Diamond*NIC* & Diamond*Link* Security Policy

Version 2.3

Background

This document seeks to focus attention on the security policy requirements of FIPS 140-1 as well as the validation requirements imposed by the Derived Test Requirements[1] (DTR). The following are some of the key statements made concerning a cryptographic module's security policy:

The requirement for a security policy is initially stated in FIPS 140-1[2] as follows:

> Documentation shall also completely specify the cryptographic module security policy, i.e., the security rules under which a module must operate. In particular, the security policy shall include the security rules derived from the security requirements of this standard and the security rules derived from any additional security requirements imposed by the manufacturer.

In reference to this statement from FIPS 140-1, the DTR make the following statement regarding documentation that the vendor is required to provide the validation laboratory:

> VE01.07.01: The vendor shall provide a separate document, or section of a document, that specifies the security policy (i.e., the security rules under which a module must operate) enforced by the cryptographic module.

In Appendix A of the DTR a statement of purpose for the security policy is given:

> There are three major reasons for developing and following a precise cryptographic module security policy:
>
> 1. To induce the cryptographic module vendor to think carefully and precisely about who he wants to access the cryptographic module, the way different system elements can be accessed, and which system elements to protect.
>
> 2. To provide a precise specification of the cryptographic security to allow individuals and organizations (e.g., validators) to determine whether the cryptographic module, as implemented, does obey (satisfy) a stated security policy.
>
> 3. To describe to the cryptographic module user (organization, or individual operator) the capabilities, protections, and access rights they will have when using the cryptographic module.

From the above, a number of important points can be drawn:

- The statement of the security policy must be a clear and concise specification of the principles to be used guide the design decisions.

---

[1] The Derived Test Requirements for FIPS 140-1 is a document developed for NIST by MITRE. This document provides guidance to a validation lab on the interpretation of the FIPS standard and defines actual tests to be performed by the lab during the validation process. This Derived Test Requirements will be referred to as the DTR in the remainder of this document.

[2] FIPS 140-1 § 4.1 paragraph 4.

- Though the DTR states that it can be a separate section of a document, it needs to be in a form that can be provided to potential users. This suggests that the security policy should be a separate document.
- The security policy must address all of the applicable requirements of FIPS 140-1.
- The security policy must address any additional security requirements imposed by the manufacturer to achieve the goals of their design.

A. Scope of Document

This document defines the cryptographic security policy for the Diamond*NIC* and Diamond*Link* network interface devices. The following figure shows a block diagram of the Diamond*NIC* and denotes the cryptographic boundary as dashed lines.

