS.	Green
OPERATE	On in POST, DIAGNOSTICS, and DISABLED (driver not attached) state. Flashing in IDLE, OPERATIONAL, and FAILSAFE states.	Green
INIT	On if the security officer has initialized the board with vcaadm. See “Initializing the Board With vcaadm” on page 68. Flashing if the ZEROIZE jumper is present.	Green
FIPS	On when operating in FIPS 140-2 level 3 certified mode. Off when in non-FIPS mode.	Green
LINK	On when the link is up.	Green

Sun Crypto Accelerator 4000 UTP Adapter

The Sun Crypto Accelerator 4000 UTP adapter is a single-port Gigabit Ethernet copper-based PCI bus card. It can be configured to operate in 10, 100, or 1000 Mbps Ethernet networks.

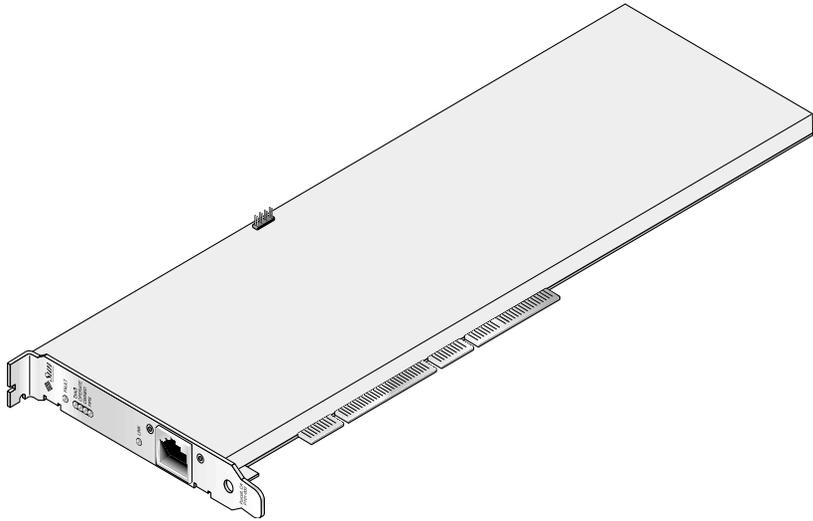


FIGURE 1-2 Sun Crypto Accelerator 4000 UTP Adapter

LED Displays

TABLE 1-6 Front Panel Display LEDs for the UTP Adapter

Label	Meaning if Lit	Color
FAULT	On when the board is HALTED (fatal error) state or low level hardware initialization failed. Flashing if an error occurred during the boot process.	Red
DIAG	On in POST, DIAGNOSTICS, and FAILSAFE (firmware not upgraded) state. Flashing when running DIAGNOSTICS.	Green
OPERATE	On in POST, DIAGNOSTICS, and DISABLED (driver not attached) state. Flashing in IDLE, OPERATIONAL, and FAILSAFE states.	Green
INIT	On if the security officer has initialized the board with vcaadm. See “Initializing the Board With vcaadm” on page 68. Flashing if the ZEROIZE jumper is present.	Green
FIPS	On when operating in FIPS 140-2 level 3 certified mode. Off when in non-FIPS mode.	Green
1000	On when using Gigabit Ethernet.	Green
ACTIVITY (no label)	On when the link is transmitting or receiving.	Amber
LINK (no label)	On when the link is up.	Green

Note – The service pack numbers (SP9 or SP1) are implied whenever Sun ONE Web Server 4.1 or 6.0 is mentioned.

Dynamic Reconfiguration and High Availability

The Sun Crypto Accelerator 4000 hardware and associated software provides the capability to work effectively on Sun platforms supporting Dynamic Reconfiguration (DR) and hot-plugging. During a DR or hot-plug operation, the Sun Crypto Accelerator 4000 software layer automatically detects the addition or removal of a board, and adjusts the scheduling algorithms to accommodate the change in hardware resources.

For High Availability (HA) configurations, multiple Sun Crypto Accelerator 4000 boards can be installed within a system or domain to insure that hardware acceleration is continuously available. In the unlikely event of a Sun Crypto Accelerator 4000 hardware failure, the software layer detects the failure and removes the failed board from the list of available hardware cryptographic accelerators. Sun Crypto Accelerator 4000 software adjusts the scheduling algorithms to accommodate the reduction in hardware resources. Subsequent cryptographic requests are scheduled to the remaining boards.

Note that the Sun Crypto Accelerator 4000 hardware provides a source for high-quality entropy for the generation of long-term keys. If all the Sun Crypto Accelerator 4000 boards within a domain or system are removed, long-term keys are generated with lower-quality entropy.

Load Sharing

The Sun Crypto Accelerator 4000 software distributes load across as many boards as are installed within the Solaris domain or system. Incoming cryptographic requests are distributed across the boards based on fixed-length work queues. Cryptographic requests are directed to the first board, and subsequent requests stay directed to the first board until it is running at full capacity. Once the first board is running at full capacity, further requests are queued to the next board available that can accept the request of this type. The queueing mechanism is designed to optimize throughput by facilitating request coalescing at the board.

Hardware and Software Requirements

TABLE 1-7 provides a summary of the hardware and software requirements for the Sun Crypto Accelerator 4000 adapter.

TABLE 1-7 Hardware and Software Requirements

Hardware and Software	Requirements
Hardware	Sun Fire™ V120, V210, V240, 280R, V480, V880, 4800, 4810, 6800, 12K, 15K; Netra™ 20 (lw4); Sun Blade™ 100, 150, 1000, 2000
Operating Environment	Solaris 8 2/02 and future compatible releases (Solaris 9 is required for IPsec acceleration.)

Required Patches

Refer to the *Sun Crypto Accelerator 4000 Board Version 1.1 Release Notes* for detailed required patch information.

The following patches are required to run the Sun Crypto Accelerator 4000 board on your system. Solaris updates contain patches to previous releases. Use the `showrev -p` command to determine whether the listed patches have already been installed.

You can download the patches from the following web site:

<http://sunsolve.sun.com>

Install the latest version of the patches. The dash number (-01, for example) becomes higher with each new revision of the patch. If the version on the web site is higher than that shown in the following tables, it is simply a later version.

If the patch you need is not available at the SunSolveSM web site, contact your local sales or service representative.

Apache Web Server Patch

If you plan to use the Apache Web Server with Solaris 8, you must install Patch 109234-09 before installing the Sun Crypto Accelerator 4000 software. Once the SUNWkcl2a package is added, the system will be configured with Apache Web Server mod_ssl 1.3.26.

Solaris 8 Patches

TABLE 1-8 lists the required Solaris 8 patches for the Sun Crypto Accelerator 4000 software.

TABLE 1-8 Required Solaris 8 Patches

Patch ID	Description
110383-01	libnvpair
108528-23	KU-05 (nvpair support)
112438-01	/dev/random
110900-10	pcifg, SunFire 15K support, and DR
110824-04	DR

TABLE 1-8 Required Solaris 8 Patches (*Continued*)

Patch ID	Description
110842-11	Bus speed and DR
110839-04	Minor node and DLPI provider names
109234-09	Apache support

Solaris 9 Patches

TABLE 1-9 lists the required Solaris 9 patches for the Sun Crypto Accelerator 4000 software.

TABLE 1-9 Required Solaris 9 Patches

Patch ID	Description
113068-04	Bus speed, Sun Fire 15K support, and DR
112838-08	<code>pcicfg</code> , DR, and Sun Fire 15K support
113218-08	Gigabit performance and <code>vca</code> memory leak
112904-08	Gigabit performance
114758-01	Minor node and DLPI provider names
112233-08	(only required for Solaris releases prior to Solaris 9 9/04)

Installing the Sun Crypto Accelerator 4000 Board

This chapter describes how to install the Sun Crypto Accelerator 4000 hardware and also how to install and remove the software with automated scripts. This chapter includes the following sections:

- “Handling the Board” on page 15
- “Installing the Board” on page 16
- “Installing the Sun Crypto Accelerator 4000 Software” on page 18
- “Directories and Files” on page 22

Once you have installed the hardware and software of the board, you need to initialize the board with configuration and keystore information. See “Initializing the Board With vcaadm” on page 68 for information on how to initialize the board.

Handling the Board

Each board is packed in a special antistatic bag to protect it during shipping and storage. To avoid damaging the static-sensitive components on the board, reduce any static electricity on your body before touching the board by using one of the following methods:

- Touch the metal frame of the computer.
- Attach an antistatic wrist strap to your wrist and to a grounded metal surface.



Caution – To avoid damaging the sensitive components on the board, wear an antistatic wrist strap when handling the board, hold the board by its edges only, and always place the board on an antistatic surface (such as the plastic bag it came in).

Installing the Board

Installing the Sun Crypto Accelerator 4000 board involves inserting the board into the system and loading the software tools. The hardware installation instructions include only general steps for installing the board. Refer to the documentation that came with your system for specific installation instructions.

▼ To Install the Hardware

1. **As superuser, follow the instructions that came with your system to shut down and power off the computer, disconnect the power cord, and remove the computer cover.**
2. **Locate an unused PCI slot (preferably a 64-bit, 66 MHz slot).**
3. **Attach an antistatic wrist strap to your wrist, and attach the other end to a grounded metal surface.**
4. **Using a Phillips-head screwdriver, remove the screw from the PCI slot cover.**
Save the screw to hold the bracket in Step 5.
5. **Holding the Sun Crypto Accelerator 4000 board by its edges only, take it out of the plastic bag and insert it into the PCI slot, and then secure the screw on the rear bracket.**
6. **Replace the computer cover, reconnect the power cord, and power on the system.**
7. **Verify that the board is properly installed by issuing the `show-devs` command at the OpenBoot PROM `ok` prompt:**

```
ok show-devs
.
/chosen
/packages
/upa@8,480000/SUNW,ffb@0,0
/pci@8,600000/network@1
/pci@8,600000/SUNW,qlc@4
/pci@8,600000/SUNW,qlc@4/fp@0,0
.
```

In the preceding example, the `/pci@8,600000/network@1` identifies the device path to the Sun Crypto Accelerator 4000 board. There is one such line for each board in the system.

To determine whether the Sun Crypto Accelerator 4000 device properties are listed correctly: from the ok prompt, navigate to the device path and type .properties to display the list of properties.

```

ok cd /pci@8,600000/network@1
ok .properties
assigned-addresses      82000810 00000000 00102000 00000000 00002000
                        81000814 00000000 00000400 00000000 00000100
                        82000818 00000000 00200000 00000000 00200000
                        82000830 00000000 00400000 00000000 00100000
d-fru-len               00 00 00 00
d-fru-off               00 00 e8 00
d-fru-dev               eeprom
s-fru-len               00 00 08 00
s-fru-off               00 00 e0 00
s-fru-dev               eeprom
compatible              70 63 69 38 30 38 36 2c 62 35 35 35 2e 31 30 38
reg                     00000800 00000000 00000000 00000000 00000000
                        02000810 00000000 00000000 00000000 00002000
                        02000814 00000000 00000000 00000000 00000100
                        02000818 00000000 00000000 00000000 00200000
                        02000830 00000000 00000000 00000000 00100000
address-bits            00 00 00 30
max-frame-size          00 00 40 00
network-interface-type  ethernet
device-type             network
name                    network
local-mac-address       00 03 ba 0e 99 ca
version                 Sun PCI Crypto Accelerator 4000 1000Base-T FCode
FCODE 2.11.13 03/03/04
phy-type                mif
board-model             501-6039
model                   SUNW,pci-vca
fcode-rom-offset        00000000
66mhz-capable
fast-back-to-back
devsel-speed            00000001
class-code              00100000
interrupts              00000001
max-latency             00000040
cache-line-size         00000010
max-latency             00000040
min-grant               00000040
subsystem-vendor-id     0000108e
subsystem-id            00003de8
revision-id             00000002
device-id               0000b555
vendor-id               00008086

```

Installing the Sun Crypto Accelerator 4000 Software

The Sun Crypto Accelerator 4000 software is included on the Sun Crypto Accelerator 4000 CD. You may need to download patches from the SunSolve web site. See “Required Patches” on page 12 for more information.

There are two methods to install the software: manually or with the `install` script. This section describes how to install the software with the `install` script. To install the software manually, refer to Appendix B.

▼ To Install the Software

1. **Insert the Sun Crypto Accelerator 4000 CD into a CD-ROM drive that is connected to your system.**
 - If your system is running Sun Enterprise Volume Manager™, it should automatically mount the CD-ROM to the `/cdrom/cdrom0` directory.
 - If your system is not running Sun Enterprise Volume Manager, mount the CD-ROM as follows:

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /cdrom
```

You see the following files and directories in the /cdrom/cdrom0 directory.

TABLE 2-1 Files in the /cdrom/cdrom0 Directory

File or Directory	Contents
Copyright	U.S. copyright file
FR_Copyright	French copyright file
install	Script that installs the Sun Crypto Accelerator 4000 software
remove	Script that removes the Sun Crypto Accelerator 4000 software
Docs	<i>Sun Crypto Accelerator 4000 Board Version 1.1 Installation and User's Guide</i> <i>Sun Crypto Accelerator 4000 Board Release Notes</i>
Packages	Contains the Sun Crypto Accelerator 4000 software packages: SUNWkc12r Cryptography Kernel Components SUNWkc12u Cryptographic Administration Utility and Libraries SUNWkc12a SSL Support for Apache (<i>optional</i>) SUNWkc12m Cryptographic Administration Manual Pages (<i>optional</i>) SUNWvcar VCA Crypto Accelerator (root) SUNWvcau VCA Crypto Accelerator (usr) SUNWvcaa VCA Administration SUNWvcaw VCA Firmware SUNWvcamn VCA Crypto Accelerator Manual Page (<i>optional</i>) SUNWvcav SunVTS Test of VCA Crypto Accelerator (<i>optional</i>) SUNWkc12o SSL Development Tools and Libraries (<i>optional</i>) SUNWkc12i.u IPsec Acceleration with KCLv2 Crypto (<i>optional</i>)

This installation script installs the required packages in a specific order and these packages must be installed before installing any optional packages. Once the required packages are installed, you can install and remove the optional packages in any order.

Install the optional SUNWkc12a package only if you plan to use Apache as your web server.

Install the optional SUNWkc12o package only if you plan to relink to another version of Apache Web Server.

Install the optional SUNWvcav package only if you plan to perform the SunVTS tests. You must have SunVTS 4.4 or later up to 5.x installed to install the SUNWvcav package.

Note – The optional SUNWkc12i.u package has the .u extension only on the Sun Crypto Accelerator 4000 CD. Once this package is installed, the name changes to SUNWkc12i. The .u extension of this package on the CD, defines the package as sun4u architecture-specific.

2. Install the required software by typing:

```
# cd /cdrom/cdrom0
# ./install
```

The install script analyzes the system to determine which required patches need to be installed, installs those patches, installs the main software, and optionally installs the optional software. For example:

Note – The copyright and license information was omitted from the following example. Refer to Appendix E for copyright and software licenses.

```
# ./install
This program installs the software for the Sun Crypto Accelerator
4000, Version 1.1.

*** Checking if Sun Crypto Accelerator support is already installed...
*** Checking for required OS patch(es):
    113146-01 112838-07 113068-04 113449-02 113453-04 114758-01
*** Checking for incompatible OS patch(es) ...
*** Checking for optional package dependencies...

Do you wish to install the optional Crypto IPsec Acceleration software
(SUNWkcl2i.u)? [y,n,?,q]

Do you wish to install the optional Crypto Apache Support (SSL) (SUNWkcl2a
SUNWkcl2o)? [y,n,?,q] y

Do you wish to install the optional Crypto QA Tools (SUNWkcl2q SUNWvcaq)?
[y,n,?,q] n

Do you wish to install the optional VCA Crypto Accelerator/Gigabit Ethernet
SunVTS Diagnostics (SUNWvcav)? [y,n,?,q] n

This script is about to take the following actions:
- Install Sun Crypto Accelerator 4000 support for Solaris 9
- Install Optional Crypto IPsec Acceleration software
- Install Optional Crypto Apache Support (SSL) software

To cancel installation of this software, press 'q' followed by a Return.
**OR**
Press Return key to begin installation:

*** Installing Sun Crypto Accelerator 4000 software for Solaris 9...
Installing required packages:
```

```
SUNWkcl2r SUNWkcl2u SUNWkcl2m SUNWvcar SUNWvcau SUNWvcaa SUNWvcamn SUNWvcaf
```

```
Installation of <SUNWkcl2u> was successful.
Installation of <SUNWkcl2m> was successful.
Installation of <SUNWvcar> was successful.
Installation of <SUNWvcau> was successful.
Installation of <SUNWvcaa> was successful.
Installation of <SUNWvcamn> was successful.
Installation of <SUNWvcaf> was successful.
*** Installing selected optional software for Solaris 9...
Installing optional package(s):
  SUNWkcl2i.u SUNWkcl2a SUNWkcl2o
Installation of <SUNWkcl2i> was successful.

Checking operating environment requirements...
Determining package requirements...
Verifying required packages are installed...
All required packages installed.
Determining patch requirements...
Verifying required patches are installed...
Requirement for 113146-01 met by 113146-01.
All required patches installed.

Installation of <SUNWkcl2a> was successful.

Installation of <SUNWkcl2o> was successful.
*** Installation complete.
```

Choosing the Optional Packages to Install

To install only the optional packages that provide the SSL support for the Apache Web Server and the Sun Crypto Accelerator 4000 online manual pages, select `SUNWkcl2a` and `SUNWkcl2m`.

To install all of the optional software packages, select the following: `SUNWkcl2a`, `SUNWkcl2m`, `SUNWvcamn`, `SUNWvcav`, `SUNWkcl2o`, and `SUNWkcl2i.u`.

See TABLE 2-1 for a description of the package contents of the optional packages in the previous examples.

Directories and Files

TABLE 2-2 shows the directories created by the default installation of the Sun Crypto Accelerator 4000 software.

TABLE 2-2 Sun Crypto Accelerator 4000 Directories

Directory	Contents
/etc/opt/SUNWconn/vca/keydata	Keystore data (encrypted)
/opt/SUNWconn/cryptov2/bin	Utilities
/opt/SUNWconn/cryptov2/lib	Support libraries
/opt/SUNWconn/cryptov2/sbin	Administrative commands

FIGURE 2-1 shows the hierarchy of these directories and files.

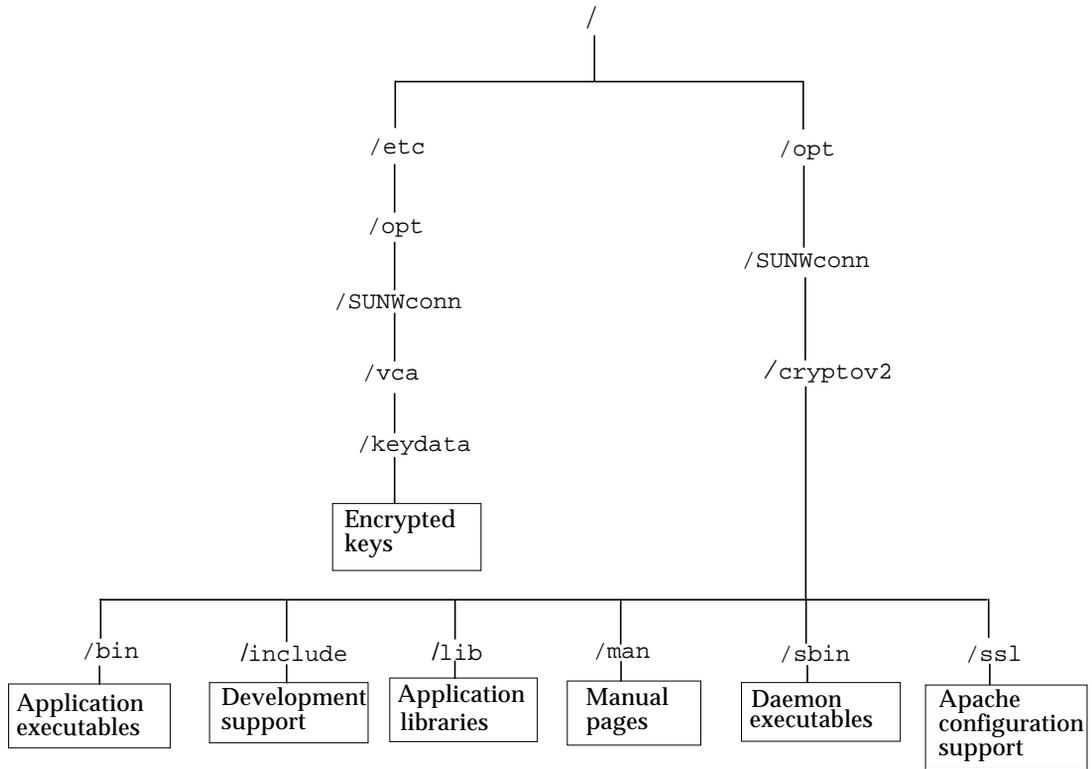


FIGURE 2-1 Sun Crypto Accelerator 4000 Directories and Files

Note – Once you install the Sun Crypto Accelerator 4000 hardware and software, you need to initialize the board with configuration and keystore information. See “Initializing the Board With vcaadm” on page 68 for information on how to initialize the board.

Removing the Sun Crypto Accelerator 4000 Software

There are three methods to remove the software: the `remove` script on the CD-ROM, the `/var/tmp/crypto_acc.remove` script on the server, or the `pkgrm` command. This section describes how to remove the software with the two removal scripts. For instructions on removing the software with the `pkgrm` command refer to Appendix B.

Use the `remove` script for software removal if you used the `install` script to install the software. Use the `/var/tmp/crypto_acc.remove` script if you installed the software manually (Appendix B).

▼ To Remove the Software With the `remove` Script

- Type the following with the Sun Crypto Accelerator 4000 CD-ROM inserted:

```
# cd /cdrom/cdrom0
# ./remove
```

▼ To Remove the Software With the `/var/tmp/crypto_acc.remove` Script

A log of this installation can be found at:

```
/var/tmp/crypto_acc.install.2003.10.13
```

- Type the following:

```
# /var/tmp/crypto_acc.remove
```

Configuring Driver Parameters

This chapter describes how to configure the `vca` device driver parameters used by both the Sun Crypto Accelerator 4000 UTP and MMF Ethernet adapters. This chapter contains the following sections:

- “Ethernet Device Driver (`vca`) Parameters” on page 25
- “Setting `vca` Driver Parameters” on page 34
- “Enabling Autonegotiation or Forced Mode for Link Parameters With the OpenBoot PROM” on page 42
- “Cryptographic and Ethernet Driver Operating Statistics” on page 44
- “Network Configuration” on page 54

Ethernet Device Driver (`vca`) Parameters

The `vca` device driver controls the Sun Crypto Accelerator 4000 UTP and MMF Ethernet devices. The `vca` driver is attached to the UNIX `pci` name property `pci108e,3de8` for the Sun Crypto Accelerator 4000 (108e is the vendor ID and 3de8 is the PCI device ID).

You can manually configure the `vca` device driver parameters to customize each Sun Crypto Accelerator 4000 device in your system. This section provides an overview of the capabilities of the Sun Crypto Accelerator 4000 Ethernet device used in the board, lists the available `vca` device driver parameters, and describes how to configure these parameters.

The Sun Crypto Accelerator 4000 Ethernet UTP and MMF PCI adapters are capable of the operating speeds and modes listed in “Enabling Autonegotiation or Forced Mode for Link Parameters With the OpenBoot PROM” on page 42. By default, the `vca` device operates in autonegotiation mode with the remote end of the link (link partner) to select a common mode of operation for the speed, duplex, and link-

clock parameters. The link-clock parameter is applicable only if the board is operating at 1000 Mbps. The vca device can also be configured to operate in forced mode for each of these parameters.



Caution – To establish a proper link, both link partners must operate in either autonegotiation or forced mode for each of the speed, duplex, and link-clock (1000 Mbps only) parameters. If both link partners are not operating in the same mode for each of these parameters, network errors will occur. See “Enabling Autonegotiation or Forced Mode for Link Parameters With the OpenBoot PROM” on page 42.

Driver Parameter Values and Definitions

TABLE 3-1 describes the parameters and settings for the vca device driver.

TABLE 3-1 vca Driver Parameter, Status, and Descriptions

Parameter	Status	Description
instance	Read and write	Device instance
adv-autoneg-cap	Read and write	Operational mode parameter
adv-1000fdx-cap	Read and write	Operational mode parameter (MMF adapter only)
adv-1000hdx-cap	Read and write	Operational mode parameter
adv-100fdx-cap	Read and write	Operational mode parameter (UTP adapter only)
adv-100hdx-cap	Read and write	Operational mode parameter (UTP adapter only)
adv-10fdx-cap	Read and write	Operational mode parameter (UTP adapter only)
adv-10hdx-cap	Read and write	Operational mode parameter (UTP adapter only)
adv-asmppause-cap	Read and write	Flow control parameter
adv-pause-cap	Read and write	Flow control parameter
pause-on-threshold	Read and write	Flow control parameter
pause-off-threshold	Read and write	Flow control parameter
link-master	Read and write	1 Gbps speed forced mode parameter
enable-ipg0	Read and write	Enable additional delay before transmitting a packet
ipg0	Read and write	Additional delay before transmitting a packet
ipg1	Read and write	Interpacket Gap parameter
ipg2	Read and write	Interpacket Gap parameter

TABLE 3-1 vca Driver Parameter, Status, and Descriptions (Continued)

Parameter	Status	Description
rx-intr-pkts	Read and write	Receive interrupt blanking values
rx-intr-time	Read and write	Receive interrupt blanking values
red-dv4to6k	Read and write	Random early detection and packet drop vectors
red-dv6to8k	Read and write	Random early detection and packet drop vectors
red-dv8to10k	Read and write	Random early detection and packet drop vectors
red-dv10to12k	Read and write	Random early detection and packet drop vectors
tx-dma-weight	Read and write	PCI Interface parameter
rx-dma-weight	Read and write	PCI Interface parameter
infinet-burst	Read and write	PCI Interface parameter
disable-64bit	Read and write	PCI Interface parameter

Advertised Link Parameters

The following parameters determine the transmit and receive speed and duplex link parameters to be advertised by the vca driver to its link partner. TABLE 3-2 describes the operational mode parameters and their default values.

Note – If a parameter’s initial setting is 0, it cannot be changed. If you try to change an initial setting of 0, it reverts back to 0. By default, these parameters are set to the capabilities of the vca device.

The Sun Crypto Accelerator 4000 UTP adapter advertised link parameters are different from those of the Sun Crypto Accelerator 4000 MMF adapter as shown in TABLE 3-2.

TABLE 3-2 Operational Mode Parameters

Parameter	Description	UTP Adapter	MMF Adapter
adv-autoneg-cap	Local interface capability advertised by the hardware 0 = Forced mode 1 = Autonegotiation (default)	X	X
adv-1000fdx-cap	Local interface capability advertised by the hardware 0 = Not 1000 Mbps full-duplex capable 1 = 1000 Mbps full-duplex capable (default)		X
adv-1000hdx-cap	Local interface capability advertised by the hardware 0 = Not 1000 Mbps half-duplex capable 1 = 1000 Mbps half-duplex capable (default)	X	X
adv-100fdx-cap	Local interface capability advertised by the hardware 0 = Not 100 Mbps full-duplex capable 1 = 100 Mbps full-duplex capable (default)	X	
adv-100hdx-cap	Local interface capability advertised by the hardware 0 = Not 100 Mbps half-duplex capable 1 = 100 Mbps half-duplex capable (default)	X	
adv-10fdx-cap	Local interface capability advertised by the hardware 0 = Not 10 Mbps full-duplex capable 1 = 10 Mbps full-duplex capable (default)	X	
adv-10hdx-cap	Local interface capability advertised by the hardware 0 = Not 10 Mbps half-duplex capable 1 = 10 Mbps half-duplex capable (default)	X	

If all of the parameters in TABLE 3-2 are set to 1, autonegotiation uses the highest speed possible. If all of these parameters are set to 0, you receive the following error message:

```
NOTICE: Last setting will leave vca0 with no link capabilities.
WARNING: vca0: Restoring previous setting.
```

Note – In the previous example, `vca0` is the Sun Crypto Accelerator 4000 device name where the string, `vca`, is used for every Sun Crypto Accelerator 4000 board. This string is always immediately followed by the device instance number of the board. Thus, the device instance number of the `vca0` board is 0.

Flow Control Parameters

The vca device is capable of sourcing (transmitting) and terminating (receiving) pause frames conforming to the IEEE 802.3x Frame Based Link Level Flow Control Protocol. In response to received flow control frames, the vca device is capable of reducing its transmit rate. Alternately, the vca device is capable of sourcing flow control frames, requesting the link partner to reduce its transmit rate if the link partner supports this feature. By default, the driver advertises both transmit and receive pause capability during autonegotiation.

TABLE 3-3 provides flow control keywords and describes their function.

TABLE 3-3 Read-Write Flow Control Keyword Descriptions

Keyword	Description																																			
adv-asmopause-cap	Both the MMF and UTP adapters support asymmetric pause; therefore, the vca device can pause only in one direction. 0=Off (default) 1=On																																			
adv-pause-cap	This parameter has two meanings depending on the value of adv-asmopause-cap. (Default=0) <table border="1"> <thead> <tr> <th>Parameter Value</th> <th>+</th> <th>Parameter Value</th> <th>=</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>adv-asmopause-cap=</td> <td></td> <td>adv-pause-cap=</td> <td></td> <td></td> </tr> <tr> <td>1</td> <td></td> <td>1 or 0</td> <td></td> <td>adv-pause-cap determines which direction pauses operate on.</td> </tr> <tr> <td>1</td> <td></td> <td>1</td> <td></td> <td>Pauses are received but are not transmitted.</td> </tr> <tr> <td>1</td> <td></td> <td>0</td> <td></td> <td>Pauses are transmitted but are not received.</td> </tr> <tr> <td>0</td> <td></td> <td>1</td> <td></td> <td>Pauses are sent and received.</td> </tr> <tr> <td>0</td> <td></td> <td>1 or 0</td> <td></td> <td>adv-pause-cap determines whether the pause capability is on or off.</td> </tr> </tbody> </table>	Parameter Value	+	Parameter Value	=	Description	adv-asmopause-cap=		adv-pause-cap=			1		1 or 0		adv-pause-cap determines which direction pauses operate on.	1		1		Pauses are received but are not transmitted.	1		0		Pauses are transmitted but are not received.	0		1		Pauses are sent and received.	0		1 or 0		adv-pause-cap determines whether the pause capability is on or off.
Parameter Value	+	Parameter Value	=	Description																																
adv-asmopause-cap=		adv-pause-cap=																																		
1		1 or 0		adv-pause-cap determines which direction pauses operate on.																																
1		1		Pauses are received but are not transmitted.																																
1		0		Pauses are transmitted but are not received.																																
0		1		Pauses are sent and received.																																
0		1 or 0		adv-pause-cap determines whether the pause capability is on or off.																																
pause-on-threshold	Defines the number of 64-byte blocks in the receive (RX) FIFO which causes the board to generate an XON-PAUSE frame.																																			
pause-off-threshold	Defines the number of 64-byte blocks in the RX FIFO which causes the board to generate an XOFF-PAUSE frame.																																			

Gigabit Forced Mode Parameter

For Gigabit links, this parameter determines the `link-master`. Generally, switches are enabled as a link master; in which case, this parameter can remain unchanged. If this is not the case, then the `link-master` parameter can be used to enable the `vca` device as a link master.

TABLE 3-4 Gigabit Forced Mode Parameter

Parameter	Description
<code>link-master</code>	When set to 1 this parameter enables master operation, assuming the link partner is a slave. When set to 0 this parameter enables slave operation, assuming the link partner is a master (default).

Interpacket Gap Parameters

The `vca` device supports the `enable-ipg0` programmable mode.

Before transmitting a packet with `enable-ipg0` enabled (default), the `vca` device adds an additional time delay. This delay, set by the `ipg0` parameter, is in addition to the delay set by the `ipg1` and `ipg2` parameters. The additional `ipg0` delay reduces collisions.

If `enable-ipg0` is disabled, the value of `ipg0` is ignored and no additional delay is set. Only the delays set by `ipg1` and `ipg2` are used. Disable `enable-ipg0` if other systems keep sending a large number of continuous packets. Systems that have `enable-ipg0` enabled might not have enough time on the network. You can add the additional delay by setting the `ipg0` parameter from 0 to 255, which is the media byte-time delay. TABLE 3-5 defines the `enable-ipg0` and `ipg0` parameters.

TABLE 3-5 Parameters Defining `enable-ipg0` and `ipg0`

Parameter	Values	Description
<code>enable-ipg0</code>	0 1	<code>enable-ipg0</code> enable <code>enable-ipg0</code> disable (Default=1)
<code>ipg0</code>	0 to 255	The additional time delay (or gap) before transmitting a packet (after receiving the packet) (Default=8)

The `vca` device supports the programmable interpacket gap (IPG) parameters `ipg1` and `ipg2`. The total IPG is the sum of `ipg1` and `ipg2`. The total IPG is 0.096 microseconds for the link speed of 1000 Mbps.

TABLE 3-6 lists the default values and allowable values for the IPG parameters.

TABLE 3-6 Read-Write Interpacket Gap Parameter Values and Descriptions

Parameter	Values (Byte-time)	Description
<code>ipg1</code>	0 to 255	Interpacket gap 1 (Default=8)
<code>ipg2</code>	0 to 255	Interpacket gap 2 (Default=4)

By default, the driver sets `ipg1` to 8-byte time and `ipg2` to 4-byte time, which are the standard values. (Byte time is the time it takes to transmit one byte on the link, with a link speed of 1000 Mbps.)

If your network has systems that use longer IPG (the sum of `ipg1` and `ipg2`), and if those machines seem to be slow in accessing the network, increase the values of `ipg1` and `ipg2` to match the longer IPGs of other machines.

Interrupt Parameters

TABLE 3-7 describes the receive interrupt blanking values.

TABLE 3-7 RX Blanking Register for Alias Read

Field Name	Values	Description
<code>rx-intr-pkts</code>	0 to 511	Interrupts after this number of packets have arrived since the last packet was serviced. A value of zero indicates no packet blanking (Default=3).
<code>rx-intr-time</code>	0 to 524287	Interrupts after 4.5 microseconds (Usecs) have elapsed since the last packet was serviced. A value of zero indicates no time blanking (Default=3).

Random Early Drop Parameters

These parameters provide the ability to drop packets based on the fullness of the receive FIFO. By default, this feature is disabled. When FIFO occupancy reaches a specific range, packets are dropped according to the preset probability. The probability should increase when the FIFO level increases. Control packets are never dropped and are not counted in the statistics.

TABLE 3-8 RX Random Early Detecting 8-Bit Vectors

Field Name	Values	Description
red-dv4to6k	0 to 255	Random early detection and packet drop vectors for a FIFO threshold greater than 4096 bytes and less than 6,144 bytes. Probability of drop can be programmed on a 12.5 percent granularity. For example, if bit 0 is set, the first packet out of every eight is dropped in this region (Default=0).
red-dv6to8k	0 to 255	Random early detection and packet drop vectors for a FIFO threshold greater than 6,144 bytes and less than 8,192 bytes. Probability of drop can be programmed on a 12.5 percent granularity. For example, if bit 8 is set, the first packet out of every eight is dropped in this region (Default=0).
red-dv8to10k	0 to 255	Random early detection and packet drop vectors for a FIFO threshold greater than 8,192 bytes and less than 10,240 bytes. Probability of drop can be programmed on a 12.5 percent granularity. For example, if bit 16 is set, the first packet out of every eight is dropped in this region (Default=0).
red-dv10to12k	0 to 255	Random early detection and packet drop vectors for a FIFO threshold greater than 10,240 bytes and less than 12,288 bytes. Probability of drop can be programmed on a 12.5 percent granularity. For example, if bit 24 is set, the first packet out of every eight is dropped in this region (Default=0).

PCI Bus Interface Parameters

These parameters enable you to modify PCI interface features to gain better PCI interperformance for a given application.

TABLE 3-9 PCI Bus Interface Parameters

Parameter	Description
<code>tx-dma-weight</code>	Determines the multiplication factor for accrediting the transmit (TX) side during a heavy round robin arbitration; the values are 0 to 3 (Default=0). Zero means no extra weight. The other values use an exponent of two for heavy traffic. For example, if <code>tx-dma-weight</code> = 0 and <code>rx-dma-weight</code> = 3, then as long as RX traffic is continuously arriving, the priority of RX traffic will be 8 times greater than the priority of TX traffic to access the PCI.
<code>rx-dma-weight</code>	Determines the multiplication factor for granting credit to the RX side during a weighted round robin arbitration. The values are 0 to 3 (Default=0).
<code>infinite-burst</code>	If enabled, this parameter allows the infinite burst capability to be used if the system supports infinite burst. The adapter does not free the bus until complete packets are transferred across the bus. The values are 0 or 1 (Default=0).
<code>disable-64bit</code>	Switches off 64-bit capability of the adapter. Note: for UltraSPARC® III based platforms, this parameter might be set to 1 by default. For UltraSPARC II based platforms, the default is 0. The values are 0 or 1 (Default=0, which enables 64-bit capability).

Setting vca Driver Parameters

You can set the vca device driver parameters in two ways:

- Using the `ndd` utility
- Using the `vca.conf` file

If you use the `ndd` utility, the parameters are valid only until you reboot the system. This method is good for testing parameter settings.

To set parameters so they remain in effect after you reboot the system, create a `/kernel/drv/vca.conf` file and add parameter values to this file when you need to set a particular parameter for a device in the system. See “To Set Driver Parameters Using a vca.conf File” on page 39 for details.

Setting Parameters Using the `ndd` Utility

Use the `ndd` utility to configure parameters that are valid until you reboot the system.

The following sections describe how you can use the vca driver and the `ndd` utility to modify (with the `-set` option) or display (without the `-set` option) the parameters for each vca device.

▼ To Specify Device Instances for the `ndd` Utility

Before you use the `ndd` utility to get or set a parameter for a vca device, you must specify the device instance for the utility.

1. **Check the `/etc/path_to_inst` file to identify the instance number associated with a particular device. Refer to the online manual pages for `path_to_inst(4)`.**

```
# grep vca /etc/path_to_inst
"/pci@8,600000/network@1" 0 "vca"
"/pci@8,700000/network@1" 1 "vca"
```

In the previous example, the three Sun Crypto Accelerator 4000 Ethernet instances are from the installed adapters. The instance numbers are 0 and 1.

2. **Use the instance number to select the device.**

```
# ndd -set /dev/vcaN
```

Note – In the examples in this user’s guide, *N* represents the instance number of the device.

The device remains selected until you change the selection.

Noninteractive and Interactive Modes

You can use the `ndd` utility in two modes:

- Noninteractive
- Interactive

In noninteractive mode, you invoke the utility to execute a specific command. Once the command is executed, you exit the utility. In interactive mode, you can use the utility to get or set more than one parameter value. Refer to the `ndd(1M)` online manual page for more information.

Using the `ndd` Utility in Noninteractive Mode

This section describes how to modify and display parameter values.

- **To modify a parameter value, use the `-set` option.**

If you invoke the `ndd` utility with the `-set` option, the utility passes *value*, which must be specified to the named `/dev/vcaN` driver instance, and assigns it to the parameter:

```
# ndd -set /dev/vcaN parameter value
```

When you change any `adv` parameter, a message similar to the following appears:

```
- link up 1000 Mbps half duplex
```

- **To display the value of a parameter, specify the parameter name and omit the value.**

When you omit the `-set` option, a query operation is assumed and the utility queries the named driver instance, retrieves the value associated with the specified parameter, and prints it:

```
# ndd /dev/vcaN parameter
```

Note – In the previous example, *N* is the instance number of the `vca` device. This number should reflect the instance number of the board for which you are running the `kstat` command.

Using the `ndd` Utility in Interactive Mode

- **To modify a parameter value in interactive mode, specify `ndd /dev/vcaN`, as shown below.**

The `ndd` utility then prompts you for the name of the parameter:

```
# ndd /dev/vcaN
name to get/set? (Enter the parameter name or ? to view all
parameters)
```

Note – In the previous example, *N* is the instance number of the `vca` device. This number should reflect the instance number of the board for which you are running the `kstat` command.

After typing the parameter name, the `ndd` utility prompts you for the parameter value (see TABLE 3-1 through TABLE 3-9).

- To list all the parameters supported by the `vca` driver, type `ndd /dev/vcaN`.
(See TABLE 3-1 through TABLE 3-9 for parameter descriptions.)

```
# ndd /dev/vcaN
name to get/set ? ?
?                               (read only)
instance                         (read and write)
adv-autoneg-cap                   (read and write)
adv-1000fdx-cap                   (read and write)
adv-1000hdx-cap                   (read and write)
adv-100fdx-cap                    (read and write)
adv-100hdx-cap                    (read and write)
adv-10fdx-cap                     (read and write)
adv-10hdx-cap                     (read and write)
adv-asmppause-cap                 (read and write)
adv-pause-cap                     (read and write)
pause-on-threshold                (read and write)
pause-off-threshold               (read and write)
link-master                       (read and write)
enable-ipg0                       (read and write)
ipg0                              (read and write)
ipg1                              (read and write)
ipg2                              (read and write)
rx-intr-pkts                      (read and write)
rx-intr-time                      (read and write)
red-p4k-to-6k                     (read and write)
red-p6k-to-8k                     (read and write)
red-p8k-to-10k                   (read and write)
red-p10k-to-12k                  (read and write)
tx-dma-weight                     (read and write)
rx-dma-weight                     (read and write)
infinite-burst                    (read and write)
disable-64bit                     (read and write)
name to get/set ?
#
```

Note – In the previous example, *N* is the instance number of the `vca` device. This number should reflect the instance number of the board for which you are running the `kstat` command.

Setting Autonegotiation or Forced Mode

The following link parameters can be set to operate in either autonegotiation or forced mode:

- speed
- duplex
- link-clock

By default, autonegotiation mode is enabled for these link parameters. When either of these parameters are in autonegotiation mode, the vca device communicates with the link partner to negotiate a compatible value and flow control capability. When a value other than `auto` is set for either of these parameters, no negotiation occurs and the link parameter is configured in forced mode. In forced mode, the value for the `speed` parameter must match between link partners. See “Enabling Autonegotiation or Forced Mode for Link Parameters With the OpenBoot PROM” on page 42.

▼ To Disable Autonegotiation Mode

If your network equipment does not support autonegotiation, or if you want to force your network `speed`, `duplex`, or `link-clock` parameters, you can disable the autonegotiation mode on the vca device.

1. Set the following driver parameters to the values that are described in the documentation delivered with your link partner device (for example, a switch):

- `adv-1000fdx-cap`
- `adv-1000hdx-cap`
- `adv-100fdx-cap`
- `adv-100hdx-cap`
- `adv-10fdx-cap`
- `adv-10hdx-cap`
- `adv-asmpause-cap`
- `adv-pause-cap`

See TABLE 3-2 for the descriptions and possible values of these parameters.

2. Set the `adv-autoneg-cap` parameter to 0.

```
# ndd -set /dev/vcaN adv-autoneg-cap 0
```

When you change any `ndd` link parameter, a message similar to the following appears:

```
link up 1000 Mbps half duplex
```

Note – If you disable autonegotiation, you must enable the `speed`, `duplex`, and `link-clock` (1000 Mbps only) parameters to operate in forced mode. For instructions, see “Enabling Autonegotiation or Forced Mode for Link Parameters With the OpenBoot PROM” on page 42.

Setting Parameters Using the `vca.conf` File

You can also specify the driver parameter properties by adding entries to the `vca.conf` file in the `/kernel/drv` directory. The parameter names are the same names listed in “Driver Parameter Values and Definitions” on page 26.



Caution – Do not remove any of the default entries in the `/kernel/drv/vca.conf` file.

The online manual pages for `prtconf(1)` and `driver.conf(4)` include additional details. The next procedure shows an example of setting parameters in a `vca.conf` file.

Variables defined in the previous section apply to known devices in the system. To set a variable for a Sun Crypto Accelerator 4000 board with the `vca.conf` file, you must know the following three pieces of information for the device: device name, device parent, and device unit address.

▼ To Set Driver Parameters Using a `vca.conf` File

1. Obtain the hardware path names for the `vca` devices in the device tree.

- a. Check the `/etc/driver_aliases` file to identify the name associated with a particular device.

```
# grep vca /etc/driver_aliases
vca "pci108e,3de8"
```

In the previous example, the device name associated with the Sun Crypto Accelerator 4000 software driver (`vca`) is `"pci108e,3de8"`.

b. Locate the device parent name and device unit address in the `/etc/path_to_inst` file.

Refer to the online manual pages for `path_to_inst(4)`.

```
# grep vca /etc/path_to_inst
"/pci@8,600000/network@1" 0 "vca"
"/pci@8,700000/network@1" 1 "vca"
```

In the previous example, there are three columns of output: Device path name, instance number, and software driver name.

The device path name in the first line of the previous example is `"/pci@8,600000/network@1"`. Device path names are made up of three parts: Device parent name, device node name, and device unit address. See TABLE 3-10.

TABLE 3-10 Device Path Name

Entire Device Path Name	Parent Name Portion	Node Name Portion	Unit Address Portion
<code>"/pci@8,600000/network@1"</code>	<code>/pci@8,600000</code>	<code>network</code>	<code>1</code>
<code>"/pci@8,700000/network@1"</code>	<code>/pci@8,700000</code>	<code>network</code>	<code>1</code>

To identify a PCI device unambiguously in the `vca.conf` file, use the entire device path name (parent name, node name, and the unit address) for the device. Refer to the `pci(4)` online manual page for more information about the PCI device specification.

2. Set the parameters for the `vca` devices in the `/kernel/drv/vca.conf` file.

In the following entry, the `adv-autoneg-cap` parameter is disabled for a particular Sun Crypto Accelerator 4000 Ethernet device.

