The ATM Forum

Technical Committee

Addendum to PNNI v1.0 - Secure Routing

AF-RA-0171.000

November 2001

The information in this publication is believed to be accurate as of its publication date. Such information is subject to change without notice and The ATM Forum is not responsible for any errors. The ATM Forum does not assume any responsibility to update or correct any information in this publication. Notwithstanding anything to the contrary, neither The ATM Forum nor the publisher make any representation or warranty, expressed or implied, concerning the completeness, accuracy, or applicability of any information contained in this publication. No liability of any kind shall be assumed by The ATM Forum or the publisher as a result of reliance upon any information contained in this publication.

The receipt or any use of this document or its contents does not in any way create by implication or otherwise:

- Any express or implied license or right to or under any ATM Forum member company's patent, copyright, trademark or trade secret rights which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor
- Any warranty or representation that any ATM Forum member companies will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor
- Any form of relationship between any ATM Forum member companies and the recipient or user of this document.

Implementation or use of specific ATM standards or recommendations and ATM Forum specifications will be voluntary, and no company shall agree or be obliged to implement them by virtue of participation in The ATM Forum.

The ATM Forum is a non-profit international organization accelerating industry cooperation on ATM technology. The ATM Forum does not, expressly or otherwise, endorse or promote any specific products or services.

NOTE: The user's attention is called to the possibility that implementation of the ATM interoperability specification contained herein may require use of an invention covered by patent rights held by ATM Forum Member companies or others. By publication of this ATM interoperability specification, no position is taken by The ATM Forum with respect to validity of any patent claims or of any patent rights related thereto or the ability to obtain the license to use such rights. ATM Forum Member companies agree to grant licenses under the relevant patents they own on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. For additional information contact:

The ATM Forum
Worldwide Headquarters
P.O. Box 29920
572 B Ruger Street
San Francisco, CA 94129-0920
Tel: +1.415.561.6110
Fax: +1.415.561.6120

Acknowledgments

The production of this specification would not be possible without the enormous amount of effort provided by many individuals. Special acknowledgements for their hard work and dedication go to Richard Graveman and Wolfgang Klasen, the current chair and vice-chair.

In addition, the following individuals (listed alphabetically), among others, contributed their time and expertise to the development of this specification:

Carter Bullard
Kim Hebda
Chris Kubic
John Rutemiller
Jeffery See
Linda Shields

Gary Buda
Michael Pierce
Editors, ATM Forum Security Working Group

Table of Contents

Preface

This specification uses three levels for indicating the degree of compliance necessary for specific functions, procedures, or coding. They are indicated by the use of key words as follows:

?? **Requirement:** “Shall” indicates a required function, procedure, or coding necessary for compliance. The word “shall” used in text indicates a conditional requirement when the operation described is dependent on whether or not an objective or option is chosen.

?? **Objective:** “Should” indicates an objective which is not required for compliance, but which is considered desirable.

?? **Option:** “May” indicates an optional operation without implying a desirability of one operation over another. That is, it identifies an operation that is allowed while still maintaining compliance.

1. Introduction

This addendum to [3] supplements the Routing Specification portion (Sections 3 and 5) and also changes the introductory material in Sections 1 and 2, as already modified by prior addenda. This addendum is intended not to modify the existing requirements for operation of PNNI Routing in an insecure environment. It is the intention that implementations compliant with the modified specification may continue to be used for such insecure operation.

This routing security addendum is based on the principles described in [1] and depends on the control plane services defined in [2]. Section 4 of [2] describes how to use the mechanisms defined in this addendum.

However, in cases of discrepancies between this document and [2], this document takes precedence.

The security features added by this addendum are designed to counter the principal security threats to PNNI routing of:

?? Unauthorized introduction of routing information,

?? Unauthorized modification of routing information,

?? Disclosure of routing information.

Exploitation of these attacks could cause a disruption of network services, including a complete loss of routing in an ATM network. The fundamental strategy for defense against these attacks is to provide strong authentication at PNNI peer discovery (using either shared secret key or public key authentication) and a secure transport mechanism for PNNI peer entity communication (using symmetric cryptographic message integrity and, optionally, confidentiality techniques).