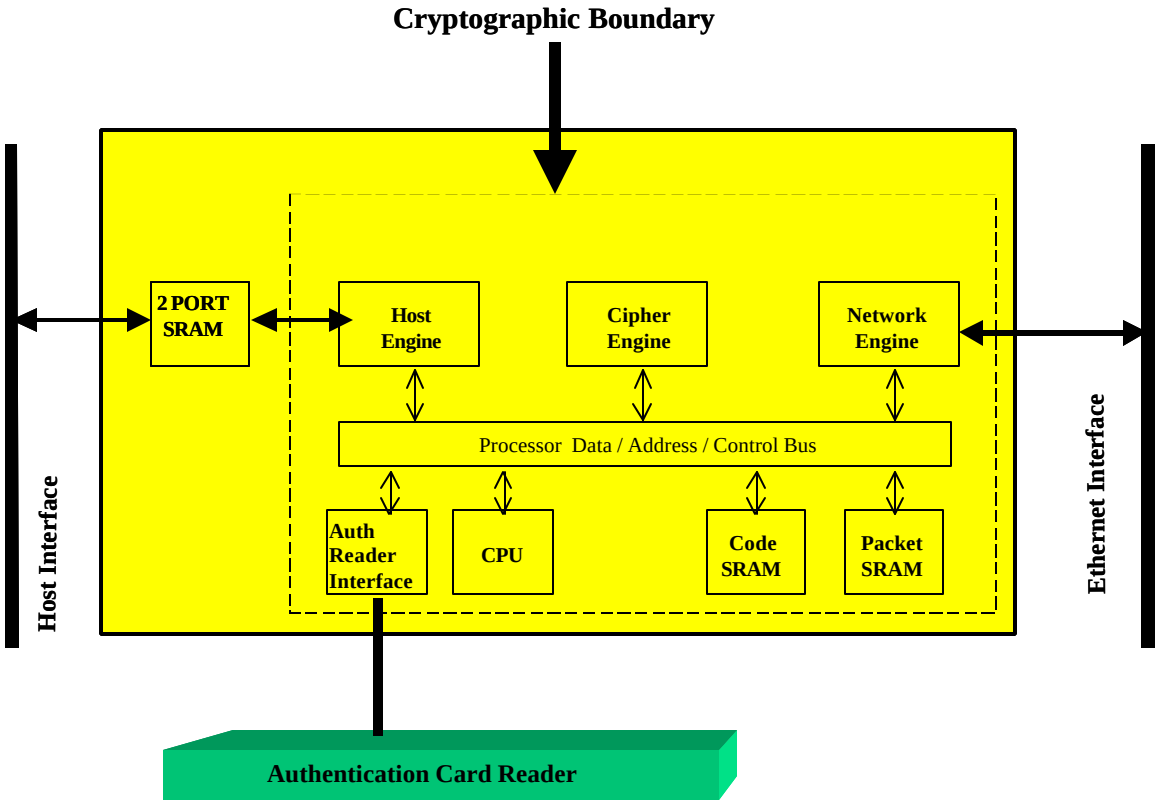

Figure 1. Diamond*NIC* Cryptographic Boundary (Multi-chip embedded module)

As can be seen from the above figure, the Diamond*NIC* is a multi-chip embedded module as described in FIPS 140-1. The design of the Diamond*NIC* includes multiple host buses including; PCI, ISA, and Sbus. In addition, a device called a Diamond*Link* is one of the configurations that utilize the Diamond*NIC* architecture. The Diamond*Link* is a multi-chip standalone module as described in FIPS 140-1. Figure 2 shows a block diagram of the Diamond*Link*.