```
name="pci108e,3de8" parent="/pci@8,700000" unit-address="1" adv-autoneg-cap=0;
```

- 3. Save the `vca.conf` file.**
- 4. Save and close all files and programs, and exit the windowing system.**
- 5. Shut down and reboot the system.**

Setting Parameters for All Sun Crypto Accelerator 4000 `vca` Devices With the `vca.conf` File

If you omit the device path name (parent name, node name, and the unit address), the variable is set for all instances of all Sun Crypto Accelerator 4000 Ethernet devices.

▼ To Set Parameters for All Sun Crypto Accelerator 4000 `vca` Devices With the `vca.conf` File

1. Add a line in the `vca.conf` file to change the value of a parameter for all instances by entering `parameter=value;`

The following example sets the `adv-autoneg-cap` parameter to 1 for all instances of all Sun Crypto Accelerator 4000 Ethernet devices:

```
adv-autoneg-cap=1;
```

Example `vca.conf` File

The following is an example `vca.conf` file:

```
#
# Copyright 2003 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
#ident "@(#)vca.conf 1.3 03/10/13 SMI"

#
# Use the new Solaris 9 ddi-no-autodetach property to prevent the
# driver from being unloaded by the cleanup modunload -i 0.
#
ddi-no-autodetach=1;
```

Enabling Autonegotiation or Forced Mode for Link Parameters With the OpenBoot PROM

The following parameters can be configured to operate in autonegotiation or forced mode at the OpenBoot PROM interface:

TABLE 3-11 Local Link Network Device Parameters

Parameter	Description
speed	This parameter can be set to <code>auto</code> , <code>1000</code> , <code>100</code> , or <code>10</code> ; the syntax is as follows: <ul style="list-style-type: none">• <code>speed=auto</code> (default)• <code>speed=1000</code>• <code>speed=100</code>• <code>speed=10</code>
duplex	This parameter can be set to <code>auto</code> , <code>full</code> , or <code>half</code> ; the syntax is as follows: <ul style="list-style-type: none">• <code>duplex=auto</code> (default)• <code>duplex=full</code>• <code>duplex=half</code>
link-clock	This parameter is applicable only if the <code>speed</code> parameter is set to <code>1000</code> or if you are using a 1000 Mbps MMF Sun Crypto Accelerator 4000 board. The value for this parameter must correspond to the value on the link partner—for example, if the local link has a value of <code>master</code> , the link partner must have a value of <code>slave</code> . This parameter can be set to <code>master</code> , <code>slave</code> , or <code>auto</code> ; the syntax is as follows: <ul style="list-style-type: none">• <code>link-clock=auto</code> (default)• <code>link-clock=master</code>• <code>link-clock=slave</code>

To establish a proper link, the `speed`, `duplex`, and `link-clock` (1000 Mbps only) parameters must be configured correctly between the local link and the link partner. Both link partners must operate in either autonegotiation or forced mode for each of the `speed`, `duplex`, and `link-clock` (1000 Mbps only) parameters. A value of `auto` for any of these parameters configures the link to operate in autonegotiation mode for that parameter. The absence of a parameter at the OpenBoot PROM `ok` prompt configures that parameter to have a default value of `auto`. A value other than `auto` configures the local link to operate in forced mode for that parameter.

When the local link is operating in autonegotiation mode for the `speed` and `duplex` parameters at 100 Mbps and below, and both full and half duplexes, then the link partner uses either the 100 Mbps or 10 Mbps speeds with either duplex.

When the `speed` parameter is operating in forced mode, the value must match the `speed` value of the link-partner. If the `duplex` parameter does not match between the local link and the link partner, the link might come up; however, traffic collisions will occur.

When the local link `speed` parameter is set to autonegotiation and the link partner `speed` parameter is set to forced, the link might come up depending on whether the `speed` value can be negotiated between the local link and the link partner. The interface in autonegotiation mode always tries to establish a link (if there is a speed match) at half duplex by default. Because one of the two interfaces is not in autonegotiation mode, the interface in autonegotiation mode detects only the `speed` parameter; the `duplex` parameter is not detected. This method is called parallel-detection.



Caution – The establishment of a link with a duplex conflict always leads to traffic collisions.

For a local link parameter to operate in forced mode, the parameter must have a value other than `auto`. For example, to establish a forced mode link at 100 Mbps with half duplex, type the following at the OpenBoot PROM `ok` prompt:

```
ok boot net:speed=100,duplex=half
```

Note – In the examples in this section, `net` is an alias for the default, integrated network interface device path. You can configure other network devices by specifying a device path instead of using `net`.

To establish a forced mode link at 1000 Mbps with half duplex that is a clock master, type the following command at the OpenBoot PROM `ok` prompt:

```
ok boot net:speed=1000,duplex=half,link-clock=master
```

Note – The `link-clock` parameter must have a value that corresponds to the `link-clock` value of the link partner. For example, if the `link-clock` value on the local link is set to `master`, the `link-clock` value on the link partner must be set to `slave`.

To establish a forced mode for a speed of 10 Mbps and an autonegotiation mode for duplex, type the following at the OpenBoot PROM `ok` prompt:

```
ok boot net:speed=10,duplex=auto
```

You could also type the following at the OpenBoot PROM `ok` prompt to establish the same local link parameters as the previous example:

```
ok boot net:speed=10
```

Refer to the IEEE 802.3 documentation for further details.

Cryptographic and Ethernet Driver Operating Statistics

This section describes the statistics presented by the `kstat(1M)` command.

Cryptographic Driver Statistics

TABLE 3-12 describes the cryptographic driver statistics.

TABLE 3-12 Cryptographic Driver Statistics

Parameter	Description	Stable or Unstable
<code>vs-mode</code>	The values are <code>FIPS</code> , <code>standard</code> , or <code>uninitialized</code> . <code>FIPS</code> indicates that the board is in FIPS mode. <code>standard</code> indicates that the board is not in FIPS mode. <code>uninitialized</code> indicates that the board is not initialized.	Stable
<code>vs-status</code>	The values are <code>ready</code> , <code>faulted</code> , or <code>failsafe</code> . <code>ready</code> indicates that the board is operating normally. <code>faulted</code> indicates that the board is not operating. <code>failsafe</code> indicates failsafe mode, which is the original factory state of the board.	Stable

Ethernet Driver Statistics

TABLE 3-13 describes the Ethernet driver statistics.

TABLE 3-13 Ethernet Driver Statistics

Parameter	Description	Stable or Unstable
<code>ipackets</code>	Number of inbound packets.	Stable
<code>ipackets64</code>	64-bit version of <code>ipackets</code> .	Stable
<code>ierrors</code>	Total packets received that could not be processed because they contained errors (long).	Stable
<code>opackets</code>	Total packets requested to be transmitted on the interface.	Stable
<code>opackets64</code>	Total packets requested to be transmitted on the interface (64-bit).	Stable
<code>oerrors</code>	Total packets that were not successfully transmitted because of errors (long).	Stable
<code>rbytes</code>	Total bytes successfully received on the interface.	Stable
<code>rbytes64</code>	Total bytes successfully received on the interface (64-bit).	Stable
<code>obytes</code>	Total bytes requested to be transmitted on the interface.	Stable
<code>obytes64</code>	Total bytes requested to be transmitted on the interface (64-bit).	Stable
<code>multircv</code>	Multicast packets successfully received, including group and functional addresses (long).	Stable
<code>multixmt</code>	Multicast packets requested to be transmitted, including group and functional addresses (long).	Stable
<code>brdstrcv</code>	Broadcast packets successfully received (long).	Stable
<code>brdstxmt</code>	Broadcast packets requested to be transmitted (long).	Stable
<code>norcvbuf</code>	Times that a valid incoming packet was known to be discarded because a buffer could not be allocated for the receive packet (long).	Stable
<code>noxmtbuf</code>	Packets discarded on output because transmit buffer was busy, or no buffer could be allocated for transmit (long).	Stable

TABLE 3-14 describes the transmit and receive MAC counters.

TABLE 3-14 TX and RX MAC Counters

Parameter	Description	Stable or Unstable
tx-collisions	16-bit loadable counter increments for every frame transmission attempt that resulted in a collision.	Stable
tx-first-collisions	16-bit loadable counter increments for every frame transmission that experienced a collision on the first attempt, but was successfully transmitted on the second attempt.	Unstable
tx-excessive-collisions	16-bit loadable counter increments for every frame transmission that has exceeded the Attempts Limit.	Unstable
tx-late-collisions	16-bit loadable counter increments for every frame transmission that has experienced a collision. The parameter indicates the number of frames that the TxMAC has dropped due to collisions that occurred after transmitting at least the Minimum Frame Size number of bytes. Usually this is an indication that at least one station on the network violates the maximum allowed span of the network.	Unstable
tx-defer-timer	16-bit loadable timer increments when the TxMAC is deferring to traffic on the network while it is attempting to transmit a frame. The time base for the timer is the media byte clock divided by 256.	Unstable
tx-peak-attempts	8-bit register indicates the highest number of consecutive collisions per successfully transmitted frame, that have occurred since this register was last read. The maximum value that this register can attain is 255. A maskable interrupt is generated to the software if the number of consecutive collisions per successfully transmitted frame exceeds 255. This register is automatically cleared at 0 after it is read.	Unstable

TABLE 3-14 TX and RX MAC Counters (Continued)

Parameter	Description	Stable or Unstable
tx-underrun	16-bit loadable counter increments after a valid frame has been received from the network.	Unstable
rx-length-err	16-bit loadable counter increments after a frame, whose length is greater than the value that was programmed in the Maximum Frame Size Register, has been received from the network.	Unstable
rx-alignment-err	16-bit loadable counter increments when an alignment error is detected in a receive frame. An alignment error is reported when a receive frame fails the cyclic redundancy checksum (CRC) checking algorithm, <i>and</i> the frame contains a noninteger number of bytes (that is, the frame size in bits is not equal to zero).	Unstable
rx-crc-err	16-bit loadable counter increments when a receive frame fails the CRC checking algorithm, <i>and</i> the frame contains an integer number of bytes (that is, the frame size in bits modulo 8 is equal to zero).	Unstable
rx-code-violations	16-bit loadable counter increments when an Rx_Err indication is generated by the XCVR over the MII, while a frame is being received. This indication is generated by the transceiver when it detects an invalid code in the received data stream. A receive code violation is not counted as an FCS or an Alignment error.	Unstable
rx-overflows	Number of Ethernet frames dropped due to lack of resources.	Unstable
rx-no-buf	Number of times the hardware cannot receive data because there is no more receive buffer space.	Unstable
rx-no-comp-wb	Number of times the hardware cannot post completion entries for received data.	Unstable
rx-len-mismatch	Number of received frames where the asserted length does not match the actual frame length.	Unstable

The following Ethernet properties (TABLE 3-15) are derived from the intersection of device capabilities and the link partner capabilities.

TABLE 3-15 Current Ethernet Link Properties

Parameter	Description	Stable or Unstable
ifspeed	1000, 100, or 10 Mbps	Stable
link-duplex	0=half, 1=full	Stable
link-pause	Current pause setting for the link, see “Flow Control Parameters” on page 29	Stable
link-asmPause	Current pause setting for the link, see “Flow Control Parameters” on page 29	Stable
link-up	1=up, 0=down	Stable
link-status	1=up, 0=down	Stable
xcvr-inuse	Type of transceiver in use: 1=internal MII, 2=external MII, 3=external PCS	Stable

TABLE 3-16 describes the read-only Media Independent Interface (MII) capabilities. These parameters define the capabilities of the hardware. The Gigabit Media Independent Interface (GMII) supports all of the following capabilities.

TABLE 3-16 Read-Only vca Device Capabilities

Parameter	Description	Stable or Unstable
cap-autoneg	0 = Not capable of autonegotiation 1 = Autonegotiation capable	Stable
cap-1000fdx	Local interface full-duplex capability 0 = Not 1000 Mbps full-duplex capable 1 = 1000 Mbps full-duplex capable	Stable
cap-1000hdx	Local interface half-duplex capability 0 = Not 1000 Mbps half-duplex capable 1 = 1000 Mbps half-duplex capable	Stable
cap-100fdx	Local interface full-duplex capability 0 = Not 100 Mbps full-duplex capable 1 = 100 Mbps full-duplex capable	Stable
cap-100hdx	Local interface half-duplex capability 0 = Not 100 Mbps half-duplex capable 1 = 100 Mbps half-duplex capable	Stable
cap-10fdx	Local interface full-duplex capability 0 = Not 10 Mbps full-duplex capable 1 = 10 Mbps full-duplex capable	Stable

TABLE 3-16 Read-Only vca Device Capabilities (*Continued*)

Parameter	Description	Stable or Unstable
cap-10hdx	Local interface half-duplex capability 0 = Not 10 Mbps half-duplex capable 1 = 10 Mbps half-duplex capable	Stable
cap-asm-pause	Local interface flow control capability 0 = Not asymmetric pause capable 1 = Asymmetric pause (from the local device) capable (See “Flow Control Parameters” on page 29)	Stable
cap-pause	Local interface flow control capability 0 = Not Symmetric pause capable 1 = Symmetric pause capable (See “Flow Control Parameters” on page 29)	Stable

Reporting the Link Partner Capabilities

TABLE 3-17 describes the read-only link partner capabilities.

TABLE 3-17 Read-Only Link Partner Capabilities

Parameter	Description	Stable or Unstable
lp-cap-autoneg	0 = No autonegotiation 1 = Autonegotiation	Stable
lp-cap-1000fdx	0 = No 1000 Mbps full-duplex transmission 1 = 1000 Mbps full-duplex	Stable
lp-cap-1000hdx	0 = No 1000 Mbps half-duplex transmission 1 = 1000 Mbps half-duplex	Stable
lp-cap-100fdx	0 = No 100 Mbps full-duplex transmission 1 = 100 Mbps full-duplex	Stable
lp-cap-100hdx	0 = No 100 Mbps half-duplex transmission 1 = 100 Mbps half-duplex	Stable
lp-cap-10fdx	0 = No 10 Mbps full-duplex transmission 1 = 10 Mbps full-duplex	Stable

TABLE 3-17 Read-Only Link Partner Capabilities *(Continued)*

Parameter	Description	Stable or Unstable
lp-cap-10hdx	0 = No 10 Mbps half-duplex transmission 1 = 10 Mbps half-duplex	Stable
lp-cap-asm-pause	0 = Not asymmetric pause capable 1 = Asymmetric pause towards link partner capability (See “Flow Control Parameters” on page 29)	Stable
lp-cap-pause	0 = Not symmetric pause capable 1 = Symmetric pause capable (See “Flow Control Parameters” on page 29)	Stable

If the link partner is not capable of autonegotiation (when `lp-cap-autoneg` is 0), the remaining information described in TABLE 3-17 is not relevant and the parameter value is 0.

If the link partner is capable of autonegotiation (when `lp-cap-autoneg` is 1), then the speed and mode information is displayed when you use autonegotiation and the link partner capabilities.

TABLE 3-18 describes the driver-specific parameters.

TABLE 3-18 Driver-Specific Parameters

Parameter	Description	Stable or Unstable
lb-mode	Copy of the loopback mode the device is in, if any.	Unstable
promisc	When enabled, the device is in promiscuous mode. When disabled, the device is not in promiscuous mode.	Unstable

Ethernet Transmit Counters

tx-wsrsv	Count of the number of times the transmit ring is full.	Unstable
tx-msgdup-fail	Attempt to duplicate packet failure.	Unstable
tx-allocb-fail	Attempt to allocate memory failure.	Unstable
tx-queue0	Number of packets queued for transmission on the first hardware transmit queue.	Unstable
tx-queue1	Number of packets queued for transmission on the second hardware transmit queue.	Unstable
tx-queue2	Number of packets queued for transmission on the third hardware transmit queue.	Unstable

TABLE 3-18 Driver-Specific Parameters (*Continued*)

Parameter	Description	Stable or Unstable
tx-queue3	Number of packets queued for transmission on the fourth hardware transmit queue.	Unstable
<i>Ethernet Receive Counters</i>		
rx-hdr-pkts	Number of packets received that were less than 256 bytes.	Unstable
rx-mtu-pkts	Number of packets received that were greater than 256 bytes and less than 1514 bytes.	Unstable
rx-split-pkts	Number of packets that were split across two pages.	Unstable
rx-nocanput	Number of packets dropped due to failures on delivery to the IP stack.	Unstable
rx-msgdup-fail	Number of packets that could not be duplicated.	Unstable
rx-allocb-fail	Number of block allocation failures.	Unstable
rx-new-pages	Number of pages that were replaced during reception.	Unstable
rx-new-hdr-pages	Number of pages that were filled with packets less than 256 bytes that were replaced during reception.	Unstable
rx-new-mtu-pages	Number of pages that were filled with those packets greater than 256 bytes and less than 1514 that got replaced during reception.	Unstable
rx-new-nxt-pages	Number of pages that contained packets that were split across pages that were replaced during reception.	Unstable
rx-page-alloc-fail	Number of page allocation failures.	Unstable
rx-mtu-drops	Number of times a whole page of packets greater than 256 bytes and less than 1514 was dropped because the driver was unable to map a new one to replace the page.	Unstable
rx-hdr-drops	Number of times a whole page of packets less than 256 bytes was dropped because the driver was unable to map a new one to replace the page.	Unstable
rx-nxt-drops	Number of times a page with a split packet was dropped because the driver was unable to map a new one to replace the page.	Unstable

TABLE 3-18 Driver-Specific Parameters (*Continued*)

Parameter	Description	Stable or Unstable
rx-rel-flow	Number of times the driver was told to release a flow.	Unstable
<i>Ethernet PCI Properties</i>		
rev-id	Revision ID of the Sun Crypto Accelerator 4000 Ethernet device useful for recognition of a device being used in the field.	Unstable
pci-err	Sum of all PCI errors.	Unstable
pci-rta-err	Number of target aborts received.	Unstable
pci-rma-err	Number of master aborts received.	Unstable
pci-parity-err	Number of PCI parity errors detected.	Unstable
pci-drto-err	Number of times the delayed transaction retry time-out was reached.	Unstable
dma-mode	Used by the Sun Crypto Accelerator 4000 driver (vca).	Unstable

▼ To Check Link Partner Settings

- As superuser, type the `kstat vca:N` command:

```
# kstat vca:N
module: vca                instance: 0
name:   vca0               class:   misc
```

Where *N* is the instance number of the `vca` device. This number should reflect the instance number of the board for which you are running the `kstat` command.

IPsec In-Line Acceleration Statistics

TABLE 3-19 describes the kernel statistics that are incremented when the board is configured for in-line IPsec hardware acceleration. See “Enabling In-Line IPsec Acceleration” on page 57 for instructions on how to configure the board to use the in-line IPsec configuration.

TABLE 3-19 Cryptographic Driver Statistics for In-Line IPsec Acceleration

Parameter	Description	Stable or Unstable
<code>ipsec_ierrors</code>	Total IPsec packets received that could not be processed because they contained errors (long)	Stable
<code>ipsec_ipackets</code>	Number of inbound IPsec packets	Stable
<code>ipsec_ipackets64</code>	Number of inbound IPsec packets (64-bit)	Stable
<code>ipsec_obytes</code>	Total IPsec bytes requested to be transmitted on the interface	Stable
<code>ipsec_obytes64</code>	Total IPsec bytes requested to be transmitted on the interface (64-bit)	Stable
<code>ipsec_oerrors</code>	Total IPsec packets that were not successfully transmitted because of errors (long)	Stable
<code>ipsec_opackets</code>	Total IPsec packets requested to be transmitted on the interface	Stable
<code>ipsec_opackets64</code>	Total IPsec packets requested to be transmitted on the interface (64-bit)	Stable
<code>ipsec_rbytes</code>	Total IPsec bytes successfully received on the interface	Stable
<code>ipsec_rbytes64</code>	Total IPsec bytes successfully received on the interface (64-bit)	Stable
<code>sadb_cache_misses</code>	Number of firmware cache misses	Stable
<code>sadb_cache_overflows</code>	Number of firmware cache overflows	Stable
<code>sadb_entries</code>	Number of entries in the SADB driver	Stable
<code>sadb_operations</code>	Number of SADB operations sent from Solaris IPsec to the driver	Stable

Note – The IPsec kernel statistics listed in TABLE 3-19 are only incremented for IPsec packets that are actually processed in-line by the hardware. Receive packets of less than 256 bytes are not processed in-line and the IPsec kernel statistics will not be incremented for these packets. These kernel statistics also do not apply to out-of-band IPsec traffic (See “Configuring IPsec Hardware Acceleration” on page 56). If `snoop` is enabled, these counters are not incremented. Out-of-band packets will increment the regular network kernel statistics and any applicable cryptographic statistics, that is, `3desbytes` and `3desjobs`.

Network Configuration

This section describes how to edit the network host files after the adapter has been installed on your system.

Configuring the Network Host Files

After installing the driver software, you must create a `hostname.vcaN` file for the adapter’s Ethernet interface. Note that in the file name `hostname.vcaN`, `N` corresponds to the instance number of the `vca` interface you plan to use. You must also create both an IP address and a host name for its Ethernet interface in the `/etc/hosts` file.

1. **Locate the correct `vca` interfaces and instance numbers in the `/etc/path_to_inst` file.**

Refer to the online manual pages for `path_to_inst(4)`.

```
# grep vca /etc/path_to_inst
"/pci@8,600000/network@1" 0 "vca"
```

The instance number in the previous example is 0.

2. Use the `ifconfig(1M)` command to set up the adapter's `vca` interface.

Use the `ifconfig` command to assign an IP address to the network interface. Type the following at the command line, replacing *ip-address* with the adapter's IP address:

```
# ifconfig vcaN plumb ip-address up
```

Refer to the `ifconfig(1M)` man page and the Solaris documentation for more information.

- If you want a setup that will remain the same after you reboot, create an `/etc/hostname.vcaN` file, where *N* corresponds to the instance number of the `vca` interface you plan to use.

To use the `vca` interface of the example shown in Step 1, create an `/etc/hostname.vcaN` file, where *N* corresponds to the instance number of the device which is 0 in this example. If the instance number were 1, the file name would be `/etc/hostname.vca1`.

- Do not create an `/etc/hostname.vcaN` file for a Sun Crypto Accelerator 4000 interface you plan to leave unused.
- The `/etc/hostname.vcaN` file must contain the host name for the appropriate `vca` interface.
- The host name must have an IP address and must be listed in the `/etc/hosts` file.
- The host name must be different from any other host name of any other interface, for example, `/etc/hostname.vca0` and `/etc/hostname.vca1` cannot share the same host name.

The following example shows the `/etc/hostname.vcaN` file required for a system named `zardoz` that has a Sun Crypto Accelerator 4000 board (`zardoz-11`).

```
# cat /etc/hostname.hme0
zardoz
# cat /etc/hostname.vca0
zardoz-11
```

3. Create an appropriate entry in the `/etc/hosts` file for each active vca interface.

For example:

```
# cat /etc/hosts
#
# Internet host table
#
127.0.0.1    localhost
129.144.10.57 zardoz    loghost
129.144.11.83 zardoz-11
```

Configuring IPsec Hardware Acceleration

The board has two configurations of IPsec hardware acceleration: in-line and out-of-band. Both configurations accelerate IPsec cryptographic operations. However, because each method offers different advantages, overall system requirements should be evaluated to determine the appropriate configuration.

Note – IPsec acceleration is supported in Solaris 9 onward, and is not supported in Solaris 8. In-line IPsec acceleration is only supported in Solaris 9 12/03 onward (See TABLE 3-20).

TABLE 3-20 Solaris Release Requirements for IPsec Acceleration

Solaris Version	Out-of-Band Acceleration	In-Line Acceleration
All Solaris 8 releases	Not Supported	Not Supported
Solaris 9 to Solaris 9 8/03	Supported	Not Supported
Solaris 9 12/03 onward	Supported	Supported

Out-of-band is the default IPsec configuration, and is optimized for performance on a multiprocessor system. This configuration offloads DES and 3DES cryptographic functions to the board, and is the preferred configuration on a multiprocessor system for which host processing power is not an issue.

In-line IPsec configuration augments out-of-band functionality with authentication support (MD5 and SHA1), and offloads portions of the host packet processing to the board. By handling the additional packet processing, the board significantly reduces host CPU usage.

Note – Out-of-band might provide greater IPsec throughput than in-line on multiprocessor systems that only require DES or 3DES encryption algorithms.

Enabling Out-of-Band IPsec Acceleration

Solaris 9 or later is required. Out-of-band is the default configuration for the board. No IPsec configuration or tuning is required to use the board for out-of-band IPsec acceleration in Solaris 9. You simply install the Sun Crypto Accelerator 4000 packages and reboot.

Enabling In-Line IPsec Acceleration

Solaris 9 12/03 or later is required. To configure in-line acceleration, you must change configuration files in both the Solaris software and the `vca` driver.

▼ To Enable In-Line IPsec Hardware Acceleration

1. **Enable in-line acceleration in the Solaris software by adding the following entry to the `/etc/system` configuration file:**

```
set ip:ip_use_dl_cap=1
```

For the change in the `/etc/system` file to take effect, the system must be rebooted.

2. **Enable in-line acceleration in the `vca` driver by adding the following entry to the `/kernel/drv/vca.conf` configuration file:**

```
inline-ipsec=1;
```

For the change in the `/kernel/drv/vca.conf` file to take effect, you must either reboot the system or unload and reload the `vca` driver.

Note – In-line acceleration should not be enabled in the driver if it is not enabled in the Solaris software because doing so might degrade non-IPsec performance.

Once in-line acceleration has been enabled, the Solaris software IPsec policies can be configured for the interface with the standard IPsec configuration procedures. For information on configuring IPsec policies in Solaris refer to the *IPsec and IKE Administration Guide* available at: <http://docs.sun.com>

In-line acceleration can be used to accelerate both AH and ESP algorithms; however, multiple nested transforms (including AH+ESP) cannot be performed on the board. If multiple transforms are applied, only the outermost transform is performed in-line. The remaining transforms are performed by the Solaris IPsec configuration. These transforms may also be done in hardware (out-of-band) if the KCL IPsec acceleration (`SUNWkcl2i.u`) package has been installed on a Solaris 9 system.

When the board is configured for IPsec in-line acceleration, additional statistics presented by the `kstat(1M)` command will be incremented. See TABLE 3-19 for descriptions of the IPsec in-line acceleration `kstat` statistics.

Administering the Sun Crypto Accelerator 4000 Board

This chapter provides an overview of administering the board with the `vcaadm`, `vcad`, `vcadiag`, `pk11export`, `utilities`. The following sections are included:

- “Using the `vcaadm` Utility” on page 59
- “Logging In and Out With `vcaadm`” on page 62
- “Entering Commands With `vcaadm`” on page 66
- “Initializing the Board With `vcaadm`” on page 68
- “Managing Keystores With `vcaadm`” on page 71
- “Managing Boards With `vcaadm`” on page 78
- “Using the `vcad` Command” on page 83
- “Using the `vcadiag` Utility” on page 89
- “Using the `pk11export` Utility” on page 92
- “Using the `iplsslcfg` Script” on page 93
- “Using the `apsslcfg` Script” on page 98
- “Assigning Different MAC Addresses to Multiple Boards Installed in the Same Server” on page 103

Using the `vcaadm` Utility

The `vcaadm` utility offers a command-line interface to the Sun Crypto Accelerator 4000 board. Only users designated as security officers are permitted to use the `vcaadm` utility. When you first connect to a Sun Crypto Accelerator 4000 board with `vcaadm`, you are prompted to create an initial security officer and password.

To access the `vcaadm` utility easily, place the Sun Crypto Accelerator 4000 tools directory in your search path, for example:

```
$ PATH=$PATH:/opt/SUNWconn/bin
$ export PATH
```

The `vcaadm` command-line syntax is:

- `vcaadm [-H]`
- `vcaadm [-y] [-h hostname] [-p port] [-d vcaN] [-f filename]`
- `vcaadm [-y] [-h hostname] [-p port] [-d vcaN] [-s sec-officer] command`

Note – When using the `-d` attribute, `vcaN` is the board's device name, where the `N` corresponds to the Sun Crypto Accelerator 4000 device instance number.

TABLE 4-1 shows the options for the `vcaadm` utility.

TABLE 4-1 `vcaadm` Options

Option	Meaning
<code>-H</code>	Displays help files for <code>vcaadm</code> commands and exits.
<code>-d vcaN</code>	Connects to the Sun Crypto Accelerator 4000 board that has <code>N</code> as the driver instance number. For example, <code>-d vca1</code> connects to device <code>vca1</code> where <code>vca</code> is a string in the board's device name and <code>1</code> is the instance number of the device. This value defaults to <code>vca0</code> and must be in the form of <code>vcaN</code> , where <code>N</code> corresponds to the device instance number.
<code>-f filename</code>	Interprets one or more commands from <code>filename</code> and exits.
<code>-h hostname</code>	Connects to the Sun Crypto Accelerator 4000 board on <code>hostname</code> . The value for <code>host</code> can be a host name or an IP address, and defaults to the loopback address.
<code>-p port</code>	Connects to the Sun Crypto Accelerator 4000 board on <code>port</code> . The value for <code>port</code> defaults to <code>6870</code> .
<code>-s sec-officer</code>	Logs in as a security officer named <code>sec-officer</code> .
<code>-y</code>	Forces a yes answer to any command that would normally prompt for a confirmation.

Note – The name `sec-officer` is used throughout this user's guide as an example security officer name.

Modes of Operation

`vcaadm` can run in one of three modes. These modes differ mainly in how commands are passed into `vcaadm`. The three modes are Single-Command mode, File mode, and Interactive mode.

Note – To use `vcaadm`, you must authenticate as security officer. How often you need to authenticate as security officer is determined by which operating mode you are using.

Single-Command Mode

In Single-Command mode, you must authenticate as security officer for every command. Once the command is executed, you are logged out of `vcaadm`.

When entering commands in Single-Command mode, you specify the command to be run after all the command-line switches are specified. For example, in Single-Command mode, the following command would show all the users in a given keystore and return the user to the command shell prompt.

```
$ vcaadm show user
Security Officer Name: sec-officer
Security Officer Password:
```

The following command performs a login as the security officer, `sec-officer`, and creates the user `web-admin` in the keystore.

```
$ vcaadm -s sec-officer create user web-admin
Security Officer Password:
Enter new user password:
Confirm password:
User web-admin created successfully.
```

Note – The first password is for the security officer, followed by the password and confirmation for the new user `web-admin`.

All output from Single-Command mode goes to the standard output stream. This output can be redirected using standard UNIX shell-based methods.

File Mode

In File mode, you must authenticate as security officer for every file you run. You are logged out of `vcaadm` after the commands in the command file are executed.

To enter commands in File mode, you specify a file from which `vcaadm` reads one or more commands. The file must be ASCII text, consisting of one command per line. Begin each comment with a pound sign (#) character. If the File mode option is set, `vcaadm` ignores any command-line arguments after the last option. The following example runs the commands in the `deluser.scr` file and answers all prompts in the affirmative:

```
$ vcaadm -f deluser.scr -y
```

Interactive Mode

In Interactive mode, you must authenticate as security officer every time you connect to a board. This is the default operating mode for `vcaadm`. To log out of `vcaadm` in Interactive mode, use the `logout` command. Refer to “Logging In and Out With `vcaadm`” on page 62.

Interactive mode presents the user with an interface similar to `ftp(1)`, where commands can be entered one at a time. The `-y` option is not supported in Interactive mode.

Logging In and Out With `vcaadm`

When you use `vcaadm` from the command line and specify *host*, *port*, and *device* using the `-h`, `-p`, and `-d` attributes respectively, you are immediately prompted to log in as security officer if a successful network connection was made.

The `vcaadm` utility establishes an encrypted network connection (channel) between the `vcaadm` application and the Sun Crypto Accelerator 4000 firmware running on a specific board.

During setup of the encrypted channel, boards identify themselves by their hardware Ethernet address and an RSA public key. A trust database (`$HOME/.vcaadm/trustdb`) is created the first time `vcaadm` connects to a board. This file contains all of the boards that are currently trusted by the security officer.

Logging In to a Board With `vcaadm`

If the security officer connects to a new board, `vcaadm` notifies the security officer and prompt the following options:

1. Abort the connection
2. Trust the connection one time only (no changes to trust database)
3. Trust this board forever (adds the hardware ethernet address and RSA public key to the trust database)

If the security officer connects to a board that has a remote access key that has been changed, `vcaadm` will notify the security officer and prompt the following three options:

1. Abort the connection
2. Trust the connection one time only (no changes to trust database)
3. Replace the old public key bound to this hardware ethernet address with the new public key

Logging In to a New Board

Note – The remaining examples in this chapter were created with the Interactive mode of `vcaadm`.

When connecting to a new board, `vcaadm` must create a new entry in the trust database. The following is an example of logging in to a new board.

```
# vcaadm -h hostname
Warning: MAC ID and Public Key Not Found
-----
The MAC ID and public key presented by this board were
not found in your trust database.

MAC ID: 08:00:20:EE:EE:EE
Key Fingerprint: 29FC-7A54-4014-442F-7FD9-5FEA-8411-CFB4
-----
Please select an action:

1. Abort this connection
2. Trust the board for this session only
3. Trust the board for all future sessions

Your Choice -->
```

Logging In to a Board With a Changed Remote Access Key

When connecting to a board that has a changed remote access key, `vcaadm` must change the entry corresponding to the board in the trust database. The following is an example of logging in to a board with a changed remote access key.

```
# vcaadm -h hostname
Warning: Public Key Conflict
-----
The public key presented by the board you are connecting
to is different than the public key that is trusted for
this MAC ID.

MAC ID: 08:00:20:EE:EE:EE
New Key Fingerprint: 29FC-7A54-4014-442F-7FD9-5FEA-8411-CFB4
Trusted Key Fingerprint: A508-38D1-FED8-8103-7ACC-0D19-C9C9-11F2
-----
Please select an action:

1. Abort this connection
2. Trust the board for this session only
3. Replace the current trusted key with the new key

Your Choice -->
```

vcaadm *Prompt*

The vcaadm prompt in Interactive mode is displayed as follows:

```
vcaadm{vcaN@hostname, sec-officer}> command
```

The following table describes the vcaadm prompt variables:

TABLE 4-2 vcaadm Prompt Variable Definitions

Prompt Variable	Definition
<i>vcaN</i>	<i>vca</i> is a string that represents the Sun Crypto Accelerator 4000 board. <i>N</i> is the device instance number (unit address) that is in the device path name of the board. Refer to “To Set Driver Parameters Using a vca.conf File” on page 39 for details on retrieving this number for a device.
<i>hostname</i>	The name of the host for which the Sun Crypto Accelerator 4000 board is physically connected. <i>hostname</i> may be replaced with the physical host’s IP address.
<i>sec-officer</i>	The name of the security officer that is currently logged in to the board.

Logging Out of a Board With vcaadm

If you are working in Interactive mode, you might want to disconnect from one board and connect to another board without completely exiting vcaadm. To disconnect from a board and log out, but remain in Interactive mode, use the `logout` command:

```
vcaadm{vcaN@hostname, sec-officer}> logout  
vcaadm>
```

In the previous example, notice that the `vcaadm>` prompt no longer displays the device instance number, hostname, or security officer name. To log in to another device, type the `connect` command with the following optional parameters.

TABLE 4-3 `connect` Command Optional Parameters

Parameter	Meaning
<code>dev vcaN</code>	Connect to the Sun Crypto Accelerator 4000 board with the driver instance number of <i>N</i> . For example <code>-d vca1</code> connects to the device <code>vca1</code> ; this defaults to device <code>vca0</code> .
<code>host hostname</code>	Connect to the Sun Crypto Accelerator 4000 board on <i>hostname</i> (defaults to the loopback address). <i>hostname</i> may be replaced with the physical host's IP address.
<code>port port</code>	Connect to the Sun Crypto Accelerator 4000 board on port <i>port</i> (defaults to 6870).

Example:

```
vcaadm{vcaN@hostname, sec-officer}> logout
vcaadm> connect host hostname dev vca2
Security Officer Login: sec-officer
Security Officer Password:
vcaadm{vcaN@hostname, sec-officer}>
```

`vcaadm` does not let you issue the `connect` command if you are already connected to a Sun Crypto Accelerator 4000 board. You must first log out and then issue the `connect` command.

Each new connection causes `vcaadm` and the target Sun Crypto Accelerator 4000 firmware to renegotiate new session keys to protect the administrative data that is sent.

Entering Commands With `vcaadm`

The `vcaadm` utility has a command language that must be used to interact with the Sun Crypto Accelerator 4000 board. Commands are entered using all or part of a command (enough to uniquely identify that command from any other command). Entering `sh` instead of `show` would work, but `re` is ambiguous because it could be `reset` or `rekey`.

The following example shows entering commands using entire words:

```
vcaadm{vcaN@hostname, sec-officer}> show user
User                                     Status
-----
web-admin                               enabled
Tom                                     enabled
-----
```

The same information can be obtained in the previous example using partial words as commands, such as `sh us`.

An ambiguous command produces an explanatory response:

```
vcaadm{vcaN@hostname, sec-officer}> re
Ambiguous command: re
```

Getting Help for Commands

`vcaadm` has built-in help functions. To get help, you must enter a question mark (?) character following the command you want more help on. If an entire command is entered and a “?” exists anywhere on the line, you get the syntax for the command, for example:

```
vcaadm{vcaN@hostname, sec-officer}> create ?
Sub-Command          Description
-----
so                   Create a new security officer
user                 Create a new user

vcaadm{vcaN@hostname, sec-officer}> create user ?
Usage: create user [<username>]

vcaadm{vcaN@hostname, sec-officer}> set ?
Sub-Command          Description
-----
passreq              Set password requirements
password             Change an existing security officer password
timeout              Set the auto-logout time
```

You can also enter a question mark at the `vcaadm` prompt to see a list of all of the `vcaadm` commands and their description, for example:

```
vcaadm{vcaN@hostname, sec-officer}> ?
```

Sub-Command	Description
-----	-----
backup	Backup master key
connect	Begin admin session with firmware
create	Create users and accounts
delete	Delete users and accounts
diagnostics	Run diagnostic tests
disable	Disable a user
enable	Enable a user
exit	Exit vcaadm
loadfw	Load new firmware
logout	Logout current session
quit	Exit vcaadm
rekey	Generate new system keys
reset	Reset the hardware
set	Set operating parameters
show	Show system settings
zeroize	Delete all keys and reset board

When not in `vcaadm` Interactive mode, the “?” character could be interpreted by the shell in which you are working. In this case, be sure to use the command shell escape character before the question mark.

Quitting the `vcaadm` Utility in Interactive Mode

Two commands allow you to exit from `vcaadm`: `quit` and `exit`. The Ctrl-D key sequence also exits from `vcaadm`.

Initializing the Board With `vcaadm`

The first step in configuring a Sun Crypto Accelerator 4000 board is to initialize it. When you initialize a board it is necessary to create a keystore. (See “Concepts and Terminology” on page 106.) When you first connect to a Sun Crypto Accelerator 4000 board with `vcaadm`, you are prompted to initialize the board with a new keystore, or to initialize the board to use an existing keystore, which is stored in a backup file. `vcaadm` prompts you for all the required information for either type of board initialization.

▼ To Initialize the Board With a New Keystore

1. Enter `vcaadm` at a command prompt of the system with the board installed or enter `vcaadm -h hostname` if the system is remote, and select 1 to initialize the board:

```
# vcaadm -h hostname
Please select an action:

1. Abort this connection
2. Trust the board for this session only.
3. Replace the trusted key with the new key.

Your Choice --> 2
This board is uninitialized.
You will now initialize the board. You may either
completely initialize the board and start with a new
keystore or initialize the board to use an existing
keystore, providing a backup file in the process.

1. Initialize the board with a new keystore
2. Initialize the board to use an existing keystore

Your Choice (0 to exit) --> 1
```

2. Create a keystore name (See “Naming Requirements” on page 72.):

```
Keystore Name: keystore-name
```

3. Select FIPS 140-2 mode or non-FIPS mode.

When in FIPS mode the board is FIPS 140-2, level 3 compliant. FIPS 140-2 is a Federal Information Processing standard that requires tamper-resistance and a high level of data integrity and security. Refer to the FIPS 140-2 document located at: <http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf>

```
Run in FIPS 140-2 mode? (Y/Yes/N/No) [No]: y
```

4. Create an initial security officer name and password (See “Naming Requirements” on page 72.):

```
Initial Security Officer Name: sec-officer
Initial Security Officer Password:
Confirm Password:
```

Note – Before an essential parameter is changed or deleted, or before a command is executed that may have drastic consequences, `vcaadm` prompts you to enter Y, Yes, N, or No to confirm. These values are not case sensitive; the default is No.

5. Verify the configuration information:

```
Board initialization parameters:
-----
Initial Security Officer Name: sec-officer
Keystore name: keystore-name
Run in FIPS 140-2 Mode: Yes
-----

Is this correct? (Y/Yes/N/No) [No]: y
Initializing crypto accelerator board... This may take a few
minutes...Done.
```

Initializing the Board to Use an Existing Keystore

If you are adding multiple boards to a single keystore, you might want to initialize all of the boards to use the same keystore information. In addition, you might want to restore a Sun Crypto Accelerator 4000 board to the original keystore configuration. This section describes how to initialize a board to use an existing keystore which is stored in a backup file.

You must first create a backup file of an existing board configuration before performing this procedure. Creating and restoring a backup file requires a password to encrypt and decrypt the data in the backup file. (See “Backing Up the Master Key” on page 77.)

▼ To Initialize the Board to Use an Existing Keystore

1. Enter `vcaadm` at a command prompt of the system with the Sun Crypto Accelerator 4000 board installed or enter `vcaadm -h hostname` if the system is remote, and select 2 to restore the board from a backup:

```
# vcaadm -h hostname
This board is uninitialized.
You will now initialize the board.  You may
either completely initialize the board and
start with a new keystore or restore the board
using a backup file.

1. Initialize the board with a new keystore
2. Initialize the board to use an existing keystore

Your Choice (0 to exit) --> 2
```

2. Enter the path and password to the backup file:

```
Enter the path to the backup file: /tmp/board-backup
Password for restore file:
```

3. Verify the configuration information:

```
Board restore parameters:
-----
Path to backup file: /tmp/board-backup
Keystore name: keystore-name
-----

Is this correct? (Y/Yes/N/No) [No]: y
Restoring data to crypto accelerator board...
```

Managing Keystores With `vcaadm`

A keystore is a repository for key material. Associated with a keystore are security officers and users. Keystores not only provide storage, but a means for key objects to be owned by user accounts. This enables keys to be hidden from applications that do not authenticate as the owner. Keystores have three components:

- **Key objects** – Long-term keys that are stored for applications such as the Sun ONE Web Server.
- **User accounts** – These accounts provide applications a means to authenticate and access specific keys.
- **Security officer accounts** – These accounts provide access to key management functions through `vcaadm`.

Note – A single Sun Crypto Accelerator 4000 board must have exactly one keystore. Multiple boards can be configured to collectively work with the same keystore to provide additional performance and fault-tolerance.

Naming Requirements

Security officer names, user names, and keystore names must meet the following requirements:

TABLE 4-4 Security Officer Name, User Name, and Keystore Name Requirements

Name Requirement	Description
Minimum length	At least one character
Maximum length	63 characters for user names and 32 characters for keystore names
Valid characters	Alphanumeric, underscore (<code>_</code>), dash (<code>-</code>), and dot (<code>.</code>)
First character	Must be alphabetic

Password Requirements

Password requirements vary based on the current `set passreq` setting (low, med, or high).

Setting the Password Requirements

Use the `set passreq` command to set the password requirements for the Sun Crypto Accelerator 4000 board. This command sets the password character requirements for any password prompted by `vcaadm`. There are three settings for password requirements, as shown in the following table:

TABLE 4-5 Password Requirement Settings

Password Setting	Requirements
low	Does not require any password restrictions. This is the default while the board is in non-FIPS mode.
med	Requires six characters minimum: Three characters must be alphabetic and one character must be nonalphabetic. This is the default setting while the board is in FIPS 140-2 mode and is the minimum password requirement allowed in FIPS 140-2 mode.
high	Requires eight characters minimum: Three characters must be alphabetic, and one character must be nonalphabetic. This is not a default setting and must be configured manually.

To change the password requirements, enter the `set passreq` command followed by `low`, `med`, or `high`. The following commands set the password requirements for a Sun Crypto Accelerator 4000 board to `high`:

```
vcaadm{vcaN@hostname, sec-officer}> set passreq high  
  
vcaadm{vcaN@hostname, sec-officer}> set passreq  
Password security level (low/med/high): high
```

Populating a Keystore With Security Officers

There may be more than one security officer for a keystore. Security officer names are known only within the domain of the Sun Crypto Accelerator 4000 board and do not need to be identical to any user name on the host system.

When creating a security officer, the name is an optional parameter on the command line. If the security officer name is omitted, `vcaadm` prompts you for the name. (See “Naming Requirements” on page 72.)

```
vcaadm{vcaN@hostname, sec-officer}> create so Alice
Enter new security officer password:
Confirm password:
Security Officer Alice created successfully.

vcaadm{vcaN@hostname, sec-officer}> create so
New security officer name: Bob
Enter new security officer password:
Confirm password:
Security Officer Bob created successfully.
```

Populating a Keystore With Users

These user names are known only within the domain of the Sun Crypto Accelerator 4000 board and do not need to be identical to the UNIX user name for the web server process.

When creating a user, the user name is an optional parameter on the command line. If the user name is omitted, `vcaadm` prompts you for the user name. (See “Naming Requirements” on page 72.)

```
vcaadm{vcaN@hostname, sec-officer}> create user web-admin
Enter new user password:
Confirm password:
User web-admin created successfully.

vcaadm{vcaN@hostname, sec-officer}> create user
New user name: Tom
Enter new user password:
Confirm password:
User Tom created successfully.
```

Users must use this password when authenticating during a web server startup.



Caution – Users must remember their password so they can access their keys. There is no way to retrieve a lost password.

Note – The user account is logged out if no commands are entered for more than five minutes. This is a tunable option. See “Setting the Auto-Logout Time” on page 79 for details.

Listing Users and Security Officers

To list users or security officers associated with a keystore, enter the `show user` or `show so` commands.

```
vcaadm{vcaN@hostname, sec-officer}> show user
User                                     Status
-----
web-admin                               Enabled
Tom                                      Enabled
-----

vcaadm{vcaN@hostname, sec-officer}> show so
Security Officer
-----
sec-officer
Alice
Bob
-----
```

Changing Passwords

Only security officer passwords may be changed with `vcaadm`. Security officers can change their own password. Use the `set password` command to change security officer passwords.

```
vcaadm{vcaN@hostname, sec-officer}> set password
Enter new security officer password:
Confirm password:
Security Officer password has been set.
```

User passwords may be changed through the PKCS#11 interface with the Sun ONE Web Server `modutil` utility. Refer to the Sun ONE Web Server documentation for details.

Enabling or Disabling Users

Note – Security officers cannot be disabled. Once a security officer is created, it is enabled until it is deleted.

By default each user is created in the enabled state. Users may be disabled. Disabled users cannot access their key material with the PKCS#11 interface. Enabling a disabled user restores access to all of that user's key material.

When enabling or disabling a user, the user name is an optional parameter on the command line. If the user name is omitted, `vcaadm` prompts you for the user name.