This approach is designed to offer protection with minimal configuration requirements. Security for the complete PNNI routing infrastructure relies on an explicit chain of trust, which requires that each node take responsibility for the data that it summarizes and transmits.

This approach specifies the use of either shared secret key or public key cryptographic techniques for peer entity authentication as described in [2].

This addendum does not provide capabilities to counter the problem of a trusted PNNI peer entity introducing inappropriate or malicious information or omitting valid information. Nor does this specification counter a denial-of-service attack launched by nodes to which an insecure adjacency has been formed.

1.1. References

[1] ATM Forum Technical Committee, “ATM Security Specification,” Version 1.1, AF-SEC-0100.002, October 2000.

[2] ATM Forum Technical Committee, “Control Plane Security,” AF-SEC-CPS-0172.000, August 2001.

[3] ATM Forum Technical Committee, “Private Network-Network Interface Specification,” Version 1.0, AF-PNNI-0055.000, March 1996.

[4] ATM Forum Technical Committee, “PNNI V1.0 Errata and PICS,” AF-PNNI-0081.000, May 1997.

1.2. Definitions

Tagged-Secure – A status bit that indicates whether the node that created a PNNI packet considered it to be secure, based on the sources of the information used to create the information. This bit is never changed after it is initially determined.

Transmit-Secure – A status bit that indicates whether or not the PNNI packet was ever transmitted between nodes over an insecure link. A packet is transmit-secure unless it has been transmitted over an insecure link, which means that a locally created PNNI packet is always transmit-secure, even if it is not tagged-secure. It is the responsibility of the recipient of a message received over an insecure link to reset this bit.

AddSecurePort – An event in the PNNI state machine that indicates that a new inside secure link to the neighboring peer node has come up (i.e., has reached the Hello state 2-Way Inside).

DropSecurePort – An event in the PNNI state machine that indicates that a secure link to the neighboring peer has gone down (i.e., exited the Hello state 2-Way Inside).

2. Changes to PNNI version 1.0

The following additions and changes are made to [3] (as previously modified by other addenda).

2.1. Overview

An overview of the mechanisms for secure PNNI routing, contained in Section 4 of [2], describes the concepts of certification hierarchy, keys, policy, and secure tags.

2.2. Secure Indicators and Their Handling

To provide indications of the security status of information summarized, two indicators are defined and procedures are added to describe their use.

2.2.1. Definition of Security Indicators

To provide the definitions of these two indicators of the security status of PNNI Routing information, the following changes are made:

In Section 5.8.2.2, add the following new second paragraph to define the tagged-secure and transmit-secure indicators:

The topology database of a node contains a copy of every PTSE received and those locally created. Associated with each PTSE, but not an actual part of it, are two indicators of the security status of that PTSE. These indicators are:

?? Tagged-secure, which indicates whether the node that created the PTSE considered it to be secure, based on the sources of the information used to create the PTSE. This bit is never changed after it is initially determined.

?? Transmit-secure, which indicates whether or not the PTSE was ever transmitted between nodes over an insecure link. It is transmit-secure unless it has been transmitted over an insecure link, which means that a locally created PTSE is always transmit-secure, even if it is not tagged-secure. It is the responsibility of the recipient of a message received over an insecure link to reset this bit.

2.2.2. Secure Indicator Procedures

To provide a description of the procedures to be followed to create these security indicators when data are originated and to pass the correct values to the next node, the following procedures for their use should be added:

In Section 5.8.3.7, Origination of a New PTSE or a New PTSE Instance, add the following new steps III and IV after the existing step II and renumber the remaining steps:

III. The node sets the tagged-secure status of the PTSE to the logical AND of the following:

  A. The tagged-secure and transmit-secure indicators of all exchanges and summarized PSTEs used to form this PTSE, and

  B. The security status of all other information used, e.g., whether all Hello exchanges used to form this PTSE were secured.

IV. The node sets the transmit-security status of the PTSE to transmit-secure (since it is new and has never been sent over an insecure link).

In Section 5.8.3.2, Sending PTSPs, add the following new paragraph after the current fourth paragraph (the one beginning "In general. ":

PTSEs with the same security status shall be grouped by encapsulating them within a single Security IG, which specifies the security status of the group. (Note that the transmit-secure status associated with a PTSE is not dependent on the link over which it will later be sent. That is, a PTSE marked transmit-secure may be sent over a link that is not secure.)