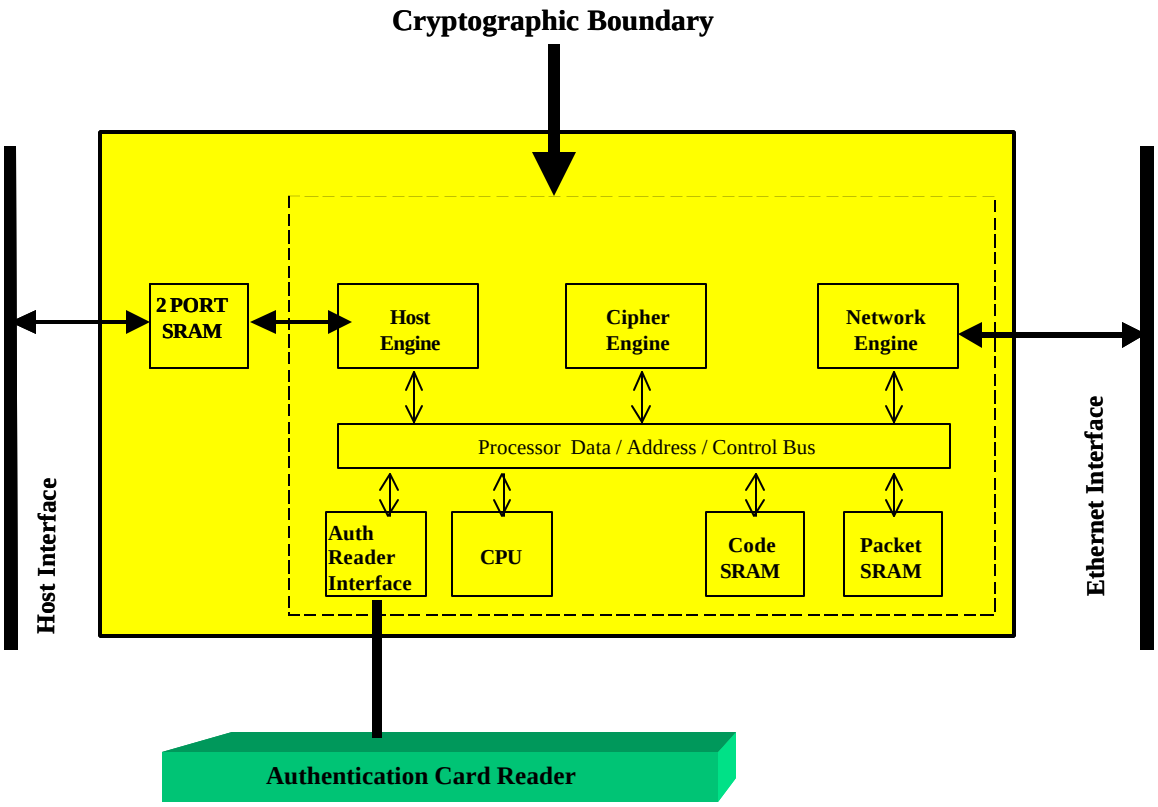

Figure 2. Diamond*Link* Cryptographic Boundary (Multi-chip standalone module)

From this point forward in this document, all references to a DiamondNIC also are also references to the Diamond*Link*.

B. Security Level

The cryptographic module meets the overall requirements applicable to Level 2 security of FIPS 140-1.

Table 1. Module Security Level Specification

Security Requirements Section	Level
Cryptographic Module	1
Module Interfaces	1
Roles and Services	2
Finite State Machine	1
Physical Security	1
Software Security	1
Operating System Security	N/A
Key Management	1
Cryptographic Algorithms	1
EMI/EMC	1
Self Test	1

C. Roles and Services

The cryptographic module shall support three distinct operator roles. These operator roles are:

1. User Role
2. Cryptographic Officer Role
3. Administrator Role

The *User Role* shall provide all of the services necessary for the secure transport of data over an insecure network. This includes the following services:

- Device Initialization

  The Diamond*Link* prohibits the flow of network datagrams until the Diamond*Central* manager instructs it to transition to the *ONLINE* state. The only exception to this policy is for Dynamic Host Configuration Protocol (DHCP) protocol data units (PDU). DHCP traffic will flow through the Diamond*Link* if the following conditions are met:

  (1) The Diamond*Link* has been configured as a DHCP node (during installation time) and,

  (2) The host behind the Diamond*Link* is sending DHCP messages, namely the Discovery Message PDU.

  The Diamond*Link* permits the following types of DHCP PDU on outbound traffic: Discovery, Request, Inform, Decline, Release and the following types on inbound traffic: Offer, ACK, and NACK. The Diamond*Link* ascertains its address by monitoring the DHCP traffic between the host that it's protecting and the DHCP server. When the host sends an ACK PDU in response to an DHCP Offer PDU, the Diamond*Link* uses the IP address and subnet mask fields. Once addressed the Diamond*Link* can validate the user and hold the policy from the Diamond*Central* manager.

- Process Transmit Packet.

  The packet transmit process is instigated by the placement of a packet into the shared memory of the Diamond*NIC* and an interrupt from the host to indicate that the packet is ready for transmit.

The Diamond*NIC* brings the packet into internal only addressable memory and performs a lookup for the destination address. If the address is not found in the allowable address table, then the packet is audited and discarded by the Diamond*NIC*. If the packet is addressed to an approved destination, the label associated with the packet is checked to determine if it is allowable for transmit. If the label is inconsistent with the classification window of the originating Diamond*NIC*, the packet is audited and discarded. If the destination of the packet is to a non Diamond*NIC*, the sending Diamond*NIC* will check its tables to determine if the destination node can receive the label associated with the outgoing packet. If the label is not consistent with the label associated with the destination host, then the packet is audited and discarded.

If the Diamond*NIC* determines that the packet can be sent to the destination address and the packet is not to be encrypted (based on the encryption policy provided by the Diamond*Central*), it shall send the packet onto the network.