```
vcaadm{vcaN@hostname, sec-officer}> disable user Tom
User Tom disabled.
vcaadm{vcaN@hostname, sec-officer}> disable user
User name: web-admin
User web-admin disabled.
```

To disable a user account, enter the `disable user` command.

To enable an account, enter the `enable user` command.

```
vcaadm{vcaN@hostname, sec-officer}> enable user Tom
User Tom enabled.

vcaadm{vcaN@hostname, sec-officer}> enable user
User name: web-admin
User web-admin enabled.
```

Deleting Users

Issue the `delete user` command and specify the user to be deleted. When deleting a user, the user name is an optional parameter on the command line. If the user name is omitted, `vcaadm` prompts you for the user name.

```
vcaadm{vcaN@hostname, sec-officer}> delete user web-admin
Delete user web-admin? (Y/Yes/N/No) [No]: y
User web-admin deleted successfully.

vcaadm{vcaN@hostname, sec-officer}> delete user
User name: Tom
Delete user Tom? (Y/Yes/N/No) [No]: y
User Tom deleted successfully.
```

Deleting Security Officers

Issue the `delete so` command and specify the security officer to be deleted. When deleting a security officer, the security officer name is an optional parameter on the command line. If the security officer name is omitted, `vcaadm` prompts you for the security officer name.

```
vcaadm{vcaN@hostname, sec-officer}> delete so Bob
Delete Security Officer Bob? (Y/Yes/N/No) [No]: y
Security Officer Bob deleted.

vcaadm{vcaN@hostname, sec-officer}> delete so
Security Officer name: Alice
Delete Security Officer Alice? (Y/Yes/N/No) [No]: y
Security Officer Alice deleted.
```

Backing Up the Master Key

Keystores are stored on the disk and encrypted in a master key. This master key is stored in the Sun Crypto Accelerator 4000 firmware and can be backed up by a security officer.

To back up the master key, use the `backup` command. The `backup` command requires a path name to a backup file where the backup will be stored. This path name can be placed on the command line or if omitted, `vcaadm` prompts you for the path name.

A password must be set for the backup data. This password is used to encrypt the master key that is in the backup file.

```
vcaadm{vcaN@hostname, sec-officer}> backup /opt/SUNWconn/vca/backups/bkup.data
Enter a password to protect the data:
Confirm password:
Backup to /opt/SUNWconn/vca/backups/bkup.data successful.
```



Caution – Choose a password that is very difficult to guess when making backup files, because this password protects the master key for your keystore. You must also remember the password you enter. Without the password, you cannot access the master key backup file. There is no way to retrieve the data protected by a lost password.

Locking the Keystore to Prevent Backups

A site might have a strict security policy that does not permit the master key for a Sun Crypto Accelerator 4000 board to leave the hardware. This can be enforced using the `set lock` command.



Caution – Once this command is issued, all attempts to back up the master key will fail. This lock persists even if the master key is rekeyed. The only way to clear this setting is to zeroize the Sun Crypto Accelerator 4000 board with the `zeroize` command. (See “Performing a Software Zeroize on the Board” on page 82.)

```
vcaadm{vcaN@hostname, sec-officer}> set lock
WARNING: Issuing this command will lock the
         master key.  You will be unable to back
         up your master key once this command
         is issued.  Once set, the only way to
         remove this lock is to zeroize the board.
Do you wish to lock the master key? (Y/Yes/N/No) [No]: y
The master key is now locked.
```

Managing Boards With `vcaadm`

This section describes how to manage Sun Crypto Accelerator 4000 boards with the `vcaadm` utility.

Setting the Auto-Logout Time

To customize the amount of time before a security officer is automatically logged out of the board, use the `set timeout` command. To change the auto-logout time, enter the `set timeout` command followed by the number of minutes before a security officer is automatically logged out. A value of 0 disables the automatic logout feature. The maximum delay is 1,440 minutes (1 day). A newly initialized board defaults to 5 minutes.

The following command changes the auto-logout time for a security officer to 10 minutes:

```
vcaadm{vcaN@hostname, sec-officer}> set timeout 10
```

Displaying Board Status

To get the current status of a Sun Crypto Accelerator 4000 board, issue the `show status` command. This command displays the hardware and firmware versions for that board, the MAC address of the network interface, the status (Up versus Down, speed, duplex, and so on) of the network interface, and the keystore name and ID.

```
vcaadm{vcaN@hostname, sec-officer}> show status
Board Status
-----
Hardware Version: 1.0
Firmware Version: 1.0
Bootstrap Firmware Version: VCA Crypto Accelerator 1.0 March 2003
Current Firmware Version: VCA Crypto Accelerator 1.0 March 2003
MAC Address: 00:03:ba:0e:96:aa
Interface information: Link up, 1000Mbps, Full Duplex
Keystore Name: keystore-name
Keystore ID: 832aece03e654790
Login Session Timeout (in minutes): 10
Password policy security level: HIGH
Number of master key backups: 0
* Device is in FIPS 140-2 Mode
-----
```

Determining if the Board is Operating in FIPS 140-2 Mode

If the Sun Crypto Accelerator 4000 board is operating in FIPS 140-2 mode, the `show status` command prints the following line:

```
* Device is in FIPS 140-2 Mode
```

If the board is not operating in FIPS 140-2 mode, the `show status` command does not print a line specifying FIPS 140-2 mode.

You can also use the `kstat(1M)` utility to determine if the board is operating in FIPS 140-2 mode. The `kstat(1M)` parameter, `vs-mode`, returns a value of `FIPS` if the board is operating in FIPS 140-2 mode. See “Cryptographic and Ethernet Driver Operating Statistics” on page 44 and the online manual page for `kstat(1M)`.

Loading New Firmware

You can update the firmware for the Sun Crypto Accelerator 4000 board as new features are added. To load firmware, issue the `loadfw` command and provide a path to the firmware file.

A successful update of the firmware requires you to manually reset the board with the `reset` command. When you reset the board, the currently logged in security officer is logged out.

```
vcaadm{vcaN@hostname, sec-officer}> loadfw /opt/SUNWconn/cryptov2/firmware/sca4000fw
Security Officer Login: sec-officer
Security Officer Password:
WARNING: This command will load new firmware onto the
         the target device. You must issue a reset
         command and log back into the target device in
         order to use the new firmware.

Proceed with firmware update? (Y/Yes/N/No) [No]: y
```

Resetting the Board

In certain situations, it might be necessary to reset the board. To do this, you must issue the `reset` command. You are asked if this is what you wish to do. Resetting a Sun Crypto Accelerator 4000 board might temporarily cease the acceleration of cryptography on the system unless there are other active Sun Crypto Accelerator

4000 boards able to take over the load. Also, this command automatically logs you out of `vcaadm`, so you must reconnect to the device by logging back into `vcaadm` if you wish to continue administering it.

```
vcaadm{vcaN@hostname, sec-officer}> reset
WARNING: Issuing this command will reset the
         the board and close this connection.

Proceed with reset? (Y/Yes/N/No) [No]: y
Reset successful.
```

Rekeying the Board

If your security policy changes, you might want to use new keys as the master key or remote access key. The `rekey` command enables you to regenerate either of these keys, or both.

Rekeying the master key also causes the keystore to be reencrypted under the new key, and invalidates older backed up master key files with the new keystore file. Make a backup of the master key whenever it is rekeyed. If you have multiple Sun Crypto Accelerator 4000 boards using the same keystore, you need to backup this new master key and restore it to the other boards.

Rekeying the remote access key logs the security officer out, forcing a new connection that uses the new remote access key.

You may specify one of three key types when issuing the `rekey` command:

TABLE 4-6 Key Types

Key Type	Action
master	Rekey the master key.
remote	Rekey the remote access key. Logs the security officer out.
all	Rekeys both master and remote access keys.

The following is an example of entering a key type of all with the rekey command:

```
vcaadm{vcaN@hostname, sec-officer}> rekey

Key type (master/remote/all): all
WARNING: Rekeying the master key will render all old board backups
         useless with the new keystore file. If other boards use this
         keystore, they will need to have this new key backed up and
         restored to those boards. Rekeying the remote access key will
         terminate this session and force you to log in again.

Rekey board? (Y/Yes/N/No) [No]: y
Rekey of master key successful.
Rekey of remote access key successful. Logging out.
```

Performing a Software Zeroize on the Board

There are two methods of clearing a board of all its key material. The first method is with a hardware jumper (shunt); this form of zeroizing returns the board to its original factory state (Failsafe mode). (See “Zeroizing the Sun Crypto Accelerator 4000 Hardware to the Factory State” on page 253.) The second method is to use the zeroize command.

Note – The zeroize command removes the key material, and leaves any updated firmware intact. This command also logs the security officer out upon successful completion.

To perform a software zeroize on a board with the zeroize command, enter the command and confirm it:

```
vcaadm{vcaN@hostname, sec-officer}> zeroize
WARNING: Issuing this command will zeroize all keys
         on the board. Once zeroized, these keys
         cannot be recovered unless you have
         previously backed up your master key.

Proceed with zeroize? (Y/Yes/N/No) [No]: y
All keys zeroized successfully.
```

Using the `vcaadm diagnostics` Command

Diagnostics can be performed from the `vcaadm` utility and from the SunVTS software. The `diagnostics` command in `vcaadm` covers three major categories in the Sun Crypto Accelerator 4000 hardware: general hardware, cryptographic subsystem, and network subsystem. Tests for general hardware cover DRAM, flash memory, the PCI bus, the DMA controller, and other hardware internals. Tests for the cryptographic subsystem cover random number generators and cryptographic accelerators. Tests on the network subsystem cover the `vca` device.

```
vcaadm{vcaN@hostname, sec-officer}> diagnostics
Performing diagnostic tests...Done.
Diagnostic Results
-----
General Hardware:                PASS
Cryptographic Subsystem:         PASS
Network Subsystem:               PASS
-----
```

Using the `vcad` Command

The `vcad` command configures and starts the `vcad` daemon, which provides cryptographic keystore services for `vcaadm(1M)` and other cryptographic applications. The `vcad` daemon also handles reading and writing of keystore data for the driver and hardware.

To access the `vcad` command easily, place the Sun Crypto Accelerator 4000 tools directory in your search path, for example:

```
$ PATH=$PATH:/opt/SUNWconn/cryptov2/sbin/
$ export PATH
```

The command-line syntax for the `vcad` command is:

```
/opt/SUNWconn/cryptov2/sbin/vcad [-dF1V] [-f config-file]
[-h host-address] [-k keystore-dir] [-L logfile] [-p port] [-s max-size]
[-t seconds][-u username]
```

TABLE 4-7 describes the supported options for the `vcad` command.

TABLE 4-7 `vcad` Command Options

Option	Description
<code>-d</code>	Turns on debugging. Each message contains the process ID for <code>vcad</code> , current thread ID, and message category in addition to the actual message itself. Multiple <code>-d</code> options increase the verbosity (maximum 2). When using multiple <code>-d</code> options, one <code>-d</code> is equivalent to setting the <code>DebugLevel</code> parameter in the configuration file to <code>INFO</code> , <code>-dd</code> is equivalent to setting it to <code>DEBUG</code> .
<code>-f config-file</code>	Specifies the location of the configuration file. The default location for this configuration file is <code>/etc/opt/SUNWconn/vca/vcad.conf</code> . If this option is used and the file cannot be opened, <code>vcad</code> does not start.
<code>-F</code>	Performs <code>vcad</code> in the foreground and sends log output to <code>stderr</code> . This behavior overrides a <i>logfile</i> chosen with the <code>-L</code> flag.
<code>-h host-address</code>	Specifies the host IPv4 or IPv6 address for <code>vcad</code> to bind and listens for incoming connections. More than one host or IP address can be specified with additional <code>-h</code> options. If this option is not used the default behavior for <code>vcad</code> is to listen on all available interfaces for incoming connections. When specific hosts or IP addresses are specified for binding, connections can only be established on interfaces answering those addresses and <code>localhost</code> . Any addresses or hosts specified with the <code>-h</code> flag are overridden by the <code>-l</code> option.
<code>-k keystore-dir</code>	Uses <i>keystore-dir</i> as the directory for all keystore data. If the daemon runs as a nonsuperuser, this directory must be readable and writable by that user, as should the keystore data files themselves. The default directory for keystore data is <code>/etc/opt/SUNWconn/vca/keydata</code> .
<code>-l</code>	Accepts only incoming connections from administrative clients that originate on the local host. This option overrides any command-line or <code>.conf</code> file directive that would have the daemon listen on any other interface.
<code>-L logfile</code>	Sends logging output to <i>logfile</i> instead of the standard location for system logs.
<code>-p port</code>	Binds using <i>port</i> for incoming connections. The default port used for 6870.
<code>-s max-size</code>	Enables commands with data of length up to <i>max-size</i> bytes to be passed down to the Sun Crypto Accelerator. Administrators can use this facility to prevent large volumes of data from being sent down through the kernel in single commands. The default maximum size for a single command is 4 MegaBytes (4194304 bytes).

TABLE 4-7 vcad Command Options (*Continued*)

Option	Description
-t <i>seconds</i>	Sets <i>seconds</i> as the number of <i>seconds</i> before vcad stops waiting for data from the client. If this timer expires, the connection between vcad and the client is closed.
-u <i>username</i>	Performs vcad as <i>username</i> . If no username is specified, vcad attempts to run as the user who started vcad. If a username is specified and that username cannot be found on the system, vcad fails to start. If vcad runs as superuser or any other account with a user ID of 0, vcad issues a warning. See “vcad Daemon Security” on page 87 for recommendations on running vcad as a nonsuperuser.
-V	Displays the version information for vcad.

vcad Configuration File

The vcad daemon obtains operating parameters from a configuration file. By default the daemon looks for this configuration file in `/etc/opt/SUNWconn/vca/vcad.conf`, though other files may be specified with the `-f` flag of the vcad command when invoking the vcad daemon. If the `-f` flag is not used and the default configuration file cannot be found or read, the vcad daemon attempts to start using all default values. In this case a warning message is sent to the standard error output.

The configuration file contains one directive per line. Each directive must have a value associated with it. Comments may be used and must start with the pound sign (#). Directive names are case-insensitive, but their values might be case-sensitive. See the descriptions of each directive in TABLE 4-8 for more detail.

Configuration file directives may be superseded by the use of a command-line option for the same operating parameter. For example, you can supersede the “Port” configuration file directive with the `-p` option. Operating parameters that are not

specified with a command-line option or a configuration file directive use a built-in default value. TABLE 4-8 describes the supported command-line directives for the `vcad` command.

TABLE 4-8 Command-Line Directives For the `vcad` Command

Directives	Description
DebugLevel <i>level</i>	Enables the user to set the one of three debug levels in the configuration file. These three levels, from least verbose to most, are Notice, Info, and Debug. Notice level is the default.
HostBind <i>host/IP</i>	Tells <code>vcad</code> to bind and listen on the specified IPv4 or IPv6 address, or the IP address that <code>host</code> resolves to. Multiple <code>HostBind</code> directives enable <code>vcad</code> to listen on more than one address. If no <code>HostBind</code> entries are in a configuration file, the default behavior is to listen on all interfaces for connections. Note that the <code>-l</code> command-line flag supersedes all <code>HostBind</code> entries.
KeyStoreDir <i>directory</i>	Enables the administrator to select an alternate directory for the storage of keystore files. This directory must have read and write permission for the user for which <code>vcad</code> runs (See the <code>User</code> directive). The default location for the keystore directory is <code>/etc/opt/SUNWconn/vca/keydata</code> .
LogFile <i>logfile</i>	Uses <i>logfile</i> as the location where all logging data is to be written. By default, logging data is written to <code>syslog</code> . If the <code>-F</code> (run in foreground) command-line flag is used, this directive is ignored and <code>vcad</code> logging data is sent to the standard error device.
MaxData <i>size</i>	Sets the maximum allowable data to be sent in a single command to be <i>size</i> bytes. By default this value is 4 MegaBytes (4194304 bytes). If the data sent exceeds this value, <code>vcad</code> returns an error to the client and closes the connection.

TABLE 4-8 Command-Line Directives For the `vcad` Command (Continued)

Directives	Description
Port <i>port</i>	Sets the listen port. The default port <code>vcad</code> listens on is 6870. If an administrator needs to have <code>vcad</code> listen on a privileged port (usually a port under 1024), <code>vcad</code> must run as a user with superuser privileges. See “ <code>vcad</code> Daemon Security” on page 87 for security relevant notes.
Timeout <i>seconds</i>	Enables the administrator to set a timeout value for command data once the first byte of that data has been received. This timeout value prevents stalled reads from locking access to specific cards. This timeout does not apply to <code>vcad</code> when it is waiting for a connected client to send a new command. Firmware timeout values cover this issue. (See “Setting the Auto-Logout Time” on page 79.) The default timeout is 300 seconds (five minutes).
User <i>username</i>	Sets <code>vcad</code> to run as <code>username</code> . The daemon attempts to set its real user ID to the UID associated with <code>username</code> . The default value for this directive is the user who started the <code>vcad</code> process.

`vcad` Daemon Security

Because the `vcad` daemon listens on a TCP port, certain security recommendations that should be considered.

When running `vcad`, the process should be run as a user ID that does not have superuser privileges, that is, not a `UID0` account. You must not be able to directly log in to this user account from the network. This account should have either no password or a locked password and no login shell. The entry in the `/etc/shadow` file for this account would have `NP` or `*LK*`.

By default, the `vcad` daemon will attempt to start as the daemon user account. The `vcad` daemon will start correctly even if this account is disabled, but the account should be present on the system. Perform the following steps to manually configure `vcad` to run as a different username.

▼ To Configure the `vcad` Daemon to Run as a Different Username

1. Configure read/write access to `/dev/vcactl`.

The `vcad` daemon communicates directly with `/dev/vcactl` to relay command data and get keystore I/O commands from the Sun Crypto Accelerator 4000 firmware. Permissions and ownership should be set such that only the user account in which `vcad` runs can read and write to `/dev/vcactl`. By default, the `vcactl`

module is added such that minor nodes are owned by the daemon with owner read and write permissions only. The safest way to change these permissions is to use `rem_drv(1m)` and `add_drv(1m)` to reregister the `vcactl` module:

```
rem_drvvcactl
add_drv-m '* MODE USERGROUP' vcactl
```

The `USER` and `GROUP` place holders should contain the user and group ownership desired for the device minor node. `MODE` is the file mode for the device minor node. `0600` is the recommended mode for the `vcactl` module. See the `add_drv(1m)` man pages for more details.

2. Configure Read/Write access to keystores.

For the `vcad` daemon to perform keystore I/O operations, it must be able to access the keystore directory specified in its configuration. The keystore directory must have read, write, and execute permissions available only to the user account in which `vcad` is running. Keystore files in this directory should only allow read and write permissions for that user.

3. Run the `vcad` daemon on a nonprivileged TCP port.

If the `vcad` daemon is running without superuser privileges, it cannot bind to a privileged port. Typically nonprivileged ports are 1024 and higher. Use `ndd` to determine the value of the `tcp_smallest_nonpriv_port` parameter if this value is not 1024 on a given system. By default, the `vcad` daemon uses port 6870.

Examples

Example 1: Start the `vcad` daemon to listen on port 5525.

```
# /opt/SUNWconn/cryptov2/sbin/vcad -p 5525
```

Example 2: Start the `vcad` daemon with extra debugging information and send the information to the screen.

```
# /opt/SUNWconn/cryptov2/sbin/vcad -Fdd
```

This starting method yields the following output on startup:

```
vcad[1679/1]: [debug] got exclusive lock
vcad[1679/1]: [info] Security daemon starting up
vcad[1679/1]: [debug] Starting file handling thread
vcad[1679/1]: [debug] Starting TCPserver
vcad[1679/1]: [debug] TCP socket bound on port 6870
vcad[1679/1]: [debug] fd is 6
```

The `vcad` daemon also provides notices when new connections are opened and closed when running with two levels of debug output.

Example 3: Start the `vcad` daemon and use an alternate configuration file.

```
# /opt/SUNWconn/criptov2/sbin/vcad -f /etc/opt/SUNWconn/vca/alt-vcad.conf
```

Using the `vcadiag` Utility

The `vcadiag` utility provides a command-line interface to the Sun Crypto Accelerator 4000 board that enables superusers to perform administrative tasks without authenticating as security officer. Command-line options determine the actions that `vcadiag` performs.

To access the `vcadiag` utility easily, place the Sun Crypto Accelerator 4000 tools directory in your search path, for example:

```
$ PATH=$PATH:/opt/SUNWconn/bin
$ export PATH
```

The `vcadiag` command-line syntax is:

- `vcadiag [-D] vcaN`
- `vcadiag [-F] vcaN`
- `vcadiag [-K] vcaN`
- `vcadiag [-Q]`
- `vcadiag [-R] vcaN`
- `vcadiag [-Z] vcaN`

Note – When using the [-DFKRZ] options, `vcaN` is the board's device name where the `N` corresponds to the Sun Crypto Accelerator 4000 device instance number.

TABLE 4-9 describes the supported options for the `vcadiag` utility.

TABLE 4-9 `vcadiag` Options

Option	Meaning
-D <code>vcaN</code>	Performs diagnostics on the Sun Crypto Accelerator 4000 board.
-F <code>vcaN</code>	Displays the public key fingerprint used by the Sun Crypto Accelerator 4000 board for securing administration sessions.
-K <code>vcaN</code>	Displays the public key and the public key fingerprint used by the Sun Crypto Accelerator 4000 board for securing administration sessions.
-Q	Provides information about Sun Crypto Accelerator 4000 devices and software components. Output is a colon-separated list of the following information: <ul style="list-style-type: none">• Device• Internal function• Keystore name• Keystore serial number• Keystore reference count. You can use this option to determine the association between devices and keystores.
-R <code>vcaN</code>	Resets the board.
-Z <code>vcaN</code>	Zeroizes the board.

The following is an example of the `-D` option:

```
# vcadiag -D vca0
Running vca0 on-board diagnostics.
Diagnostics on vca0 PASSED.
```

The following is an example of the `-F` option:

```
# vcadiag -F vca0
5f26-b516-83b4-d254-a75f-c70d-0544-4de6
```

The following is an example of the `-K` option:

```
# vcdiag -K vca0
Device: vca0
Key Length: 1024 bits
Key Fingerprint: 5f26-b516-83b4-d254-a75f-c70d-0544-4de6
Modulus:
    b7215a99 8bb0dfe9 389363a0 44dac2b0 7c884161
    20ee8c8b d751437d 4e6a5cdb 76fdc2ba ad353c0b
    248edc1d 3c76591d dbca5997 f6ee8022 e8bb5a6d
    465a4f8c 601d46be 573e8681 506e5d8d f240a0db
    11d5c095 2d237061 df27b2de c353900f f531092b
    7d9a755b c5d79782 95a1180b e17303bb aca939ef
    006c73f7 74469031
Public Exponent:
    00010001
```

The following is an example of the `-Q` option:

```
# vcdiag -Q
vca0:cb
vca0:cb:keystore-name:83097c2b3e35ef5b:1
vca0:ca
vca0:ca:keystore-name:83097c2b3e35ef5b:1
kcl2pseudo
vca0:om
vca0:om:keystore-name:83097c2b3e35ef5b:1
libkcl
```

The following is an example of the `-R` option:

```
# vcdiag -R vca0
Resetting device vca0, this may take a minute.
Please be patient.
Device vca0 reset ok.
```

The following is an example of the `-Z` option:

```
# vcdiag -Z vca0
Zeroizing device vca0, this may take a few minutes.
Please be patient.
Device vca0 zeroized.
```

Using the `pk11export` Utility

The `pk11export` utility extracts keys and certificates from key databases and places them into the PKCS#12 importable format. This utility requires a PKCS#11 interface to extract the objects, and place the keys and certificates into a PKCS#12 file. Only one key and certificate pair may be extracted at a time.

This utility works with different PKCS#11 providers, if the interface is contained within a dynamic library. The `pk11export` utility exports keys through a PKCS#11 provider while the following requirements are met:

- The PKCS#11 interface must implement the `C_WrapKey` PKCS#11 function.
- The PKCS#11 interface must implement the `CKM_DES3_CBC_PAD` and `CKM_SHA_1` PKCS#11 mechanisms.
- The key to be exported must have the `CKA_EXTRACTABLE` attribute set.

The command-line syntax for `pk11export` is as follows:

- `/opt/SUNWconn/cryptov2/bin/pk11export -v`
- `/opt/SUNWconn/cryptov2/bin/pk11export -l [-p pkcs11-lib]`
- `/opt/SUNWconn/cryptov2/bin/pk11export [-n friendly-name] [-o filename] [-p pkcs11-lib] token-name`

TABLE 4-10 describes the supported options for the `pk11export` utility.

TABLE 4-10 `pk11export` Options

Option	Description
<code>-l</code>	Lists all available tokens recognized by a given PKCS#11 library.
<code>-n <i>friendly-name</i></code>	Specifies which key and certificate pair is to be exported. The <i>friendly-name</i> is a string value.
<code>-o <i>filename</i></code>	Place the resulting PKCS#12 file in the <i>filename</i> file. If no output <i>filename</i> is given, the PKCS#12 file is placed in the current directory with the filename of <code>pkcs12file</code> .
<code>-p <i>pkcs11-lib</i></code>	Specifies the PKCS#11 library from which to extract keys and certificates. This option requires a full path to a dynamic library, provided in the <i>pkcs11-lib</i> variable. By default, <code>pk11export</code> uses the Sun Crypto Accelerator 1000 PKCS#11 library (<code>/opt/SUNWconn/crypto/lib/libpkcs11.so</code>), but any PKCS#11 library can be specified with the <i>pkcs11-lib</i> variable in this option.
<code>-v</code>	Display the version information for <code>pk11export</code> .

Examples

Example 1: List the tokens for a PKCS#11 implementation.

```
# pk11export -l -p /opt/SUNWconn/cryptov2/bin/libvpkcs11.so
0. SUNW acceleration only
1. arf
```

Example 2: Export the Server-Cert certificate from the PKCS#11 token nobody@webserv and place in the /tmp/webserv-export.p12 file.

```
example% pk11export -o /tmp/webserv-export.p12 nobody@webserv
Enter password for nobody@webserv:
Enter password for pkcs12 file:
Re-enter password for pkcs12 file:
/tmp/webserv-export.p12 was created successfully
```

Using the `iplsslcfg` Script

Options 1 and 2 of the `iplsslcfg` script install the modules to configure and register the board with Sun ONE Web and Application Server software. Options 3 and 4 of the script export and import Sun ONE Web Server keys to and from the PKCS#12 format.

▼ To Use Option 1 of the `iplsslcfg` Script for Sun ONE Web Server 4.1

- See “Configuring Sun ONE Web Server 4.1” on page 114.

▼ To Use Option 1 of the `iplsslcfg` Script for Sun ONE Web Server 6.0

- See “Configuring Sun ONE Web Server 6.0” on page 124.

▼ To Use Option 2 of the `iplsslcfg` Script

1. Type the following to execute the `iplsslcfg` script:

```
# /opt/SUNWconn/cryptov2/bin/iplsslcfg
```

2. Type **2** for the Sun ONE Application Server and enter the binary and domain paths.

```
Sun Crypto Accelerator Sun ONE Installation
-----
This script will install the Sun Crypto Accelerator
cryptographic modules for Sun ONE Products.

Please select what you wish to do:
-----
1. Configure Sun ONE Web Server for SSL
2. Configure Sun ONE Application Server for SSL
3. Export Sun ONE Web Server keys to PKCS#12 format
4. Import keys from PKCS#12 format for Sun ONE Web Server

Your selection (0 to quit): 2

You will now be prompted for four pieces of information:
 1. The location of the Sun ONE Application Server binaries
 2. The location where Sun ONE Server domains are stored
 3. The Application Server domain (e.g. domain1)
 4. The Application Server server name (e.g. server1)

Full path to Application Server binaries: [/opt/SUNWappserver7]:
/opt/SUNWappserver7
Full path to Application Server domains: [/var/opt/SUNWappserver7]:
/var/opt/SUNWappserver7
Application Server domain: domain1
Application Server server name: server1
This script will update your Sun ONE Application Server installation
in /opt/SUNWappserver7 to use the Sun Crypto Accelerator.
You will need to restart your admin server after this has completed.
Ok to proceed? [Y/N]: y
Using database directory
/var/opt/SUNWappserver7/domains/domain1/server1/config...
Module "Sun Crypto Accelerator 4000" added to database.
/opt/SUNWappserver7 has been configured to use
the Sun Crypto Accelerator.

<Press ENTER to continue>
```

3. Type 0 to quit.

▼ To Use Option 3 of the `iplsslcfg` Script

This option exports SSL certificates and keys from the Sun ONE Web Server internal database into PKCS#12 format. These certificates can then be reimported into the Sun Crypto Accelerator 4000 module.

1. Type the following to execute the `iplsslcfg` script:

```
# /opt/SUNWconn/criptov2/bin/iplsslcfg
```

2. Type 3 to export Sun ONE Web Server keys to PKCS#12 format, and press Return.

```
Sun Crypto Accelerator Sun ONE Installation
-----
This script will install the Sun Crypto Accelerator
cryptographic modules for Sun ONE Products.

Please select what you wish to do:
-----
1. Configure Sun ONE Web Server for SSL
2. Configure Sun ONE Application Server for SSL
3. Export Sun ONE Web Server keys to PKCS#12 format
4. Import keys from PKCS#12 format for Sun ONE Web Server

Your selection (0 to quit): 3
```

3. Type the path of the Sun ONE server directory.

The `iplsslcfg` utility searches for any potential certificates and key databases from which you can export keys.

```
Please enter the full path of the web server
root directory [/usr/iplanet/servers]: /usr/iplanet/servers
```

4. Type in the name from the list provided.

```
The following certificate databases were found:
https-machine.domain.com-webserv1-
https-machine.domain.com-webserv2-
Which certificate database do you wish to export from?
https-machine.domain.com-webserv1-
```

5. Provide the server certificate friendly name you wish to export.

By default, this name is Server-Cert.

```
Please provide the name for the certificate you wish to export.
If you wish to export from a hardware device, you will need to
provide the token name followed by a ":" and the certificate name.
Not all external tokens will allow keys to be exported.
Certificate Name [Server-Cert]: Server-Cert
```

6. Specify the path and filename for the PKCS#12 file.

```
Please specify the path where the PKCS#12 file will be stored:
/tmp/export.p12
```

7. Enter passwords

Upon successful authentication, you are asked to set the password for the PKCS#12 file. Once this password is created, the PKCS#12 file is written to the filename you chose in Step 6.

```
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
Successfully created the PKCS#12 file.
<Press ENTER to continue>
```

8. Type 0 to quit.

▼ To Use Option 4 of the ip1sslcfg Script

This option imports keys and certificates from PKCS#12 format into the board.

1. Type the following to execute the `iplsslcfg` script:

```
# /opt/SUNWconn/cryptov2/bin/iplsslcfg
```

2. Type `4` to import keys from PKCS#12 format for the Sun ONE Web Server, and press Return.

```
Sun Crypto Accelerator Sun ONE Installation
-----
This script will install the Sun Crypto Accelerator
cryptographic modules for Sun ONE Products.

Please select what you wish to do:
-----
1. Configure Sun ONE Web Server for SSL
2. Configure Sun ONE Application Server for SSL
3. Export Sun ONE Web Server keys to PKCS#12 format
4. Import keys from PKCS#12 format for Sun ONE Web Server

Your selection (0 to quit): 4
```

3. Type the path to the Sun ONE server directory.

```
Please enter the full path of the web server
root directory [/usr/iplanet/servers]: /usr/iplanet/servers
```

4. Type the path to the PKCS#12 file you wish to import.

```
Enter the path to the PKCS#12 file: /tmp/export.p12
```

5. Answer yes to the following question.

```
Will you be importing to a hardware device? [Y/N]: Y
```

6. Type the keystore name that you configured the board to use during initialization.

```
Enter the token name: vca0
```

7. Type the `username:password` string to authenticate successfully. See TABLE 5-1.

```
Enter Password or Pin for "vca0":
```

8. Type the password used to protect the PKCS#12 file.

```
Enter password for PKCS12 file:  
Import successful.  
  
<Press ENTER to continue>
```

Using the `apsslcfg` Script

Option 1 of the `apsslcfg` script configures the Apache Web Server for SSL. Option 2 configures keys for Apache Web Servers.

Note – The `apsslcfg` script supports Apache Web Server 1.3.26 only.

▼ To Use Option 1 of the `apsslcfg` Script

- See “Configuring Apache Web Server 1.3x” on page 176.

Using Option 2 of the `apsslcfg` Script

Option 2 has 3 subsequent options as follows:

1. Generate a keypair and request a certificate for Apache
2. Export Apache (PEM encoded X.509) keys to PKCS#12 format
3. Import keys from PKCS#12 format to Apache (PEM encoded X.509)

▼ To Generate a Keypair and Request a Certificate for Apache

This option generates RSA keys and certificate requests that can be submitted to a certification authority.

1. Type 1 to select this option.

2. Type the path for the binaries and Apache modules, and the path for the configuration files.

```
Please enter the directory where the Apache
binaries and libraries exist [/usr/apache]: /usr/apache
```

```
Please enter the directory where the Apache
configuration files exist [/etc/apache]: /etc/apache
```

3. Type the path for the keys.

```
Where would you like the keys stored? [/etc/apache/keys]: /etc/apache/keys
```

4. Type the base name for the key and certificate request files.

This name is prepended to the filenames. For example, if you choose `cert1` the key filename is `cert1-key.pem` and the certificate request filename is `cert1-certreq.pem`.

```
Please choose a base name for the key and request file: cert1
```

5. Select the size of the RSA key to be generated.

Once the bit size has been chosen, the RSA key is generated.

```
What size would you like the RSA key to be [1024]? 1024
```

6. Type the password that encrypts the key file.

Use a strong password and do not forget it.

```
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

7. Type the certificate name components for your request.

The certificate request is written to a file that can be submitted to a certification authority.

```
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]: US
State or Province Name (full name) [Some-State]: California
Locality Name (eg, city) []: San Diego
Organization Name (eg, company) []: Company
Organizational Unit Name (eg, section) []: Department
SSL Server Name (eg, www.company.com) []: www.company.com
Email Address []: admin@domain.com

The keyfile is stored in /etc/apache/keys/cert1-key.pem.
The certificate request is in /etc/apache/keys/cert1-certreq.pem.

<Press ENTER to continue>
```

▼ To Export Apache (PEM Encoded X.509) Keys to PKCS#12 Format

This option enables you to place Apache Web Server keys and certificates into a PKCS#12 file.

1. **Type 2 to select the option.**
2. **Type the paths to the key and certificate files.**

If the key and certificate files are the same file, type the same path twice.

Note – Key and certificate data may be stored in the same file or in separate files. However, when stored in different files, the filenames must be the same.

```
Enter the path to the key file:
Enter the path to the certificate file:
```

3. Type the path for the output PKCS#12 file.

```
Please specify the path where the PKCS#12
file will be stored:
```

4. Type a friendly name for the certificate.

This name uniquely identifies certificates and key pairs.

```
Please provide a friendly name for the PKCS#12 being
built. This friendly name is necessary when
importing your PKCS#12 file for use by other web servers.
Friendly Name [Server-Cert]:
```

5. Type the password that protects the key that to be placed into the PCKS#12 file.

```
Enter pass phrase for /etc/apache/keys/ap1-key.pem:
```

6. Type the password to protect the key data in the PKCS#12 file.

The PKCS#12 file is written to the file specified above.

```
Enter Export Password:
Verifying - Enter Export Password:
Your PKCS#12 file has been created successfully and is in
/tmp/exp.p12

<Press ENTER to continue>
```

▼ **To Import Keys From PKCS#12 Format to Apache (PEM encoded X.509)**

This option enables you to extract keys and certificates from PKCS#12 files and use them with an Apache Web Server.

1. Type 3 to select the option.

2. Type the path and filename of a PKCS#12 file.

```
Enter the path to the PKCS#12 file:
```

3. Type the path for the extracted key and certificate.

```
Enter the directory where keys and certificates
will be stored:
```

4. Type a filename for the key and certificate file.

Both the encrypted key and the certificate will be contained in the same file.

```
Please choose a name for the key and
Certificate file. This file will contain
both the encrypted key and the certificate:
```

5. Type the password used to protect the PKCS#12 file.

```
Enter Import Password:
MAC verified OK
```

6. Type a new password to protect the extracted key file in an Apache-readable format.

The key and certificate data is written to the file specified in Step 4.

```
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:

The keys have been successfully extracted to the file
/etc/apache/key2/yakstuff.pem.

<Press ENTER to continue>
```

Assigning Different MAC Addresses to Multiple Boards Installed in the Same Server

There are two methods to assign different MAC addresses to multiple boards in a single server. The first method is at the operating-system level, and the second is at the OpenBoot PROM level.

▼ To Assign Different MAC Addresses From a Terminal Window

1. Enter the following command:

```
# eeprom "local-mac-address?"=true
```

Note – With the “local-mac-address?” parameter set to `true`, all nonintegrated network interface devices use the local MAC address assigned to the product at the manufacturing facility.

2. Reboot the system.

▼ To Assign Different MAC Addresses From the OpenBoot PROM Level

1. Enter the following command at the OpenBoot PROM `ok` prompt:

```
ok setenv local-mac-address? true
```

Note – With the “local-mac-address?” parameter set to `true`, all nonintegrated network interface devices use the local MAC address assigned to the product at the manufacturing facility.

2. Boot the operating system.

Installing and Configuring Sun ONE Server Software

This chapter describes how to configure the Sun Crypto Accelerator 4000 board for use with Sun ONE servers. This chapter includes the following sections:

- “Administering Security for Sun ONE Web Servers” on page 105
- “Configuring Sun ONE Web Servers” on page 110
- “Configuring Sun ONE Web Servers to Start Up Without User Interaction on Reboot” on page 113
- “Installing and Configuring Sun ONE Web Server 4.1” on page 113
- “Installing and Configuring Sun ONE Web Server 6.0” on page 123
- “Installing and Configuring Sun ONE Application Server 7” on page 133
- “Installing and Configuring Sun ONE Directory Server 5.2” on page 146
- “Installing and Configuring Sun ONE Messaging Server 5.2” on page 158
- “Installing and Configuring Sun ONE Portal Server 6.2” on page 169

Note – The Sun ONE servers described in this manual were previously named iPlanet™ Servers.

Administering Security for Sun ONE Web Servers

This section provides an overview of the security features of the Sun Crypto Accelerator 4000 board as it is administered with Sun ONE Web Servers.

Note – To manage keystores, you must have access to the system administrator account for your system.

Concepts and Terminology

Keystores and users must be created for applications that communicate with the Sun Crypto Accelerator 4000 board through a PKCS#11 interface, such as the Sun ONE Web Server.

Note – The Apache Web Server (Chapter 6) does not use the keystore or user account features described in this chapter.

Within the context of the Sun Crypto Accelerator 4000 board, users are owners of cryptographic keying material. Each key is owned by a single user. Each user may own multiple keys. A user might want to own multiple keys to support different configurations, such as a `production` key and a `development` key (to reflect the organizations the user is supporting).

Note – The term *user* or *user account* refers to Sun Crypto Accelerator 4000 users created in `vcaadm`, not traditional UNIX user accounts. There is no fixed mapping between UNIX user names and Sun Crypto Accelerator 4000 user names.

A keystore is a repository for key material. Associated with a keystore are security officers and users. Keystores provide not only storage, but a means for key objects to be owned by user accounts. This enables keys to be hidden from applications that do not authenticate as the owner. Keystores have three components:

- **Key objects** – Long-term keys that are stored for applications such as the Sun ONE Web Server.
- **User accounts** – Accounts that provide applications a means to authenticate and access specific keys
- **Security officer accounts** – Accounts that provide access to key management functions through `vcaadm`.

Note – A single Sun Crypto Accelerator 4000 board must have exactly one keystore. Multiple Sun Crypto Accelerator 4000 boards can be configured to collectively work with the same keystore to provide additional performance and fault-tolerance.

A typical installation contains a single keystore with three users. For example, such a configuration could consist of a single keystore *sca4000-ks-1* and three users within that keystore, *webserv*, *dirserv*, and *mailserv*. This would enable the three users to own and maintain access control of their server keys within that single keystore. FIGURE 5-1 illustrates an overview of a typical installation.

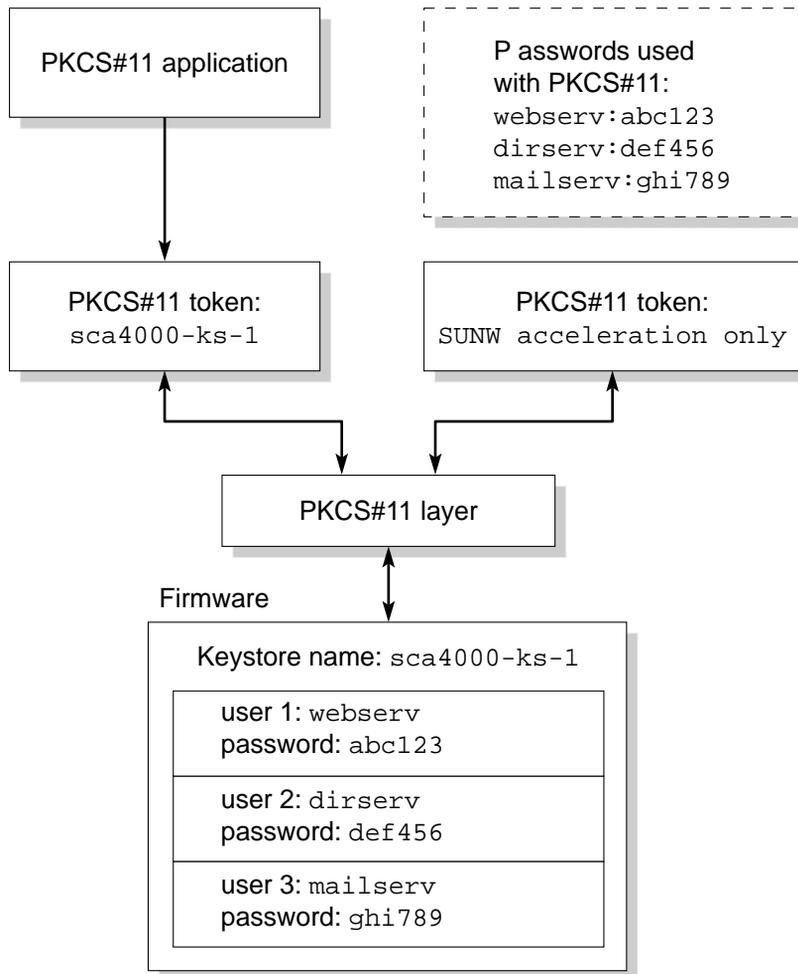


FIGURE 5-1 Keystore and Users Overview

An administrative tool, *vcaadm*, is used to manage Sun Crypto Accelerator 4000 keystores and users. Refer to “Managing Keystores With *vcaadm*” on page 71.

Tokens and Token Files

Keystores appear to Sun ONE Web Servers as *tokens*. Token files enable Sun Crypto Accelerator 4000 administrators to selectively present only specific tokens to a given application.

Example

If three keystores are created, *engineering*, *finance*, and *legal*, by default, the three tokens are presented to the Sun ONE Web Server:

- engineering
- finance
- legal

Token Files

To override the default case, a token file must exist. Some applications cannot handle multiple tokens. Token files are text files that contain one or more token names, one per line.

Note – Token names and keystore names are the same.

A Sun ONE Web Server presents only the tokens listed in the token file. The methods of specifying token files are as follows (in order of precedence):

1. The file named by the environment variable `SUNW_PKCS11_TOKEN_FILE`

Some application software suppresses environment variables, in which case this approach might not be feasible.

2. The file `$HOME/.SUNWconn_cryptov2/tokens`

This file must exist in the home directory of the UNIX user for which the Sun ONE Web Server runs. The Sun ONE Web Server may run as a UNIX user who has no home directory, in which case this approach might not be feasible.

3. The file `/etc/opt/SUNWconn/cryptov2/tokens`

If no token file exists, the Sun Crypto Accelerator 4000 software presents all tokens to Sun ONE Web Servers.

The following is an example of a token file:

```
=====  
# This is an example token file  
  
engineering # Comments are acceptable on the same line  
  
legal  
  
# Because the finance keystore is not listed, the Sun Crypto  
# Accelerator will not present it to the Sun ONE Web Server.  
  
...  
=====
```

Note – Comments are preceded by a pound sign (#). Empty lines are acceptable.

If none of the files described in this subsection are found, then the default method described in “Tokens and Token Files” on page 108 is used.

Enabling and Disabling Bulk Encryption

The bulk encryption feature for SunONE server software is disabled by default. You might want to enable this feature for securely transferring primarily large files.

To enable Sun ONE server software to use bulk encryption on the Sun Crypto Accelerator 4000 board, you simply create an empty file in the `/etc/opt/SUNWconn/cryptov2/` directory named `sslreg`, and restart the server software.

```
# touch /etc/opt/SUNWconn/cryptov2/sslreg
```

To disable the bulk encryption feature, you must delete the `sslreg` file and restart the server software.

```
# rm /etc/opt/SUNWconn/cryptov2/sslreg
```

Configuring Sun ONE Web Servers

This section describes the following topics:

- “Passwords” on page 110
- “Populating a Keystore” on page 111
- “Overview of Enabling Sun ONE Web Servers” on page 112

Passwords

You are asked for several passwords in the course of enabling a Sun ONE Web Server. TABLE 5-1 provides a description of each. These passwords are referred to throughout this chapter.

TABLE 5-1 Passwords Required for Sun ONE Web Servers

Type of Password	Description
Sun ONE Web Server Administration Server	Required to start up the Sun ONE Web Server Administration Server. This password was assigned during the Sun ONE Web Server setup.
Web Server Trust Database	Required to start the internal cryptographic module when running in secure mode. This password was assigned when creating a trust database through the Sun ONE Web Server Administration Server. This password is also required when requesting and installing certificates into the internal cryptographic module.
Security Officer	Required when performing <code>vcaadm</code> privileged operations.
<i>username:password</i>	Required to start the Sun Crypto Accelerator 4000 module when running in secure mode. This password is also required when requesting and installing certificates into the internal cryptographic module (<i>keystore-name</i>). This password consists of the <i>username</i> and <i>password</i> of a keystore user that was created in <code>vcaadm</code> . The keystore <i>username</i> and <i>password</i> are separated by a colon (:).

Populating a Keystore

Before you can enable the board for use with a Sun ONE Web Server, you must first initialize the board and populate the board's keystore with at least one user. The keystore for the board is created during the initialization process. You can also initialize Sun Crypto Accelerator 4000 boards to use an existing keystore. Refer to "Initializing the Board With `vcaadm`" on page 68.