In Section 5.8.3.3, Receiving a PTSP, add the following new paragraph at the end:

The security status associated with each PTSE is examined and processed as follows:

1. If the message by which the PTSE was received did not pass CPS security checks, the transmit-secure status of this PTSE is reset to not-transmit-secure. Otherwise, the transmit-secure status of the PTSE is not changed.

Note: The CPS services for a RCC shall include integrity and replay-reorder detection. The confidentiality service is optional and determined by policy. The use or failure to use the confidentiality service has no affect on the transmit-secure indicator.

2. The comparisons described in Section 5.8.2.2.4 are performed to determine whether the newly received PTSE is to replace the current PTSE in this node's topology database.
3. If so, the tagged-secure indication from the received PTSE and the transmit-secure indication resulting from step 1 above are associated with the new PTSE in this node's topology database.

A secure node shall have a configuration option to discard insecure information without further action or notification.

2.2.3. Changes to Comparison of PTSE Instances

Due to the extensions in packet formats and the new security properties of PTSEs, changes to the PTSE comparison procedure are necessary. This is because secure PTSEs are more desirable than insecure ones, the same way newer versions are of more interest than older ones.

In Section 5.8.2.2.3, add a bullet in the second paragraph as follows:

The additional information that distinguishes one instance of a PTSE from another instance of the same PTSE is:

?? PTSE sequence number

?? PTSE remaining lifetime

?? Security status of PTSE

?? PTSE checksum.

In Section 5.8.2.2.4 replace paragraph 2 with:

When two instances of the same PTSE exist simultaneously, they must be compared to see whether they are separate instances and, if so, which instance is to be retained. This decision is based on the security status and freshness of the two instances. The precedence between security status and freshness is a matter of security policy. The security status of an instance of a PTSE is based on the tagged-secure and transmit-secure flags. The freshness part of this comparison consists of the following steps:

Note: The consequences of giving freshness priority over security status include the possibility that an attacker can replace a secured instance of a PTSE with a more recent looking but bogus instance. On the other hand, requiring the transmit-secure status for all instances limits one's ability to obtain current topology information, and requiring the tagged-secure status for all instances limits an originating node's ability to report connections with insecure neighbors.

2.3. New Security Information Group (IG)

This section defines the changes to add a new information group to carry security-related information.

2.3.1. Security IG

To allow the tagging of information as secure or insecure, a new Security IG is added to PNNI Routing. This new IG is used either:

1. To encapsulate those existing IGs that need to be marked with security information, or
2. To be included in those existing IGs that need to be marked.

Note: The method used depends on what the desired procedures are for a receiving node that has not implemented this security addendum.

Add the following new section and table:

5.14.16 Security Unrestricted Information Group

The security status of information contained in an IG or packet may be indicated by the use of the Security IG. This unrestricted IG may be used in either of the two following ways:

?? It may be contained in any PNNI packet or IG and indicates the security status of all other information contained in that packet or IG.

?? It may contain one or more other IGs and indicates the status of all information contained within itself, which includes all IGs included at a lower level.

The contents of the Security IG are shown in Table 5-48. A field in the Security IG indicates which of these two applications is being used. If identified as "scope = higher level," then this Security IG may appear only once within the top level of a packet or IG. If identified as "scope = included IGs," this Security IG may appear multiple times within a packet or IG.

In the event that a packet or IG contains multiple, conflicting Security IGs at different levels, the security status of each individual IG in the packet is determined by the lowest level Security IG that applies to it.

Examples of the use of this IG to indicate the security status of various portions of other packets and IGs are provided in Appendix A.

Table 5-48: The Security IG.

Offset	Size (Octets)	Name	Function/Description
0	2	Type	Type = 641 (security)
2	2	Length	Length of the entire IG
4	1	Scope	Indicates what information this IG applies to: 0 = Not used 1 = This IG applies to the higher level IG or packet and all its contents. (This Security IG may not contain any further Igs.) 2 = This IG applies to included IGs. It applies only to IGs contained within this Security IG 3–255 Reserved
5	1	Application	0 = not used 1 = tagged & transmit secure flags 2–239 Reserved 240–255 User Defined
6	1	Transmit-security status	Indicates the transmit and tagged security status of the information: Bit 1 = transmit secure Bit 2 = tagged secure Bits 3–8 reserved
If the scope = included IGs, this is followed by any other TLV groups that would be allowed within the packet or TLV group at the place where this TLV is located.			