If the Diamond*NIC* determines that the packet can be sent to the destination address and the packet is to be encrypted (based on the encryption policy provided by the Diamond*Central*), it shall check to determine if there is a cipher/authentication key set for the destination/label pair. If a key does not exist, the Diamond*NIC* shall discard the packet and instigate an IKE key generation mechanism to create keys for use in the encryption and authentication of the packets from the attached host and the destination address. The key generation process will attempt to generate both transmit and receive keys but may not be able to complete the process based on the policy being enforced by the destination node.

Once the keys are available, the next packet for the destination address with the same label will find that the keys are available. The keys are retrieved from the local table (not shared with the host system) and are used to format (IPSec ESP), encrypt, and authenticate the packet.

- Process Receive Packet

The packet receive process is instigated by the reception of a packet from the network. This event is recognized by the Diamond*NIC* firmware by the issuance of an interrupt from the network coprocessor to the Diamond*NIC* CPU. The issuance of the interrupt signifies that a packet has been receive by the coprocessor and placed in the network/Diamond*NIC* CPU shared memory area.

If the received packet is clear text, then the discretionary access control list (which is downloaded to the Diamond*NIC* from the Diamond*Central*) is checked to determine if the Diamond*NIC* is allowed to receive clear text packets from the source address. If clear text packets are not to be received from the source address, the packet is audited and discarded. If clear text is allowable for the source address, the packet is copied into the host shared memory and notification is posted to the host that a receive packet is available.

If the received packet is encrypted, then the discretionary access control list (which is downloaded to the Diamond*NIC* from the Diamond*Central*) is checked to determine if the Diamond*NIC* is allowed to receive encrypted packets from the source address. If encrypted packets are not to be received from the source address, the packet is audited and discarded. If the packet is encrypted, then the information pertaining to the source address is checked to access the appropriate key. If a key does not exist, the Diamond*NIC* shall discard the packet and instigate an IKE key generation mechanism to create keys for use in the encryption and authentication of the packets from the attached host and the source address. The key generation process will attempt to generate both transmit and receive keys but may not be able to complete the process based on the policy being enforced by the source node. If the key set does exist, then the packet is authenticated and decrypted. If the authentication fails, the packet is discarded and an audit event is generated. If the authentication succeeds, the packet is then copied into the host shared memory and the host is notified.

- Change the state of the Diamond*NIC*

  The user can cause the Diamond*NIC* to transition to an operational state (suspended or on-line) by inserting the authentication card. The insertion of the authentication card instigates communication with the Diamond*Central* to determine which state (suspended, on-line, or offline) to which the Diamond*NIC* should transition.

  To cause the Diamond*NIC* to transition from an operational state (suspended or on-line) to the off-line state, the user can remove the authentication card from the card reader.

  When the user re-inserts the authentication card, the Diamond*NIC* will perform its self-tests. This allows the user to initiate the self-tests performed by the system at any time.

- Cause the device to perform self-test.

  The user can cause the Diamond*NIC* to perform its self-test at any time by removing and re-inserting the authentication card. The act of inserting an authentication card automatically causes the Diamond*NIC* to perform self-tests.

The *Cryptographic Officer Role* is provided by the use of a special cryptographic officer authentication card. The cryptographic officer role shall provide the service to:

- Load static user credentials in flash memory.

- Load configuration data; including the IP address of the Diamond*Central*, the IP address of the default router, the approved authentication mechanisms and the approved encryption algorithms that can be used by the device.

- Load Diamond*Central* shared secret - Load the shared secret used to create the encrypted channel (administrator interface) between the Diamond*NIC* and the Diamond*Central*.

The *Administrator Role* commu*nic*ates with the Diamond*NIC* via an encrypted channel over the network. The administrator roll shall include the following services:

- Update authentication values

  The administrator, via the Diamond*Central*, shall update the shared secret and user authentication information upon successful initialization of the Diamond*NIC*.