Note – Only one keystore per Sun Crypto Accelerator 4000 board can be configured and you must configure one keystore per board. You can configure multiple Sun Crypto Accelerator 4000 boards to collectively work with the same keystore to provide additional performance and fault-tolerance.

▼ To Populate a Keystore

1. If you have not already done so, place the Sun Crypto Accelerator 4000 tools directory in your search path, for example:

```
$ PATH=$PATH:/opt/SUNWconn/bin
$ export PATH
```

2. Access the `vcaadm` utility with the `vcaadm` command or enter `vcaadm -h hostname` to connect `vcaadm` to a board on a remote host.

See "Using the `vcaadm` Utility" on page 59.

```
$ vcaadm -h hostname
```

3. Populate the board's keystore with users.

These user names are known only within the domain of the Sun Crypto Accelerator 4000 board and do not need to be identical to the UNIX user name that the web server process is using. Before attempting to create the user, remember that you must first log in as a `vcaadm` security officer.

4. Create a user with the `create user` command.

```
vcaadm{vcaN@hostname, sec-officer}> create user username
Initial password:
Confirm password:
User username created successfully.
```

The *username* and *password* created here collectively make the *username:password* (See TABLE 5-1). You must use this password when authenticating during a web server startup. This is the keystore password for a single user.



Caution – Users must remember this *username:password*. Without this password, users cannot access their keys. There is no way to retrieve a lost password.

5. Exit `vcaadm`.

```
vcaadm{vcaN@hostname, sec-officer}> exit
```

Overview of Enabling Sun ONE Web Servers

To enable Sun ONE Web Servers you must complete the following procedures, which are explained in detail in the next two sections.

1. Install the Sun ONE Web Server.
2. Create a trust database.
3. Request a certificate.
4. Install the certificate.
5. Configure the Sun ONE Web Server.



Caution – These procedures must be followed in the order given. Failure to do so could result in an incorrect configuration.

- If you are using Sun ONE Web Server 4.1, go to “Installing and Configuring Sun ONE Web Server 4.1” on page 113.
- If you are using Sun ONE Web Server 6.0, go to “Installing and Configuring Sun ONE Web Server 6.0” on page 123.

Configuring Sun ONE Web Servers to Start Up Without User Interaction on Reboot

You can enable the Sun ONE Web Servers to perform an unattended startup at reboot with an encrypted key.

▼ To Create an Encrypted Key for Automatic Startup of Sun ONE Web Servers on Reboot

1. Navigate to the `config` subdirectory for your Sun ONE Web Server instance—for example, `/usr/iplanet/servers/https-webserver-instance-name/config`.
2. Create a `password.conf` file with only the following lines (See TABLE 5-1 for password definitions):

```
internal:trust-db-password  
keystore-name:username:password
```

3. Set the file ownership of the password file to the UNIX user ID that the web server runs as, and set the file permissions to be readable only by the owner of the file:

```
# chown web-server-UNIX-user-ID password.conf  
# chmod 400 password.conf
```

Installing and Configuring Sun ONE Web Server 4.1

This section describes how to install and configure Sun ONE Web Server 4.1 to use the board. You must perform these procedures in order. Refer to the Sun ONE Web Server documentation for more information about installing and using Sun ONE Web Servers. This section includes the following procedures:

- “To Install Sun ONE Web Server 4.1” on page 114

- “Configuring Sun ONE Web Server 4.1” on page 114
- “To Create a Trust Database” on page 115
- “To Register the Board With the Web Server” on page 116
- “To Generate a Server Certificate” on page 117
- “To Install the Server Certificate” on page 120
- “To Enable the Web Server for SSL” on page 121

▼ To Install Sun ONE Web Server 4.1

1. Download the Sun ONE Web Server 4.1 software.

You can find the web server software at the following URL:
<http://www.sun.com/>

2. Change to the installation directory and extract the web server software.

3. Install the web server with the `setup` script from the command-line.

The default path name for the server is `/usr/netscape/server4`.

This chapter refers to the default paths. If you decide to install the web server software in a different location, be sure to note where you installed it.

```
# ./setup
```

4. Answer the prompts from the installation script.

Except for the following prompts, you can accept the default.

- a. Agree to accept the license terms by typing `yes`.
- b. Enter a fully qualified domain name.
- c. Enter the Sun ONE Web Server 4.1 Administration Server password twice.
- d. Press Return when prompted.

Configuring Sun ONE Web Server 4.1

These procedures create a trust database for the web server instance; register the board with the web server; generate and install a server certificate; and enable the web server for SSL.

The Sun ONE Web Server Administration Server must be up and running during the configuration process.

▼ To Create a Trust Database

1. Start the Sun ONE Web Server 4.1 Administration Server.

Instead of running `startconsole` as `setup` requests, start a Sun ONE Web Server 4.1 Administration Server by typing the following command:

```
# /usr/netcape/server4/https-admserv/start
SunONE-WebServer-Enterprise/4.1SP9 BB1-08/23/2001 05:50
startup: listening to http://hostname.domain, port 8888 as root
```

The response provides the URL for connecting to your servers.

2. Start the Administration graphical user interface (GUI) by opening up a web browser and typing:

```
http://hostname.domain:admin-port
```

In the authentication dialog box enter the Sun ONE Web Server 4.1 Administration Server user name and password you selected while running `setup`.

Note – If you used the default settings during the Sun ONE Web Server setup, type `admin` for the user ID or the Sun ONE Web Server 4.1 Administration Server user name.

3. Select OK.

The Sun ONE Web Server 4.1 Administration Server window is displayed.

4. Create the trust database for the web server instance.

a. Click the **Servers** tab in the Sun ONE Web Server 4.1 Administration Server window.

b. Select a server and click the **Manage** button.

c. Click the **Security** tab near the top of the page and click the “**Create Database**” link.

d. Enter a password (web server trust database; see TABLE 5-1) in the two dialog boxes and select **OK**.

Choose a password of at least eight characters. You use this password to start the internal cryptographic modules when the Sun ONE Web Server runs in secure mode.

You might want to enable security on more than one web server instance. If so, repeat Step 1 through Step 4 for each web server instance.

Note – If you want to run Secure Socket Layer (SSL) on the Sun ONE Web Server 4.1 Administration Server server as well, the process of setting up a trust database is similar. Refer to the *iPlanet Web Server, Enterprise Edition Administrator's Guide* at <http://docs.sun.com> for more information.

▼ To Register the Board With the Web Server

1. Execute the following script to register the board with the web server:

```
# /opt/SUNWconn/bin/iplsslcfg
```

This script prompts you to choose a server and installs the Sun Crypto Accelerator 4000 cryptographic modules for the Sun ONE server you choose. The script then updates the configuration files to enable the board.

2. Type 1 to configure the Sun ONE Web Server to use SSL and press Return.

Note – This procedure assumes that you choose option 1 at this prompt. If you want to choose options 2, 3 or 4, refer to “Using the iplsslcfg Script” on page 93.

```
Sun Crypto Accelerator Sun ONE Installation
-----
This script will install the Sun Crypto Accelerator
cryptographic modules for Sun ONE Products.

Please select what you wish to do:
-----
1. Configure Sun ONE Web Server for SSL
2. Configure Sun ONE Application Server for SSL
3. Export Sun ONE Web Server keys to PKCS#12 format
4. Import keys from PKCS#12 format for Sun ONE Web Server

Your selection (0 to quit): 1
```

3. Enter the path of the web server root directory when prompted and press Return.

```
Please enter the full path of the web server
root directory [/usr/netscape/server4]: /usr/netscape/server4
```

4. Type *y* and press Return when prompted.

```
This script will update your Sun ONE Web Server installation
in /usr/netscape/server4 to use the Sun Crypto Accelerator
You will need to restart your admin server after this has
completed.
Ok to proceed? [Y/N]: y

Using database directory /usr/netscape/server4/alias...
Module "Sun Crypto Accelerator 4000" added to database.
/usr/netscape/server4 has been configured to use
the Sun Crypto Accelerator.

<Press ENTER to continue>
```

5. Type *0* to quit.

▼ **To Generate a Server Certificate**

1. Restart the Sun ONE Web Server 4.1 Administration Server by typing the following commands:

```
# /usr/netscape/server4/https-admserv/stop
# /usr/netscape/server4/https-admserv/start
```

The response provides the URL for connecting to your servers.

2. Start the Administration GUI by opening up a web browser and typing:

```
http://hostname.domain:admin-port
```

In the authentication dialog box, enter the Sun ONE Web Server 4.1 Administration Server user name and password you selected while running setup.

Note – If you used the default settings during Sun ONE Web Server setup, type *admin* for the User ID or the Sun ONE Web Server 4.1 Administration Server user name.

3. Select OK.

The Sun ONE Web Server 4.1 Administration Server window is displayed.

4. To request the server certificate, select the **Security** tab near the top of the Sun ONE Web Server 4.1 Administration Server window (FIGURE 5-2).

The Create Trust Database page is displayed.

5. Select the “Request a Certificate” link on the left panel (FIGURE 5-2).

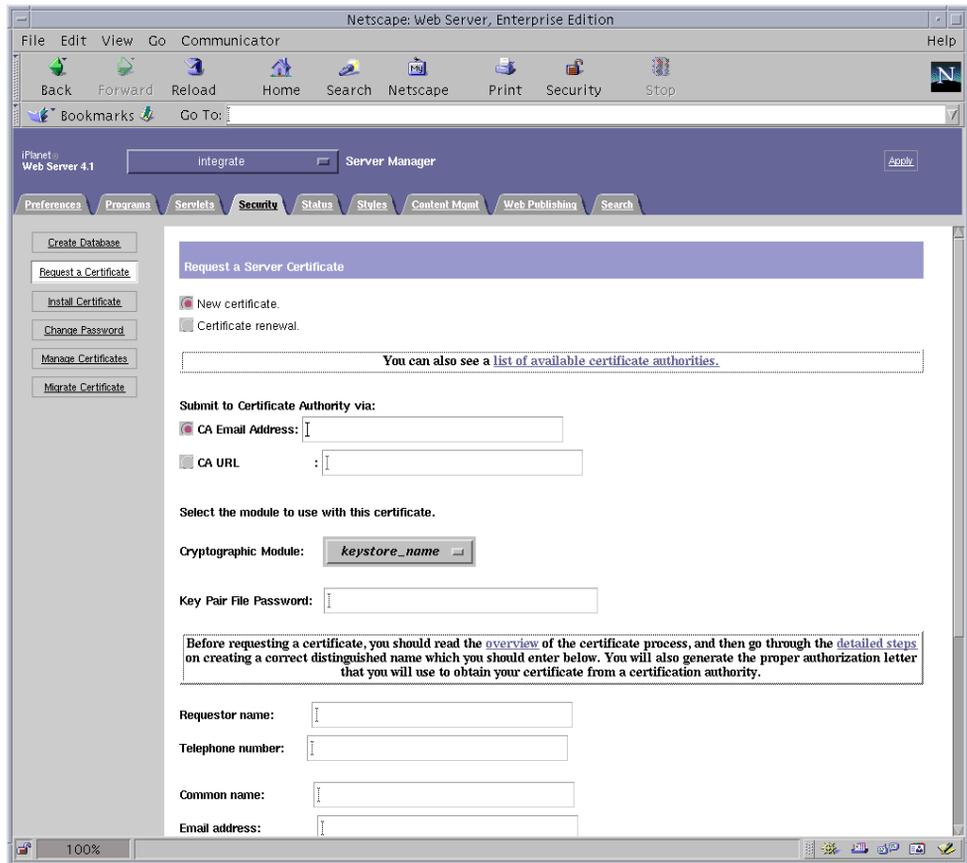


FIGURE 5-2 Sun ONE Web Server 4.1 Administration Server Request a Server Certificate Dialog Box

6. Fill out the form to generate a certificate request, using the following information:

- a. Select a New Certificate.

If you can directly post your certificate request to a web-capable certificate authority or registration authority, select the “CA URL” (Certificate Authority URL) link. Otherwise, select “CA Email Address” and enter an email address where you would like the certificate request to be sent.

b. Select the “Cryptographic Module” you want to use.

Each keystore has its own entry in this pull-down menu. Be sure that you select the correct keystore. Do not select “SUNW acceleration only.”

c. In the “Key Pair File Password” dialog box, provide the password for the user that will own the key.

This password is the *username:password* (TABLE 5-1).

d. Type the appropriate information for the requestor information fields in

TABLE 5-2.

TABLE 5-2 Requestor Information Fields

Field	Description
Requestor Name	Contact information for the requestor
Telephone Number	Contact information for the requestor
Common Name	Web site domain that is typed in a visitor’s browser
Email Address	Contact information for the requestor
Organization	Company name
Organizational Unit	(Optional) Department of the company
Locality	(Optional) City, county, principality, or country
State	(Optional) Full name of the state
Country	Two-letter ISO code for the country (for example, the United States is US)

e. Click OK to submit the information.

7. Use a certificate authority to generate the certificate.

- If you choose to post your certificate request to a CA URL, the certificate request is automatically posted there.
- If you choose the “CA Email Address,” copy the certificate request that was emailed to you with the headers and hand it off to your certificate authority.

8. Once the certificate is generated, copy it, along with the headers, to the clipboard.

Note – The certificate is different from the certificate request and is usually presented to you in text form. Keep this data on the clipboard for Step 5 of the following procedure.

▼ To Install the Server Certificate

1. Select the “Install Certificate” link on the left side of the Sun ONE Web Server 4.1 Administration Server window.

Once your request has been approved by a certificate authority and a certificate has been issued, you must install the certificate in the Sun ONE Web Server.

2. Click the Security tab.
3. On the left panel, choose the “Install Certificate” link.

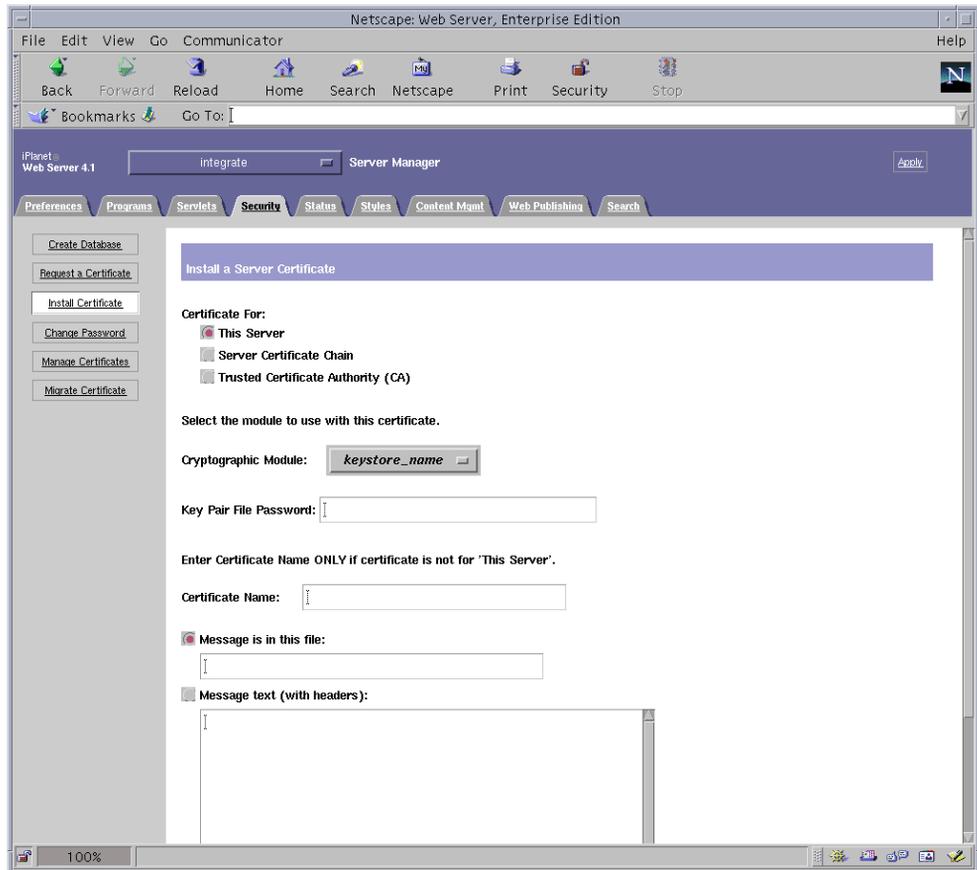


FIGURE 5-3 Sun ONE Web Server 4.1 Administration Server Install a Server Certificate Dialog Box

4. Fill out the form to install your certificate:

TABLE 5-3 Fields for the Certificate to Install

Fields	Description
Certificate For	This server
Cryptographic Module	Each keystore has its own entry in this pull-down menu. Be sure to select the correct keystore name. To use the board, you must select a module with the same name you assigned the keystore.
Key Pair File Password	This password is the <i>username:password</i> (TABLE 5-1).
Certificate Name	In most cases, you can leave this blank. If you provide a name, it alters the name the web server uses to access the certificate and key when running with SSL support. The default for this field is <code>Server-Cert</code> .

5. Paste the certificate you copied from the certificate authority (in Step 8 of “To Generate a Server Certificate” on page 117) into the Message box.

You are shown some basic information about the certificate.

6. Click OK.

7. If everything looks correct, select the “Add Server Certificate” button.

On-screen messages tell you to restart the server. This is not necessary because the web server instance has been shut down the entire time.

You are also notified that in order for the web server to use SSL, the web server must be configured to do so. Use the following procedure to configure the web server.

Note – Refer to the `mod_ssl` and OpenSSL documentation for information on how to self-sign a certificate for testing.

Now that your web server and the server certificate are installed, you must enable the web server for SSL.

▼ To Enable the Web Server for SSL

1. From the main Sun ONE Web Server 4.1 Administration Server page, select the web server instance you want to work with and select **Manage**.
2. If the **Preferences** tab is not selected at the top of the page, click the **Preferences** tab.
3. Select the “**Encryption On/Off**” link on the left side of the page.

4. Set encryption to On.

The Port field in the dialog box should update to the default SSL port number 443. Alter the port number if necessary.

5. Click the OK button.

6. Apply these changes by clicking the Save button.

The web server is now configured to run in secure mode.

7. Edit the `/usr/netscape/server4/https-hostname/config/magnus.conf` file (*hostname is the name of the web server*) by adding the following line:

```
CERTDefaultNickname keystore-name:Server-Cert
```

By default, the certificate you generated is named `Server-Cert`. If your certificate has a different name, be sure to use the name you chose instead of `Server-Cert`.

8. Select the server you want to administer and click the Apply button in the far upper right corner of the page.

This selection applies the changes through the Sun ONE Web Server 4.1 Administration Server.

9. Click the “Load Configuration Files” button to apply the changes you just made to the `magnus.conf` file.

You are redirected to a page that enables you to start your web server instance.

If you select the Apply Changes button when the server is off, an authentication dialog box prompts you for the `username:password`. This window is not resizable, and you might have a problem submitting the change.

There are two workarounds for this problem:

- Select the Load Configuration Files instead.
- Start up the web server first, and click the Apply Changes button.

10. In the Sun ONE Web Server 4.1 Administration Server window, select the On/Off link on the left side of the window.

11. Enter the passwords for the servers and select the OK button.

You are prompted for one or more passwords. At the Module Internal prompt, provide the password for the web server trust database.

At the Module `keystore-name` prompt, enter the `username:password` for that keystore.

Enter the `username:password` for other keystores as prompted.

12. Verify the new SSL-enabled web server at the following URL:

```
https://hostname.domain:server-port/
```

Note – The default *server-port* is 443.

Installing and Configuring Sun ONE Web Server 6.0

This section describes how to install and configure Sun ONE Web Server 6.0 to use the board. You must perform these procedures in order. Refer to the Sun ONE Web Server documentation for more information about installing and using Sun ONE Web Servers. This section includes the following procedures:

- “To Install Sun ONE Web Server 6.0” on page 123
- “Configuring Sun ONE Web Server 6.0” on page 124
- “To Create a Trust Database” on page 124
- “To Register the Board With the Web Server” on page 125
- “To Generate a Server Certificate” on page 127
- “To Install the Server Certificate” on page 130
- “To Enable the Web Server for SSL” on page 131

▼ To Install Sun ONE Web Server 6.0

1. Download the Sun ONE Web Server 6.0 software.

You can find the web server software at the following URL:
<http://www.sun.com/>

2. Change to the installation directory and extract the web server software.

3. Install the web server with the setup script from the command-line.

The default path name for the server is: `/usr/iplanet/servers`.

This chapter refers to the default paths. If you decide to install the software in a different location, be sure to note where you installed it.

```
# ./setup
```

4. Answer the prompts from the installation script.

Except for the following prompts, you can accept the defaults:

- a. Agree to accept the license terms by typing `yes`.

- b. Enter a fully qualified domain name.
- c. Enter the Sun ONE Web Server 6.0 Administration Server password twice.
- d. Press Return when prompted.

Configuring Sun ONE Web Server 6.0

These procedures create a trust database for the web server instance; register the board with the web server; generate and install a server certificate; and enable the web server for SSL.

The Sun ONE Web Server Administration Server must be up and running during the configuration process.

▼ To Create a Trust Database

1. Start the Sun ONE Web Server 6.0 Administration Server.

To start a Sun ONE Web Server 6.0 Administration Server, use the following command (instead of running `startconsole` as `setup` requests):

```
# /usr/iplanet/servers/https-admserv/start
SunONE-WebServer-Enterprise/6.0SP1 B08/20/2001 00:58
warning: daemon is running as super-user
[LS ls1] http://hostname.domain/port 8888 ready to accept requests
startup: server started successfully
```

The response provides the URL for connecting to your servers.

2. Start the Administration GUI by opening up a web browser and typing:

```
http://hostname.domain:admin-port
```

In the authentication dialog box, enter the Sun ONE Web Server 6.0 Administration Server user name and password you selected while running `setup`.

Note – If you used the default settings during Sun ONE Web Server setup, enter `admin` for the User ID or the Sun ONE Web Server 6.0 Administration Server user name.

3. Click OK.

The Sun ONE Web Server 6.0 Administration Server window is displayed.

4. Create the trust database for the web server instance.

You might want to enable security on more than one web server instance. If so, repeat Step 1 through Step 4 for each web server instance.

Note – If you want to run SSL on the Sun ONE Web Server 6.0 Administration Server as well, the process of setting up a trust database is similar. Refer to the *iPlanet Web Server, Enterprise Edition Administrator's Guide* at <http://docs.sun.com> for more information.

- a. Click the Servers tab in the Sun ONE Web Server 6.0 Administration Server dialog box.
- b. Select a server and click the Manage button.
- c. Click the Security tab near the top of the page and click the “Create Database” link.
- d. Enter a password (web server trust database, see TABLE 5-1) in the two dialog boxes and click OK.

Choose a password of at least eight characters. This will be the password used to start the internal cryptographic modules when the Sun ONE Web Server runs in secure mode.

▼ To Register the Board With the Web Server

1. Execute the following script to register the board with the web server:

```
# /opt/SUNWconn/criptov2/bin/iplsslcfg
```

This script prompts you to choose a server and installs the Sun Crypto Accelerator 4000 cryptographic modules for the Sun ONE server you choose. The script then updates the configuration files to enable the board.

2. Type 1 to configure your Sun ONE Web Server to use SSL and press Return.

Note – This procedure assumes that you choose option 1 at this prompt. If you want to choose options 2, 3 or 4, see “Using the iplsslcfg Script” on page 93.

```
Sun Crypto Accelerator Sun ONE Installation
-----
This script will install the Sun Crypto Accelerator
cryptographic modules for Sun ONE Products.

Please select what you wish to do:
-----
1. Configure Sun ONE Web Server for SSL
2. Configure Sun ONE Application Server for SSL
3. Export Sun ONE Web Server keys to PKCS#12 format
4. Import keys from PKCS#12 format for Sun ONE Web Server

Your selection (0 to quit): 1
```

3. Enter the path of the web server root directory when prompted and press Return.

```
Please enter the full path of the web server
root directory [/usr/iplanet/servers]: /usr/iplanet/servers
```

4. Type *y* and press Return when prompted, if you want to proceed.

```
This script will update your Sun ONE Web Server installation
in /usr/iplanet/servers to use the Sun Crypto Accelerator
You will need to restart your admin server after this has
completed.
Ok to proceed? [Y/N]: y

Using database directory /usr/iplanet/servers/alias...
Module "Sun Crypto Accelerator 4000" added to database.
/usr/iplanet/servers has been configured to use
the Sun Crypto Accelerator.

<Press ENTER to continue>
```

5. Type 0 to quit.

▼ To Generate a Server Certificate

1. Restart the Sun ONE Web Server 6.0 Administration Server by typing the following commands:

```
# /usr/iplanet/servers/https-admserv/stop
# /usr/iplanet/servers/https-admserv/start
```

The response provides the URL for connecting to your servers.

2. Start the Administration GUI by opening up a web browser and typing:

```
http://hostname.domain:admin-port
```

In the authentication dialog box enter the Sun ONE Web Server 6.0 Administration Server user name and password you selected while running `setup`.

Note – If you used the default settings during Sun ONE Web Server setup, enter `admin` for the user ID or the Sun ONE Web Server 6.0 Administration Server user name.

3. Click **OK**.

The Sun ONE Web Server 6.0 Administration Server window is displayed.

4. To request the server certificate, select the **Security** tab near the top of Sun ONE Web Server 6.0 Administration Server window.

The Create Trust Database window is displayed.

5. Click the “Request a Certificate” link on the left panel of the Sun ONE Web Server 6.0 Administration Server window.

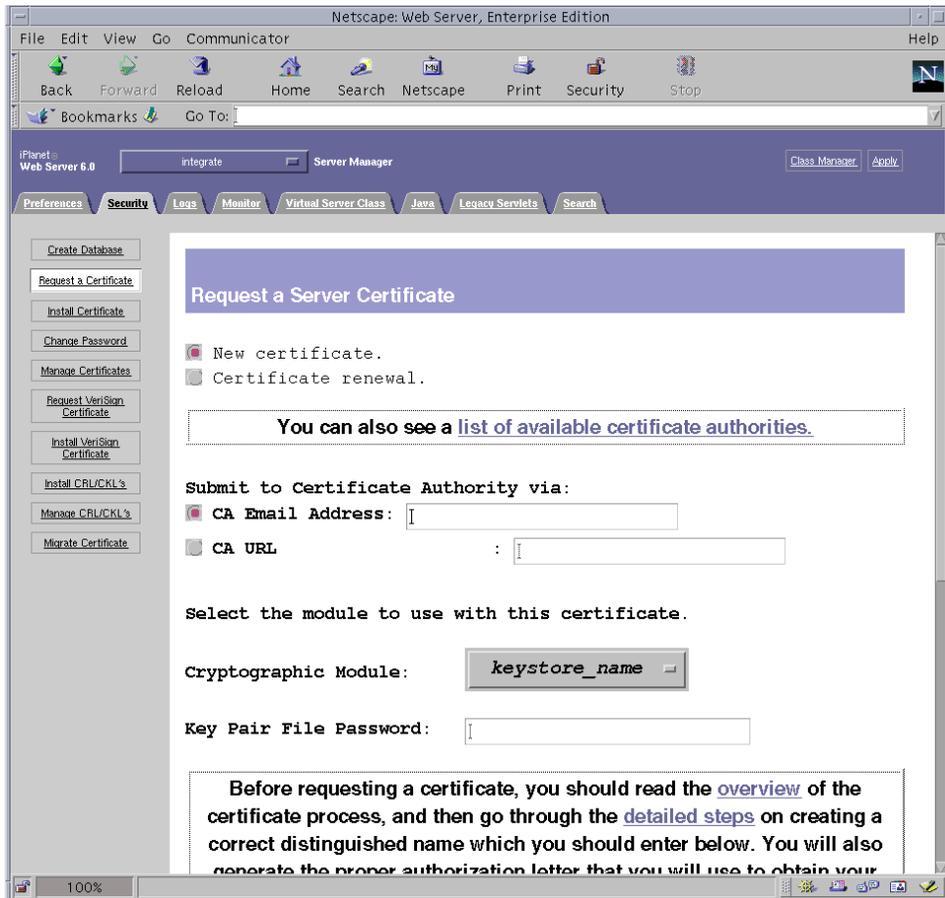


FIGURE 5-4 Sun ONE Web Server 6.0 Administration Server Request a Server Certificate Dialog Box

6. Fill out the form to generate a certificate request, using the following information:

- a. Select a New Certificate.

If you can directly post your certificate request to a web-capable certificate authority or registration authority, select the CA URL link. Otherwise, select CA Email Address and enter an email address where you would like the certificate request to be sent.

- b. Select the “Cryptographic Module” you want to use.

Each keystore has its own entry in this pull-down menu. Be sure that you select the correct keystore. Do not select “SUNW acceleration only.”

- c. In the “Key Pair File Password” dialog box, provide the password for the user that will own the key.

This password is the *username:password* (TABLE 5-1).

- d. Type the appropriate information for the requestor information fields in TABLE 5-4.

TABLE 5-4 Requestor Information Fields

Field	Description
Requestor Name	Contact information for the requestor
Telephone Number	Contact information for the requestor
Common Name	Web site domain that is typed in a visitor's browser
Email Address	Contact information for the requestor
Organization	Company name
Organizational Unit	(Optional) Department of the company
Locality	(Optional) City, county, principality, or country
State	(Optional) Full name of the state
Country	Two-letter ISO code for the country (for example, the United States is US)

- e. Click OK to submit the information.

7. Use a certificate authority to generate the certificate.

- If you choose to post your certificate request to a CA URL, the certificate request is automatically posted there.
- If you choose the CA Email Address, copy the certificate request that was emailed to you with the headers and hand it off to your certificate authority.

8. Once the certificate is generated, copy it, along with the headers, to the clipboard.

Note – The certificate is different from the certificate request and is usually presented to you in text form. Keep this data on the clipboard for Step 5 of “To Install the Server Certificate” on page 130.

▼ To Install the Server Certificate

1. Select the “Install Certificate” link on the left side of the Sun ONE Web Server 6.0 Administration Server window.

Once your request has been approved by a certificate authority and a certificate has been issued, you must install the certificate in the Sun ONE Web Server.

2. Click the Security tab.
3. On the left panel, click the “Install Certificate” link.

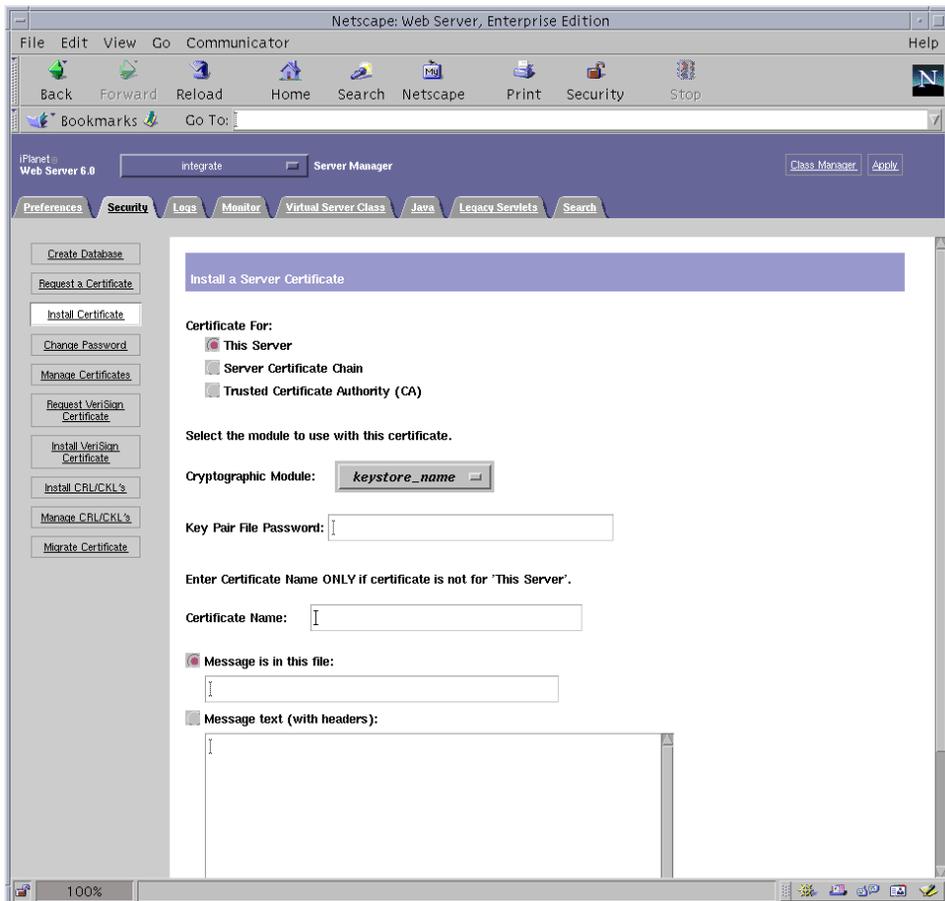


FIGURE 5-5 Sun ONE Web Server 6.0 Administration Server Install a Server Certificate Dialog Box

4. Fill out the form to install your certificate:

TABLE 5-5 Fields for the Certificate to Install

Fields	Description
Certificate For	This server
Cryptographic Module	Each keystore has its own entry in this pull-down menu. Ensure that you select the correct keystore name. To use the board, you must select a module in the form of <i>keystore-name</i> .
Key Pair File Password	This password is the <i>username:password</i> (TABLE 5-1).
Certificate Name	In most cases, you can leave this blank. If you provide a name, it alters the name the web server uses to access the certificate and key when running with SSL support. The default for this field is <code>Server-Cert</code> .

5. Paste the certificate you copied from the certificate authority (in Step 8 of the “To Generate a Server Certificate” on page 127) into the Message text box.

You are shown some basic information about the certificate.

6. Click OK.

7. If everything looks correct, click the “Add Server Certificate” button.

On-screen messages tell you to restart the server. This is not necessary because the web server instance has been shut down the entire time.

You are also notified that in order for the web server to use SSL, the web server must be configured to do so. Use the following procedure to configure the web server.

Note – Refer to the `mod_ssl` and OpenSSL documentation for information on how to self-sign a certificate for testing.

Now that your web server and the Server Certificate are installed, you must enable the web server for SSL.

▼ To Enable the Web Server for SSL

1. Select the Preferences tab near the top of the page.

2. Select the “Edit Listen Sockets” link on the left panel.

The main panel lists all the listen sockets set for the web server instance.

a. Alter the following fields:

- **Port:** Set to the port on which you will be running your SSL-enabled web server (usually this is port 443).
- **Security:** Set to On.

b. Click OK to apply these changes.

In the security field of the Edit Listen Sockets page, there should now be an Attributes link.

3. Select the Attributes link.

4. Enter the *username:password* to authenticate to the keystore on the system.

5. If you want to change the default set of ciphers, select the cipher suites under the Ciphers heading.

A dialog box is displayed for changing the cipher settings. You can select either “Cipher Default” settings, SSL2, or SSL3/TLS. If you select the “Cipher Default,” you are not shown the default settings. The other two choices require you to select the algorithms you want to enable in a pop-up dialog box. Refer to your Sun ONE documentation on cipher selection.

6. Select the certificate for the keystore followed by: *Server-Cert* (or the name you chose).

Only keys that the appropriate keystore user owns appear in the Certificate Name field. This keystore user is the user that is authenticated with the *username:password*.

7. When you have chosen a certificate and confirmed all the security settings, click OK.

8. Select the Apply link in the far upper right corner to apply these changes before you start your server.

9. Select the “Load Configuration Files” link to apply the changes.

You are redirected to a page that allows you to start your web server instance.

If you click the “Apply Changes” button when the server is off, an authentication dialog box prompts you for the *username:password*. This window is not resizable, and you might have a problem submitting the change.

There are two workarounds for this problem:

- Select “Load Configuration Files” instead.
- Start up the web server first, and click “Apply Changes.”

10. In the Sun ONE Web Server 6.0 Administration Server window, select the On/Off link on the left side of the window.

11. Enter the passwords for the servers and click OK.

You are prompted for one or more passwords. At the Module Internal prompt, provide the password for the web server trust database.

At the Module *keystore-name* prompt, enter the *username:password*.

Enter the *username:password* for other keystores as prompted.

12. Verify the new SSL-enabled web server at the following URL:

`https://hostname.domain:server-port/`

Note – The default *server-port* is 443.

Installing and Configuring Sun ONE Application Server 7

This section describes how to install and configure Sun ONE Application Server 7 to use the board. The application server Add-Ons software must be installed in addition to the application server software. You must perform these procedures in order. Refer to the Sun ONE Application Server documentation for more information about installing and using Sun ONE Application Servers. This section includes the following procedures:

- “To Install Sun ONE Application Server 7” on page 133
- “Configuring Sun ONE Application Server 7” on page 135
- “To Create a Trust Database” on page 136
- “To Register the Board With the Application Server” on page 137
- “To Generate a Server Certificate” on page 139
- “To Install the Server Certificate” on page 141
- “To Enable the Application Server for SSL” on page 142

▼ To Install Sun ONE Application Server 7

1. Download the Sun ONE Application Server 7 software.

You can find the application server software at the following URL:

`http://www.sun.com/`

There are different distributions of Sun ONE Application Server 7, each with unique features.

2. Change to the installation directory and extract the application server software.

The default path for the installation directory is different for each distribution of the Sun ONE Application Server 7 software.

3. Run the `setup` program to start the GUI-based installation.

Note – You can also run the `setup -console` program from a terminal window to start a command-line based installation. The examples in this procedure assume you are using the GUI-based installation.

```
# ./setup
```

4. Answer the prompts in the installation script.

Except for the following prompts, you can accept the defaults:

- a. Agree to accept the license terms by typing `yes`.
- b. When prompted for the location of the JDK (Java™ Development Kit), you can either choose: **Use Existing Installation if it is supported, or Install From the Appserver Build.**
- c. Enter the Sun ONE Application Server Administration Server username (you can choose any name).
- d. Enter the Sun ONE Application Server Administration Server password twice.

Note – Perform the following step only if you are using the Solaris 8 OE.

5. If you are using Solaris 8, install the Solaris 8 Sun ONE Application Server patch (109326-08).

This patch is not required for Solaris 9. Download the Solaris 8 Sun ONE Application Server patch from the SunSolve web site:
<http://sunsolve.sun.com>

Add the patch as follows:

```
# cd patch-location/SUNWappserver7/patches
# cd patches/109326-08
# ./patchadd .
```

6. Reboot the system.

▼ To Install the Sun ONE Application Server Add-Ons Software

1. Download the Sun ONE Application Server 7 Add-Ons software.

You can find the application server software at the following URL:

```
http://www.sun.com/
```

2. Extract the application server Add-Ons software.

3. Change to the `./AddOns/SSLUtils` directory

4. Create the directory where the `iplsslcfg` script invokes the `modutil` security tool.

```
# mkdir /usr/bin/mps
```

This path is where the `iplsslcfg` script expects to find the `modutil` security tool.

5. Copy the `modutil`, `certutil`, and `pk12util` binaries to the `/usr/bin/mps/` path.

```
# cp modutil /usr/bin/mps/  
# cp certutil /usr/bin/mps/  
# cp pk12util /usr/bin/mps/
```

6. Enable the execute permission to the binaries in the `/usr/bin/mps/` directory.

```
# chmod 544 /usr/bin/mps/*
```

Configuring Sun ONE Application Server 7

These procedures create a trust database for the application server instance; register the board with the application server; generate and install a server certificate; and enable the application server for SSL and TLS.

The Sun ONE Application Server Administration Server must be up and running during the configuration process.

▼ To Create a Trust Database

1. **Start the Sun ONE Application Server and the Sun ONE Application Server Administration Server.**

```
# installation-directory/bin/asadmin start-appserv
```

Note – Messages appear indicating that the application server is running.

2. **Start the Administration GUI by opening up a web browser and entering the following URL.**

```
http://hostname:4848
```

In the authentication dialog box, enter the Sun ONE Application Server user name and password you created during the `setup` program.

Note – If you used the default settings during Sun ONE Application Server setup, enter `admin` for the User ID or the Sun ONE Application Server Administration Server user name.

3. **Click OK.**

4. **Create the trust database for the application server instance.**

You might want to enable security on more than one application server instance. If so, repeat Step 1 through Step 4 for each application server instance.

Note – If you want to run SSL on the Sun ONE Application Server Administration Server as well, the process of setting up a trust database is similar. Refer to the *Sun ONE Application Server 7 Administrator's Guide* at <http://docs.sun.com/source/816-7158-10/> for more information.

- a. **Navigate to the “Manage Database” section of the Administration GUI.**

Select the Security link on the left panel and click the Manage Database tab on the right panel.

- b. **Type a password of at least eight characters in the two text boxes and click OK.**

This password is the trust database password of the Sun ONE Application Server. This password is used to start the internal cryptographic modules when the application server runs in secure mode.

▼ To Register the Board With the Application Server

1. Execute the `iplsslcfg` script to register the board with the application server.

```
# /opt/SUNWconn/cryptov2/bin/iplsslcfg
```

This script prompts you to choose a server and installs the Sun Crypto Accelerator 4000 cryptographic modules for the Sun ONE server you choose. The script then updates the configuration files to enable the board.

2. Type **2** for the Sun ONE Application Server, and enter the binary and domain paths.

Note – The procedures in this section assume that you choose option 1 at this prompt. If you wish to choose options 3 or 4, refer to “Using the `iplsslcfg` Script” on page 93.

```
Sun Crypto Accelerator Sun ONE Installation
-----
This script will install the Sun Crypto Accelerator
cryptographic modules for Sun ONE Products.

Please select what you wish to do:
-----
1. Configure Sun ONE Web Server for SSL
2. Configure Sun ONE Application Server for SSL
3. Export Sun ONE Web Server keys to PKCS#12 format
4. Import keys from PKCS#12 format for Sun ONE Web Server

Your selection (0 to quit): 2
```

3. Type the location of the binaries and domains, and the domain and server name.

```
You will now be prompted for four pieces of information:
 1. The location of the Sun ONE Application Server binaries
 2. The location where Sun ONE Server domains are stored
 3. The Application Server domain (e.g. domain1)
 4. The Application Server server name (e.g. server1)

Full path to Application Server binaries: [/opt/SUNWappserver7]:
/opt/SUNWappserver7
Full path to Application Server domains:
[/var/opt/SUNWappserver7]: /var/opt/SUNWappserver7
Application Server domain: domain1
Application Server server name: server1
This script will update your Sun ONE Application Server
installation in /opt/SUNWappserver7 to use the Sun Crypto
Accelerator.
You will need to restart your admin server after this has
completed.
Ok to proceed? [Y/N]: y
Using database directory
/var/opt/SUNWappserver7/domains/domain1/server1/config...
Module "Sun Crypto Accelerator 4000" added to database.
/opt/SUNWappserver7 has been configured to use
the Sun Crypto Accelerator.

<Press ENTER to continue>
```

Note – The default installation directory might be different depending on your Sun ONE Application Server 7 distribution.

4. Type 0 to quit.

▼ To Generate a Server Certificate

1. Navigate to the “Certificate Management” section of the Administration GUI.

Select the Security link on the left panel and select the “Certificate Management” tab on the right panel. You are now in the Request submenu window of the “Certificate Management” section of the Administration GUI.

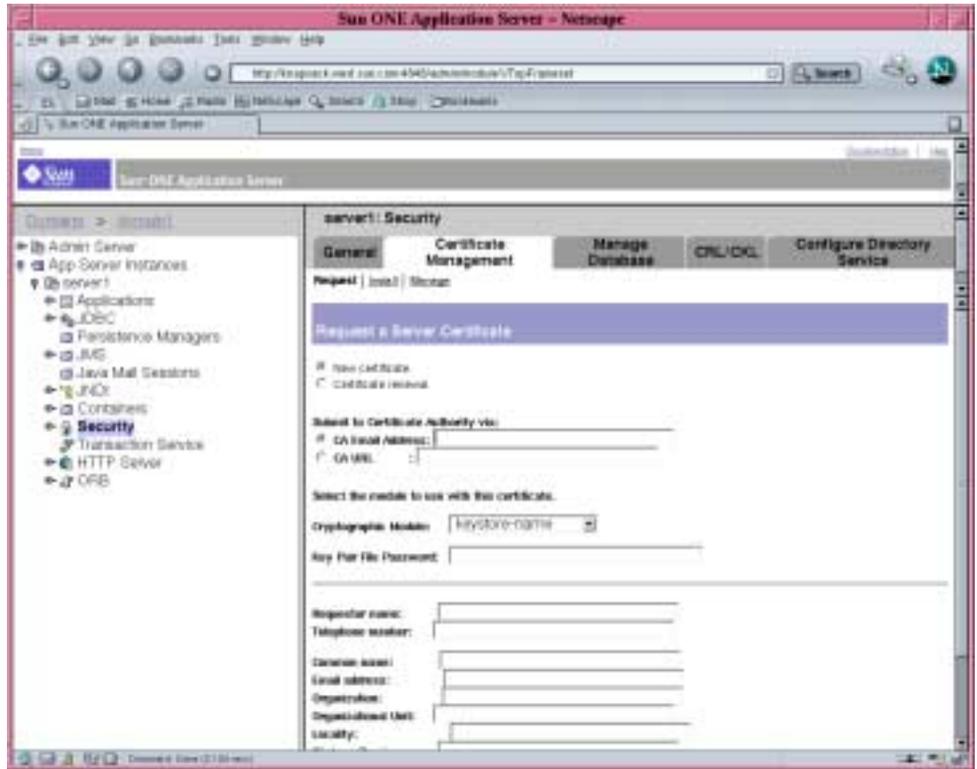


FIGURE 5-6 Sun ONE Application Server Administration Server Request a Server Certificate Dialog Box

2. Fill out the form to generate a certificate request, using the following information:

a. Select a new certificate.

If you can directly post your certificate request to a web-capable certificate authority or registration authority, select the CA URL link. Otherwise, select CA Email Address and enter an email address where you would like the certificate request to be sent.

b. Select the “Cryptographic Module” you want to use.

Each keystore has its own entry in this pull-down menu. Be sure that you select the correct keystore. Do not select “SUNW acceleration only.”

- c. In the “Key Pair File Password” dialog box, provide the password for the user that will own the key.

This password is the *username:password* (See TABLE 5-1).

- d. Type the appropriate information for the requestor information fields in TABLE 5-6.

TABLE 5-6 Requestor Information Fields

Field	Description
Requestor Name	Contact information for the requestor
Telephone Number	Contact information for the requestor
Common Name	Web site domain that is typed in a visitor’s browser
Email Address	Contact information for the requestor
Organization	Company name
Organizational Unit	(Optional) Department of the company
Locality	(Optional) City, county, principality, or country
State	(Optional) Full name of the state
Country	Two-letter ISO code for the country (for example, the United States is US)

- e. Click OK to submit the information.