2.3.2. Changes to Information Group Summary

References to the new Security IG are added as appropriate to the summary tables in Section 5.14.3 and several other places. The following changes are made:

In Section 5.14.3, add the following row to the first part of Table 5-18, Information Group Summary:

Type	IG Name	Contains IGs one level down
641	Security	Note: If marked as scope = higher level, this IG may not contain any other IGs one level down. If marked as scope = included IGs, this IG may contain any IGs one level down that would be contained in the IG one level up or in the packet at the same position occupied by this Security IG.

In Section 5.14.3 in Table 5-18 (first part), add the following to each row that currently has entries in the third column:

Security (641)

In Section 5.14.3, add the following row to the second part of Table 5-18 Information Group Summary continued:

Type	IG Name	Contained in IGs one level up	Contained in packets
641	Security	All	All packets

In Section 5.14.3, add the following to each current row of Table 5-19, Information Groups in PNNI Packets:

Security (641)

Modify Section 5.14.9.2, Unrestricted Information Group, as follows:

The Systems Capabilities and the Security IGs are ~~is~~ the only unrestricted IGs ~~used~~ currently defined in PNNI~~Phase 1~~. Other unrestricted IGs may be defined in later versions of PNNI.~~Those foreseen are for authentication and access control.~~

2.3.3. PNNI Hellos

It is necessary to allow the Security IG to be in the Hello packet to indicate the security status of the uplink information attributes.

In Section 5.14.8, PNNI Hellos, add the following new paragraph to the last cell of Table 5-27 (after the paragraph beginning : "Hellos sent between LGNs ..."):

The Security IG may be included in any Hello packet. It may then contain any of the above listed IGs.

2.3.4. PNNI Topology State Packets (PTSP)

To avoid confusion about the PTSE, the following change should be made, because there is no authentication field defined in this specification.

In Section 5.14.9, PNNI Topology State Packets (PTSP), modify the text immediately after Table 5.31 as follows:

Each PTSP consists of multiple PNNI Topology State Elements (PTSEs), all from the same originating node. Each PTSE includes its own checksum~~and authentication field (null in PNNI Phase 1)~~, allowing for PTSEs from the same originating node ...

2.3.5. Database Summary Packet

Since the new Security IG may be used in the Database Summary packet to indicate the security status of the PTSE summaries that this packet carries, a note about its possible inclusion is added to the table to show the format of this packet.

In Section 5.14.11, Database Summary Packet, add the following note (as underlined) in Table 5-42 after the header (between field at offset 12 and the nodal PTSE summaries IG):

Offset	**Size (Octets)**	**Name**	**Function/Description**
...	...	...	...
12	4	DS sequence number	
A Security IG may be used here once to indicate the security status of the entire packet or multiple times to indicate the status of a set of one or more included Nodal PTSE summaries.			
Repeat for each set of PTSEs in the topology database:			
	2	Type	Type = 512 (Nodal PTSE summaries)
...	...	...	...

2.3.6. PTSEType Field

The definitions of the PTSEType field in the PTSP are modified to include the possibility that the new Security IG is the top-level IG in the PTSE in the PTSP. It should be understood that this does not mean that the same Security IG is actually stored as a part of the PTSE in the node's topology database.

In Section 5.8.2.2.1, modify the final bullet as follows:

?? PTSEType
The PTSEType field indicates which restricted information ~~groups are allowed to appear~~ group appears inside of the PTSE, or that no restricted information groups ~~are allowed~~ appear in the originating node's topology database (see Section 5.14.9 for details).