- Configure the Diamond*NIC* per a predefined policy

  Upon successful authentication, the Diamond*Central* shall download via the administrator interface the security policy to be implemented by the Diamond*NIC*. This policy includes data associations (node to node communication) defined to be clear text and encrypted. All data associations not defined are deemed invalid and will cause the packet to be destroyed and audited.

- Change the state of the Diamond*NIC* from on-line to suspended, suspended to on-line, on-line to off-line.

  The Diamond*Central* will command the Diamond*NIC* to transition to various states (suspended, on-line, off-line) via the administrator interface.

- Zeroize the Diamond*NIC*

  The Diamond*Central* can zeroize all data keys as well as the shared secret used for communication with the Diamond*Central* via the administrator interface.

- Update Diamond*NIC* firmware

  The Diamond*Central* can send to the Diamond*NIC* a firmware update which is authenticated with a key and the DES MAC mechanism. The key for the DES MAC is sent to the Diamond*NIC* via the administrator interface.

.

D. Security Rules

This section documents the security rules enforced by the cryptographic module (Diamond*NIC*) to implement the security requirements of this FIPS 140-1 Level 1 module[3].

Upon the application of power or when commanded by the cryptographic module shall perform the following tests:

- *RAM test*
- *FLASH checksum test*
- *Cipher known answer tests*
- *Network diagnostic tests*
- *Timer diagnostic tests*
- *Exponentiation known answer test*
- *Continuous Random Number Generator Test as specified in FIPS 140-1 §4.11.2 paragraph 5*

The user shall be capable of commanding the module to perform the power-up self test by removing and re-insertion of the authentication card.

The cryptographic module shall provide three distinct operator roles. These are the User Role, the Cryptographic Officer Role, and the Administrator Role.

The DiamondNIC shall provide role based authentication.

- Possession of the user authentication card provides access to the Diamond*NIC* (not necessarily the network) for the user role. Possession of the crypto officer card provides authentication for the crypto officer role. Possession of the shared secret (which is 32 bytes – 24 for triple des operation and 8 for DES-MAC authentication) provides authentication for the administrator role.

The DiamondNIC shall not communicate with the DiamondCentral to allow a user to login to the device until after it has been initialized with a crypto officer card and a user authentication card has been inserted.

- After reading the configuration information from the crypto officer card and updating the Diamond*Central* shared secret and communication data, the Diamond*NIC* will transition to the offline state and await the insertion of another card.

---

[3] Rules are contained in the number paragraphs and are shown in italics. Other information is included for background purposes only.

The DiamondNIC shall recognize a user authentication card and attempt to initialize with the DiamondCentral using data on the DiamondCentral shared secret, authentication card and the profile selected by the user.

The user is disallowed after one invalid attempt to initialize with the DiamondCentral.

The user shall not have access to any cryptographic services unless the DiamondNIC has been commanded to transition to the online state by the DiamondCentral.

The DiamondNIC shall have a bypass mode that is enabled by setting a switch on the board, insert a valid principal card into the card reader and selecting profile 99. In this mode, no cryptographic services will be available to the user.

The DiamondCentral shall, before allowing the DiamondNIC to transition to the online state, download a transmit and receive mandatory access control policy to the DiamondNIC. This policy shall include a maximum and minimum transmit window as well as an allowable and mandatory transmit and receive category set.

- All outgoing packets shall have a security level between the maximum and minimum transmit level and a category set that is a superset of the mandatory and a subset of the allowable category values.
- All incoming packets shall have a security level between the maximum and minimum transmit classification level and a category set that is a superset of the mandatory and a subset of the allowable category values.

The DiamondCentral shall download communication rules (DAC policy) to the DiamondNIC. The policy shall be reconfigurable by the DiamondCentral at any time. These rules define the communication paths as follows:

- Valid destination addresses for packets sent from the attached host to the network.
- Valid source addresses for packets being sent to the attached host from the network.
- Allowable/prohibited TCP and UDP port values for transmission and reception by the host.
- The encryption algorithm used to secure the IPSec packet (DES or 3DES).
- The authentication mechanism used to secure the IPSec packet (DES MAC or SHA-1).