3. Use a certificate authority to generate the certificate.

- If you chose to post your certificate request to a CA URL, the certificate request is automatically posted there.
- If you chose the CA Email Address, copy the certificate request that was emailed to you with the headers and hand it off to your certificate authority.

4. Once the certificate is generated, copy it, along with the headers, to the clipboard.

Note – The certificate is different from the certificate request and is usually presented to you in text form. Keep this data on the clipboard for Step 4 of “To Install the Server Certificate” on page 141.

▼ To Install the Server Certificate

1. Select the Install link in the right panel of the “Certificate Management” section of the Administration GUI.

You are now in the Install submenu window of the “Certificate Management” section of the Administration GUI.

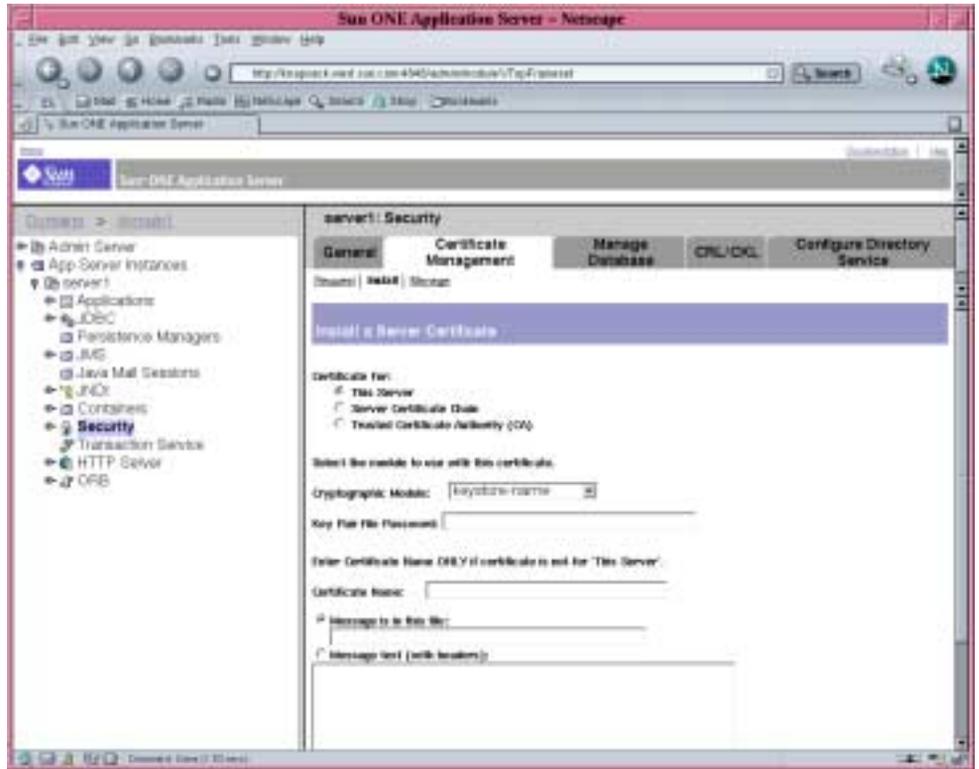


FIGURE 5-7 Sun ONE Application Server Administration Server Install a Server Certificate Dialog Box

2. Fill out the form to install your certificate:

TABLE 5-7 Fields for the Certificate to Install

Fields	Description
Certificate For	This server
Cryptographic Module	Each keystore has its own entry in this pull-down menu. Ensure you select the correct keystore name. To use the Sun Crypto Accelerator 4000 board, you must select the module with the same name that you chose when you requested the certificate.
Key Pair File Password	This password is the <i>username:password</i> .
Certificate Name	In most cases, you can leave this field blank. If you provide a name, it will alter the name the application server uses to access the certificate and key when running with SSL support. The default for this field is <i>Server-Cert</i> .

3. Select the Message text (with headers) radio button.

4. Click the “Message text (with headers):” radio button, and paste the certificate you copied from the certificate authority (in Step 4 of “To Generate a Server Certificate” on page 139) into the text box provided underneath the radio button.

5. Click OK.

You are shown some basic information about the certificate.

6. If everything looks correct, click “Add Server Certificate.”

You are prompted to restart the application server. Do not restart the application server yet, it will be restarted after SSL configuration is complete. You are also notified that in order for the application server to use SSL, the application server must be configured to do so.

▼ To Enable the Application Server for SSL

1. Type the following command in a terminal window.

You must also type the Sun ONE Application Server Administration Server password after executing this command.

Note – You can omit the `--host hostname --port administration-server-port` arguments if you are running the command on the local host, and if the Sun ONE Application Server Administration Server is configured to use the default port of 4848.

```
# installation-directory/bin/asadmin create-ssl --user app-admin --host
hostname --port administration-server-port --type http-listener --certname
keystore-name:server-certificate-name --instance server-name http-listener
password>
```

- 2. In the left panel of the Administration GUI, select the expander icon to the left of the HTTP Server link.**

The HTTP Server submenu items appear.

- 3. Select the “HTTP Listeners” submenu item under the “HTTP Server” link.**

4. In the right panel, select the HTTP listener that you wish to configure for SSL/TLS and select the associated link of the HTTP listener.

A window appears in which you can edit the properties for the HTTP listener.

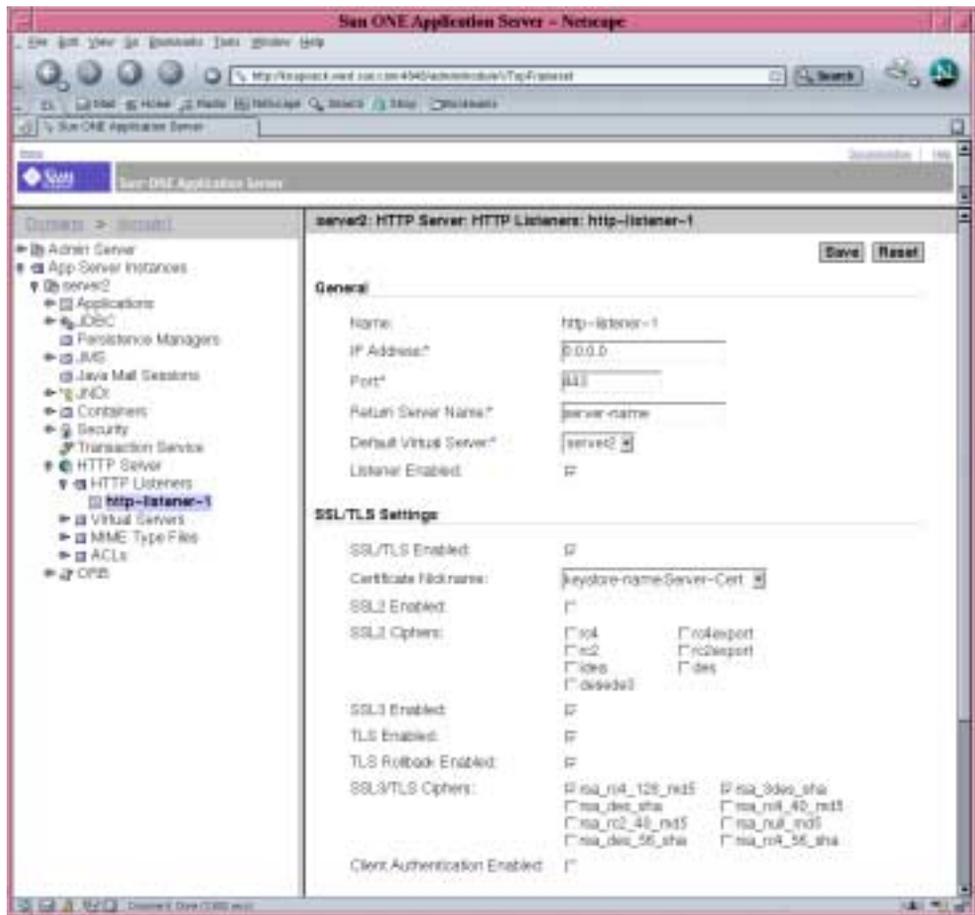


FIGURE 5-8 Sun ONE Application Server Administration Server HTTP Listener Properties Dialog Box

5. For the SSL/TLS Settings, verify the Certificate Nickname matches the certificate nickname you chose with the `--certname` option of the command in Step 1 of “To Enable the Application Server for SSL” on page 142.

6. Check the following boxes at minimum:

- SSL/TLS Enabled
- SSL3 Enabled
- TLS Enabled
- TLS Rollback Enabled

- SSL3/TLS Ciphers: `rsa_rc4_128_md5` and `rsa_3des_sha`
7. Set the port—this is typically 443.
 8. For rollback, TLS must also be enabled on the browser seeking access to your server.
 - For Netscape Navigator 6.0, check both TLS and SSL3.
 - For Microsoft Internet Explorer 5.0 and 5.5, use the TLS Rollback option.
 - For TLS Rollback, check TLS and make sure both SSL3 and SSL2 are disabled.
 9. Click Save.
 10. Select “App Server Instances” and select your server instance in the left panel, then select “Apply Changes” in the right panel.
 11. Stop and start the server to make the changes take effect.

The `init.conf` file is automatically modified to show security on, and all virtual servers are automatically assigned the default security parameters.

After you have enabled SSL on a server, its URLs use `https` instead of `http`. URLs that point to documents on an SSL-enabled server have the following format:

```
https://server-name.domain.dom:port-number
```

For example:

```
https://admin.sun.com:443
```

Note – If you use the default secure HTTP port number (443), you do not need to enter the port number in the URL.

Refer to the Enabling SSL/TLS section of the *Sun ONE Application Server 7 Administrator's Guide to Security* at:

<http://docs.sun.com/source/816-7158-10/sgencryp.html#14403>

Installing and Configuring Sun ONE Directory Server 5.2

This section describes how to install and configure Sun ONE Directory Server 5.2 to use the board. You must perform these procedures in order. Refer to the Sun ONE Directory Server documentation for more information about installing and using Sun ONE Directory Servers. This section includes the following procedures:

- “Installing Sun ONE Directory Server 5.2” on page 146
- “Configuring Sun ONE Directory Server 5.2” on page 147
- “To Create a Trust Database” on page 147
- “To Register the Board With the Directory Server (32-Bit)” on page 149
- “To Register the Board With the Directory Server (64-Bit)” on page 150
- “Generating and Installing a Server Certificate” on page 151
- “Viewing and Installing Root CA Certificates” on page 152
- “To Enable the Directory Server for SSL” on page 154

Installing Sun ONE Directory Server 5.2

This procedure installs the directory server software from the command-line.

▼ To Install Sun ONE Directory Server 5.2

1. Download the Sun ONE Directory Server 5.2 software.

You can find the directory server software at the following URL:
<http://www.sun.com/>

2. Change to the installation directory.

3. Execute the `./idsktune` command to ensure the recommended patches are installed.

4. Extract the directory server software.

5. Execute the `setup` script to install the software.

Note – There is no need to install individual packages because the `setup` script installs all of them.

After installation, the Sun ONE Directory Server and Administration Server start automatically.

To Start the Directory Server Manually

1. Change to the startup directory.

```
# cd /var/Sun/mps
```

2. Execute the `start-admin` command.

```
# ./start-admin
```

3. Change to the `slapd-servername` directory.

```
# cd slapd-servername
```

Where *servername* is the instance name.

4. Type the `start-slapd` command.

```
# ./start-slapd
```

Configuring Sun ONE Directory Server 5.2

These procedures create a trust database for the directory server instance; register the board with the directory server; generate and install a server certificate; view and install root CA certificates; and enable the directory server for SSL.

The configuration directory and the Sun ONE Directory Server Administration Server must be up and running during the configuration process.

▼ To Create a Trust Database

This procedure adds the Sun Crypto Accelerator 4000 module, and is the same for both 32-bit and 64-bit installations.

1. Start the directory server console.
2. Select the directory server instance you wish to configure and select **Open** in the main console window.
3. In the new window that appears, select **Console**→**Security**→**Manage Certificates**.

This step creates a trust database for the directory server instance.

- a. Select a password and place it in both boxes, then click OK (See FIGURE 5-9).
- b. Close the “Manage Certificates” dialog box that follows.

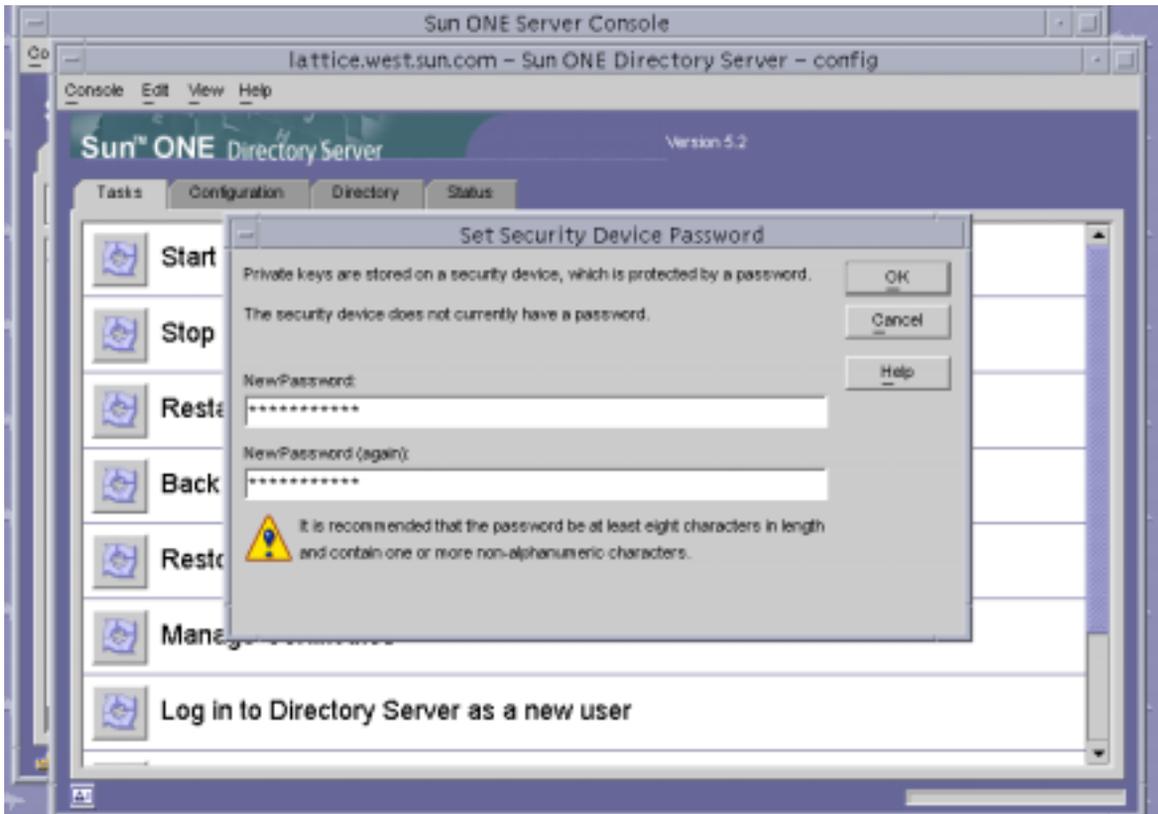


FIGURE 5-9 Sun ONE Directory Server Set Security Device Password Dialog Box

4. In the new window that pops up, select Console→Security→Configure Security Modules.
 - a. Click Install.
 - b. Type the following path in the *Enter the PKCS#11 module driver filename* entry:

```
/opt/SUNWconn/cryptov2/lib/libvpkcs11.so
```

5. Type a name in the *Enter an identifying name for this module* entry, for example:

Sun Crypto Accelerator 4000

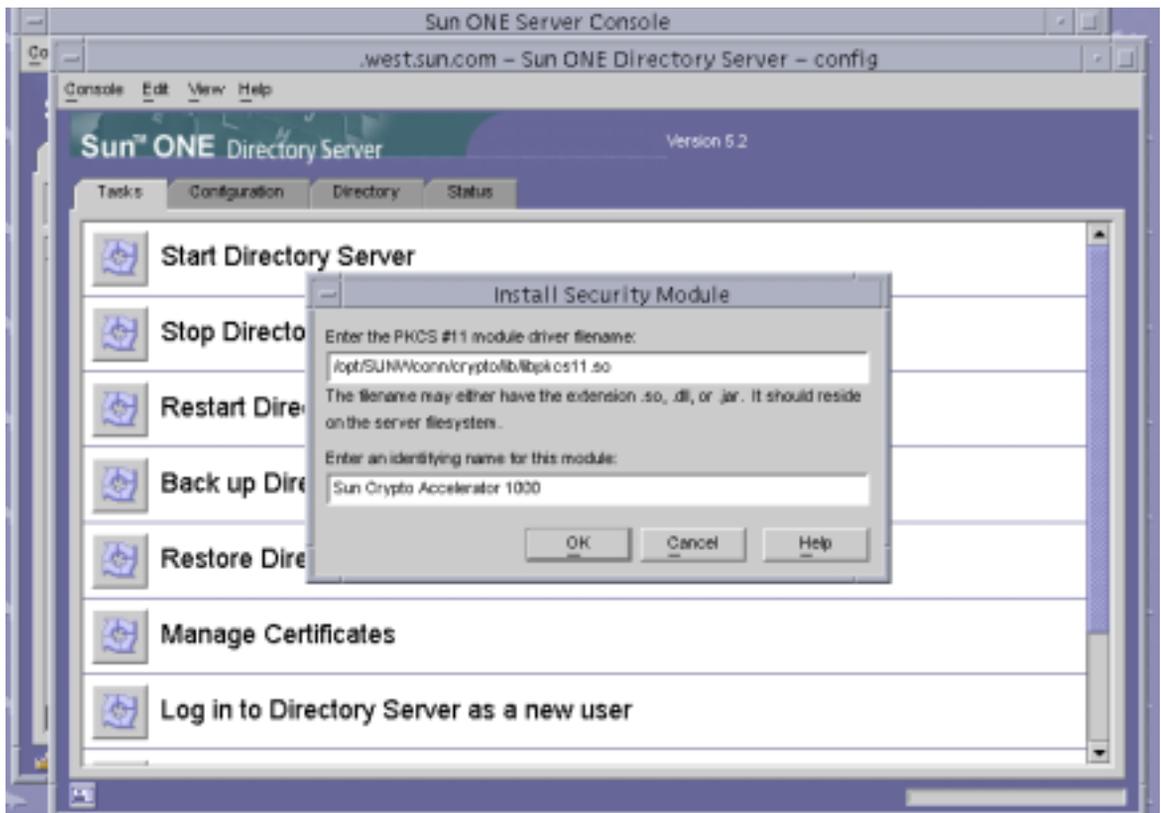


FIGURE 5-10 Sun ONE Directory Server Install Security Module Dialog Box

6. Click OK.

▼ To Register the Board With the Directory Server (32-Bit)

This procedure adds the 32-bit board module from the command-line.

1. Type the following command to set the appropriate path.

```
# setenv LD_LIBRARY_PATH server-inst/lib:${LD_LIBRARY_PATH}
```

2. Add the board to the `secmod.db` database.

a. Change to the following directory:

```
# cd server-inst/alias
```

b. Add the library with the `modutil` utility.

```
# server-inst/shared/bin/modutil -dbdir . -nocertdb -add "Sun Crypto Acclerator 4000" -libfile /opt/SUNWconn/cryptov2/lib/libvpkcs11.so
```

▼ To Register the Board With the Directory Server (64-Bit)

This procedure adds the 64-bit board module from the command-line.

1. Obtain the 64-bit versions of the Netscape Security Services (NSS) utilities from <http://www.mozilla.org>.

```
ftp://ftp.mozilla.org/pub/mozilla.org/security/nss/releases/NSS_3_3_2_RTM/SunOS5.8_64_OPT.OBJ/
```

Save the `nss-3.3.2.tar.gz` tar file.

2. Type the following command to set the appropriate path.

Note – Throughout this section *server-inst* refers to the root installation directory of the product, and *nss64-inst* refers to the location that you installed the 64-bit versions of the NSS tools.

```
# setenv LD_LIBRARY_PATH server-inst/lib/64:${LD_LIBRARY_PATH}
```

3. Add the board to the `secmod.db` database.

a. Change to the `alias` directory:

```
# cd server-inst/alias
```

b. Add the library.

```
# nss64-inst/bin/modutil -dbdir . -nocertdb -add "Sun Crypto Acclerator 4000" -libfile /opt/SUNWconn/cryptov2/lib/64/libvpkcs11.so
```

Generating and Installing a Server Certificate

With the exception of the different path variables described in TABLE 5-8, this procedure is the same for both 32-bit and 64-bit versions of the PKCS#11 library installed.

TABLE 5-8 32- and 64-Bit Path Variable Differences

Variable Definition	32-Bit	64-Bit
LD_LIBRARY_PATH	<i>server-inst/lib</i>	<i>server-inst/lib/64</i>
Location of the NSS tools	<i>server-inst/shared/bin</i>	<i>nss64-inst</i> (wherever you installed the NSS tools)

TABLE 5-9 describes the variables used for the `certutil` commands in this section.

TABLE 5-9 `certutil` Variable Descriptions

Variable	Descriptions
<i>token-name</i>	Name of the PKCS#11 token; this is the name of the keystore you chose when you initialized the board.
<i>subject-name</i>	Name asserted on the digital certificate, typically of the form: <i>CN=Fully-Qualified-Domain-Name, OU=Organization-Unit, O=Organization.</i> Names may vary with the organization.
<i>output-file</i>	Location for the certificate request.
<i>certfile</i>	Location for the ASCII-encoded certificate.
<i>instname</i>	Directory server instance name.
<i>nickname</i>	Server certificate friendly name chosen by the user.

▼ To Generate a Server Certificate

1. Change to the following directory.

```
# cd server-inst/alias
```

2. Request a certificate.

```
# certutil -R -d . -h token-name -s "subject-name" -a -o output-file [-g key-size] -P slapd-instname-
```

3. **Submit the certificate request in *output-file* to a Certificate Authority of your choice.**

Place the base64-encoded certificate in a text file named *certfile*.

▼ To Install the Server Certificate

1. **Install the server certificate.**

```
# certutil -A -d . -h token-name -t "Pu,Pu,Pu" -P slapd-instance- -a -i certfile -n  
nickname
```

Viewing and Installing Root CA Certificates

Sun ONE Directory Server includes several publicly known Root Certificate Authority certificates that are currently trusted. If your server certificate was issued by one of these well known Root CAs, skip this procedure.

▼ To View Root CA Certificates Known to the Directory Server

1. **From the directory server console window, open the directory server instance for the board.**
2. **From the menu at the top of the console window, select Console→Security→Manage Certificates**
3. **Select the CA Certs tab at the top of the “Manage Certificates” window.**

A list of CA certificates known to the Sun ONE Directory Server instance is displayed. You can view more detailed information about a given CA certificate by highlighting an entry and clicking the Detail button.

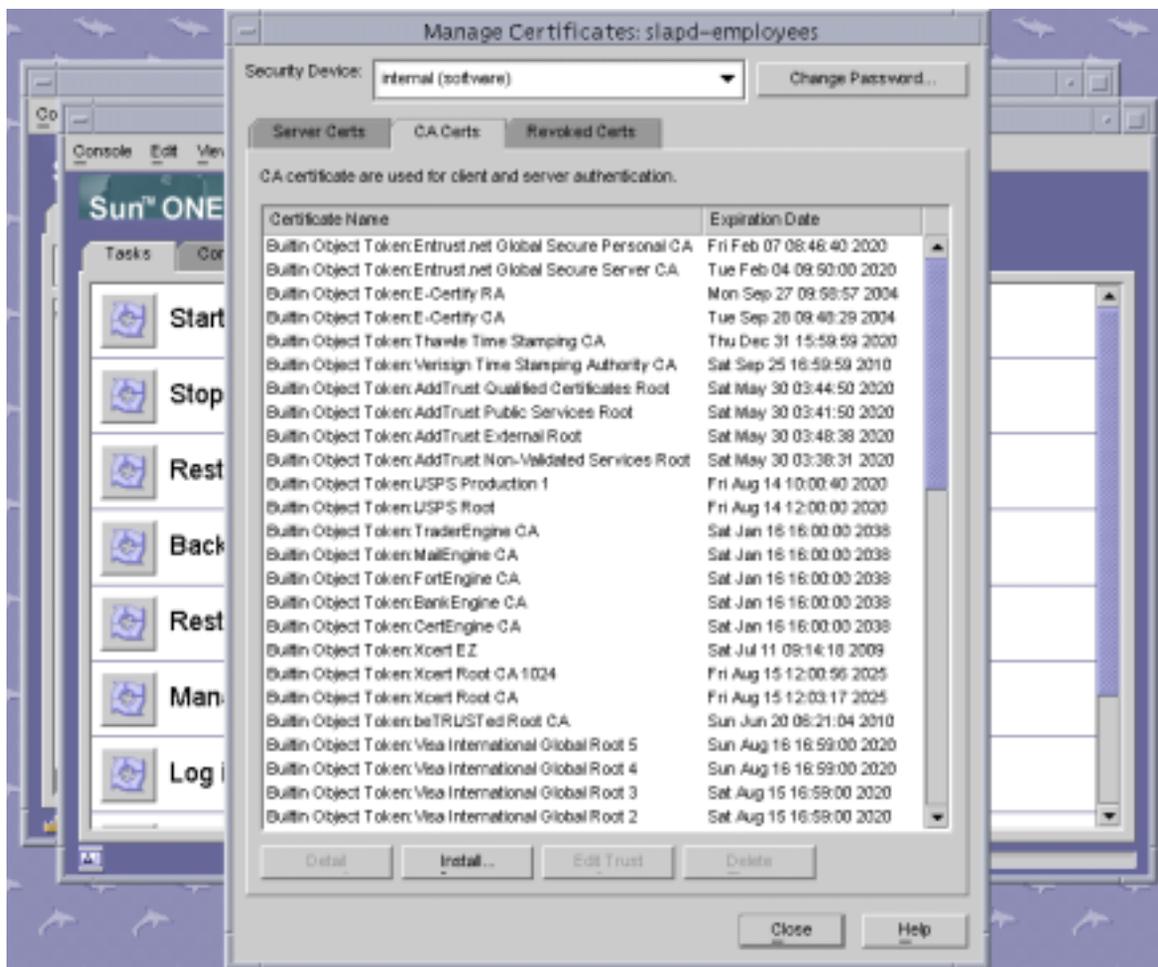


FIGURE 5-11 Sun ONE Directory Server Managing Certificates Dialog Box

▼ To Install Root CA Certificates

Perform the following procedure only if you retrieve your certificates from a proprietary PKI. That is, do not perform this procedure if you use VeriSign, Thawte, or GTE. This procedure is for cases where certificates issued by major vendors have an intermediate CA that has not been installed in the Sun ONE default trusted CA list.

1. Change to the `alias` directory.

```
# cd server-inst/alias
```

2. Install the root CA certificate.

Note – If you are installing more than one CA certificate, use different `-n` values. If you use the same `-n` value, the certificates overwrite each other. Replace `CA-Cert` with the CommonName component of the CA certificate's subject name (look for `CN=` in the SubjectName).

```
# certutil -A -d . -P slapd-instance- -n "CA-Cert" -t "CT,CT,CT" -a -i path-to-ca-cert
```

▼ To Enable the Directory Server for SSL

1. Start the directory server console if not started already.

```
# ./cd server-root  
# ./startconsole
```

2. Open the directory server instance by double-clicking the directory server instance of the board in the left panel of the main console window.

3. Click the Directory tab in the main console window.

4. Open the `cn=config` entry in the left panel of the Directory tab and modify the following parameters (See FIGURE 5-12):

a. Set `nsslapd-security` to **on**.

b. Set `nsslapd-secureport` to the desired port (default 636).

c. Click OK.

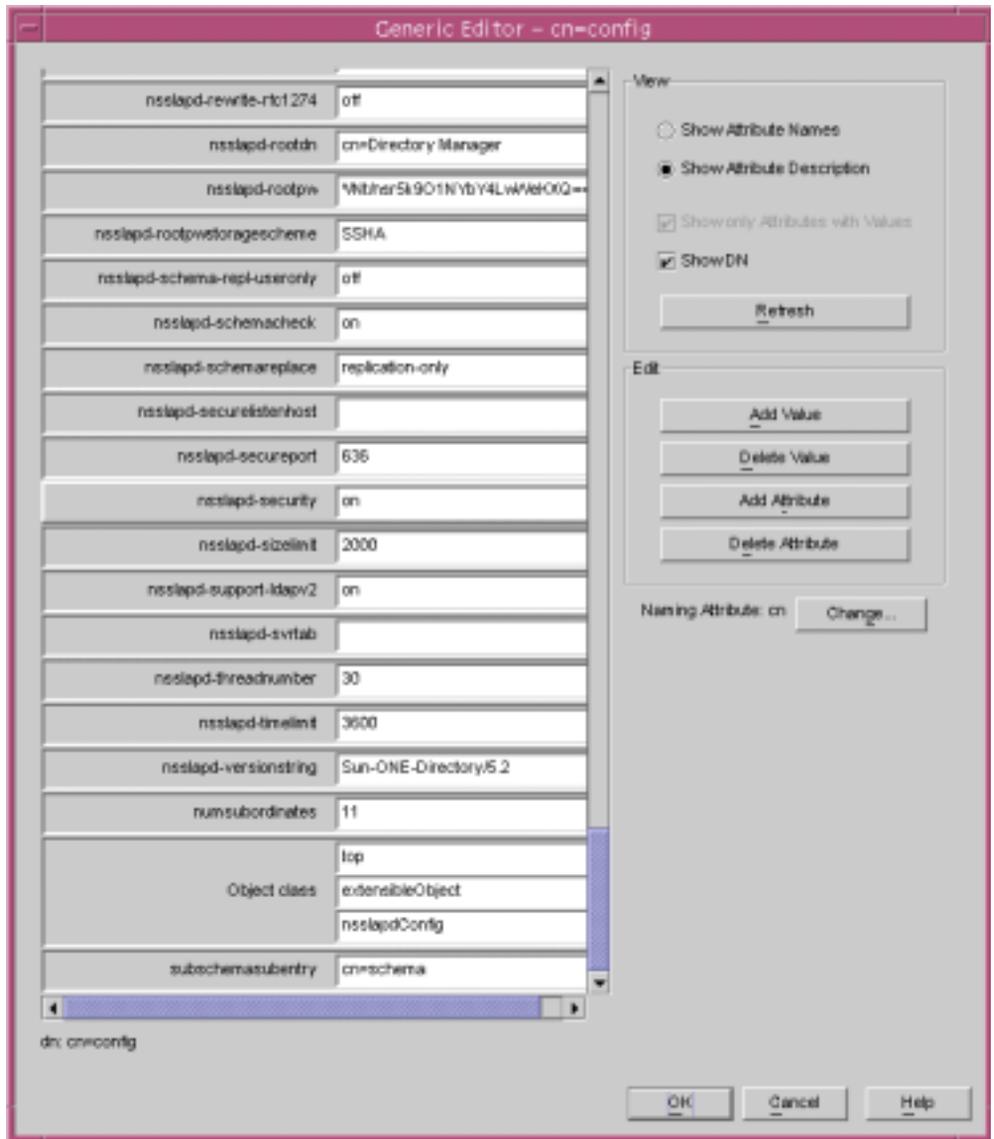


FIGURE 5-12 Sun ONE Directory Server cn=config Editor Dialog Box

5. Open the `cn=encryption,cn=config` entry in the left panel of the main console window and modify the following parameters (See FIGURE 5-13):

a. Set `nsssl3` to on.

- b. Use the “Add Attribute” button to add nsCertFile with a value of alias/slapd-*instance*-cert8.db
- c. Use the “Add Attribute” button to add nsKeyFile with a value of alias/slapd-*instance*-key3.db

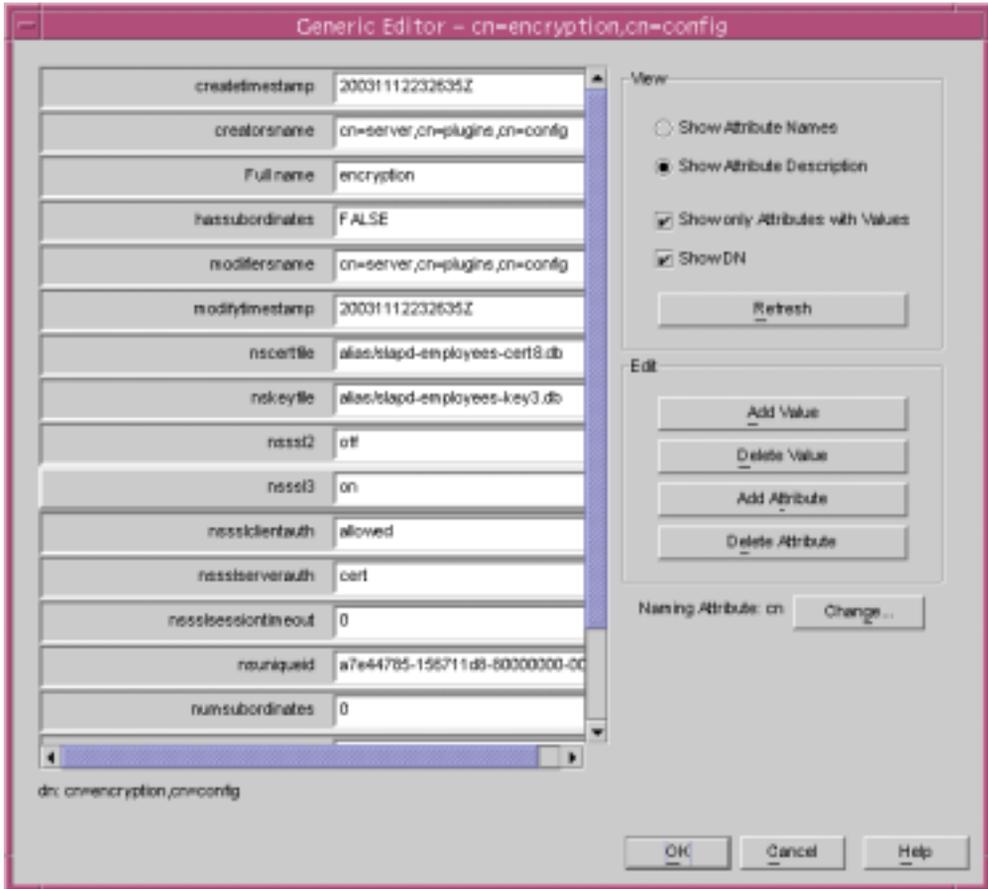


FIGURE 5-13 Sun ONE Directory Server cn=encryption,cn=config Dialog Box

- d. Click OK.
- 6. Create a new entry in the database under cn=encryption,cn=config
 - a. In the main window, right click on the encryption icon, and select New→Other from the menu.
 - b. Select nsEncryptionModule.

- c. Change the value of the “Full Name” attribute to “RSA” (Remote Security Access) from “New” (See FIGURE 5-14).

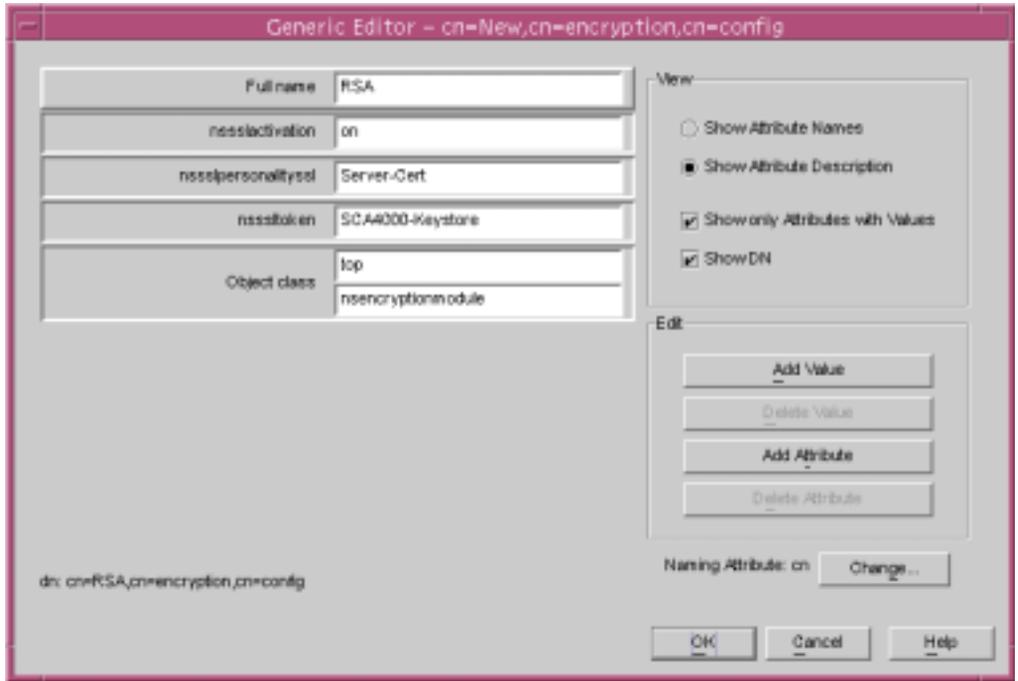


FIGURE 5-14 Sun ONE Directory Server nsEncryption Module Dialog Box

- d. Use the “Add Attribute” button to add the following attributes and values:

nssltoken	<i>token-name</i>
nsslpersonalityssl	<i>nickname</i>
nsslactivation	on

- e. Click OK.

Installing and Configuring Sun ONE Messaging Server 5.2

This section describes how to install and configure Sun ONE Messaging Server 5.2 to use the board. You must perform these procedures in order. Refer to the Sun ONE Messaging Server documentation for more information about installing and using Sun ONE Messaging Servers. This section addresses the following topics:

- “Installing Sun ONE Messaging Server 5.2” on page 158
- “Configuring Sun ONE Messaging Server 5.2” on page 158
- “To Create a Trust Database” on page 159
- “To Register the Board With the Messaging Server” on page 160
- “To Generate a Server Certificate” on page 160
- “To Install the Server Certificate” on page 165
- “To Enable the Messaging Server for SSL” on page 168

Installing Sun ONE Messaging Server 5.2

This procedure installs the Sun ONE Messaging Server 5.2 from the command-line.

▼ To Install Sun ONE Messaging Server 5.2

1. Download the Sun ONE Messaging Server 5.2 software.

You can find the messaging server software at the following URL:
<http://www.sun.com/>

2. Change to the installation directory and extract the messaging server software.

3. Install the messaging server software with the `setup` script.

- a. Type the install path when prompted.
- b. Type the components you wish to install when prompted.
- c. Execute the `./setup` command to install the components.

Configuring Sun ONE Messaging Server 5.2

These procedures create a trust database for the messaging server instance; register the board with the messaging server; generate and install a server certificate; and enable the messaging server for SSL.

The configuration directory and the Sun ONE Messaging Server Administration Server must be up and running during the configuration process.

▼ To Create a Trust Database

1. Start the messaging server console.
2. Open the Sun ONE Messaging server instance.

The menu in FIGURE 5-15 appears:

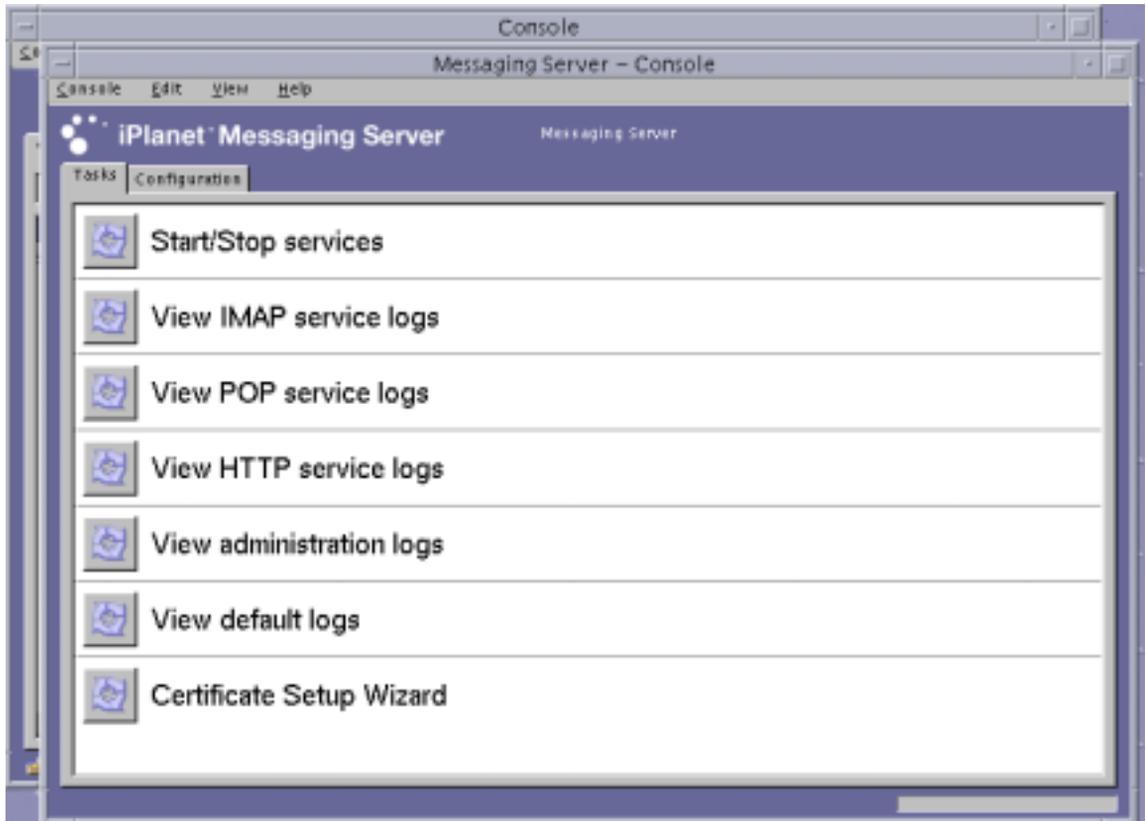


FIGURE 5-15 Sun ONE Messaging Server Main Console Window

3. Select Console→Certificate Setup Wizard

The Certificate Setup Wizard appears.

- a. Click Next.
- b. Select the “internal (software)” token.

- c. Select “Do not install a certificate” and click Next.
- d. Click Next.
- e. Set the password for the internal database and click Next.
- f. Click Done.

▼ To Register the Board With the Messaging Server

1. Change to the following directory.

```
# cd server-root/shared/bin
```

2. Ensure the LD_LIBRARY_PATH variable is set properly.

```
# setenv LD_LIBRARY_PATH server-root/lib:${LD_LIBRARY_PATH}
```

3. Add the board module to the secmod.db database.

```
# ./modutil -dbdir ../../admin-serv/config \  
-nocertdb \  
-add "Sun Crypto Accelerator 4000" \  
-libfile "/opt/SUNWconn/cryptov2/lib/libvpkcs11.so"
```

▼ To Generate a Server Certificate

1. Use the messaging server console to request a certificate by opening up the Certificate Setup Wizard; select Console -> Certificate Setup Wizard.
 - a. Click Next
 - b. Select the token that matches the Sun Crypto Accelerator 4000 token in which you want to store your keys, as shown in FIGURE 5-16.

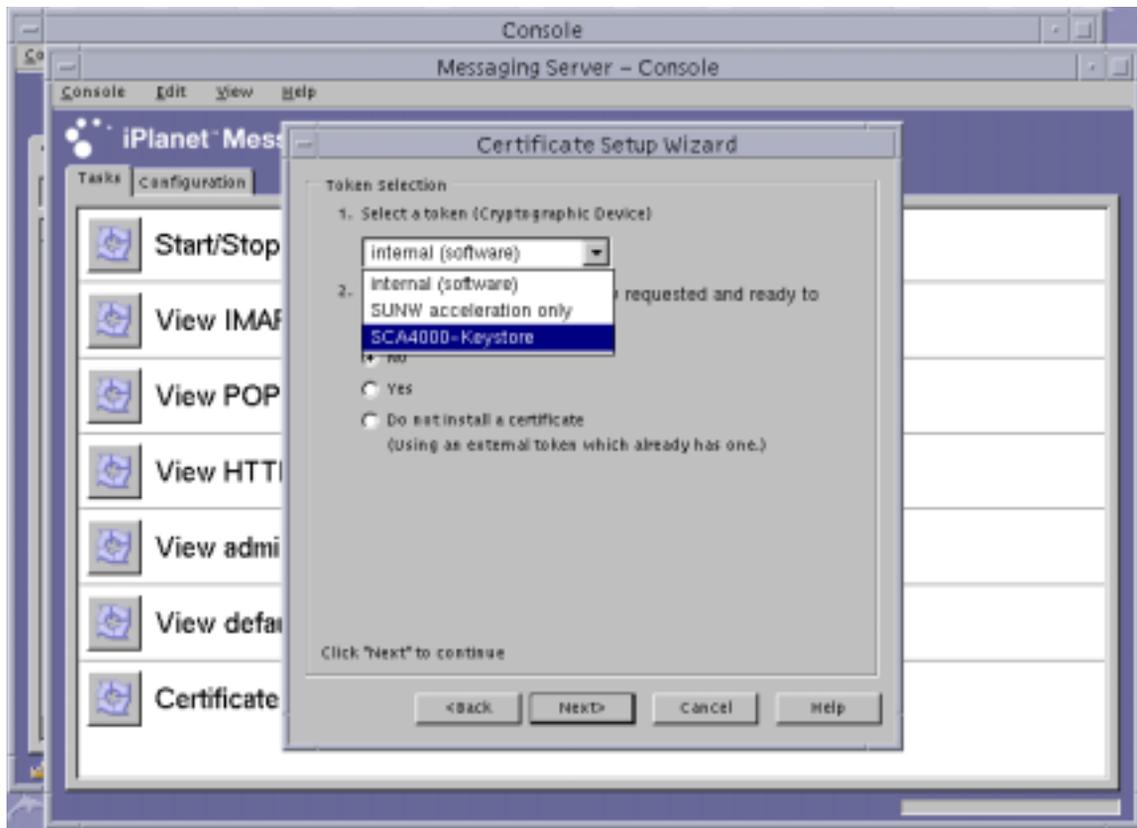


FIGURE 5-16 Sun ONE Messaging Server Certificate Setup Wizard Token Selection Dialog Box

- c. Answer No to “Is the certificate already requested and ready to install?” and click Next.
- d. Click Next.

- e. Select “New Certificate” and choose which method (either email or HTTPS) to submit the certificate request to a certificate authority (FIGURE 5-17), and click Next.