In Section 5.14.9, PTSP, change Table 5-32 as follows:

Offset	Size (Octets)	Name	Function/Description
4	2	PTSEType	Identifies the type of PTSE contained in this packet and in the topology database of the node sending this packet. It indicates the top level Information Group in the PTSE in the sending node's database. It is also the same as the top level Information Group in the PTSE in this packet, except when the optional unrestricted Security Information Group is included at the top level above the PTSE, in which case, it indicates the type of IG contained at the top level within the Security IG. PTSEType must be one of the type codes of a restricted IG or NoRestrictedIG (type=0). In this PTSE, that particular restricted IG may appear, and also any unrestricted IGs. Restricted IGs other than the one mentioned are not allowed.~~.~~ If the type is NoRestrictedIG, then no restricted IGs are allowed. Note that this is not aiming to influence the types of TLVs embedded inside of the restricted information groups. Only the "top-level" restricted information groups in the PTSE have to conform to this rule. Since the type of Information Group identified here must be consistent with the Information Group contained later in the same packet, this value may be ignored at the receiver. It is included here only for consistency with the format of the Database Summary Packet given in Table 5-42. This field contains the following: **Low order 12 bits:** 0 = No Restricted IG or one of the values of Restricted IG as listed in Table 5.18. **Bits 13-16** Reserved

In Section 5.14.11, Database Summary Packets, change Table 5-42 as follows:

Offset	Size (Octets)	Name	Function/Description
	2	PTSEType	Identifies the type of PTSE in the sender's topology database. The format and contents of this field are as defined in Table 5-32.

2.3.7. Information Group Tags

In Section 5.14.2.6, modify the first sentence of the final paragraph as follows:

All PNNI 1.0 information groups shall be originated with their information group tags set to optional, summarizable, and non-transitive with ~~two~~ three exceptions:

1. the Transit network ID IG shall have information group tag values optional, summarizable, and transitive;
2. the System Capabilities IG may have any combination of IG tags;

3. the Security IG shall have the tag values optional, not summarizable, and transitive.

2.3.8. Examples of the Use of the New Security IG

Examples of a number of potential uses of the Security IG are added to indicate how this IG is included in the current structure of packets and information groups, both to encapsulate the information to be tagged and as a part of an information group, to specify particular handling by a node that does not understand this new Security IG.

Add a new Appendix A (numbered as appropriate) as contained at the end of this Addendum.

2.4. Procedures for AddPort and DropPort

The neighboring peer state machine described in Section 5.7 of [3] requires modification to support a mixture of secure and insecure links to a neighboring node. The Port ID List, the AddPort event, and the DropPort event need to be recorded according to their secure and insecure parts. The establishment of the first secure port requires that the database be re-synchronized, because some of the already requested and received insecure PTSEs can now be received as secure.

The following changes are made to do this:

In Section 5.6.2.1, modify references to AddPort and DropPort in the second paragraph to read:

For lowest-level neighbor nodes with parallel physical links and~~/or~~ VPCs between them, there will be multiple instances of the Hello protocol. However, for the purposes of database synchronization and flooding of PTSEs, there is only one instance of the neighboring peer data structure and associated neighbor peer state machine for all insecure links and one for all secure links. In order to describe the interaction between the multiple Hello conversations and the single neighboring peer conversation (for database synchronization and flooding procedures), reference is made to a neighboring peer state machine and to its events AddPort, AddSecurePort, DropPort, and DropSecurePort. AddPort indicates that a new inside insecure link to the neighboring peer node has come up (i.e., has reached the Hello state 2-Way Inside), and AddSecurePort, indicates the same for a new secure link. The event DropPort indicates that an insecure link to the neighboring peer has gone down (i.e., exited the Hello state 2-Way Inside), and event DropSecurePort, indicates the same for a secure link. The description of the Hello state machine includes indication of when the events AddPort, AddSecurePort, DropPort, and DropSecurePort, must be sent, thus describing the interaction between the multiple Hello ~~conversions~~ state machines and corresponding neighboring peer~~conversation~~ state machine.

In Section 5.6.2.1.4, in HP0 through HP20, change each occurrence of "AddPort" to "AddPort or AddSecurePort as appropriate" and change each occurrence of "DropPort" to "DropPort or DropSecurePort as appropriate" and add a note after HP20 to read:

AddPort or AddSecurePort shall be generated depending on the security status of the link.

In Section 5.6.3.1, modify references to AddPort and DropPort in the fourth through seventh sentences to read:

The Hello protocol used to monitor the status of the SVCC triggers the AddPort, AddSecurePort, ~~and~~ DropPort, and DropSecurePort events ~~in~~to the neighboring peer state machine that controls database synchronization between the LGNs. This is similar to the relationship between the Hello protocol and the neighboring peer state machines run between lowest-level neighbors. The event AddPort or AddSecurePort in the neighboring peer state machine -is triggered when the Hello state machine for the SVCC reaches the 2-Way Inside state. The event DropPort or DropSecurePort in the neighboring peer state machine is triggered when the Hello state machine for the SVCC falls out of the 2-Way Inside state.