The DiamondNIC shall generate audits for all attempted Mandatory and Discretionary Access Control (MAC and DAC) violations.

The DiamondNIC shall generate audits for all received encrypted packets that do not pass the message authentication code test.

The DiamondNIC shall send all auditable events to the DiamondCentral for logging.

The DiamondCentral shall download a non-security auditing policy to include statistical, broadcast and TCP Open/Close events. These audit events shall be sent to the DiamondCentral for logging.

The shared secret between the DiamondNIC and the DiamondCentral shall be changed during each initialization.

The DiamondNIC shall determine the encryption and authentication algorithms and keys based on the shared secret method of the IKE standard.

There shall be a different key for each host/ label of data combination.

The DiamondNIC shall accept a firmware update from the DiamondCentral if the update passes a DES Message Authentication Code check using firmware update key sent to the DiamondNIC from the DiamondCentral via the encrypted control communication.

The DiamondNIC shall accept state control commands (suspend, online, and shutdown) commands from the DiamondCentral via the encrypted control communication.

The DiamondCentral shall be capable of zeroizing the DiamondCentral shared secret stored in the DiamondNIC.

If the user authentication card is removed from the DiamondNIC, the cryptographic module shall notify the DiamondCentral and change its state to offline via the encrypted control communication.

The data communication keys shall be zeriozed when the authentication card is removed.

While not in bypass mode, the switch that governs the bypass mechanism is off.

E. Definition of Security Relevant Data Items

The DTR specifies the following test.

> TE01.07.01: The tester shall review the security policy specification provided by the vendor. Specifically, he or she must determine that it identifies all roles, services, and security relevant data items of the cryptographic module, and specifies what access, if any, a user, performing a service within the context of a given role, has to each of the security relevant data items. The specification should be complete, and detailed enough to be able to answer the following question: "What access does operator X, performing service Y while in role Z have to security relevant data item K?" for every role, service, and security relevant data item contained in the cryptographic module.

This and the following two sections address information that must be included in the security policy to address this and other similar tests.

There are 7 types of cryptographic security relevant data items (SRDIs). These are:

Diamond*Central* shared secret (**DCSS**) – Used to provide encrypted communication between the Diamond*NIC* and the Diamond*Central* for the administrator interface.

Traffic encryption keys (**TEK**s) – Used to encrypt the traffic between the Diamond*NIC* and another Diamond*NIC* or other IPSec device. These are generated as part of the IKE key generation process.

Traffic authentication keys (**TAK**s) – Used to authenticate traffic between the Diamond*NIC* and another Diamond*NIC* or other IPSec device. These are generated as part of the IKE key generation process.

DH private keys (**DHPK**) – Generated by the Diamond*NIC* for each used level of classification and used as part of the IKE key generation process.

Firmware update key (**FWUK**) – Sent to the Diamond*NIC* by the Diamond*Central* as part of the firmware update sequence. The firmware is stored in RAM and a DES_MAC is calculated on the firmware using the update key. If the computed value is the same as the value sent from the Diamond*Central* then the firmware in the flash is replaced by the new firmware.

Association Table (**DAT**) – The list of approved source and destination addresses (IP address and TCP/UDP port numbers).

Node authentication values (**NAV**) – A shared secret used as the authentication mechanism for the IKE key generation process.

F. Definitions of SRDI Modes of Access

The table below defines the relationship between access to SRDIs and the different module services. The modes of access shown in the table are defined as follows:

a) Transmit Packet Processing: The operation to transmit a packet shall first access the current state (**DS**) of the Diamond*NIC*. If the Diamond*NIC* is not on-line, then the packet is not processed until the state changes to on-line. If the Diamond*NIC* is on-line, then the discretionary access control list (**DAT**) is checked to determine if communication is allowable. If the destination is not allowable (because of IP address or TCP/UDP port number) then the packet is destroyed and an audit event is generated.