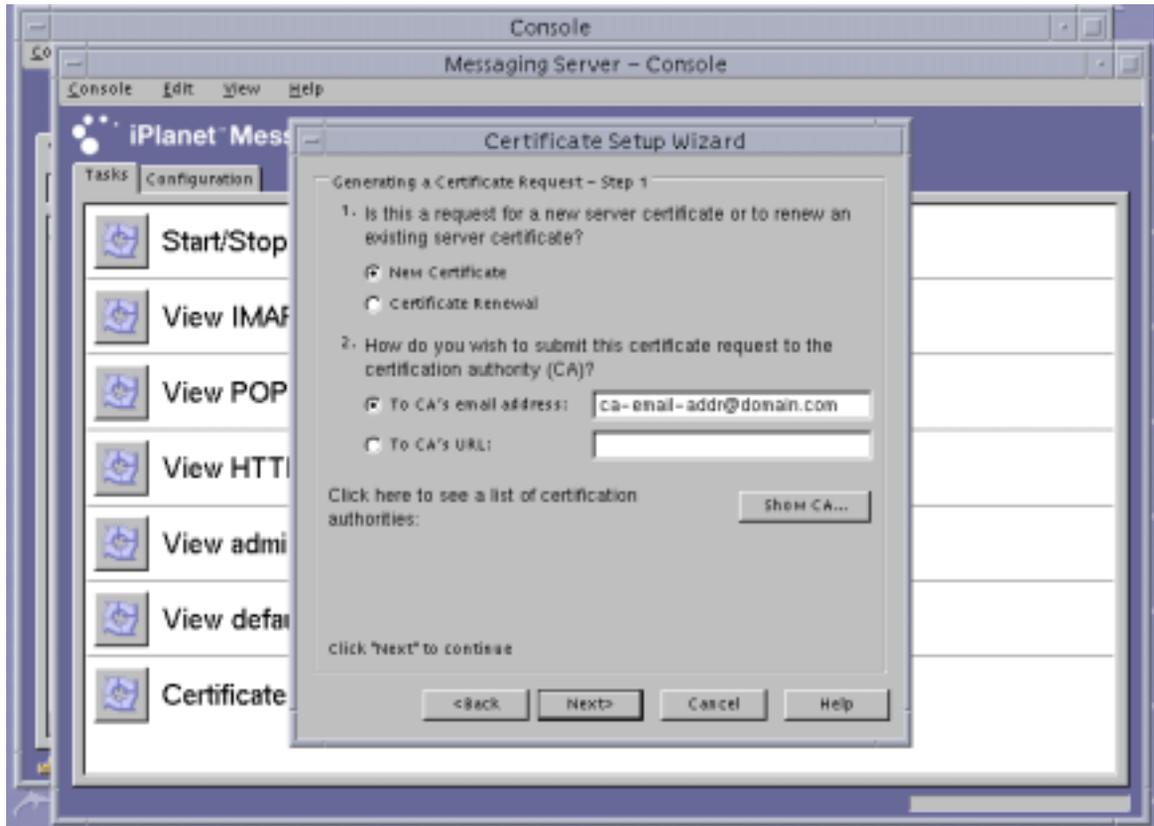


FIGURE 5-17 Sun ONE Messaging Server Certificate Setup Wizard Certificate Request Dialog Box

- f. Type the appropriate information for the requestor information fields in TABLE 5-10, and click Next.

TABLE 5-10 Requestor Information Fields

Field	Description
Requestor Name	Contact information for the requestor
Telephone Number	Contact information for the requestor
Common Name	Web site domain that is typed in a visitor's browser
Email Address	Contact information for the requestor

TABLE 5-10 Requestor Information Fields

Field	Description
Organization	Company name
Organizational Unit	(Optional) Department of the company
Locality	(Optional) City, county, principality, or country
State	(Optional) Full name of the state
Country	Two-letter ISO code for the country (for example, the United States is US)

g. The screen requests you to enter the password you used when creating a trust database. Instead, enter the password for the keystore user (*username:password*) and click Next.

See TABLE 5-1 for details on *username:password*.

- h. If you selected the HTTPS method in Step e, the request should already be sent to the CA. If you selected the email method in Step e, click “Copy to Clipboard” and click Next (FIGURE 5-18).

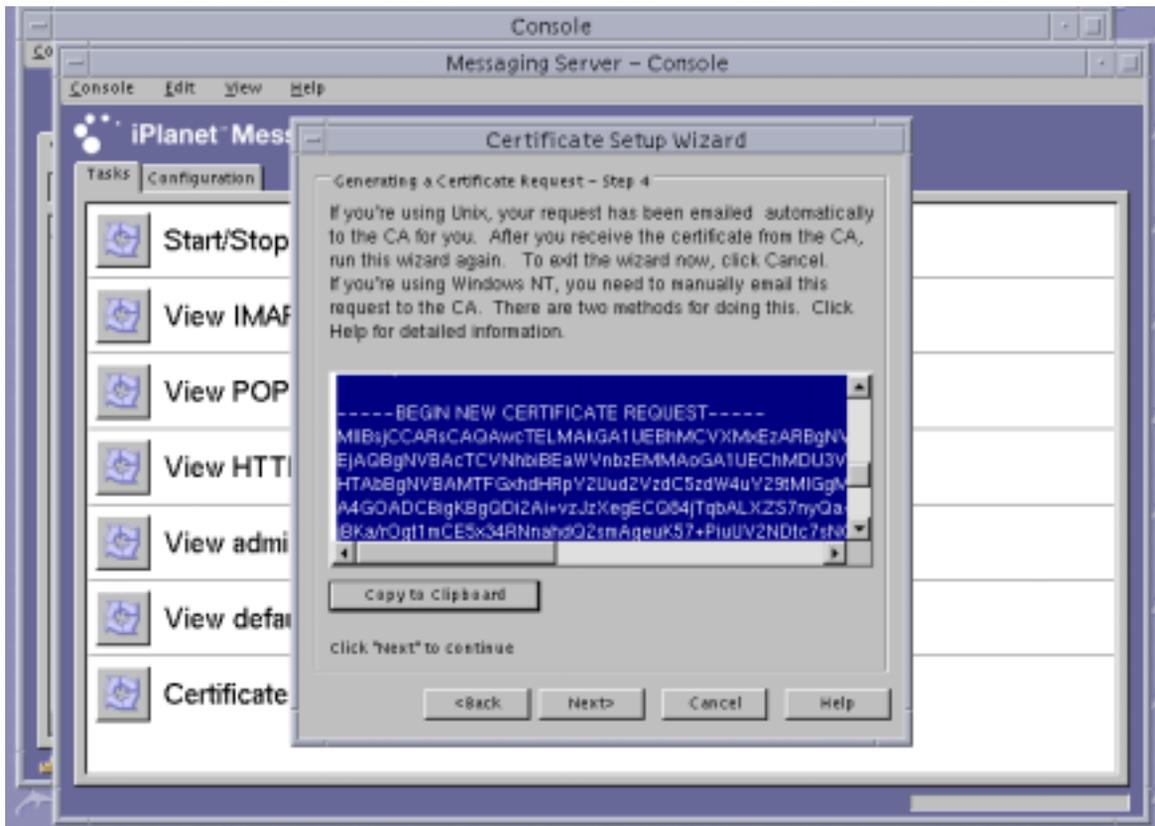


FIGURE 5-18 Sun ONE Messaging Server Certificate Setup Wizard Certificate Delivery Dialog Box

- i. Click Next.

Note – After requesting a certificate, the Certificate Setup Wizard will continue and allow you to install the issued certificate into the Sun Crypto Accelerator 4000 keystore. If you exited the Certificate Setup Wizard after the certificate was generated, but before it was installed, you can restart the Certificate Setup Wizard and pick up where you left off.

▼ To Install the Server Certificate

1. If you exited the Certificate Setup Wizard during the Generating a Server Certificate procedure, restart the Wizard by selecting Console -> Certificate Setup Wizard and click Next on the first screen.
2. Select the token that matches the Sun Crypto Accelerator 4000 token in which you want to install the certificate.
This token must be the same token from which you generated the request.
3. Answer Yes to the question that asks if the server certificate is ready to install, and click Next.
4. Click Next.
5. Install the certificate for “This Server,” and input the keystore password (*username:password*) if not already provided by the Wizard, and click Next (See FIGURE 5-19).

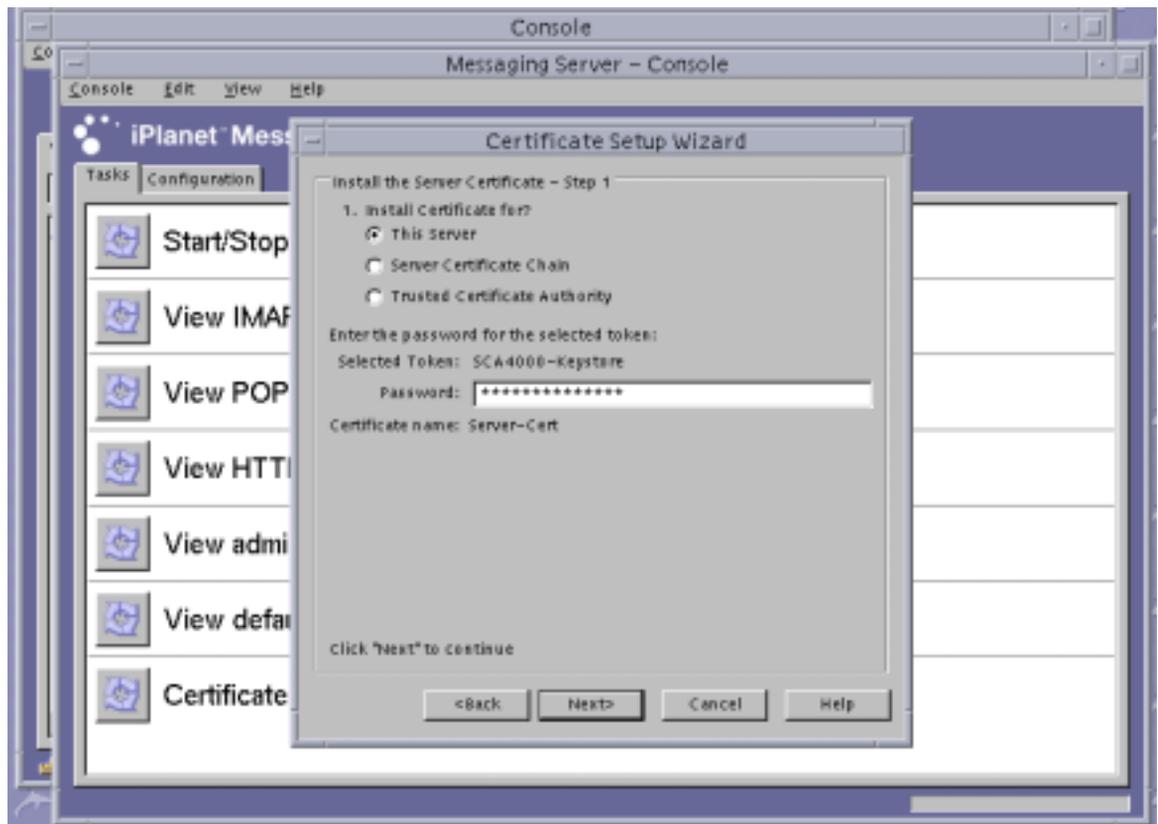


FIGURE 5-19 Sun ONE Messaging Server Certificate Setup Wizard Password Dialog Box

Note – The default certificate name is `Server-Cert`.

6. Copy the base 64-encoded certificate to the clipboard and paste it into the text box labeled “The certificate is located in the following text field,” and click Next (See FIGURE 5-20).

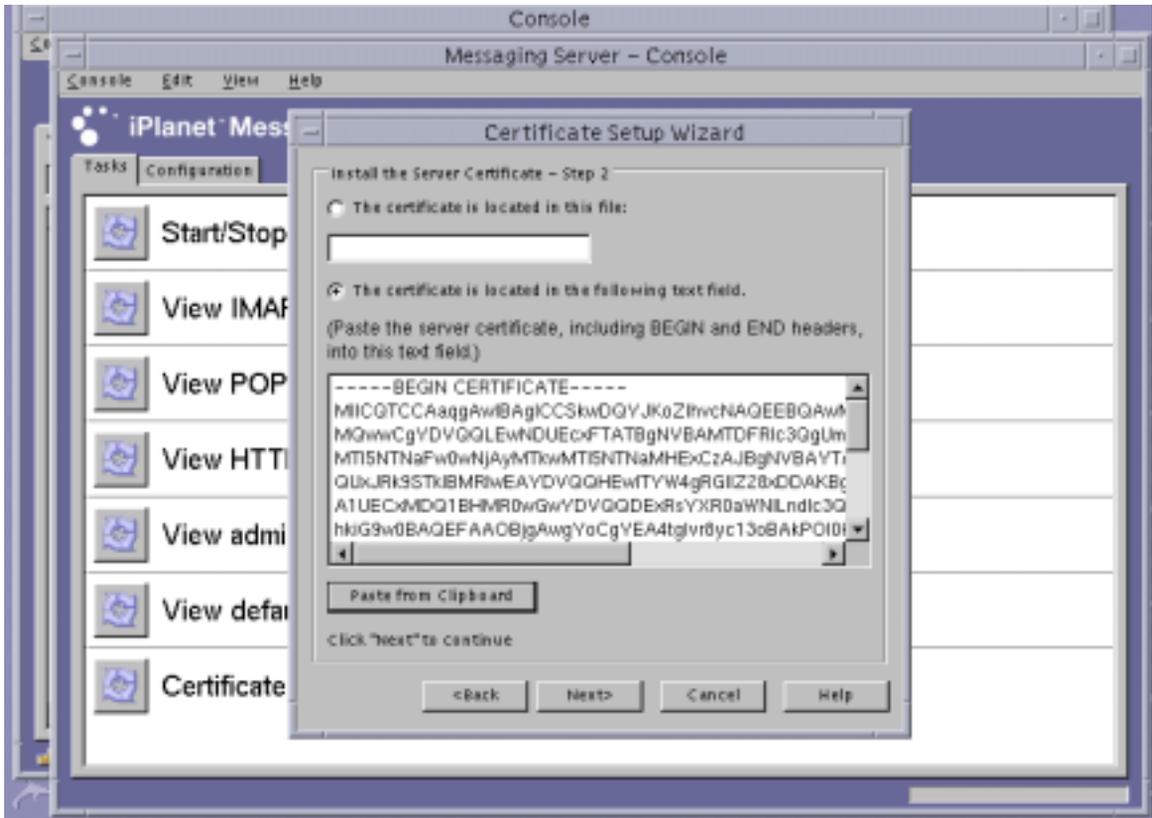


FIGURE 5-20 Sun ONE Messaging Server Certificate Setup Wizard Certificate Entry Dialog Box

- a. Click **Add** to add the certificate.
 - b. Click **Done**.
7. Add the root CA certificate (only if not from a root certificate authority already trusted by the messaging server).
Use the Certificate Setup Wizard for this step.
 - a. From the messaging server console, select **Console**→**Certificate Setup Wizard**.

- b. Click Next.
- c. Select "internal (software)" as the token and click Yes to "Is the certificate already requested and ready to install?" and click Next.
- d. Click Next.
- e. Select "Trusted Certificate Authority" and click Next.
- f. Copy the base 64-encoded CA certificate to the clipboard and paste it into the text box labeled "The certificate is located in the following text field," and click Next.
- g. Click Add to add the certificate (FIGURE 5-21).

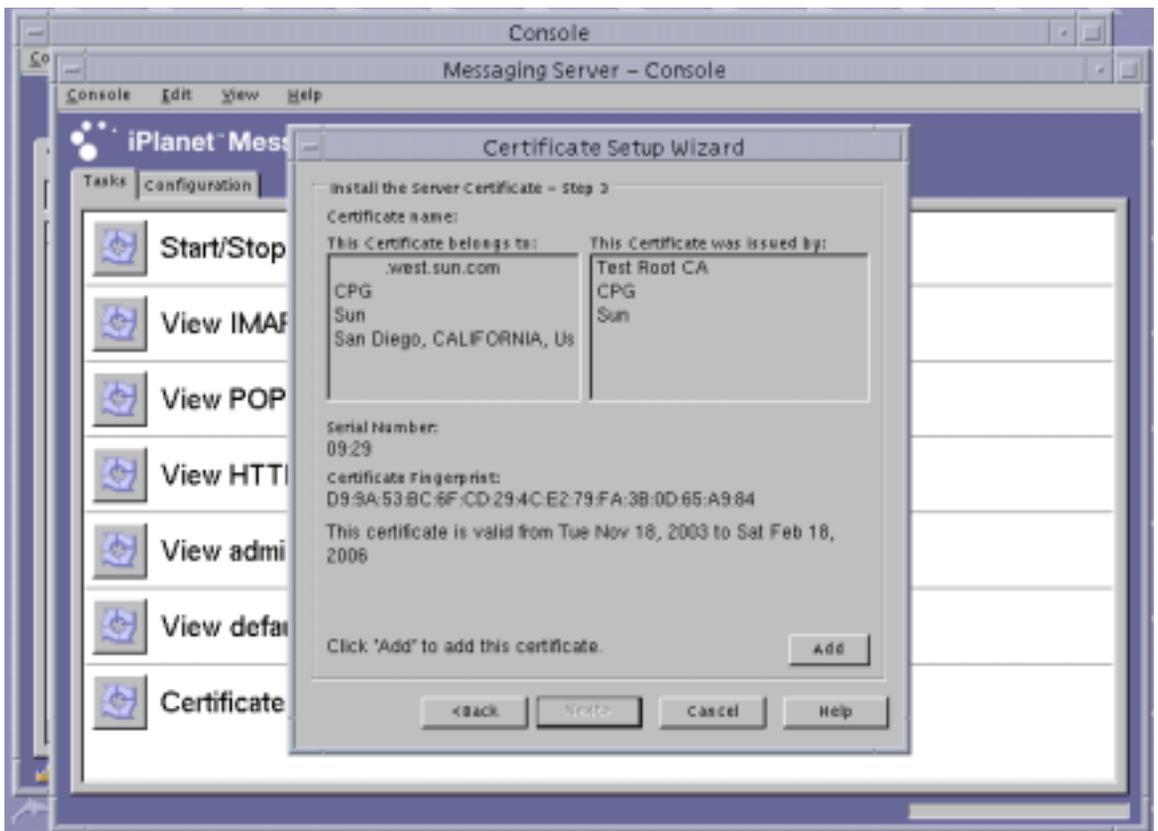


FIGURE 5-21 Sun ONE Messaging Server Certificate Setup Wizard Password Dialog Box

- h. Click Done.

▼ To Enable the Messaging Server for SSL

1. Use the `su` command to become the user for which you chose to run the messaging server.

If you do not remember this username, you can search the `server-root/msg-instname/config/msg.conf` file for the `local.serveruid` property and retrieve the username.

```
# cd server-root/msg-instname
# su username
```

2. Use the `configutil` tool to set SSL parameters for the messaging server.

TABLE 5-11 describes the variable definitions used with the `configutil` tool.

TABLE 5-11 `configutil` Variable Descriptions

Variable	Definition
<i>keystorename</i>	Name of the keystore used in Step 1.
<i>certname</i>	Friendly name of the certificate to be used. The default is <code>Server-Cert</code> .
<i>portnumber</i>	Port number to run POP3 over SSL; this is typically 995.

```
# ./configutil -o nssserversecurity -v on
# ./configutil -o encryption.rsa.nssslactivation -v on
# ./configutil -o encryption.rsa.nsssltoken -v keystorename
# ./configutil -o encryption.rsa.nssslpersonalityssl -v certname
# ./configutil -l -o service.pop.enablesslport -v yes
# ./configutil -l -o service.pop.sslport -v portnumber
```

3. In the messaging server console, click the **Configuration** tab for the console window used to administer the Sun ONE Messaging Server instance. Click the **System** tab under **Messaging Server** -> **Services** -> **IMAP**.
4. In the previous window, set the port number for “Use separate port for IMAP over SSL.” By default this port is 993.

5. Configure the `sslpassword.conf` file for the messaging server instance.

```
# cd server-root/msg-instname/config
# vi sslpassword.conf
```

Replace the `Internal (Software) token:netscape!` line with `tokenname:username:password`. Where `tokenname` is the keystore name. This `tokenname` is the name of the token on which you chose to generate the key in Step 1. The `username:password` is what you use to authenticate to that token. See TABLE 5-1 for details about `username:password`.

6. Change ownership and permissions for the `sslpassword.conf` file.

Because the `sslpassword.conf` file contains password information used to authenticate to key material, the file must be owned by the user for which the daemon runs, and that file must be readable by that user only.

```
# cd server-root/msg-instname/config
# chown msg-user sslpassword.conf
# chmod 0400 sslpassword.conf
```

7. Restart the server from the command line.

```
# cd server-root
# msg-instname/start-msg
```

Installing and Configuring Sun ONE Portal Server 6.2

This section describes how to install and configure Sun ONE Portal Server 6.2 to use the board. You must perform these procedures in order. Refer to the Sun ONE Portal Server documentation for more information about installing and using Sun ONE Portal Servers. This section includes the following procedures:

- “Installing Sun ONE Portal Server 6.2” on page 170
- “Configuring Sun ONE Portal Server 6.2” on page 171
- “To Register the Board With the Portal Server” on page 171
- “To Generate a Server Certificate” on page 117
- “To Install the Server Certificate” on page 120
- “To View Root CA Certificates Known to the Portal Server” on page 173

- “To Install Root CA Certificates” on page 173
- “To Enable the Portal Server for SSL” on page 174

This section describes how to install and configure Sun ONE Portal Server 6.2 to use the board. You must perform these procedures in order. Refer to the Sun ONE Portal Server documentation for more information about installing and using Sun ONE Portal Servers.

The Sun ONE Portal Server 6.2 includes Sun ONE Web Server 6.0. You must install and configure the Sun ONE Web Server software before installing and configuring the portal server (See “Installing and Configuring Sun ONE Web Server 6.0” on page 123).

Note – When installing and configuring the Sun ONE Web Server for use with the portal server, use the following installation path: `/opt/SUNWam/servers`.

Installing Sun ONE Portal Server 6.2

This section describes how to install the Sun ONE Portal Server 6.1 from the command-line.

▼ To Install Sun ONE Portal Server 6.2

1. Download the Sun ONE Portal Server 6.1 software.

You can find the portal server software at the following URL:
<http://www.sun.com/>

2. Change to the installation directory and extract the portal server software.

3. Install the portal server software with the `setup` script.

- a. Enter the install path when prompted.
- b. Enter the components you wish to install when prompted.
- c. Execute the `./setup` command to install the components.

Note – A trust database is automatically created during installation.

Configuring Sun ONE Portal Server 6.2

These procedures configure the portal server secure remote access (SRA) gateway; register the board with the portal server; generate and install a server certificate; and enable the portal server for SSL.

Before beginning, ensure that SRA has been installed and a gateway server certificate (self-signed or issued by any CA) has been installed. The Sun ONE Portal Server Administration Server must be up and running during the configuration process.

▼ To Register the Board With the Portal Server

1. **Create a new user account for the board with the `vcaadm` utility (see “Using the `vcaadm` Utility” on page 59).**

```
vcaadm{vca0@localhost, sec-officer}> create user
New user name: username
Enter new user password:
Confirm password:
User crypta created successfully.
```

2. **Load the Sun Crypto Accelerator 4000 module.**

The `LD_LIBRARY_PATH` variable must point to the following:

```
/usr/lib/mps/secv2/
```

- a. **Load the module.**

```
# /usr/bin/mps/modutil -dbdir /etc/opt/SUNWps/cert/default -add "Sun Crypto
Accelerator 4000" -libfile /opt/SUNWconn/cryptov2/lib/libvpkcs11.so
```

- b. **Verify that this module is loaded.**

```
# /usr/bin/mps/modutil -list -dbdir /etc/opt/SUNWps/cert/default -nocertdb
```

Generating and Installing a Server Certificate

During these procedures, the `LD_LIBRARY_PATH` environment variable must point to the following:

```
/usr/lib/mps/secv1/
```

TABLE 5-9 describes the variables used for the `certutil` commands in this section.

TABLE 5-12 `certutil` Variable Descriptions

Variable	Descriptions
<i>token-name</i>	Name of the PKCS#11 token; this is the name of the keystore you chose when you initialized the board.
<i>subject-name</i>	Name asserted on the digital certificate, typically of the form: <i>CN=Fully-Qualified-Domain-Name, OU=Organization-Unit, O=Organization.</i> Names may vary with the organization.
<i>output-file</i>	Location for the certificate request.
<i>certfile</i>	Location for the ASCII-encoded certificate.
<i>instname</i>	Portal server instance name.
<i>nickname</i>	Server certificate friendly name chosen by the user.

▼ To Generate a Server Certificate

1. Change to the following directory.

```
# cd /etc/opt/SUNWps/cert/default
```

2. Request a certificate.

```
# /usr/bin/mps/bin/certutil -R -d . -h token-name -s "subject-name" -a -o output-file  
[-g key-size]
```

3. Submit the certificate request in *output-file* to a Certificate Authority of your choice.

Place the base64-encoded certificate in a text file named *certfile*.

▼ To Install the Server Certificate

1. Install the server certificate.

```
# /usr/bin/mps/certutil -A -d . -h token-name -t "Pu,Pu,Pu" -a -i certfile -n nickname
```

Viewing and Installing Root CA Certificates

Sun ONE Portal Server includes several publicly known Root Certificate Authority certificates that are currently trusted. If your server certificate was issued by one of these well known Root CAs, skip this procedure.

▼ To View Root CA Certificates Known to the Portal Server

● Type the following command:

```
# /usr/bin/mps/certutil -L -d /etc/opt/SUNWps/cert/default
```

▼ To Install Root CA Certificates

Perform the following procedure only if you retrieve your certificates from a proprietary PKI. That is, do not perform this procedure if you use VeriSign, Thawte, or GTE. This procedure is for cases where certificates issued by major vendors have an intermediate CA that has not been installed in the Sun ONE default trusted CA list.

1. Change to the certificate database directory.

```
# cd /etc/opt/SUNWps/cert/default
```

2. Install the root CA certificate.

Note – If you are installing more than one CA certificate, use different `-n` values. If you use the same `-n` value, the certificates overwrite each other. Replace `CA-Cert` with the CommonName component of the CA certificate's subject name (look for `CN=` in the SubjectName).

```
# /usr/bin/mps/certutil -A -d . -n "CA-Cert" -t "CT,CT,CT" -a -i path-to-ca-cert
```

▼ To Enable the Portal Server for SSL

1. Create a `/etc/opt/SUNWps/cert/default/.nickname` file.

```
# vi /etc/opt/SUNWps/cert/default/.nickname
```

The file must contain only the following line with no spaces:

```
keystore-name: server-cert
```

2. Select the acceleration ciphers.

Note – The `/etc/opt/SUNWconn/cryptov2/sslreg` file must be present for the DES and 3DES algorithms to be accelerated in the Sun Crypto Accelerator 4000 hardware. See “Enabling and Disabling Bulk Encryption” on page 109.

The board accelerates RSA functions but supports acceleration only for DES and 3DES ciphers. To enable one of these ciphers do the following:

```
Gateway >> Security >> Enable SSL Cipher Selection: >> SSL3  
Ciphers: >>  
SSL3_RSA_WITH_3DES_EDE_CBC_SHA or  
SSL3_RSA_WITH_DES_CBC_SHA
```

3. Modify the `/etc/opt/SUNWps/platform.conf` *gateway-profile-name* to enable the board.

```
gateway.enable.accelerator=true
```

4. From a terminal window, restart the gateway.

```
# InstallDir/SUNWps/bin/gateway -n gateway-profile-name start
```

The gateway prompts you to enter the keystore password. Enter the password or pin for `sra-keystore:username:password`.

Installing and Configuring Apache Web Server Software

This chapter describes how to install and configure Apache Web Servers to use the board and includes the following sections:

- “Configuring Apache Web Server 1.3x” on page 176
- “Building and Configuring Apache Web Server 2.x” on page 182
- “Configuring the Apache Web Server to Start Up Without User Interaction on Reboot” on page 186
- “Configuring the Sun Crypto Accelerator 1000 for Use With Apache After the Sun Crypto Accelerator 4000 Software is Installed” on page 187

The following are the software requirements to configure Apache Web Server to use the board:

- Apache Web Server 1.3.26 or later—the 1.3.26 version is provided with the Sun Crypto Accelerator 4000 software
- Patch 109234-09 for Solaris 8 available from <http://sunsolve.sun.com>
- Patch 113146-02 for Solaris 9 available from <http://sunsolve.sun.com>
- SUNWkc12a package included with the Sun Crypto Accelerator 4000 software

Once the SUNWkc12a package is added, the system is configured with Apache Web Server and mod_ssl 1.3.26.

Note – Apache Web Servers do not use the keystore or user account features described in Chapter 5 “Concepts and Terminology” on page 106.



Caution – Do not configure Apache Web Server for use with the Sun Crypto Accelerator 1000 board and the Sun Crypto Accelerator 4000 board at the same time. Apache will not work correctly.

Note – The bulk encryption feature for Apache software is enabled by default and cannot be disabled.

Configuring Apache Web Server 1.3x

This section describes how to use the `apsslcfg` script to configure the web server to use the board. This section also describes how to create and install a server certificate.

▼ To Configure Apache Web Server

1. **Create an `httpd` configuration file if you have not already created one.**

For Solaris systems, the `httpd.conf-example` file is usually in the `/etc/apache` directory. You can use this file as a template and copy it as follows:

```
# cp /etc/apache/httpd.conf-example /etc/apache/httpd.conf
```

2. **Replace `ServerName` with your server name in the `httpd.conf` file.**
3. **Start `apsslcfg`.**

```
# /opt/SUNWconn/cryptov2/bin/apsslcfg
```

4. **Select 1 to configure your Apache Web Server to use SSL.**

Note – This procedure assumes that you choose option 1 at this prompt. If you want to choose option 2, refer to “Using the apsslcfg Script” on page 98.

```
Sun Crypto Accelerator Apache Installation
-----
This script will install the Sun Crypto Accelerator
cryptographic modules for Apache.

Please select what you wish to do:
-----
1. Configure Apache for SSL
2. Work with Apache keys

Your selection (0 to quit): 1
```

5. Type the path of the Apache binaries.

On Solaris systems, this path is typically `/usr/apache`.

```
Please enter the directory where the Apache
binaries and libraries exist [/usr/apache]: /usr/apache
```

6. Type the path for the Apache configuration files.

On Solaris systems, this path is typically `/etc/apache`.

```
Please enter the directory where the Apache configuration files exist
[/etc/apache]: /etc/apache
```

7. Create a remote security access (RSA) keypair for your system.

If you choose not to create a keypair, you must later use `apsslcfg` to generate one.

```
Do you wish to create a new RSA keypair and certificate request? [Y/N]: Y
```

If you answer no to this question, skip to “To Generate a Server Certificate” on page 178.

8. Provide the directory for storing the keys.

If this directory does not exist, it is created.

```
Where would you like the keys stored? [/etc/apache/keys]: /etc/apache/keys
```

9. Choose a base name for the key material.

This name is appended with different suffixes to distinguish key files, certificate request files, and certificate files from each other.

```
Please choose a base name for the key and request file: base-name
```

10. Provide a key length between 512 and 2048 bits.

For most web server applications, 1024 bits is sufficiently strong, but you can choose stronger keys if preferred.

```
What size would you like the RSA key to be [1024]? 1024
Using configuration from /opt/SUNWconn/cryptov2/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to /etc/apache/keys/base-name
```

11. Create your PEM pass phrase.

This pass phrase protects the key material. Be sure to select a strong pass phrase, but one that you can remember. If you forget the pass phrase, you will be unable to access your keys.

```
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
```



Caution – You must remember the pass phrase you enter. Without the pass phrase, you cannot access your keys. There is no way to retrieve a lost pass phrase.

▼ To Generate a Server Certificate

- 1. Create a certificate request using the keys you created in Step 7 of “To Configure Apache Web Server” on page 176.**

- a. **Type the password to access your keys. Then type the appropriate information for the requestor information fields.**

TABLE 6-1 provides a description of the requestor information fields.

```

Enter PEM pass phrase:
You are about to be asked to enter information that will be incorporated into
your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:US
State or Province Name (full name) [Some-State]:.
Locality Name (eg, city) []:.
Organization Name (eg, company) []: Company
Organizational Unit Name (eg, section) []: Department
SSL Server Name (eg, www.company.com) []:www.company.com
Email Address []:admin@company.com

```

TABLE 6-1 Requestor Information Fields

Field	Description
Country Name	Two-letter ISO code for the country (for example, the United States is US)
State or Province Name	(Optional) Full name of the state, or you may enter a dot (.)
Locality	City, county, principality, or country
Organization Name	Company name
Organizational Unit Name	Department of the company
SSL Server Name	Web site domain that is typed in a visitor's browser
Email Address	Contact information for the requestor

2. Modify the `/etc/apache/httpd.conf` file as directed.

Information regarding your key and certificate files, and instructions for how to modify the `/etc/apache/httpd.conf` file appears.

```
The keyfile is stored in /etc/apache/keys/base-name-key.pem.  
The certificate request is in /etc/apache/keys/base-name-certreq.pem.
```

You will need to edit `/etc/apache/httpd.conf` for the following items:

You must specify the ports that Apache will listen to for SSL connections, as well as for non-SSL connections. One way to accomplish this is to add the following lines in the Listen section:

```
Listen 80  
Listen 443
```

In the LoadModule section, add the following:

```
LoadModule ssl_module /usr/apache/libexec/mod_ssl.so.version-number
```

In the AddModule section, add the following:

```
AddModule mod_ssl.c
```

Note – The correct *version-number* will be displayed for your configuration.

3. If you chose not to set up a VirtualHost, you must place the SSLEngine, SSLCertificateFile, and SSLCertificateKeyFile directives in the httpd.conf file just above the SSLPassPhraseDialog directive.

You may need a virtual host directive similar to what is shown below:

```
<VirtualHost _default_:443>
    SSLEngine on
    SSLCertificateFile /etc/apache/keys/base-name-cert.pem
    SSLCertificateKeyFile /etc/apache/keys/base-name-key.pem
</VirtualHost>
```

You must add the following line after all of your VirtualHost definitions:

```
SSLPassPhraseDialog exec:/opt/SUNWconn/cryptov2/bin/apgetpass
```

Other SSL-related directives and their explanations can be found in the Sun Crypto Accelerator documentation.

Other Apache-related directives may need to be configured in order to start your Apache Web Server. Please refer to your Apache documentation.

<Press ENTER to continue>

If you answered no to the question in Step 7 of “Configuring Apache Web Server 2.x” on page 184, you are given additional information on how to generate key material.

Since you did not create keys, you will need to make sure that you have a key file and a certificate file in place before enabling SSL for Apache.

You can create a new key file and certificate request by selecting the "Generate a keypair and request a certificate for Apache" option after choosing "Work with Sun ONE and Apache keys" from the apsslcfg main menu.

4. Type 0 to quit when you finish with apsslcfg.

▼ To Install the Server Certificate

1. **Copy your certificate request with the headers from the `/etc/apache/keys/base-name-certreq.pem` file (where `base-name` was set in Step 9 of “To Configure Apache Web Server” on page 176), and transfer the certificate request to your certificate authority.**
2. **Once the certificate is generated, create the certificate file `/etc/apache/keys/base-name-cert.pem` and paste your certificate into the file.**
3. **Start the Apache Web Server.**

The following path assumes your Apache binary directory is `/usr/apache/bin`. If this is not your binary directory, type the correct path.

```
# /usr/apache/bin/apachectl sslstart
```

4. **Enter your PEM pass phrase when prompted.**
5. **Verify the new SSL-enabled web server with a browser at the following URL:**
`https://server-name:server-port/`
Note that the default `server-port` is 443.

Note – Refer to the `mod_ssl` and `OpenSSL` documentation for information on how to self-sign a certificate for testing.

Building and Configuring Apache Web Server 2.x

The Sun Crypto Accelerator 4000 software does not include a `mod_ssl` library for Apache 2.x Web Servers. This section describes the options you need to include when building the web server, and describes how to configure Apache 2.x to use the board.

Building Apache 2.x Web Server

To start this process, your OpenSSL implementation must have all of the required patches. This section covers only the board specific options, and is not an exhaustive set of instructions to build the entire Apache 2.x suite. For complete instructions, refer to the documentation available at <http://www.apache.org>.

▼ To Build Apache 2.x

1. Preset the `SH_LIBS` environment variable to comply with the `configure` script.

```
sh:
# SH_LIBS="-lssl -lcrypto"
# export SH_LIBS
csh/tcsh:
# setenv SH_LIBS "-lssl -lcrypto"
```

2. Change to the installation directory and execute the `configure` script.

This script has many command-line options, the following are required to configure the web server to use the board:

```
# ./configure --enable-ssl --enable-mods-shared=ssl
--with-ssl=/opt/SUNWconn/cryptov2
```

3. Once the script has finished, do one of the following:

- a. If you are building and installing Apache 2.x for the first time, type the following.

```
# make
# make install
```

- b. If you wish to build the `mod_ssl` shared library for an existing Apache 2.x Web Server, type the following:

```
# make shared-build
# cp modules/ssl/.libs/mod_ssl.so Apache-directory/modules
```

Configuring Apache Web Server 2.x

This section describes how to configure the web server to use the board by generating and installing a server certificate and enabling the web server for SSL.

▼ To Generate a Server Certificate

1. Generate a key and certificate request.

```
# /opt/SUNWconn/cryptov2/bin/openssl req \  
-new -newkey rsa:keysize -keyout key-output-file \  
-out cert-request-output-file \  
-config /opt/SUNWconn/cryptov2/ssl/openssl.cnf  
Generating a 1024 bit RSA private key  
.....  
.....+++++  
.....+++++  
writing new private key to '/tmp/key1.pem'
```

2. Type the password to protect the key file.

```
Enter PEM pass phrase:  
Verifying - Enter PEM pass phrase:
```

3. Type the “Distinguished Name” values (See TABLE 6-2).

```
You are about to be asked to enter information that will be  
incorporated into your certificate request.  
What you are about to enter is what is called a Distinguished Name  
or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.  
-----  
Country Name (2 letter code) [US]: US  
State or Province Name (full name) [Some-State]: California  
Locality Name (eg, city) []: San Diego  
Organization Name (eg, company) []: Company  
Organizational Unit Name (eg, section) []: Company Division  
SSL Server Name (eg, www.company.com) []: www.company.com  
Email Address []: admin@domain.com
```

TABLE 6-2 Distinguished Name Fields

Field	Description
Country Name	Two-letter ISO code for the country (for example, the United States is US)
State or Province Name	(Optional) Full name of the state, or you may enter a dot (.)
Locality Name	(Optional) City, county, principality, or country
Organization Name	Company name
Organizational Unit Name	(Optional) Department of the company
SSL Server Name	Web site domain that is typed in a visitor's browser
Email Address	Contact information for the requestor

▼ To Install the Server Certificate

- **Copy your certificate request with the headers into the same directory where your key file was created in Step 1 of “To Generate a Server Certificate” on page 184.**

▼ To Enable SSL

1. **Edit the `ssl.conf` file in the `conf` subdirectory of the Apache 2.x Web Server installation directory.**

There are several directives in the `ssl.conf` file; the following directives must be configured for the web server to use the board.

```
Listen port-number
ServerName fully-qualified-domain-name
SSLEngine on
SSLCertificateFile path-to-certificate-file
SSLCertificateKeyFile path-to-key-file
```

2. **Start the Apache Web Server.**

This assumes your Apache binary directory is `/usr/apache/bin`. If this is not your binary directory, type the correct directory.

```
# /usr/apache/bin/apachectl sslstart
```

3. **Enter your PEM pass phrase when prompted for it.**

4. Verify the new SSL-enabled web server with a browser by going to the following URL:

`https://server-name:server-port/`

The default *server-port* is 443.

Note – Refer to the `mod_ssl` and OpenSSL documentation for information on how to self-sign a certificate for testing.

Configuring the Apache Web Server to Start Up Without User Interaction on Reboot

You can enable the Apache Web Server to perform an unattended startup at reboot with an encrypted key.

▼ To Create an Encrypted Key for Automatic Startup of Apache Web Server on Reboot

1. Verify that the following entry exists in the `httpd.conf` file:

```
SSLPassPhraseDialog exec:/opt/SUNWconn/cryptov2/bin/apgetpass
```

This directive retrieves a password from a protected password file in the `/etc/apache` directory.

2. Create a password file that contains only the password in the `/etc/apache` directory with the following filename convention:

```
server-name:port.KEYTYPE.pass
```

- *server-name* – The value that you placed in the `ServerName` directive in the `httpd.conf` file
- *port* – The port on which this SSL server runs (for example, 443)
- *KEYTYPE* – Either `RSA` or `DSA`

Example: For a server named `webserv101` running SSL on port 443 with an RSA key, create the following file in `/etc/apache`:

```
webserv101:443.RSA.pass
```

Change the permissions and ownership of the password file as follows:

```
# chmod 400 server-name:port.KEYTYPE.pass  
# chown root server-name:port.KEYTYPE.pass
```

Refer to the `mod_ssl` and OpenSSL documentation for more information.

Configuring the Sun Crypto Accelerator 1000 for Use With Apache After the Sun Crypto Accelerator 4000 Software is Installed

Once the `SUNWkc12a` software package is installed, the system is configured with Apache Web Server `mod_ssl` 1.3.26.

If you want to configure the Sun Crypto Accelerator 1000 board with Apache, you must have the following patches.

To configure the Sun Crypto Accelerator 1000 for use with Apache 1.3.26 on a Solaris 8 system with the `SUNWkc12a` package installed, you need the following patches:

- For Apache 1.3.26 – Patch ID 109234-09 or later
- For Sun Crypto Accelerator 1000 version 1.0 software – Patch ID 112869-02
- For Sun Crypto Accelerator 1000 version 1.1 software – Patch ID 113355-01

To configure the Sun Crypto Accelerator 1000 for use with Apache 1.3.26 on a Solaris 9 system with the `SUNWkc12a` package installed, you need the following patches:

- For Apache 1.3.26 – Patch ID 113146-01 or later
- For Sun Crypto Accelerator 1000 version 1.1 software – Patch ID 113355-01

Diagnostics and Troubleshooting

This chapter describes diagnostic tests and troubleshooting for the Sun Crypto Accelerator 4000 software. This chapter includes the following sections:

- “SunVTS Diagnostic Software” on page 189
- “Using kstat to Determine Cryptographic Activity” on page 198
- “Using the OpenBoot PROM FCode Self-Test” on page 199
- “Troubleshooting the Sun Crypto Accelerator 4000 Board” on page 202

SunVTS Diagnostic Software

The core SunVTS wrapper provides test control and a user interface to a suite of tests. Some of those tests are delivered with packages `SUNWvts` and `SUNWvtsx` to make up a bundle that is contained on the Solaris 8/9 Software Supplement CD. Other, unbundled, tests that use the SunVTS core are packaged with the driver software of the device tested.

The Sun Crypto Accelerator 4000 board can be tested by three SunVTS tests. Two of those tests, `nettest` and `netlbttest`, are bundled with the core SunVTS software beginning with the release of SunVTS 5.1 Patch Set (PS) 2. These tests operate on the Ethernet circuitry of the board.

The third SunVTS test, `vcatest`, is delivered in the `SUNWvcav` package on the Sun Crypto Accelerator 4000 CD and operates with the core SunVTS wrapper to provide diagnostics of the cryptographic circuitry of the board.

Installing SunVTS netlbttest and nettest Support for the vca Driver

TABLE 7-1 shows the method of updating installed SunVTS software to provide SunVTS netlbttest and nettest support for the vca driver.

TABLE 7-1 SunVTS netlbttest and nettest Required Software for the vca Driver

Base Solaris Software	Base SunVTS Software	Required Replacement Package	Required Overlay Patch
Solaris 8 7/01	SunVTS4.4		111854-04
Solaris 8 10/01	SunVTS4.5		112250-04
Solaris 8 2/02	SunVTS4.6	SunVTS5.1ps2	
Solaris 9 5/02	SunVTS5.0	SunVTS5.1ps2	
Solaris 9 9/02	SunVTS5.1		113614-11
Solaris 8 HW 12/02	SunVTS5.1ps1		113614-11
Solaris 9 12/02	SunVTS5.1ps1		113614-11
Solaris 8 HW 5/03	SunVTS5.1ps2		
Solaris 9 4/03	SunVTS5.1ps2		

SunVTS software is delivered on the Solaris Software Supplement CD that is distributed with each Solaris release. The version of SunVTS software listed in the Base SunVTS Software column of TABLE 7-1 is distributed on the Solaris Software Supplement CD included in the Solaris release identified on the same line.

Entries in TABLE 7-1 that begin with “SunVTS” identify the version of a set of SunVTS packages. Within each SunVTS package set, the `SUNWvts` and `SUNWvtsx` packages must be installed.

The Required Replacement Packages column in TABLE 7-1 lists the SunVTS package sets that must replace the previously installed SunVTS package set. Remove the previously installed SunVTS packages before adding the SunVTS replacement packages. The previously installed SunVTS packages must be removed by the same method you used to install them. For example, if you used the `pkgadd` command to install the packages, use the `pkgrm` command to remove the packages.

If an entry is shown in the Required Overlay Patch column in TABLE 7-1, use the `patchadd` command to install that patch over the SunVTS packages shown in the Base SunVTS Software column. Do not remove the previously installed SunVTS packages before adding the required patch.

Using the `patchadd` command to install patch 113614-11 is the equivalent of replacing the previously installed SunVTS packages with the SunVTS5.1ps2 packages.

The replacement packages are available at:
<http://www.sun.com/oem/products/vts/>

The overlay patches are available at:
<http://sunsolve.sun.com/>

Note – The required SunVTS packages and any required patches must be installed before the `SUNWvcav` package is installed. The `SUNWvcav` package contains the SunVTS test `vcatest`.

Using SunVTS Software to Perform `vcatest`, `nettest`, and `netlbttest`

Refer to the SunVTS test reference manual, user's guide, and quick reference card for instructions on how to perform and monitor these diagnostics tests. These documents are available on the Solaris on Sun Hardware Documentation Set at <http://docs.sun.com>. These documents are also provided on the Solaris Software Supplement CD that is distributed with the Solaris release on your system.

Note – SunVTS can be used only if you have installed the required SunVTS packages and any required SunVTS patches.

▼ To Perform `vcatest`

1. As superuser, start SunVTS.

```
# /opt/SUNWvts/bin/sunvts
```

Refer to the SunVTS user's guide for detailed instructions on starting SunVTS. The following instructions assume that you have started SunVTS using the CDE user interface.