In Section 5.7, modify the third paragraph as follows:

For lowest-level neighboring peers, there may be multiple parallel physical links ~~and/~~or VPCs between them. As described in Section 5.6, each physical link and~~/or~~ VPC between the two neighboring peers will run a separate Hello state machine. However, for the purposes of database synchronization and flooding, only one conversation is held between the neighboring peers. This conversation is described by the neighboring peer state machine and the neighboring peer data structure, which includes the information required to maintain database synchronization and flooding to the neighboring peer. Whenever a link reaches the Hello state 2-Way Inside, the event AddPort or AddSecurePort is sent to ~~is triggered in~~ the corresponding neighboring peer state machine. Similarly, when a link falls out of the Hello state 2-Way Inside, the event DropPort or DropSecurePort is triggered in the corresponding neighboring peer state machine. The database exchange process commences when the event AddPort or AddSecurePort is first triggered~~,~~ after the first link between the two neighboring peers comes up. When the DropPort or DropSecurePort event for the last link between the neighboring peers occurs, the neighboring peer state machine will internally generates the DropPortLast event causing all state information for the neighboring peer to be cleared.

In Section 5.7, modify the final two sentences of the fifth paragraph as follows:

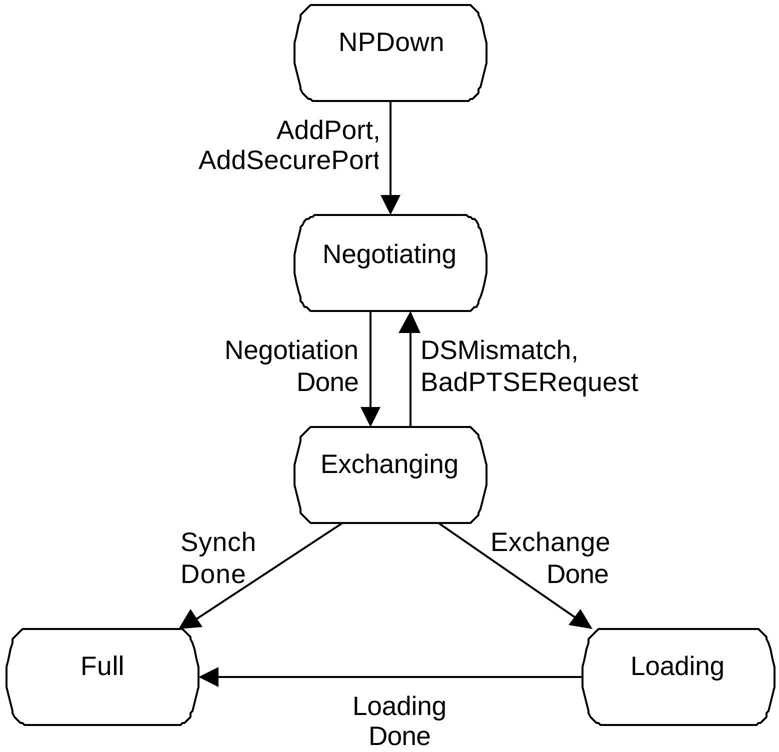
When the Hello state of the RCC reaches 2-Way Inside, the event AddPort or AddSecurePort is triggered in the neighboring peer state machine and the database exchange process commences. After an AddSecurePort event is received, all database exchange and flooding messages must occur over a link protected by CPS. When the Hello state of the RCC falls out of the state 2-Way Inside, the event DropPort or DropSecurePort is triggered in the neighboring peer state machine, causing it to transition from Full state to NPDown state.

In Section 5.7.1, modify the description of the Port ID List as follows:

Port ID List

> The Port ID List is used only in the case of lowest-level neighboring peers, which are connected by physical links ~~and/~~or VPCs. The Port ID List is a list of those insecure links to the neighboring peer that are in the state 2-Way Inside and, when the security service is being provided, a separate list of those secure links to the neighboring peer that are in the state Inside link secure. When PTSPs, PTSE acknowledgment packets, database summary packets, or PTSE request packets are transmitted or retransmitted to the neighboring peer, the secure list shall be used if it exists. Any of the links specified in ~~this~~ the appropriate list may be used.