If the **DAT** signifies that the destination is allowable and is clear text, then the transmit security window (**DSW**) is accessed to determine if the Diamond*NIC* can transmit that particular label. If the label can not be transmitted then the packet is destroyed and an audit event is generated. If the label is within the bounds of the transmit window of the Diamond*NIC*, then the **DAT** is checked to determine if the receiving address is allowed to receive the label associated with the address. If the packet label can not be received by the destination address, then the packet is destroyed and an audit event is generated. If the label can be received by the destination address, then the packet is transmitted to the network.

If the **DAT** signifies that the destination is allowable and communication is to be encrypted, then the keys associated with the destination (**TEK** and **TAK**) are accessed to determine if there is a key for the label associated with the packet.

If a key exists, then it is used to encrypt the packet and the key associated with the authentication mechanism (**TAK**) is used to perform the authentication of the packet. If the useful life of the key has been exhausted, then the keys (cipher and authentication) associated with the destination address are destroyed. After the encryption and authentication is complete, the packet is transmitted to the network.

If no key exists for the destination/label pair, then the Diamond*NIC* shall check the label of the packet against the transmit window of the Diamond*NIC* (**DSW**). If the label can not be transmitted, then it packet is destroyed and an audit event is generated. If the packet is within the bounds of the transmit window and the destination address may not be a Diamond*NIC*, then the label of the packet is checked against the label defined for the destination address in the **DAT**. If the label of the packet is not a subset of the label of the destination address, then the packet is destroyed and an audit event is generated. If the destination address is a Diamond*NIC* or the label of the packet is a subset of the label associated with the destination address, then the packet is destroyed and an IKE process is instigated.

The IKE process will utilize the list of approved encryption algorithms (**ACAL**) and the list of approved authentication algorithms (**AAAL**) to negotiate an acceptable combination to secure the information between the new nodes. If the Diamond*NIC* does not have a diffie-hellman private value generated for the classification level, then a private (**DHPK**) and public value (**DHLK**) are generated. The diffie-hellman data, the shared secret (**NAV**) associated with the destination address and random data generated as part of the IKE protocol are used to generate the keying material (**TEK** and **TAK**) to secure the communications between the Diamond*NIC* and the destination address.

b) Receive Packet Processing:. The operation to receive a packet shall first access the current state (**DS**) of the Diamond*NIC*. If the Diamond*NIC* is not on-line and the packet is not from the Diamond*Central*, then the packet thrown away and the network buffer is returned to the network coprocessor. If the Diamond*NIC* is on-line, then the discretionary access control list (**DAT**) is checked to determine if communication is allowable. If the source is not allowable (because of IP address and SPI number) then the packet is destroyed and an audit event is generated.

If the **DAT** signifies that the destination is allowable and is clear text, then the receive security window (**DSW**) is accessed to determine if the Diamond*NIC* can receive that particular label. If the label cannot be received then the packet is destroyed and an audit event is generated. If the label is within the bounds of the receive window of the Diamond*NIC*, then the **DAT** is checked to determine if the sending address is allowed to send the label associated with the address. If the packet label can not be sent by the source address, then the packet is destroyed and an audit event is generated. If the label can be sent by the source address, then the packet is passed to the host system.

If the **DAT** signifies that the source is allowable and communication is supposed to be encrypted, then the keys associated with the destination (**TEK** and **TAK**) are accessed to determine if there is a key for the label associated with the packet.

If a key exists, then it is used to decrypt the packet and the key associated with the authentication mechanism (**TAK**) is used to perform the authentication of the packet. After the authentication and is complete, the packet is checked for allowable TCP/UDP port numbers. If the protocol is not TCP/UDP or the **DAT** signifies that the port number is acceptable, then the packet is given to the host system

If no key exists for the source/label pair, then the Diamond*NIC* shall check the label of the packet against the receive window of the Diamond*NIC* (**DSW**). If the label can not be received, then it packet is destroyed and an audit event is generated. If the packet is within the bounds of the receive window and the source address may not be a Diamond*NIC*, then the label of the packet is checked against the label defined for the source address in the **DAT**. If the label of the packet is not a subset of the label of the source address, then the packet is destroyed and an audit event is generated. If the source address is a Diamond*NIC* or the label of the packet is a subset of the label associated with the source address, then the packet is destroyed and an IKE process is instigated.