2. On the SunVTS Diagnostic main window, set the System Map to Logical mode.

Note – Physical mode is supported; however, this procedure assumes you are using Logical mode.

3. **Disable all tests by clearing their check boxes.**
4. **Select the check box for Cryptography, then select the plus box for Cryptography to display all tests in the Cryptography group.**
5. **Clear check boxes in the Cryptography group that are not named `vcatest`.**
 - If a `vcatest` is displayed, then go to Step 6.
 - If a `vcatest` is not displayed, probe the system to find it by selecting Reprobe system in the Commands drop-down menu.

Refer to the SunVTS user's guide for the exact procedure. When the probe completes and a `vcatest` is displayed, continue to Step 6.
6. **Select one of the instances of `vcatest` then right-click and drag to display the Test Parameter Options dialog box.**

These options, which only pertain to the `vcatest`, are described in "Test Parameter Options for `vcatest`" on page 193.
7. **After you have made all your selections, click Apply from the Within Instance drop-down menu to change the selected instance of `vcatest`, or select Apply from the Across All Instances drop-down menu to change all checked instances of `vcatest`.**

This action removes the dialog box and returns you to the SunVTS Diagnostic main window.
8. **Select one of the instances of `vcatest` then right-click and drag to display the Test Execution Options dialog box.**

An alternate method of displaying Test Execution Options dialog box is to select the Options drop-down main menu; then select Test Executions. These options are generic SunVTS controls that affect all tests. Refer to the SunVTS user's guide for detailed information.
9. **When you have made all selections, select Apply to remove the dialog box and return to the SunVTS Diagnostic main window.**
10. **Click Start to perform the selected tests.**
11. **Click Stop to stop all tests.**

Test Parameter Options for `vcatest`

TABLE 7-2 describes the `vcatest` subtests.

TABLE 7-2 `vcatest` Subtests

Test Name	Description
CDMF	Tests CDMF bulk encryption
DES	Tests DES bulk encryption
3DES	Tests 3DES bulk encryption
RSA	Tests RSA public and private keys
DSA	Tests DSA signature verification
MD5	Tests MD5 Message Digest/Digital Signature
SHA1	Tests SHA1 Digest Key Creation
RNG	Tests random number generation

`vcatest` Command-Line Syntax

To perform `vcatest` from the command line instead of the CDE interface, specify all arguments in the command-line string.

In 32-bit mode, the path to `vcatest` is `/opt/SUNWvts/bin/`. In 64-bit mode, the path is `/opt/SUNWvts/bin/sparcv9/`.

All SunVTS standard options are supported from the command-line interface for `vcatest`. Test-specific options are specified with the `-o` argument.

Refer to the SunVTS test reference manual for a definition of the standard command-line arguments. The `vcatest` is a Functional mode test; therefore, `-f` must be included. Include `-u` to display a usage message, or `-v` for VERBOSE messages. Items enclosed in square brackets denote optional entries.

The following is an example of invoking `vcatest` in 32-bit mode as a standalone program. The following command performs all subtests on `vca0`:

```
# /opt/SUNWvts/bin/vcatest -f -o dev=vca0,t1=all
```

The following is an example of invoking `vcatest` in 64-bit mode from the SunVTS infrastructure. The following command tests RSA, DSA, and MD5 on `vca2`:

```
# /opt/SUNWvts/bin/sparcv9/vcatest -f -o dev=vca2,t1=RSA+DSA+MD5
```

When performing `vcatest` from the command line, omission of an option produces the default behavior for that option, as stated in TABLE 7-3.

TABLE 7-3 `vcatest` Command-Line Syntax

Option	Description
<code>dev=vcaN</code>	Specifies the instance of the device to test such as <code>vca0</code> or <code>vca2</code> . Defaults to <code>vca0</code> if not included. Note that <i>N</i> specifies the placement of the instance number of the device being tested.
<code>t1=testlist</code>	Specifies the list of subtests to be performed. The subtests for <code>t1</code> are separated by the + (plus) character. The supported subtests are CDMF, DES, 3DES, DSA, RSA, MD5, SHA1, and RNG, so <code>t1=CDMF+DES+3DES+DSA+RSA+MD5+SHA1+RNG</code> enables all subtests. You can also insert <code>t1=all</code> which performs all tests. Defaults to <code>all</code> if no subtests are specified.

▼ To Perform `netlbttest`

1. As superuser, start SunVTS.

```
# /opt/SUNWvts/bin/sunvts
```

Refer to the SunVTS user's guide for detailed startup instructions.

The following instructions assume that SunVTS was started using the CDE user interface.

2. On the SunVTS Diagnostic main window, set the System Map to Logical mode.

Note – Physical mode is also supported; however, this procedure assumes you are using Logical mode.

3. Disable all tests by clearing their check boxes.

4. Select the check box for Network, then select the plus box for Network to display all tests in the Network group.

5. **Clear check boxes in the Network group that are not named `vcaN(netlbttest)`.**
Note that *N* specifies the placement of the instance number of the device under test.
 - If a `vcaN(netlbttest)` is displayed, go to Step 6.
 - If a `vcaN(netlbttest)` is not displayed, probe the system to find it by selecting **Reprobe system** in the **Commands** drop-down menu.

Refer to the SunVTS user's guide for the exact procedure. When the probe completes and a `vcaN(netlbttest)` is displayed, continue to Step 6.

6. **Select the Intervention Mode button. Select one of the instances of `vcaN(netlbttest)`, then right-click and drag to display the Test Parameter Options dialog box.**

These options, which only pertain to `netlbttest`, are described in the SunVTS test reference manual.

7. **After you have made all selections, select **Apply** from the **Within Instance** drop-down menu to change the selected instance of `vcaN(netlbttest)`, or select **Apply** from the **Across All Instances** drop-down menu to change all checked instances of `vcaN(netlbttest)`.**

This action removes the dialog box and returns you to the SunVTS Diagnostic main window.

8. **Select one of the instances of `vcaN(netlbttest)` then right-click and drag to display the Test Execution Options dialog box.**

An alternate method of displaying the Test Execution Options dialog box is to select the **Options** drop-down main menu; then select **Test Executions**. These options are generic SunVTS controls that affect all tests. Refer to the SunVTS user's guide for detailed information.

9. **When you have made all selections, select **Apply** to remove the dialog box the return to the SunVTS Diagnostic main window.**
10. **Click **Start** to perform the selected tests.**
11. **Click **Stop** to stop all tests.**

▼ To Perform `nettest`

1. **As superuser, start SunVTS.**

```
# /opt/SUNWvts/bin/sunvts
```

Refer to the SunVTS user's guide for detailed startup instructions.

Note – The following instructions assume that SunVTS was started using the CDE user interface.

2. On the SunVTS Diagnostic main window, set the System Map to Logical mode.

Note – Physical mode is also supported; however, this procedure assumes you are using Logical mode.

3. Disable all tests by clearing their check boxes.

4. Select the check box for Network, then select the plus box for Network to display all tests in the Network group.

5. Clear check boxes in the Network group that are not named `vcaN(nettest)`.

Note that *N* specifies the placement of the instance number of the device under test.

- If a `vcaN(nettest)` is displayed, then go to Step 6.
- If a `vcaN(nettest)` is not displayed, enter `ifconfig -a` in another window on the server containing the `vcaN` board. There should be an entry listed as follows:

```
vcaN up inet ip-address plumb
```

If the preceding `ifconfig` entry is not listed, the `nettest` probe does not consider the device testable. Follow the `ifconfig` online manual page instructions for bringing an interface online.

Once the `ifconfig -a` produces the preceding entry, return to the SunVTS Diagnostic main window and probe the system to find `vca` by selecting Reprobe system in the Commands drop-down menu.

Refer to the SunVTS user's guide for the exact procedure. When the probe completes and a `vca0(nettest)` is displayed, continue to Step 6.

6. Select one of the instances of `vcaN(nettest)`, then right-click and drag to display the Test Parameter Options dialog box.

These options, which only pertain to `nettest`, are described in the SunVTS test reference manual.

7. After you have made all selections, select Apply from Within Instance drop-down menu to change the selected instance of `vcaN(nettest)`, or select Apply from the Across All Instances drop-down menu to change all checked instances of `vcaN(nettest)`.

This action removes the dialog box and returns you to the SunVTS Diagnostic main window.

- 8. Select one of the instances of `vcaN(nettest)`, then right-click and drag to display the Test Execution Options dialog box.**

An alternate method of displaying the Test Execution Options dialog box is to select the Options drop-down main menu; then select Test Executions. These options are generic SunVTS controls that affect all tests. Refer to the SunVTS user's guide for detailed information.

- 9. When you have made all selections, select Apply to remove the dialog box, then return to the SunVTS Diagnostic main window.**
- 10. Click Start to perform the selected tests.**
- 11. Click Stop to stop all tests.**

Note – Do not select `nettest` and `netlbttest` to be performed simultaneously.

Using `kstat` to Determine Cryptographic Activity

The Sun Crypto Accelerator 4000 board does not contain lights or other indicators to reflect cryptographic activity on the board. To determine whether cryptographic work requests are being performed on the board, use the `kstat(1M)` command to display the device usage:

```
# kstat vca:0
module: vca                instance: 0
name:   vca0                class:   misc
       3desbytes            3040
       3desjobs             5
       crtime               65.342725895
       dsassign             0
       dsverify             0
       rngbytes            10592
       rngjobs             187
       rngshalbytes        16328
       rngshaljobs         327
       rsaprivate          9
       rsapublic           0
       snaptime            106956.467004482
```

Note – In the previous example, 0 is the instance number of the `vca` device. This number should reflect the instance number of the board for which you are performing the `kstat` command.

Displaying the `kstat` information indicates whether cryptographic requests or “jobs” are being sent to the Sun Crypto Accelerator 4000 board. A change in the *jobs* values over time indicates that the board is accelerating cryptographic work requests sent to the Sun Crypto Accelerator 4000 board. If cryptographic work requests are not being sent to the board, verify your web server configuration per the web server specific configuration.

Do not attempt to interpret the kernel/driver statistic values returned by `kstat(1M)`. These values are maintained within the driver to facilitate field support. The meanings and actual names may change over time.

Note – If the `nostats` property is defined in the `/kernel/drv/vca.conf` file, the capture and display of statistics will be disabled. This property can be used to help prevent traffic analysis.

Using the OpenBoot PROM FCode Self-Test

The following tests are available to help identify problems with the adapter if the system does not boot.

You can invoke the FCode self-test diagnostics by using the `test` or `test-all` commands from the OpenBoot PROM `ok` prompt. If you encounter an error while performing diagnostics, appropriate messages will be displayed. Refer to the *OpenBoot Command Reference Manual* for more information on the `test` and `test-all` commands.

The FCode self-test exercises most functionality subsection by subsection and ensures the following:

- Connectivity during adapter board installation
- Verification that all components required for a system boot are functional

▼ Performing the Ethernet FCode Self-Test Diagnostic

To perform the Ethernet diagnostics, you must first bring the system to a stop at the OpenBoot PROM `ok` prompt after issuing a reset. If you do not reset the system, the diagnostic tests might cause the system to hang.

For more information about the OpenBoot commands in this section, refer to the *OpenBoot Command Reference Manual*.

1. Shut down the system.

Use the standard shutdown procedures described in the *Solaris Handbook for Sun Peripherals*.

2. At the OpenBoot PROM ok prompt, set the auto-boot? configuration variable to false.

```
ok setenv auto-boot? false
```

3. Reset the system.

```
ok reset-all
```

4. Type show-nets to display the list of devices and enter a selection:

You see a list of devices, similar to the example below, specific to the adapter:

```
ok show-nets
a) /pci@8,600000/network@1
b) /pci@8,700000/network@5,1
q) NO SELECTION
Enter Selection, q to quit: a
/pci@8,600000/network@1 has been selected.
Type ^Y ( Control-Y ) to insert it in the command line.
e.g. ok nvalias mydev ^Y for creating devalias mydev for
/pci@8,600000/network@1
```

Note – To perform the following self-test with the test command, the Ethernet port must be connected to a network.

5. Perform the self-test using the test command:

The following tests are performed when the test command is executed:

- vca register test (happens only when diag-switch? is true)
- Internal loopback test
- Link up/down test

Note – The Sun Crypto Accelerator 4000 UTP adapter self-test for a 1000 Mbps connection is not supported for use with an external loopback cable because the link-clock cannot be reconciled. For this test, the local and remote ports must reconcile as clock master and clock slave. If an external loopback cable is used, both the local and remote ports are identical. So, the single port cannot be both a clock master and a clock slave, because this causes the PHY link-up to fail. For a Sun Crypto Accelerator 4000 UTP adapter self-test for a 1000 Mbps connection to work, a remote 1000BASE-T port must be connected.

Type the following:

```
ok test device-path
```

If the test passes, you see the following messages:

```
ok test /pci@8,600000/network@1
Testing /pci@8,600000/network@1
Register tests: passed
Internal loopback test: passed
/pci@8,600000/network@1: 100 Mbps half duplex link up
```

If the board is not connected to a network, you see the following messages:

```
ok test /pci@8,600000/network@1
Testing /pci@8,600000/network@1
Register tests: passed
Internal loopback test: passed
/pci@8,600000/network@1: link down
```

6. After testing the adapter, type the following to return the OpenBoot PROM ok prompt interface to standard operating mode:

```
ok setenv diag-switch? false
```

7. Set the auto-boot? configuration parameter to true.

```
ok setenv auto-boot? true
```

8. Reset and reboot the system.

Troubleshooting the Sun Crypto Accelerator 4000 Board

This section describes the commands available at the OpenBoot PROM level for troubleshooting the board. Refer to the *OpenBoot Command Reference Manual* for more information on the commands described in the following subsections.

show-devs

To determine whether the Sun Crypto Accelerator 4000 device is listed in the system: from the OpenBoot PROM ok prompt, type `show-devs` to display the list of devices. You see lines in the list of devices, similar to the examples below, specific to the board:

```
ok show-devs
.
.
/chosen
/packages
/upa@8,480000/SUNW,ffb@0,0
/pci@8,600000/network@1
/pci@8,600000/SUNW,qlc@4
/pci@8,600000/SUNW,qlc@4/fp@0,0
.
.
```

In the preceding example, the `/pci@8,600000/network@1` entry identifies the device path to the board. There will be one such line for each board in the system.

.properties

To determine whether the Sun Crypto Accelerator 4000 device properties are listed correctly: from the ok prompt, type `.properties` to display the list of properties.

```
ok .properties
assigned-addresses      82000810 00000000 00102000 00000000 00002000
                        81000814 00000000 00000400 00000000 00000100
                        82000818 00000000 00200000 00000000 00200000
                        82000830 00000000 00400000 00000000 00100000
d-fru-len               00 00 00 00
d-fru-off               00 00 e8 00
d-fru-dev               eeprom
s-fru-len               00 00 08 00
s-fru-off               00 00 e0 00
s-fru-dev               eeprom
compatible              70 63 69 38 30 38 36 2c 62 35 35 35 2e 31 30 38
reg                     00000800 00000000 00000000 00000000 00000000
                        02000810 00000000 00000000 00000000 00002000
                        02000814 00000000 00000000 00000000 00000100
                        02000818 00000000 00000000 00000000 00200000
                        02000830 00000000 00000000 00000000 00100000
address-bits            00 00 00 30
max-frame-size          00 00 40 00
network-interface-type  ethernet
device-type             network
name                   network
local-mac-address       00 03 ba 0e 99 ca
version                 Sun PCI Crypto Accelerator 4000 1000Base-T
FCode 2.11.13 03/03/04
phy-type                mif
board-model             501-6039
model                  SUNW,pci-vca
fcode-rom-offset        00000000
66mhz-capable
fast-back-to-back
devsel-speed            00000001
class-code              00100000
interrupts              00000001
max-latency             00000040
min-grant                00000040
subsystem-vendor-id     0000108e
subsystem-id            00003de8
revision-id             00000002
device-id                0000b555
vendor-id                00008086
```

watch-net

To monitor a network connection: from the `ok` prompt, type the `apply watch-net` command with the device path:

```
ok apply watch-net /pci@8,600000/network@1
/pci@8,600000/network@1: 1000 Mbps full duplex link up
Watch ethernet packets
'.' is a good packet and 'X' is a bad packet
Press any key to stop
.....X...X.....X.....
```

The system monitors network traffic, displaying “.” each time it receives an error-free packet and “X” each time it receives a packet with an error that can be detected by the network hardware interface.

PKCS#11 Interface

This chapter describes the board's implementation of the PKCS#11 interface and assumes that the Sun Crypto Accelerator 4000 software is installed in the default locations. This chapter also assumes you have familiarity with the PKCS#11 interface. Information about the PKCS#11 standard and original copies of the header files `pkcs11.h`, `pkcs11f.h`, and `pkcs11t.h`, are available at: <http://www.rsasecurity.com/rsalabs/PKCS>

This chapter includes the following sections:

- “General Issues” on page 205
- “Administering the Board to Use PKCS#11” on page 206
- “Installing and Administering Applications That Use Cryptographic Services” on page 207
- “PKCS#11 and FIPS Mode” on page 208
- “Developing Applications to Use PKCS#11” on page 211

General Issues

The Sun Crypto Accelerator 4000 board and associated software provide a PKCS#11 interface. All the PKCS#11 functions necessary for most applications are provided in the Sun Crypto Accelerator 4000 software.

PKCS#11 was designed for a single-user system. The Solaris operating system is a multi-user system, and must deal with multiple concurrent mutually-distrusting users. To accommodate this, the board has added provision for identifying and authenticating multiple users without extending PKCS#11. Every PKCS#11 function that accepts a secret PIN must be given a string of the form *username:password* (See TABLE 5-1). This PIN structure typically propagates up through applications, although a few applications written specifically for the board might ask for the username and secret part separately.

PKCS#11 has a limited administrative facility with just two functions: `C_InitToken`, which initializes the token, and `C_InitPin`, which sets user PINs. The board does not use this facility, and instead uses the `vcaadm` utility.

The `vcaadm` security officer (SO) is not related to the UNIX superuser. Additionally, a board user's `userid`, which is created by the SO with `vcaadm`, is not related to any UNIX username or ID.

PKCS#11 has distinct notions of slots and tokens. Tokens are similar to smart cards, and plug into *slots*. In the Sun Crypto Accelerator 4000 system, there is no distinction between slots and tokens. This guide generally uses the term *token*; however, applications and other documentation might use the term *slot*.

Each board supports up to one *keystore*. The SO gives each keystore a name with `vcaadm`. Each keystore is presented by the board as a PKCS#11 token, with the token label being the name of the associated keystore, space-padded to 32 characters. Multiple boards may support a single keystore for high-availability.

In addition, there is one special token with the label `SUNW_acceleration_only`. This token cannot store any persistent keys, and applications cannot log into this token. Requests submitted to this token are distributed over all available boards.

Many applications display a list of tokens—generally tokens are identified by the PKCS#11 token label. (The token label is the space-padded name of the associated keystore assigned by the SO.)

Administering the Board to Use PKCS#11

The Sun Crypto Accelerator 4000 system is administered using the `vcaadm` utility (See Chapter 4). The SO names the keystore and creates user accounts, giving each account an initial password. The SO also controls whether or not the board operates in FIPS mode (See “PKCS#11 and FIPS Mode” on page 208).

The board supports many PKCS#11 mechanisms. Most of the mechanisms are unconditionally available. However the administrator has some control over the presentation of the following mechanisms:

- `CKM_SSL3_SHA1_MAC`
- `CKM_SSL3_MD5_MAC`
- `CKM_SSL3_PRE_MASTER_KEY_GEN`
- `CKM_SSL3_MASTER_KEY_DERIVE`
- `CKM_SSL3_KEY_AND_MAC_DERIVE`
- `CKM_TLS_PRE_MASTER_KEY_GEN`

- CKM_TLS_MASTER_KEY_DERIVE
- CKM_TLS_KEY_AND_MAC_DERIVE

These mechanisms are always presented by the acceleration-only token. They are presented by tokens with keystores only if the `/etc/opt/SUNWconn/cryptov2/sslreg` is present. To create this file, type the following command as superuser:

```
# touch /etc/opt/SUNWconn/cryptov2/sslreg
```

Restart the applications for this change to take affect.

Network Security Services (NSS) recognizes when these mechanisms are available. When the mechanisms are provided, NSS makes many calls to `C_DigestUpdate` with tiny buffers, which causes the performance to degrade. For this reason, these mechanisms are not provided by default.

Installing and Administering Applications That Use Cryptographic Services

The default location for the PKCS#11 library is:

```
/opt/SUNWconn/cryptov2/lib/libvpkcs11.so
```

Most applications have a configuration file or database, sometimes accessed with a GUI, that contains the location of the PKCS#11 library. Using an editor or the GUI, enter the above value for the default location.

When a key has the `CKA_SENSITIVE` attribute, operations involving that key are restricted to hardware. However, not all operations and all types of keys are supported by the hardware. If an application requests an operation that cannot be done in hardware, and the key's `CKA_SENSITIVE` attribute is true, the operation will fail. The precise rules regarding what combinations keys, operations, and mechanisms are available is described in detail in “Hardware Acceleration and Sensitive Keys” on page 209. If your application will not run because of these rules, you might be able to configure it to not mark its keys as sensitive.

Whether or not the `SSL...` and `TLS...` mechanisms are presented is controlled by the administrator. If the application requires these mechanisms, or you want to experiment with the performance effect of having these mechanisms available, see “Administering the Board to Use PKCS#11” on page 206.

If the board is in FIPS mode, it will provide only mechanisms that are FIPS-approved (See “PKCS#11 and FIPS Mode” on page 208).

PKCS#11 and FIPS Mode

When put in FIPS mode by the SO (using `vcaadm`), the Sun Crypto Accelerator 4000 board is compliant with Federal Information Processing Standard FIPS 140-2 level 3. Detailed information on FIPS 140-2 can be found at: <http://www.nist.gov/dmvp>

Operating the board in FIPS mode causes the following changes in the board’s operation:

- Only FIPS-approved mechanisms are made available by the board, itself.
- All keys and critical security parameters cross the PCI bus in encrypted form.
- Certain additional integrity checks are done at startup and when keys and random numbers are generated.
- Random numbers are generated by a FIPS-approved algorithm that combines saved state and true random data (entropy) from a thermal-noise-based generator using hashing and arithmetic. 512 bits from the thermal-noise-based generator are used for every 160 bits of output data. (In non-FIPS mode, 512 bits from the thermal-noised-based generator are SHA-1 hashed to 160 bits.)

FIPS mode applies only to the Sun Crypto Accelerator 4000 board itself. As stated above, when the board is put in FIPS mode, only FIPS-approved mechanisms are provided by the board. Notably, MD5, RC2, and RC4 are not FIPS-approved. However, because the FIPS regulations apply only to the hardware, the software will continue to make available all the mechanisms that the software normally provides.

The main difference incurred when operating in FIPS mode is that non-FIPS-approved operations will be done only in software, which has two consequences:

- Cryptographic operations using non-FIPS-approved mechanisms are not accelerated.
- If a cryptographic operation using a non-FIPS-approved mechanism involves a key whose `CKA_SENSITIVE` attribute is set to true, the operation fails because keys with the `CKA_SENSITIVE` attribute set to true can be used in the hardware only.

Hardware Acceleration and Sensitive Keys

The board chooses where to execute operations based on the capability of the hardware, on security requirements, and on performance.

PKCS#11 specifies many key types and mechanisms, not all of which are entirely supported in the hardware. When an application requests a combination of operation, key, and mechanism that is not supported entirely in hardware, it may be executed partially in software and partially in hardware, or it may be executed entirely in software.

When a key's `CKA_SENSITIVE` attribute is true, any operation that uses it must be executed securely, that is, with no key material leaving the hardware. If the hardware is not capable of securely executing the operation, it fails. Alternately, when the key's `CKA_SENSITIVE` attribute is false, the board chooses between hardware and software based on performance. This section describes the rules used to choose between hardware, software, and failing the operation.

For convenience, the following sets of keys and mechanisms are defined:

- `hardware_key_set =`
 - RSA with key size not exceeding 2048 bits
 - DSA with key size not exceeding 1024 bits
 - DES
 - 3DES
 - CDMF
- `hardware_mechanism_set =`
 - `CKM_CDMF_...` except `CKM_CDMF_ECB`
 - `CKM_DES_...` except `CKM_DES_ECB`
 - `CKM_DES3_...` except `CKM_DES3_ECB`
 - `CKM_DSA`
 - `CKM_MD5` except in FIPS mode
 - `CKM_RSA_...`
 - `CKM_SHA_1`
- `hardware_wrap_mechanism_set =`
 - `CKM_AES_CBC_PAD`
 - `CKM_CDMF_CBC_PAD`
 - `CKM_DES_CBC_PAD`
 - `CKM_DES3_CBC_PAD`
 - `CKM_RC2_CBC_PAD` except in FIPS mode

For any operation to execute securely in the hardware, the key must be in the `hardware_key_set` and the mechanism must be in the `hardware_mechanism_set`. If the key is in the `hardware_key_set` but the mechanism is not in the `hardware_mechanism_set`, the operation may be accelerated by hardware, but with software assistance.

`C_DeriveKey` may be accelerated in hardware, but requires the assistance of software and thus is not hardware-level secure.

The following table describes if and where operations involving keys are performed:

TABLE 8-1 Processing for Most Crypto Operations Involving Keys

Case	CKA_SENSITIVE=False	CKA_SENSITIVE=True
Hardware-level secure	Hardware for RSA, DSA, and large buffers; software otherwise	Hardware
Hardware acceleration is possible with software assistance	Hardware and software for RSA, DSA, and large buffers; software otherwise	Fail
Software only	Software	Fail

`C_WrapKey` and `C_UnwrapKey` involve two operations on two keys. For the `C_WrapKey`, there is an encode operation that encodes the wrapped key, followed by an encrypt operation that encrypts the encoded value using the wrapping key. `C_UnwrapKey` does the reverse, but with decrypt and decode.

If the wrapped key is an RSA or DSA key and the wrapping mechanism is the `hardware_wrap_mechanism_set`, both the encoding and encryption steps are done in hardware. The operation will be hardware-level secure for both keys.

If any of the conditions above are not satisfied, the encoding step will be done in software. The operation will be not be hardware-level secure for the wrapped key. The encryption step will be treated the same as a `C_Encrypt` operation using the wrapping key and the mechanism. Refer to TABLE 8-1.

The various cases are summarized in the following table:

TABLE 8-2 Failure Condition for C_WrapKey and C_UnwrapKey

Condition	Failures when the wrapped key is sensitive	Failures when the wrapping key is sensitive
Wrapped key is RSA or DSA and mechanism is in hardware_wrap_mechanism_set	-	-
Wrapping key is in hardware_key_set and mechanism is in hardware_mechanism_set	Fail	-
All other cases	Fail	Fail

C_Digest assembles the entire buffer in host memory. C_DigestFinal sends the entire buffer to hardware if the buffer is large, but does not exceed 65532 bytes. Otherwise the entire buffer is processed in software.

C_DigestKey brings the key material into host memory and then treats it just like ordinary data, which is then processed with C_DigestUpdate. It will fail if the key's CKA_SENSITIVE attribute is true.

Developing Applications to Use PKCS#11

The necessary header files are in /opt/SUNWconn/cryptov2/include; add this directory to the include path and include cryptoki.h. The lower-level include files, pkcs11.h, pkcs11f.h, and pkcs11t.h are available in the Sun Crypto Accelerator 4000 software. These files are identical to those available at the PKCS#11 web site (<http://www.rsasecurity.com/rsalabs/PKCS>). The pkcs11_preamble.h file is available in the include directory and must be included before any of the lower level files.

The pkcs11 library is: /opt/SUNWconn/cryptov2/lib/libvpkcs11.so.

The Sun Crypto Accelerator 4000 library can be linked as an ordinary library, or it can be dynamically opened with dlopen (3DL).

When linking as an ordinary library, use the following command:

```
cc [flags] files... -L /opt/SUNWconn/cryptov2/lib \  
-R /opt/SUNWconn/cryptov2/lib -l vpkcs11 [other libraries...]
```

The code should invoke functions directly as in the following example:

```
rv = C_Initialize(NULL);
```

When dynamically linking use the following (shown with error handling elided):

```
cc [flags] files... -ldl [ other libraries ... ]

#include "cryptoki.h"
#include <dlfcn.h>
#include <link.h>

void *cryptodlhandle;
CK_RV (*getfunctionlistp) (CK_FUNCTION_LIST_PTR *);
CK_FUNCTION_LIST *pkllfunclist; /* may need to be globally
accessible */
CK_RV rv;
/* dlopen Sun Cryptoaccelerator 4000 library */
cryptodlhandle =
    dlopen("/opt/SUNWconn/cryptov2/lib/libvpkcs11.so",
    RTLD_NOW | RTLD_LOCAL | RTLD_GROUP);
if (cryptodlhandle == NULL) ...
/* Get pointer to C_GetFunctionList function */
getfunctionlistp = dlsym(cryptodlhandle, "C_GetFunctionList");
if (getfunctionlistp == NULL) ...
/* Get libvpkcs11's cryptki function list */
rv = (*getfunctionlistp) (&pkllfunclist);
if (rv != CKR_OK) ...
```

The code should invoke functions indirectly, as follows:

```
rv = pkllfunclist -> C_Initialize(NULL);
```

The Sun Crypto Accelerator 4000 software imposes very few arbitrary limits. Most resources are limited only by host memory. The maximum number of tokens, including the acceleration-only token, is 1024.

To prevent a denial of service attack by a faulty or malicious program consuming excessive kernel memory, the software limits the amount of kernel memory any one Solaris user (not process) may consume to no more than 16 Mbytes. This limit is not configurable.

You can avoid kernel memory exhaustion problems by observing the following recommendations:

- Do not abandon multistep operations. Call the appropriate finalizing function (for example, `C_EncryptFinal`) or close the session when you are done.
- Do not abandon objects that are not needed. Either close the creating session (effective for volatile objects only) or call `C_DestroyObject` when you are done.
- Do not submit extremely large (multimegabyte) chunks of data at one time. (This does not apply to digest operations, because large digest operations are always done in software.)

The PKCS#11 administrative functions `C_InitToken` and `C_InitPin` are not implemented. The `C_Login` function with the `CKU_SO` (security officer) flag is rejected.

In PKCS#11, *public token* objects are persistent objects that are visible and deletable without authentication. Because the users known by the Sun Crypto Accelerator 4000 software are unrelated to Solaris users, and because the software does not ascertain user identity until `C_Login` succeeds, these objects would need to be globally visible to all users, and therefore deletable by any user. Because this behavior is not acceptable, public token objects are not allowed. Any attempt to create a public token object will fail.

The number of volatile (session) objects is limited by virtual memory only. Persistent objects must all fit in the RAM on the board, but this is not a limitation for any practical use. Consistent with this concept, the fields of the `CK_TOKEN_INFO` structure (returned by the `C_GetTokenInfo` function) that indicate maximum memory sizes are all set to `CK_EFFECTIVELY_INFINITE`. The `C_GetObjectSize` function is not implemented.

The optional *dual operation* functions (`C_DigestEncryptUpdate`, `C_DecryptDigestUpdate`, `C_SignEncryptUpdate`, and `C_DecryptVerifyUpdate`) are not implemented, and the `CKF_DUAL_OPERATIONS_FLAG` in the `flags` field returned by `C_GetTokenInfo` is false.

Only a limited implementation of `C_GetOperationState` and its companion function `C_SetOperationState` is provided. `C_GetOperationState` succeeds only when the operation is `C_Digest` and the size of the accumulated input data does not exceed 65532 bytes.

The tokens provided by the Sun Crypto Accelerator 4000 system are considered nonremovable. Thus the `CKF_REMOVABLE_DEVICE` flag returned by `CK_GetSlotInfo` is false.

The `C_WaitForSlotEvent` function is not implemented, and the Sun Crypto Accelerator 4000 system never calls the callback function passed as the `Notify` parameter to `C_OpenSession`. The software never surrenders control back to the calling application with the `pApplication` parameter of `C_OpenSession`.

The Sun Crypto Accelerator 4000 board contains a high-quality true random number generator. It does not need to be seeded, and, in fact, `C_SeedRandom` will be rejected with `CKR_RANDOM_SEED_NOT_SUPPORTED`.

Functions with an implementation that depends on critical fields being in host memory in the clear will fail when they involve a key that has been created with the `CKA_SENSITIVE` attribute set to true. The precise rules are as follows:

- `C_DigestKey` fails if the key has `CKA_SENSITIVE` set to true.
- `C_DeriveKey` fails for all mechanisms if the base key or the key to be derived has `CKA_SENSITIVE` set to true.
- `C_WrapKey` and `C_UnwrapKey` fails if the key to be wrapped or unwrapped has `CKA_SENSITIVE` set to true, and if any of the following conditions are true:
 - The key is anything other than an RSA or DSA key.
 - The mechanism is anything other than `CKM_DES_CBC_PAD`, `CKM_DES3_CBC_PAD`, `CKM_RC2_CBC_PAD`, or `CKM_AES_CBC_PAD`.
- Any operation involving the following mechanisms fails if the key has `CKA_SENSITIVE` set to true:
 - `CKM_AES...`
 - `CKM_CDMF_ECB`
 - `CKM_DES_ECB`
 - `CKM_DES3_ECB`
 - `CKM_DH...`
 - `CKM_MD5_HMAC...`
 - `CKM_RC2...`
 - `CKM_RC4...`
 - `CKM_SHA_1_HMAC...`
 - `CKM_SSL3...`
 - `CKM_TLS...`
- Any operation involving an RSA key larger than 2048 bits or a DSA key larger than 1024 bits fails if `CKA_SENSITIVE` is set to true.

The `CKA_EXTRACTABLE` attribute defaults to true. The `CKA_SENSITIVE` attribute defaults to the opposite of `CKA_EXTRACTABLE`. An attempt to set both `CKA_SENSITIVE` and `CKA_EXTRACTABLE` to false will fail with `CKR_TEMPLATE_INCONSISTENT`.

Inconsistent attributes are generally not detected. For example if a template contains the same attribute more than once, the implementation simply uses the last value. Attributes not associated with the key type are simply ignored. Not all invalid attributes are detected.

The `CKA_LOCAL`, `CKA_ALWAYS_SENSITIVE`, and `CKA_NEVER_EXTRACTABLE` attributes are not implemented.

The error codes returned by the software are not always what might be expected. In particular, `CKR_MECHANISM_INVALID` is returned for many errors where other values might seem more appropriate. The return code `CKR_HOST_MEMORY` usually means that an internal call to the `malloc(3c)` command failed. After this error is returned, important state has probably not been properly saved, and attempting to continue, except by calling `C_Finalize`, may be futile.

To reduce overhead, the software's implementation of `C_EncryptInit` and similar functions sometimes defers sending the key to the board until there is actual data to encrypt. A consequence of this deferral is that certain errors that PKCS#11 declares should be reported by `C_EncryptInit` (and similar functions) are actually reported on the first subsequent call to `C_EncryptUpdate` (and similar functions).

The mechanisms known by the following PKCS#11 designators are available in the Sun Crypto Accelerator 4000 software. The `CKM_SSL3...` and `CKM_TLS...` mechanisms, although shown in the list, are available on tokens with keystores only if the file `/etc/opt/SUNWconn/CRYPTOV2/SSLREG` is present (See "Administering the Board to Use PKCS#11" on page 206).

- `CKM_AES_CBC`
- `CKM_AES_CBC_PAD`
- `CKM_AES_ECB`
- `CKM_AES_KEY_GEN`
- `CKM_CDMF_CBC`
- `CKM_CDMF_CBC_PAD`
- `CKM_CDMF_ECB`
- `CKM_CDMF_KEY_GEN`
- `CKM_DES2_KEY_GEN`
- `CKM_DES3_CBC`
- `CKM_DES3_CBC_PAD`
- `CKM_DES3_ECB`
- `CKM_DES3_KEY_GEN`
- `CKM_DES_CBC`
- `CKM_DES_CBC_PAD`
- `CKM_DES_ECB`
- `CKM_DES_KEY_GEN`
- `CKM_DH_PKCS_DERIVE`
- `CKM_DH_PKCS_KEY_PAIR_GEN`
- `CKM_DSA`
- `CKM_DSA_KEY_PAIR_GEN`
- `CKM_MD5`
- `CKM_MD5_HMAC`
- `CKM_MD5_HMAC_GENERAL`
- `CKM_RC2_CBC`
- `CKM_RC2_CBC_PAD`
- `CKM_RC2_ECB`
- `CKM_RC2_KEY_GEN`
- `CKM_RC4`

- CKM_RC4_KEY_GEN
- CKM_RSA_PKCS
- CKM_RSA_PKCS_KEY_PAIR_GEN
- CKM_RSA_X_509
- CKM_SHA_1
- CKM_SHA_1_HMAC
- CKM_SHA_1_HMAC_GENERAL
- CKM_SSL3_KEY_AND_MAC_DERIVE
- CKM_SSL3_MASTER_KEY_DERIVE
- CKM_SSL3_MD5_MAC
- CKM_SSL3_PRE_MASTER_KEY_GEN
- CKM_SSL3_SHA1_MAC
- CKM_TLS_KEY_AND_MAC_DERIVE
- CKM_TLS_MASTER_KEY_DERIVE
- CKM_TLS_PRE_MASTER_KEY_GEN

RSA, DSA, and Diffie-Hellman keys have the following maximum key sizes:

TABLE 8-3 Maximum Key Sizes

Key	Nonsensitive Maximum Key Size	Sensitive Maximum Key Size
RSA	4096	2048
DSA	4096	1024
DH	2048	Not available

Do not assume that object handles or session handles are small integers or are sequentially allocated. These handles may be any unsigned long.

The mutex callback function pointers that can be passed to `C_Initialize` are ignored.

In many cases, operations on small amounts of data are processed by the host processor rather than the board because the cost of sending the operation to the board exceeds the cost of doing it in the host. However, all operations involving objects with the `CKA_SENSITIVE` attribute set to true are done in the board.

If the accumulated size of all `C_DigestUpdate` buffers exceeds 65532 bytes, the digesting is processed by software in the host. The same characteristic applies to `C_Digest`. So both small and very large amounts of data are processed by software.

Information about persistent objects is brought into a process when the user successfully executes the `C_Login` function and it remains cached. Subsequent creation, deletion, or modification of persistent objects by another process might not be observed. Operations that take place in the board will use the current state of the key. (Operations are performed in the board if the board is capable and the key is

sensitive, or the board is capable and the buffer is large enough to justify it.) All other cases, plus `C_FindObjects` functions, are processed in software with the cached state of the key.



Caution – Do not depend on the above key-caching behavior remaining unchanged in future releases.

As required by the PKCS#11 standard, all persistent object handles become invalid when the user calls the `C_Logout` function or closes the last PKCS#11 session. The software purges the token objects from the software's cache. A subsequent successful `C_Login` function brings in all the then-current token objects. Note that this login could be for a different user and thus bring in a different set of token objects. However, even if this login is for the same user, the token objects might not get the same handles as they had before.

Specifications

This appendix lists the specifications for the Sun Crypto Accelerator 4000 MMF and UTP adapters. It contains the following sections:

- “Sun Crypto Accelerator 4000 MMF Adapter” on page 219
- “Sun Crypto Accelerator 4000 UTP Adapter” on page 222

Sun Crypto Accelerator 4000 MMF Adapter

This section provides the specifications for the Sun Crypto Accelerator 4000 MMF adapter.

Connectors

FIGURE A-1 shows the connector for the Sun Crypto Accelerator 4000 MMF adapter.



FIGURE A-1 Sun Crypto Accelerator 4000 MMF Adapter Connector

TABLE A-1 lists the characteristics of the SC connector (850 nm).

TABLE A-1 SC Connector Link Characteristics (IEEE P802.3z)

Characteristic	62.5 Micron MMF	50 Micron MMF
Operating range	Up to 260 meters	Up to 550 meters

Physical Dimensions

TABLE A-2 Physical Dimensions

Dimension	Measurement	Metric Measurement
Length	12.283 inches	312.00 mm
Width	4.200 inches	106.68 mm

Performance Specifications

TABLE A-3 Performance Specifications

Feature	Specification
PCI clock	33/66 MHz max
PCI data burst transfer rate	Up to 64-byte bursts
PCI data/address width	32/64-bit
PCI modes	Master/slave
1 Gbps, 850 nm	1000 Mbps (full duplex)

Power Requirements

TABLE A-4 Power Requirements

Specification	Measurement
Maximum power consumption	6.25 W @ 5V 12.75 W @ 3.3V
Voltage tolerance	5V +/- 5% 3.3V +/- 5%

Interface Specifications

TABLE A-5 Interface Specifications

Feature	Specification
PCI clock	33 MHz or 66 MHz
Host interface	PCI 2.1 with support for 33 MHz or 66 MHz clock rate and 3.3V or 5V power
PCI bus width	32 bits or 64 bits

Environmental Specifications

TABLE A-6 Environmental Specifications

Condition	Operating Specification	Storage Specification
Temperature	0° to +55° C, +32° to +131° F	-40° to +75° C, -40° to +167° F
Relative humidity	5 to 85% noncondensing	0 to 95% noncondensing

Sun Crypto Accelerator 4000 UTP Adapter

This section provides the specifications for the Sun Crypto Accelerator 4000 UTP adapter.

Connectors

FIGURE A-1 shows the connector for the Sun Crypto Accelerator 4000 UTP adapter.



FIGURE A-2 Sun Crypto Accelerator 4000 UTP Adapter Connector

TABLE A-7 lists the characteristics of the Cat-5 connector used by the Sun Crypto Accelerator 4000 UTP adapter.

TABLE A-7 Cat-5 Connector Link Characteristics

Characteristic	Description
Operating range	Up to 100 meters

Physical Dimensions

TABLE A-8 Physical Dimensions

Dimension	Measurement	Metric Measurement
Length	12.283 inches	312.00 mm
Width	4.200 inches	106.68 mm

Performance Specifications

TABLE A-9 Performance Specifications

Feature	Specification
PCI clock	33/66 MHz max
PCI data burst transfer rate	Up to 64-byte bursts
PCI data/address width	32/64-bit
PCI modes	Master/slave
1 Gbps	1000 Mbps (Full Duplex)
100 Mbps	100 Mbps (Full and Half Duplex)
10 Mbps	10 Mbps (Full and Half Duplex)

Power Requirements

TABLE A-10 Power Requirements

Specification	Measurement
Maximum power consumption	6.25 W @ 5V 12.75 W @ 3.3V
Voltage tolerance	5V +/- 5% 3.3V +/- 5%

Interface Specifications

TABLE A-11 Interface Specifications

Feature	Specification
PCI clock	33 MHz or 66 MHz
Host interface	PCI 2.1 with support for 33 MHz or 66 MHz clock rate and 3.3V or 5V power
PCI bus width	32 bits or 64 bits

Environmental Specifications

TABLE A-12 Environmental Specifications

Condition	Operating Specification	Storage Specification
Temperature	0° to +55° C, +32° to +131° F	-40° to +75° C, -40° to +167° F
Relative humidity	5 to 85% noncondensing	0 to 95% noncondensing

Installing the Software Without the Installation Script

This appendix describes how to install the Sun Crypto Accelerator 4000 software manually without using the installation script (`/cdrom/cdrom0/install`) provided on the product CD. The following sections are included:

- “Installing the Software Manually” on page 227
- “Directories and Files” on page 230
- “Removing the Software Manually” on page 231

Installing the Software Manually

The Sun Crypto Accelerator 4000 software is included on the product CD. You may need to download patches from the SunSolve web site (<http://sunsolve.sun.com>). See “Required Patches” on page 12 for more information.

▼ To Install the Software Manually

1. **Insert the Sun Crypto Accelerator 4000 CD into a CD-ROM drive that is connected to your system.**
 - If your system is running Sun Enterprise Volume Manager, it should automatically mount the CD-ROM to the `/cdrom/cdrom0` directory.
 - If your system is not running Sun Enterprise Volume Manager, mount the CD-ROM as follows:

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /cdrom
```

You see the following files and directories in the /cdrom/cdrom0 directory.

TABLE B-1 Files in the /cdrom/cdrom0 Directory

File or Directory	Contents
Copyright	U.S. copyright file
FR_Copyright	French copyright file
install	Installation script that installs the Sun Crypto Accelerator 4000 software
remove	Removal script that removes the Sun Crypto Accelerator 4000 software
Docs	<i>Sun Crypto Accelerator 4000 Board Version 1.1 Installation and User's Guide</i> <i>Sun Crypto Accelerator 4000 Board Release Notes</i>
Packages	Sun Crypto Accelerator 4000 software packages: SUNWkc12r Cryptography Kernel Components SUNWkc12u Cryptographic Administration Utility and Libraries SUNWkc12a SSL Support for Apache (<i>optional</i>) SUNWkc12m Cryptographic Administration Manual Pages (<i>optional</i>) SUNWvcar VCA Crypto Accelerator (root) SUNWvcau VCA Crypto Accelerator (usr) SUNWvcaa VCA Administration SUNWvcaw VCA Firmware SUNWvcamm VCA Crypto Accelerator Manual Page (<i>optional</i>) SUNWvcav SunVTS Test of VCA Crypto Accelerator (<i>optional</i>) SUNWkc12o SSL Development Tools and Libraries (<i>optional</i>) SUNWkc12i.u IPsec Acceleration with KCLv2 Crypto (<i>optional</i>)

The required packages must be installed in a specific order and must be installed before installing any optional packages. Once the required packages are installed, you can install and remove the optional packages in any order.

Install the optional SUNWkc12a package only if you plan to use Apache as your web server.

Install the optional SUNWkc12o package only if you plan to relink to another (unsupported) version of Apache Web Server.

Install the optional SUNWvcav package only if you plan to perform the SunVTS tests. You must have SunVTS 4.4 or later up to 5.x installed to install the SUNWvcav package.

Note – The optional SUNWkc12i.u package has the .u extension only on the Sun Crypto Accelerator 4000 CD. Once this package is installed, the name is changed to SUNWkc12i. The .u extension of this package on the CD, defines the package as sun4u architecture-specific.

1. Install the required software packages by typing:

```
# cd /cdrom/cdrom0/Packages
# pkgadd -d . SUNWkcl2r SUNWkcl2u SUNWkcl2m SUNWvcar SUNWvcau SUNWvcaa SUNWvcam
SUNWvcaw
```

2. (Optional) To verify that the software is installed properly, run the `pkginfo` command.

```
# pkginfo SUNWkcl2r SUNWkcl2u SUNWvcar SUNWvcau SUNWvcaa SUNWvcaw
system    SUNWkcl2r    KCLv2 Crypto (Root)
system    SUNWkcl2u    KCLv2 Crypto Support Software
system    SUNWvcaa     VCA Crypto Accelerator/Gigabit Ethernet Admin
system    SUNWvcaw     VCA Crypto Accelerator/Gigabit Ethernet firmware
system    SUNWvcar     VCA Crypto Accelerator/Gigabit Ethernet Drivers
system    SUNWvcau     VCA Crypto Accelerator/Gigabit Ethernet Daemon
```

3. (Optional) To ensure that the driver is attached, you can run the `prtdiag` command.

Refer to the `prtdiag(1m)` online manual pages.

```
# prtdiag -v
```

4. (Optional) Run the `modinfo` command to see that modules are loaded.

```
# modinfo | grep Crypto
62  1317f62  20b1f 198   1  vca (VCA Crypto/Ethernet v1.102)
63  13360e9  12510 200   1  kcl2 (Kernel Crypto Library v1.148)
197 136d5d6   19b0 199   1  vcactl (VCA Crypto Control v1.19)
```

Installing the Optional Packages

To install only the optional packages that provide the SSL support for Apache Web Server and the Sun Crypto Accelerator 4000 online manual pages, type the following:

```
# cd /cdrom/cdrom0/Packages
# pkgadd -d . SUNWkcl2a SUNWkcl2m
```

To install all of the optional software packages, type the following:

```
# cd /cdrom/cdrom0/Packages
# pkgadd -d . SUNWkc12a SUNWkc12m SUNWvcamn SUNWvcav SUNWkc12o SUNWkc12i.u
```

See TABLE B-1 for a description of the package contents of the optional packages in the previous examples.

Directories and Files

TABLE B-2 shows the directories created by the default installation of the Sun Crypto Accelerator 4000 software.

TABLE B-2 Sun Crypto Accelerator 4000 Directories

Directory	Contents
/etc/opt/SUNWconn/vca/keydata	Keystore data (encrypted)
/opt/SUNWconn/cryptov2/bin	Utilities
/opt/SUNWconn/cryptov2/lib	Support libraries
/opt/SUNWconn/cryptov2/sbin	Administrative commands

FIGURE B-1 shows the hierarchy of these directories and files.

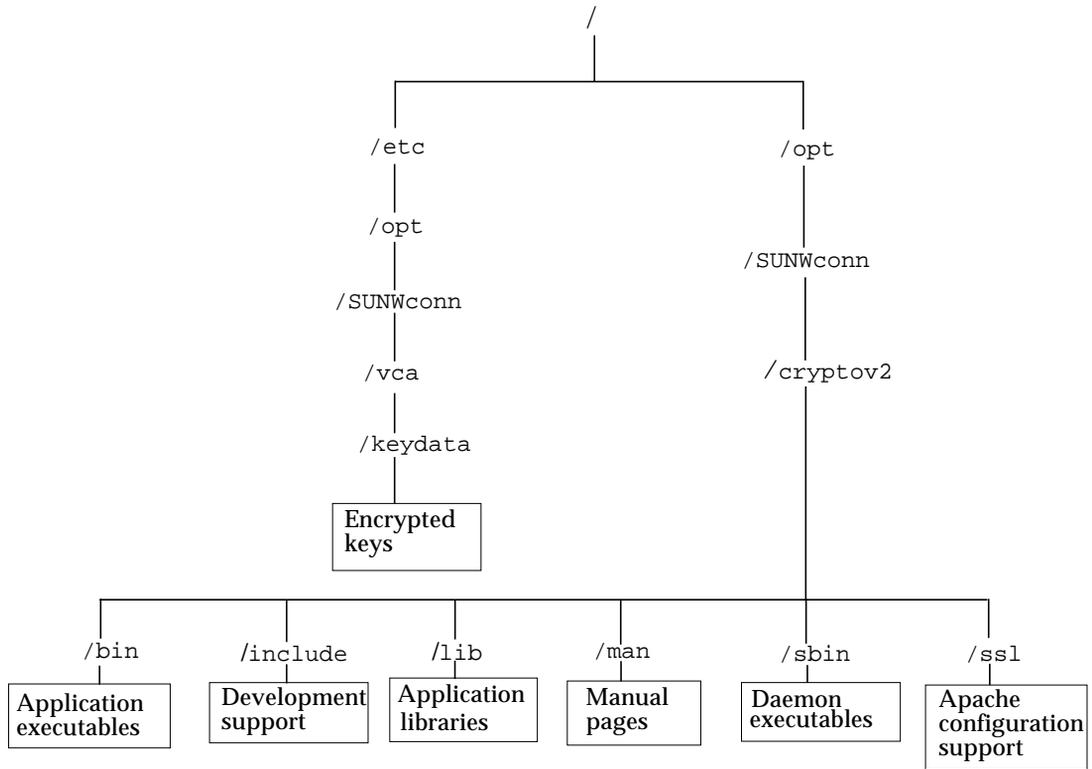


FIGURE B-1 Sun Crypto Accelerator 4000 Directories and Files

Note – Once you have installed the hardware and software of the board, you need to initialize the board with configuration and keystore information. See “Initializing the Board With vcaadm” on page 68 for information on how to initialize the board.

Removing the Software Manually

If you have created keystores (refer to “Managing Keystores With vcaadm” on page 71), you must delete the keystore information that the Sun Crypto Accelerator 4000 board is configured with before removing the software. The `zeroize` command removes all key material, but does not delete the keystore files that are stored in the filesystem of the physical host in which the board is installed. See the “Performing a Software Zeroize on the Board” on page 82 for details on the

zeroize command. To delete the keystore files stored in the system, become superuser and remove the keystore files. If you have not yet created any keystores, you can skip this procedure.



Caution – Do not delete a keystore that is currently in use or that is shared by other users and keystores. To free references to keystores, you might have to shut down the web server and/or administration server.



Caution – Before removing the Sun Crypto Accelerator 4000 software disable any web servers you have enabled for use with the Sun Crypto Accelerator 4000 board. Failure to do so leaves those web servers nonfunctional.

▼ To Remove the Software Manually

- As superuser, use the `pkgrm` command to remove only the software packages you installed.



Caution – Installed packages must be removed in the order shown. Failure to remove them in this order could result in dependency warnings and leave kernel modules loaded.

If you installed all the packages, you would remove them as follows:

```
# pkgrm SUNWkcl2o SUNWvcav SUNWvcar SUNWkcl2a SUNWkcl2u SUNWkcl2r
SUNWvcamn SUNWkcl2m SUNWkcl2i SUNWvcaa SUNWvcafz SUNWvcau
```

Note – After installing or removing the SunVTS test (`SUNWvcav`) for the Sun Crypto Accelerator 4000 board, if SunVTS is already running it might be necessary to reprobe the system to update the available tests. See your SunVTS documentation for more information.

SSL Configuration Directives for Apache Web Servers

This appendix lists directives for using Sun Crypto Accelerator 4000 software to configure SSL support for Apache Web Servers. Configure directives in your `http.conf` file. Refer to the Apache Web Server documentation for more information.

1. `SSLPassPhraseDialog` `exec:program`

Context: Global

This directive informs the Apache Web Server that the specified *program* should be executed to collect the password for key file. *program* should print the collected password to standard output.

If multiple key files are present, and have common passwords, then *program* is executed once (each collected password is tried before running *program* again.)

program is executed with two arguments, the first is the name of the server, in the form *servername:port*, for example, `www.fictional-company.com:443`. (Port 443 is the typical port for SSL based web servers.) The second argument is the type of key in the key file (*keytype*). *keytype* can be either RSA or DSA.

Note – Because this program can be executed during system startup, be sure to design it to cope with the situation where the console is not a `tty` device (that is, a `tty(3c)` returns false).

The supplied program `/opt/SUNWconn/cryptov2/bin/apgetpass` can be used for the *program* executable. This program automatically prompts for the password, suppressing the display of the password as it is entered.

The supplied `sslpassword` program also automatically searches for passwords in files, which can be used to avoid user interaction when the web server starts up. Passwords for key files are searched for in files named

`/etc/apache/servername:port.keytype.pass`. If this file is not present, then the file `/etc/apache/default.pass` is used. These password files contain only the unencrypted password on a line by itself.

Note – Password files should be protected by permissions so that only the UNIX user that the web server runs as can read the file. This user should be the same user as configured with the standard Apache `User` directive.

If not specified, the default behavior uses an internal prompting mechanism. Do not use the default; use the supplied `sslpassword` program instead, to avoid problems with interaction at system startup.

2. `SSLEngine` (`on|off`)

Context: Global, virtual host

This directive enables the SSL protocol. It is typically used in a virtual host to enable SSL on a subset of servers. One form commonly used is:

```
<VirtualHost _default_:443>
SSLEngine on
</VirtualHost>
```

This statement configures the use of SSL for any servers listening on port 443 (the standard HTTPS port). If not present, this protocol is turned off by default.

3. `SSLProtocol` [`+-`] *protocol*

Context: Global, virtual host

This directive configures the protocol(s) that the server should use for SSL transactions. The available protocols are listed and described in TABLE C-1:

TABLE C-1 SSL Protocols

Protocol	Description
SSLv2	Original standard SSL protocol from Netscape
SSLv3	Updated version of the SSL protocol, supported by most popular web browsers
TLSv1	Update to SSLv3 currently undergoing IETF standardization, with minimal browser support
all	Enable all protocols

Using the plus (+) or minus (-) signs, protocols can be added or removed. For example, to disable support for SSLv2, the following directive could be used:

```
SSLProtocol all -SSLv2
```

The preceding statement is equivalent to:

```
SSLProtocol +SSLv3 +TLSv1
```

4. SSLCipherSuite *cipher-spec*

Context: Global, virtual host, directory, `.htaccess`

The SSLCipherSuite directive is used to configure which SSL ciphers are available for use and their preference. In global context or virtual host context, this directive is used during the initial SSL handshake. In per-directory context, it forces an SSL renegotiation to use the named ciphers. The renegotiation takes place after the request is read, but before the response is sent.

The *cipher-spec* is a colon-delimited list of the ciphers described in TABLE C-2. In TABLE C-2, DH refers to Diffie-Hellman and DSS refers to the Digital Signature Standard.

TABLE C-2 Available SSL Ciphers

Cipher-Tag	Protocol	Key Exchange	Auth.	Encryption	MAC	Type
DES-CBC3-SHA	SSLv3	RSA	RSA	3DES (168-bit)	SHA1	
DES-CBC3-MD5	SSLv2	RSA	RSA	3DES (168-bit)	MD5	
RC4-SHA	SSLv3	RSA	RSA	ARCFOUR (128-bit)	SHA1	
RC4-MD5	SSLv3	RSA	RSA	ARCFOUR (128-bit)	MD5	
RC4-MD5	SSLv2	RSA	RSA	ARCFOUR (128-bit)	MD5	
RC2-CBC-MD5	SSLv2	RSA	RSA	ARCTWO (128-bit)		
DES-CBC-SHA	SSLv3	RSA	RSA	DES (56-bit)	SHA1	
RC4-64-MD5	SSLv2	RSA	RSA	ARCFOUR (64-bit)	MD5	
DES-CBC-MD5	SSLv2	RSA	RSA	DES (56-bit)	MD5	
EXP-DES-CBC-SHA	SSLv3	RSA (512 bit)	RSA	DES (40-bit)	SHA1	export
EXP-RC2-CBC-MD5	SSLv2	RSA (512 bit)	RSA	ARCTWO (40-bit)	SHA1	export
EXP-RC2-CBC-MD5	SSLv3	RSA (512 bit)	RSA	ARCTWO (40-bit)	SHA1	export
EXP-RC4-MD5	SSLv3	RSA (512 bit)	RSA	ARCFOUR (40-bit)	MD5	export

TABLE C-2 Available SSL Ciphers (*Continued*)

Cipher-Tag	Protocol	Key Exchange	Auth.	Encryption	MAC	Type
EXP-RC4-MD5	SSLv2	RSA (512 bit)	RSA	ARCFOUR (40-bit)	MD5	export
NULL-SHA	SSLv3	RSA	RSA	None	SHA1	
NULL-MD5	SSLv3	RSA	RSA	None	MD5	
ADH-DES-CBC3-SHA	SSLv3	DH	None	3DES (168-bit)	SHA1	
ADH-DES-CBC-SHA	SSLv3	DH	None	DES (56-bit)	SHA1	
ADH-RC4-MD5	SSLv3	DH	None	ARCFOUR (128-bit)	MD5	
EDH-RSA-DES-CBC3-SHA	SSLv3	DH	RSA	3DES (168-bit)	SHA1	
EDH-DSS-DES-CBC3-SHA	SSLv3	DH	DSS	3DES (168-bit)	SHA1	
EDH-RSA-DES-CBC-SHA	SSLv3	DH	RSA	DES (56-bit)	SHA1	
EDH-DSS-DES-CBC-SHA	SSLv3	DH	DSS	DES (56-bit)	SHA1	
EXP-EDH-RSA-DES-CBC-SHA	SSLv3	DH (512 bit)	RSA	DES (40-bit)	SHA1	export
EXP-EDH-DSS-DES-CBC-SHA	SSLv3	DH (512 bit)	DSS	DES (40-bit)	SHA1	export
EXP-ADH-DES-CBC-SHA	SSLv3	DH (512 bit)	None	DES (40-bit)	SHA1	export
EXP-ADH-RC4-MD5	SSLv3	DH (512 bit)	None	ARCFOUR (40-bit)	MD5	export

TABLE C-3 lists and describes the aliases that provide macro-like groupings.

TABLE C-3 SSL Aliases

Alias	Description
SSLv2	All SSL version 2.0 ciphers
SSLv3	All SSL version 3.0 ciphers
EXP	All export-grade ciphers
EXPORT40	All 40-bit export ciphers
EXPORT56	All 56-bit export ciphers
LOW	Lower strength ciphers (DES, 40-bit RC4)
MEDIUM	All 128-bit ciphers
HIGH	All ciphers using Triple DES
RSA	All ciphers using RSA key exchange
DH	All ciphers using Diffie-Hellman key exchange
EDH	All ciphers using Ephemeral Diffie-Hellman key exchange

TABLE C-3 SSL Aliases (*Continued*)

Alias	Description
ADH	All ciphers using anonymous Diffie-Hellman key exchange
DSS	All ciphers using DSS authentication
NULL	All ciphers using no encryption

The preference of ciphers can be configured using the special characters listed and described in TABLE C-4.

TABLE C-4 Special Characters to Configure Cipher Preference

Character	Description
<none>	Adds cipher to list
!	Removes a cipher from the list entirely—it cannot be added again
+	Adds cipher to list, and pull to current location (possibly demoting it)
-	Remove cipher from list (can be added later in list)

The default value of *cipher-spec* is

```
SSLCipherSuite ALL:!ADH:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP
```

The default configures all ciphers except anonymous (unauthenticated) Diffie-Hellman, giving preference to ARCFOUR and RSA, and then higher grades of encryption over the lower grades.

5. SSLCertificateFile *file*

Context: Global, virtual host

This directive specifies the location of the PEM-encoded X.509 certificate file for this server.

6. SSLCertificateKeyFile *file*

Context: Global, virtual host

This directive specifies the location of the PEM-encoded private key file for this server, corresponding to the certificate configured with the SSLCertificateFile directive.

7. SSLCertificateChainFile *file*

Context: Global, virtual host

This directive specifies the location of a file containing the PEM-encoded certificates making up the certification path of the server. You can use the directive to assist clients in verifying the server's certificate when the server's certificate is not directly signed by an authority that the client recognizes.

Certificates in the chain are assumed to be valid for client authentication as well, when client authentication (`SSLVerifyClient`) is used.

8. `SSLCACertificateFile` *file*

Context: Global, virtual host

This directive specifies the location of a file containing the concatenation of the certificates for certification authorities (CAs) used for client authentication.

9. `SSLCARevocationFile` *file*

Context: Global, virtual host

This directive specifies the location of a file containing the concatenation of the certificate revocation lists of CAs used for client authentication.

10. `SSLVerifyClient` *level*

Context: Global, virtual host, directory, `.htaccess`

This directive configures the authentication of clients to the server. Note that this is not normally needed for e-commerce applications, but has use in other applications.

Values for *level* are listed and described in TABLE C-5.

TABLE C-5 SSL Verify Client Levels

Level	Description
none	No client certificate is required
optional	Client may present a valid certificate
require	Client <i>must</i> present a valid certificate
optional_no_ca	Client may present a certificate, but it need not be valid

Typically either `none` or `require` is used. The default is `none`.

11. `SSLVerifyDepth` *depth*

Context: Global, virtual host, directory, `.htaccess`

This directive specifies the maximum certificate chain depth that the server will allow for client certificates. A value of 0 means that only self-signed certificates are eligible, whereas a value of 1 means that client certificates must be signed by a CA known directly to the server (through the `SSLCACertificateFile`).

Larger values permit delegation of the CA.

12. SSLLog *filename*

Context: Global, virtual host

This directive specifies a log file where SSL-specific information will be logged. If not specified (default), then no SSL-specific information will be logged.

13. SSLLogLevel *level*

Context: Global, virtual host

This directive specifies the verbosity of the information logged in the SSL log file. Values for *level* are listed and described in TABLE C-6.

TABLE C-6 SSL Log Level Values

Value	Description
none	No logging, but error messages are still sent to the standard Apache error log
warn	Include warning messages
info	Include information messages
trace	Include trace messages
debug	Include debugging messages

14. SSLOptions [+*-*] *option*

Context: Global, virtual host, directory, `.htaccess`

This directive configures SSL runtime options on a per-directory basis. Options can be added to the current configuration by prefixing them with a plus sign (+), or removed using a minus sign (-). If multiple options could apply to a directory, the most restrictive option is used; the options are not merged.

Options are listed and described in TABLE C-7.

TABLE C-7 Available SSL Options

Options	Description
StdEnvVars	Standard set of SSL-related CGI/SSI environment variables are created—there is a performance penalty for this.
ExportCertData	Causes the <code>SSL_SERVER_CERT</code> , <code>SSL_CLIENT_CERT</code> and <code>SSL_CLIENT_CERT_CHAINn</code> ($n = 0, 1, \dots$) environment variables to be exported. These variables contain PEM-encoded certificates for the client and server.
FakeBasicAuth	Distinguished Name (DN) of the client certificate is translated into an HTTP Basic Authentication Username, and is “faked” to have authentication. This allows the use of standard Apache access control mechanisms with SSL client authentication without prompting the user for a password. Entries for these users in the Apache password files must use the encrypted password <code>xxj3lZMTZzkVA</code> , which is just an encrypted form (<code>crypt(3c)</code>) of the word “password.”
StrictRequire	Forces a forbidden access due to <code>SSLRequireSSL</code> to be denied, even in the presence of other directives, such as <code>Satisfy Any</code> , which might override this.

15. SSLRequireSSL

Context: Directory, `.htaccess`

This directive forbids access in a given directory unless HTTPS is used. Use the directive to guard against misconfigurations that might otherwise leave a directory's contents available to unauthenticated and unencrypted accesses.

Configuring Custom Applications to Use the Board

This appendix describes the software supplied with the board. This software can be used to build OpenSSL-compatible applications to take advantage of the cryptographic acceleration features of the board. Not all OpenSSL applications benefit from being compiled in this fashion. Some applications benefit from being built with the stock OpenSSL library, which can be downloaded from <http://www.openssl.org>.

Configuring Custom Applications to Use the Board

This information on building applications to use the Sun Crypto Accelerator 4000 software and hardware is provided strictly as-is, and is not an officially supported part of this product. This information might be useful, but it is provided without any warranty. If you require a Sun-supported solution, please contact Sun Professional Services to learn about your options.

▼ To Configure Custom Applications to Use the Board

1. **Install the `SUNWkcl2o` package, which contains the required header files and libraries.**

2. Configure your application to include OpenSSL headers from

/opt/SUNWconn/cryptov2/include, such as with the following compiler flag:

```
-I/opt/SUNWconn/cryptov2/include
```

3. Direct the linker to include references to the appropriate libraries.

Most OpenSSL-compatible applications reference either or both of the `libcrypto.a` and `libssl.a` libraries. Include the Sun cryptographic libraries. The following linker attributes accomplish this:

```
-L/opt/SUNWconn/cryptov2/lib -R/opt/SUNWconn/cryptov2/lib \  
-lcrypto -lssl -lkcl
```

Software Licenses

This appendix provides the Sun Binary Code License Agreement and third-party software notices and licenses.

Note – The third-party licenses and notices provided in this appendix are included exactly as they are provided by the owners of the software licenses and notices.

Sun Microsystems, Inc.

Binary Code License Agreement

READ THE TERMS OF THIS AGREEMENT AND ANY PROVIDED SUPPLEMENTAL LICENSE TERMS COLLECTIVELY "AGREEMENT") CAREFULLY BEFORE OPENING THE SOFTWARE MEDIA PACKAGE. BY OPENING THE SOFTWARE MEDIA PACKAGE, YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCESSING THE SOFTWARE ELECTRONICALLY, INDICATE YOUR ACCEPTANCE OF THESE TERMS BY SELECTING THE "ACCEPT" BUTTON AT THE END OF THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL THESE TERMS, PROMPTLY RETURN THE UNUSED SOFTWARE TO YOUR PLACE OF PURCHASE FOR A REFUND OR, IF THE SOFTWARE IS ACCESSED ELECTRONICALLY, SELECT THE "DECLINE" BUTTON AT THE END OF THIS AGREEMENT.

1. LICENSE TO USE. Sun grants you a non-exclusive and non-transferable license for the internal use only of the accompanying software and documentation and any error corrections provided by Sun (collectively "Software"), by the number of users and the class of computer hardware for which the corresponding fee has been paid.
2. RESTRICTIONS Software is confidential and copyrighted. Title to Software and all associated intellectual property rights is retained by Sun and/or its licensors. Except as specifically authorized in any Supplemental License Terms, you may not make copies of Software, other than a single copy of Software for archival purposes. Unless enforcement is prohibited by applicable law, you may not modify, decompile, or reverse engineer Software. You acknowledge that Software is not designed,

licensed or intended for use in the design, construction, operation or maintenance of any nuclear facility. Sun disclaims any express or implied warranty of fitness for such uses. No right, title or interest in or to any trademark, service mark, logo or trade name of Sun or its licensors is granted under this Agreement.

3. **LIMITED WARRANTY.** Sun warrants to you that for a period of ninety (90) days from the date of purchase, as evidenced by a copy of the receipt, the media on which Software is furnished (if any) will be free of defects in materials and workmanship under normal use. Except for the foregoing, Software is provided "AS IS". Your exclusive remedy and Sun's entire liability under this limited warranty will be at Sun's option to replace Software media or refund the fee paid for Software.

4. **DISCLAIMER OF WARRANTY.** UNLESS SPECIFIED IN THIS AGREEMENT, ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT THESE DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

5. **LIMITATION OF LIABILITY.** TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF OR RELATED TO THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In no event will Sun's liability to you, whether in contract, tort (including negligence), or otherwise, exceed the amount paid by you for Software under this Agreement. The foregoing limitations will apply even if the above stated warranty fails of its essential purpose.

6. **TERMINATION.** This Agreement is effective until terminated. You may terminate this Agreement at any time by destroying all copies of Software. This Agreement will terminate immediately without notice from Sun if you fail to comply with any provision of this Agreement. Upon Termination, you must destroy all copies of Software.

7. **EXPORT REGULATIONS.** All Software and technical data delivered under this Agreement are subject to US export control laws and may be subject to export or import regulations in other countries. You agree to comply strictly with all such laws and regulations and acknowledge that you have the responsibility to obtain such licenses to export, re-export, or import as may be required after delivery to you.

8. **U.S. GOVERNMENT RESTRICTED RIGHTS.** If Software is being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), then the Government's rights in Software and accompanying documentation will be only as set forth in this Agreement; this is in accordance with 48 CFR 227.7201 through 227.7202-4 (for Department of Defense (DOD) acquisitions) and with 48 CFR 2.101 and 12.212 (for non-DOD acquisitions).

9. **GOVERNING LAW.** Any action related to this Agreement will be governed by California law and controlling U.S. federal law. No choice of law rules of any jurisdiction will apply.

10. **SEVERABILITY.** If any provision of this Agreement is held to be unenforceable, this Agreement will remain in effect with the provision omitted, unless omission would frustrate the intent of the parties, in which case this Agreement will immediately terminate.

11. **INTEGRATION.** This Agreement is the entire agreement between you and Sun relating to its subject matter. It supersedes all prior or contemporaneous oral or written communications, proposals, representations and warranties and prevails over any conflicting or additional terms of any quote, order, acknowledgment, or other communication between the parties relating to its subject matter during the term of this Agreement. No modification of this Agreement will be binding, unless in writing and signed by an authorized representative of each party.

For inquiries please contact: Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, California 95054

(Form ID#011801)

Sun Microsystems, Inc.

Supplemental Terms for Sun Crypto Accelerator 4000

These Supplemental Terms for the Sun Crypto Accelerator 4000 supplement the terms of the Binary Code License Agreement ("BCL"). Capitalized terms not defined herein shall have the meanings ascribed to them in the BCL. These Supplemental Terms will supersede any inconsistent or conflicting terms in the BCL. Use of the Software constitutes acceptance of the BCL as supplemented hereby.

1. **THIRD PARTY LICENSE TERMS.** Some portions of Software are provided with notices and/or licenses from other parties which govern the use of those portions.

Third Party License Terms

OPENSSL LICENSE ISSUES

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License

Copyright (c) 1998-2001 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"
4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE

OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed.

If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)" The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-).
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

```
``Ian Fleming was a UNIX fan!  
How do I know? Well, James Bond  
had the (license to kill) number 007,  
i.e. he could execute anyone."  
-- Unknown
```

MOD_SSL LICENSE

The mod_ssl package falls under the Open-Source Software label because it's distributed under a BSD-style license. The detailed license information follows.

Copyright (c) 1998-2000 Ralf S. Engelschall. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by Ralf S. Engelschall <rse@engelschall.com> for use in the mod_ssl project (<http://www.modssl.org/>)."
4. The names "mod_ssl" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact rse@engelschall.com.
5. Products derived from this software may not be called "mod_ssl" nor may "mod_ssl" appear in their names without prior written permission of Ralf S. Engelschall.
6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by Ralf S. Engelschall <rse@engelschall.com> for use in the mod_ssl project (<http://www.modssl.org/>)."

THIS SOFTWARE IS PROVIDED BY RALF S. ENGELSCHALL ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL RALF S. ENGELSCHALL OR HIS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Manual Pages

This appendix provides descriptions of the Sun Crypto Accelerator 4000 commands and utilities provided in the board's software, and lists the online manual pages for each.

The online manual pages can be viewed with the following command:

```
man -M /opt/SUNWconn/man pagename
```

TABLE F-1 lists and describes the available online manual pages.

TABLE F-1 Sun Crypto Accelerator 4000 Online Manual Pages

man page	Description
vca(7d)	Leaf driver that provides access control to the underlying hardware cryptographic accelerator
vcad(1m)	Daemon that provides keystore services
vcaadm(1m)	Utility that manipulates the configuration, account, and keying databases associated with the board
vcadiag(1m)	Utility that allows superusers to reset boards, zeroize key material, and perform basic diagnostics
kc12(7d)	kc12 is a kernel module that provides support for cryptographic hardware drivers.
apsslcfg(1m)	Configuration utility for Apache Web Servers
iplsslcfg(1m)	Configuration utility for Sun ONE Web Servers
pk11export(1m)	Key export utility that uses the PKCS#11 interface

Zeroizing the Hardware

This appendix describes how to perform a hardware zeroize of the Sun Crypto Accelerator 4000 board, which returns the board to the factory state. When the board is returned to the factory state, it is in Failsafe mode.



Caution – You should perform a hardware zeroize only if it is absolutely necessary. If you need to remove all key material only, perform a software zeroize with the `zeroize` command in the `vcaadm` program. See “Performing a Software Zeroize on the Board” on page 82 for details on the `zeroize` command. Also refer to the online manual pages for `vcadiag(4)` for removing all key material.

Note – Performing a hardware zeroize on the board removes the Sun Crypto Accelerator 4000 firmware. You will have to reinstall the firmware which is provided with the Sun Crypto Accelerator 4000 software.

Zeroizing the Sun Crypto Accelerator 4000 Hardware to the Factory State

In some situations, it might become necessary to return a board to `failsafe` mode, and clear it of all key material and configuration information. This can only be done by using a standard SCSI hardware jumper (shunt).

Note – You can use the `zeroize` command with the `vcaadm` program to remove all key material from a Sun Crypto Accelerator 4000 board. However, the `zeroize` command leaves any updated firmware intact. See “Performing a Software Zeroize on the Board” on page 82. Also refer to the `vcadiag(4)` online manual pages.

▼ To Zeroize the Sun Crypto Accelerator 4000 Board With a Hardware Jumper

1. Power off the system.

Note – For some systems, you can use dynamic reconfiguration (DR) to remove and replace the board as necessary for this procedure instead of powering off the system. Refer to the documentation delivered with your system for the correct DR procedures.



Caution – The board must not receive any electrical power while adjusting the jumper.

2. Remove the computer cover to get access to the jumper, which is located at the top middle of the board.

3. Place the jumper on pins 1 and 2 of the jumper block.

Pins 1 and 2 are the pins closest to the bracket. There are four sets of two pins. Place the jumper on the 1 and 2 pin set as shown in FIGURE G-1.



Caution – The board does not function with the jumper on pins 1 and 2.

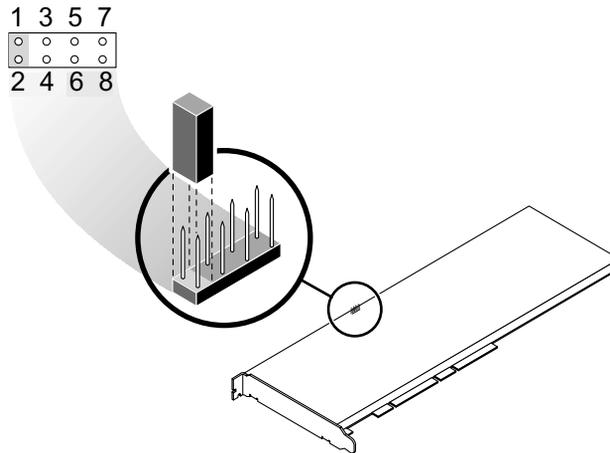


FIGURE G-1 Hardware Jumper Block Pins

4. Power on the system.



Caution – When you power on the system after adjusting the hardware jumper, all firmware, key material, and configuration information is deleted. This process returns the board to the factory state and places the board in Failsafe mode.

5. Power off the system.

6. Remove the jumper from pins 1 and 2 of the jumper block and store the jumper in the original location.

7. Power on the system.

8. Connect to the Sun Crypto Accelerator 4000 board with `vcaadm`.
`vcaadm` prompts you for a path to upgrade the firmware.

9. Type `/opt/SUNWconn/criptov2/firmware/sca4000fw` as the path for installing the firmware.

The firmware is automatically installed and you are logged out of `vcaadm`.

10. Reconnect to Sun Crypto Accelerator 4000 board with `vcaadm`.

`vcaadm` prompts you to either initialize the board with a new keystore, or initialize the board to use an existing keystore. See “Initializing the Board With `vcaadm`” on page 68.

Index

SYMBOLS

`$HOME/.vcaadm/trustdb`, 62
`.properties` command, 203
`.u` extension, 19, 228
`/etc/apache/default.pass`, 234
`/etc/apache/`
 `servername.port.keytype.pass`, 234
`/etc/driver_aliases` file, 39
`/etc/hostname.vcaN` file, 55
`/etc/hosts` file, 56
`/etc/opt/SUNWconn/vca/keydata`, 22, 230
`/etc/path_to_inst` file, 40
`/kernel/drv/vca.conf` file, 199
`/opt/SUNWconn/cryptov2/firmware/`
 `sca4000fw`, 255
`/opt/SUNWconn/cryptov2/include`, 242
`/opt/SUNWconn/cryptov2/lib`, 22, 230
`/opt/SUNWconn/cryptov2/sbin`, 22, 230

NUMERICCS

16-bit loadable counter increments, 46
8-bit vectors, 32

A

administering Sun ONE Web Servers, 105
administrative commands, 22, 230
`adv-asmpause-cap`, 29

`adv-asmpause-cap` parameter, 29
`adv-autoneg-cap`, 26
`adv-autoneg-cap` parameter, 26
advertised link parameters, 27
algorithms, 5
alias read, 31
Apache SSL directives, 233
Apache Web Servers, 19, 228
 directives, 233, 234, 235, 236, 237, 238, 239, 240
 `.htaccess`, 235
 available SSL Ciphers, 235
 cipher preference, 237
 special characters, 237
 SSL aliases, 236
 `SSLCACertificateFile`, 238
 `SSLCARevocationFile`, 238
 `SSLCertificateChainFile`, 237
 `SSLCertificateFile`, 237
 `SSLCertificateKeyFile`, 237
 `SSLCipherSuite`, 235, 237
 `SSLEngine`, 234
 `SSLLog`, 239
 `SSLLogLevel`, 239
 `SSLOptions`, 239
 `SSLPassPhraseDialog`, 233
 `sslpassword`, 233
 `SSLProtocol`, 234
 `SSLRequireSSL`, 240
 `SSLVerifyClient`, 238
 `SSLVerifyDepth`, 238
applications, building, 241
assigning an IP address, 55

- auto-boot? configuration variable, 200, 201
- autonegotiation, 25, 29
 - disabling, 38
 - pause capability, 29
 - setting, 25, 38
 - transmit and receive, 29

B

- blanking register for alias read, 31
- blanking values, 27, 31
- building applications
 - libcrypto.a, 242
 - libssl.a, 242

C

- commands
 - .properties, 203
 - driver.conf, 39
 - ifconfig, 55
 - kstat, 44, 52, 198
 - modinfo, 229
 - pkgadd, 229
 - prtconf, 39
 - prtdiag, 229
 - setenv auto-boot?, 200
 - show-devs, 202
 - show-nets, 200
 - watch-net, 204
 - zeroize, 254
- configuration, network, 54
- configuring device driver parameters, 25
- configuring Sun ONE Web Servers, 110
- configuring the network host files, 54
- cryptographic activity, 198
- cryptographic algorithm acceleration, 3
- cryptographic and Ethernet driver operating statistics, 44
- cryptographic driver operating statistics, 44
- cryptographic driver statistics, 44
- cryptographic libraries, 242
- custom applications, 241

D

- dcatest, 192
 - subtests, 193
- deleting security officers, 77
- detecting 8-bit vectors, 32
- determining cryptographic activity, 198
- device path names, 40
- diagnostic support, 3
- diagnostics tests, 191
- diag-switch? configuration variable, 200
- Diffie-Hellman, 235
- Digital Signature Standard, 235
- directories and files, 22, 230
 - hierarchy of, 22, 230
- displaying board status, 79
- driver parameters, 25
 - configuring, 25
 - forced mode, 26
 - parameters and settings, 26
 - values and definitions, 26
- driver statistic values, 198
- driver statistics, 44, 45
- driver.conf file, 39
- driver_aliases file, 39
- driver-specific parameters, 50
- drop parameters, 32
- DSS, 235
- dynamic reconfiguration, 10

E

- early detecting 8-bit vectors, 32
- early drop parameters, 32
- editing the network host files, 54
- enabling
 - Sun ONE Web Servers, 110
- enabling Sun ONE Web Servers, 112
- entropy, 11
 - high-quality, 11
 - low-quality, 11
- etc/apache/default.pass, 234
- etc/apache/
 - servername.port.keytype.pass, 234
- etc/hostname.vcaN file, 55

etc/hosts file, 56
etc/path_to_inst file, 40
Ethernet
 driver operating statistics, 44
 driver statistics, 45
 FCode self-test diagnostic, 199
 MMF, 25
 PCI properties, 52
 properties, 48
 receive counters, 51
 transmit counters, 50
 UTP, 25
example vca.conf file, 41

F

factory state, 253
Failsafe mode, 253
FCode self-test, 199
FIFO occupancy, 32
files and directories
 installation, 19, 228
FIPS 140-2 mode, 69
firmware, 255
flow control, 29
 frames, 29
 keywords, 29
forced mode of operation, 26
forced mode parameter, 30
Frame Based Link Level Flow Control Protocol, 29

G

gap parameters, 30
Gigabit forced mode parameter, 30
Gigabit media independent interface (GMII), 48

H

hardware, 11
hardware and software requirements, 11
hardware zeroize, 253
high availability, 11

high-quality entropy, 11
host files, 54
hostname.vcaN file, 55
hosts file, 56
hot-plug, 10

I

IEEE 802.3x, 29
ifconfig command, 55
infinitt-burst, 27
infinitt-burst parameter, 27
initializing the board, 23, 231
installation
 directories and files, 22, 230
 files and directories, 19, 228
 software packages, 229
installation script, 19
installing the optional packages, 21, 229
interface
 Gigabit media independent, 48
 media independent, 48
 PKCS#11, 205
 vca interface, 55
interpacket gap parameters, 30
interrupt blanking values, 27, 31
interrupt parameters, 31
ipg0, 30
ipg0 parameter, 30
ipg1, 30
ipg1 parameter, 30
ipg2, 30
ipg2 parameter, 30

K

kernel statistic values, 198
kernel/drv/vca.conf file, 199
key length, 178
key objects, 72
keystore data, 22, 230
keystores, 69, 70, 106
 managing with vcaadm, 71

kstat command, 44, 52, 198

L

libcrypto.a parameter, 242
libraries, cryptographic, 242
libssl.a parameter, 242
link capabilities, 28
link parameters, 27
link partner, 25, 29, 48, 52
 checking, 52
 settings, 52
link-master, 26
link-master parameter, 26
load balancing, 11
load sharing, 11
locking to prevent backups, 78
long-term keys, 11

M

man page descriptions, 251
media independent interface (MII), 48
MMF, 25
mode, FIPS 140-2, 69
modinfo command, 229

N

name property, 25
naming requirements, 72
ndd utility, 34
network configuration, 54
network host files, 54
nostats property, 199

O

OBP commands
 .properties, 203
 reset-all, 200
 setenv auto-boot?, 200

 setenv diag-switch?, 201
 show-devs, 202
 show-nets, 200
 test device_path, 201
 watch-net, 204

OBP configuration variables

 auto-boot?, 200, 201
 diag-switch?, 200

OBP PROM, 199, 202

occupancy, FIFO, 32

online manual pages, 251

 apsslcfg(1m), 251
 iplsslcfg(1m), 251
 kcl2(7d), 251
 vca(7d), 251
 vcaadm(1m), 251
 vcad(1m), 251
 vcadiag(1m), 251

OpenBoot PROM, 42, 199, 202

OpenBoot PROM FCode self-test, 199

OpenSSL-compatible applications, 241

operating environment, 11

operating statistics, 44

operational mode parameters, 27, 28

opt/SUNWconn/cryptov2/firmware/
 sca4000fw, 255

opt/SUNWconn/cryptov2/include, 242

optimize throughput, 11

optional packages, 19, 228

 descriptions, 19, 228
 installing, 21, 229

P

packages

 optional, 228
 required, 228

parallel-detection, 43

parameter values

 how to modify and display, 35

parameters, 27

 8-bit vectors, 32
 adv-asmopause-cap, 29
 adv-autoneg-cap, 26
 driver-specific, 50
 early detecting 8-bit vectors, 32

- early drop, 32
- flow control, 29
- forced mode, 30
- Gigabit forced mode parameter, 30
- infinet-burst, 27
- interpacket gap, 30
- interrupt, 31
- ipg0, 30
- ipg1, 30
- ipg2, 30
- libcrypto.a, 242
- libssl.a, 242
- link, 27
- link capabilities, 28
- link-master, 26
- operational mode, 28
- pause-off-threshold, 26
- PCI bus interface, 33
- RX random early detecting 8-bit vectors, 32
- rx-intr-pkts, 27, 31
- rx-intr-time, 31
- setting for all vca devices, 41
- setting with `vca.conf` file, 39, 41
- parameters and settings, 26
- password requirements, 72
- passwords
 - list required for Sun ONE Web Servers, 110
 - system administrator, 111
 - `vcaadm`, 72, 111
- patches, 12
 - required, 12
 - Solaris 8, 12
 - Solaris 9, 13
- path names, 40
- `path_to_inst` file, 40
- pause capability, 29
- pause-off-threshold, 26
- pause-off-threshold parameter, 26
- PCI adapters, 25
- PCI bus interface parameters, 33
- `pci` name property, 25
- PKCS#11 interface, 75, 205
- PKCS#11 interface definitions for users, 106
- `pkgadd` command, 229
- platforms, 11
- product features, 1

- properties
 - Ethernet, 48
 - Ethernet PCI, 52
 - `nostats`, 199
- protocols and interfaces, 2
- `prtconf` command, 39
- `prtdiag` command, 229

Q

- quitting `vcaadm`, 68

R

- random early detecting 8-bit vectors, 32
- random early drop parameters, 32
- read-only link partner capabilities, 49
- read-only `vca` device capabilities, 48
- read-write flow control, 29
- receive counters, 51
- receive interrupt blanking values, 27, 31
- receive MAC counters, 46
- receive random early detecting 8-bit vectors, 32
- register for alias read, 31
- request coalescing, 11
- required packages, 228
- required patches, 12
- RSA keypair, 177
- RX blanking register for alias read, 31
- RX MAC counters, 46
- RX random early detecting 8-bit vectors, 32
- `rx-intr-pkts`, 27, 31
- `rx-intr-pkts` parameter, 27, 31
- `rx-intr-time`, 31
- `rx-intr-time` parameter, 31

S

- security officer accounts, 72
- security officers, 73
- self-test, 199
- server certificate, 118, 127

- setenv auto-boot?, 200
- setting vca driver parameters
 - using ndd, 34, 39
 - using vca.conf, 34, 39
- show-devs command, 202
- show-nets command, 200
- software packages, 229
- Solaris 9 patches, 13
- Solaris operating environments, 11
- specifications, 220, 221, 222, 223, 224, 225
 - MMF adapter, 220, 221, 222
 - characteristics, 220
 - environmental specifications, 222
 - interface specifications, 222
 - performance specifications, 221
 - power requirements, 221
 - UTP adapter, 222, 223, 224, 225
 - characteristics, 223
 - connectors, 222
 - environmental specifications, 225
 - interface specifications, 225
 - performance specifications, 224
 - physical dimensions, 224
 - power requirements, 224
- speed=
 - 10, 42
 - 100, 42
 - 1000, 42
 - auto, 42
- SSL acceleration, 5
- SSL algorithms, 4
- standard Ethernet frame sizes, 2
- standards and protocols, 2
- statistic values, 198
- Sun cryptographic libraries, 242
- Sun ONE Application Server 7, 133
 - binary and domain paths, 94, 137
 - configuring, 135
 - installing a server certificate, 141
 - installing the add-on SSL utilities, 135
 - iplsslcfg script, 137
 - trust database, 136
- Sun ONE Directory Server 5.2
 - enabling SSL, 154
 - generating a server certificate, 151
 - installing, 146
 - installing a server certificate, 152
 - registering the board, 149
 - root CA certificates, 152, 173
 - starting manually, 147
 - trust database, 147
- Sun ONE Messaging Server 5.2
 - enabling SSL, 168
 - installing, 158
 - installing certificates, 165
 - registering the board, 160
 - server certificates, 160
 - trust database, 159
- Sun ONE Portal Server 6.2, 169
 - configuring, 171
 - enabling SSL, 174
 - generating a server certificate, 172
 - installing, 170
 - installing a server certificate, 173
- Sun ONE Web Servers
 - administering, 105
 - configuring, 110
 - creating and populating a keystore, 111
 - enabling, 112
 - passwords, 110
- Sun ONE Web Server 4.1
 - configuring, 120
 - creating a trust database, 115
 - generating a server certificate, 115
 - installing, 113
 - installing the server certificate, 120
- Sun ONE Web Server 6.0
 - creating a trust database, 124
 - generating a server certificate, 127
 - installing, 123, 133
 - installing a server certificate, 130
- token files, 108
- tokens, 108
- SunVTS, 190, 191
 - netlbttest, 194
 - nettest, 195
 - required software, 190
 - software, 189
 - vca driver, 190
 - vcatest
 - command-line syntax, 193
 - test parameter options, 193
 - vcatest, 191
- SunVTS 4.4, 19, 228
- SunVTS 5.1 Patch Set (PS) 2, 189

- SunVTS 5.x, 19, 228
- support libraries, 22, 230
- supported
 - algorithms, 5
 - cryptographic algorithms, 4
 - hardware, 11
 - operating environments, 11
 - platforms, 11
 - software, 11
 - Solaris operating environments, 11
 - SSL algorithms, 5

T

- token files, 108
- tokens, 108
- transmit and receive pause capability, 29
- transmit counters, 50
- transmit MAC counters, 46
- troubleshooting, 202
- trust database
 - creating
 - Sun ONE Web Server 4.1, 115
 - Sun ONE Web Server 6.0, 124
 - vcaadm, 62
- TX and RX MAC counters, 46
- TX MAC counters, 46

U

- UNIX `pci` name property, 25
- URL
 - for OpenSSL, 241
 - for Sun ONE software, 114, 123, 133, 135, 146, 158, 170
- user accounts, 72
- user concepts and terminology, 106
- utilities, 22, 230
- UTP, 25

V

- values and definitions, 26
- vca driver, 190

- required software, 190
- vca driver parameters
 - configuring, 25
 - forced mode, 26
 - parameters and settings, 26
 - values and definitions, 26
- vca interface, 55
- vca.conf file, 39
- vca.conf file, example, 41
- vcaadm
 - populating a keystore
 - with security officers, 73
 - with users, 74
- vcaadm
 - backups, 77
 - changing passwords, 75
 - character requirements, 72
 - command-line syntax, 60
 - deleting users, 77
 - diagnostics command, 83
 - enabling and disabling users, 76
 - entering commands, 66
 - file mode, 62
 - getting help, 67
 - initializing the board, 68
 - interactive mode, 62
 - listing security officers, 75
 - listing users, 75
 - loading new firmware, 80
 - locking to prevent backups, 78
 - logging in and out, 62
 - managing boards, 78
 - modes of operation, 61
 - naming requirements, 72
 - options, 60
 - password requirements, 72
 - prompt, 65
 - quitting, 68
 - rekeying a board, 81
 - resetting a board, 80
 - setting auto-logout, 79
 - user name requirements, 72
 - using, 59
 - utility, 59
- vcadiag
 - command-line syntax, 89
 - examples, 90, 91
 - options, 90

- using, 89
- utility, 89

vectors, 32

W

watch-net command, 204

Z

zeroize command, 254

zeroizing the hardware, 253