The IKE process will utilize the list of approved encryption algorithms (**ACAL**) and the list of approved authentication algorithms (**AAAL**) to negotiate an acceptable combination to secure the information between the new nodes. If the Diamond*NIC* does not have a diffie-hellman private value generated for the classification level, then a private (**DHPK**) and public value (**DHLK**) are generated. The diffie-hellman data, the shared secret (**NAV**) associated with the source address and random data generated as part of the IKE protocol are used to generate the keying material (**TEK** and **TAK**) to secure the communications between the Diamond*NIC* and the source address. If existing key material exists for the communications channel, then the old keying material (**TEK** and **TAK**) are zeroized and replaced with the new values.

c) Load Diamond*Central* shared secret: The load Diamond*Central* shared secret function requires the use of a crypto officer authentication card. This card identifies its user as a crypto officer and contains the shared secret used by the Diamond*NIC* for communication with the Diamond*Central*. The Diamond*NIC* will copy the information from the card and store it in its on-board FLASH memory (**DCSS**).

d) Update authentication values: The administrator (via the Diamond*Central*) shall download (under protection of the encrypted communication between the Diamond*NIC* and the Diamond*Central* using the **DCSS**) new secret values each time a user successfully logs into the Diamond*NIC*.

e) Configure the Diamond*NIC* per a predefined policy: The administrator (via the Diamond*Central*) shall download (under protection of the encrypted communication between the Diamond*NIC* and the Diamond*Central* using the **DCSS**) the defined association table (**DAT**), the defined security window (**DSW**) and node authentication values (**NAV**) each time a user successfully logs into the Diamond*NIC*. The change could be an addition or a removal of the ability to send/receive packets to other host systems. In the case of a removal, any traffic encryption keys (**TEK**) or traffic authentication keys (**TAK**) used for communication between the node and the removed destination node are zeroized.

f) Zeroize Diamond*NIC*: The administrator can zeroize the keys stored and in use by the Diamond*NIC*. The command is sent via the encrypted communication channel setup by the **DCSS**. The command will zeroize the **DCSS**, traffic keys (**TEK** and **TAK**), the diffie-hellman values (**DHPK** and **DHLK**), the association table (**DAT**), the security window (**DSW**), the node authentication values (**NAV**), approved crypto algorithm list (**ACAL**) and the approved authentication algorithm list (**AAAL**).

g) Update Diamond*NIC* firmware: The administrator (via the Diamond*Central*) can send a new version of the firmware of the Diamond*NIC* via the encrypted channel setup by the **DCSS**. The Diamond*Central* will first send an authentication key (**FWUK**) and the firmware. The Diamond*NIC* shall verify the signature of the firmware and only update the firmware if the signature is verified. Once the firmware is updated, the Diamond*NIC* will zeroize the **FWUK** and reset its self.

G. Service to SRDI Access Operation Relationship

The table on this page has been devised to show these relationships:

Table 2 Services Versus SRDI Access	DCSS	TEK	TAK	DHPK	FWUK	DAT	NAV	**U**	**C**	**A**
Process Transmit Packet		WAZ	WAZ	WA		A	A	X		
Process Receive Packet		WAZ	WAZ	WA		A	A	X		
Load DiamondCentral shared secret	W								X	
Update authentication values	W									X
Configure the DiamondNIC per a predefined policy	A	Z	Z			W	W			X
Zeroize DiamondNIC	Z	Z	Z	Z		Z	Z			X
Update Firmware	A				WAZ					X

In the above table, access to the SRDIs via the service utilizes the following abbreviations:

- A = Access (note that the actual value is never seen outside the security perimeter so it is not technically a read)
- W = Write
- Z = Zeroize

In the table above, access to the services by individuals is shown by placing an X in the appropriate column. Note that U is used to signify a User, C is used to signify crypto officer and A is used to signify administrator.