In Section 5.7.2, modify Figure 5-6 as follows:



In addition to the state transitions pictured:

?? Event DSMismatch forces Negotiating state.

?? Event BadPTSERequest forces Negotiating state.

?? Event DropPort or DropSecurePort causes no state change unless it is the last port, which forces the NPDown state.

?? Event DropPortLast forces NPDown state.

Figure 5-6: Neighboring Peer State Change (Database Synchronization).

In Section 5.7.3, modify two and add two event definitions, as follows:

AddPort
A Hello state machine for an insecure link to the neighboring peer has reached the 2-Way Inside state.

AddSecurePort
A Hello state machine for a secure link to the neighboring peer has reached the 2-Way Inside state.

DropPort
A Hello state machine for an insecure link to the neighboring peer has exited the 2-Way Inside state.

DropSecurePort
A Hello state machine for a secure link to the neighboring peer has exited the 2-Way Inside state.

In Section 5.7.4, Table 5-12, add the following:

Event	NPDown	Negotiating	Exchanging	Loading	Full
AddSecurePort	Ds1s Negotiating	Ds7s Negotiating	Ds7s Negotiating	Ds7s Negotiating	Ds8s Negotiating
DropSecurePort	FSM_ERR	Ds9s Negotiating	Ds9s Exchanging	Ds9s Loading	Ds9s NPDown

In Section 5.7.4, modify the actions Ds1, Ds7, Ds9, and Ds10 and add four new actions as follows:

Ds1:

Action: For the case of lowest-level nodes, which are connected by physical links ~~and/~~or VPCs, the port ID is added to the insecure portion of the Port ID List in the neighboring peer data structure.

[Second paragraph unchanged.]

Ds1s:

Action: For the case of lowest-level nodes, which are connected by physical links or VPCs, the port ID is added to the secure portion of the Port ID List in the neighboring peer data structure.

Ds7:

Action: For the case of lowest-level neighboring peers, which are connected by physical links ~~and/~~or VPCs, the port ID is added to the insecure portion of the Port ID list in the neighboring peer data structure.

Ds7s:

Action: For the case of lowest-level neighboring peers, which are connected by physical links or VPCs, the port ID is added to the secure portion of the Port ID list in the neighboring peer data structure.

Ds8:

Action: Same as Ds7 with the additional requirement that this action will cause a link to the neighboring peer to be added, causing a new instance of a PTSE to be originated.

Ds8s:

Action: Same as Ds7s with the additional requirement that this action will cause a link to the neighboring peer to be added, causing a new instance of a PTSE to be originated.

Ds9:

Action: The link is removed from the insecure Port ID list in the corresponding neighboring peer data structure. The action will cause a link to the neighboring peer to be removed. If there is a PTSE advertising that link, a new instance of the affected PTSE must be originated. If this was the last active link, a new instance of the affected PTSE must be originated. If this was the last active link to this neighbor, generate the DropPortLast event.

Ds9s:

Action: The link is removed from the secure Port ID list in the corresponding neighboring peer data structure. The action will cause a link to the neighboring peer to be removed. If there is a PTSE advertising that link, a new instance of the affected PTSE must be originated. If this was the last active link, a new instance of the affected PTSE must be originated. If this was the last active link to this neighbor, generate the DropPortLast event.

Ds10:

No change.

2.5. Path Computation

A secure node shall have a configuration option as to how it considers the security status of routing information when computing a path.

?? A secure node shall have the option of disregarding insecure information.

Path computation to enforce less restrictive security policies using both secure and insecure information is implementation dependent. For example :

?? Secure information may be given preferential treatment over insecure information.

?? Secure information may be considered equivalent to insecure information.

2.6. Security Labels

Delete Section 5.14.2.5.4, Security Labels.

Appendix A. Examples of Using the Security Information Group

(This Appendix does not form an integral part of this specification.)

A.1. Introduction

The Security IG defined in Section 5.14.14 provides the capability to indicate the security status of information contained within other IGs. As defined, all information to be marked is either:

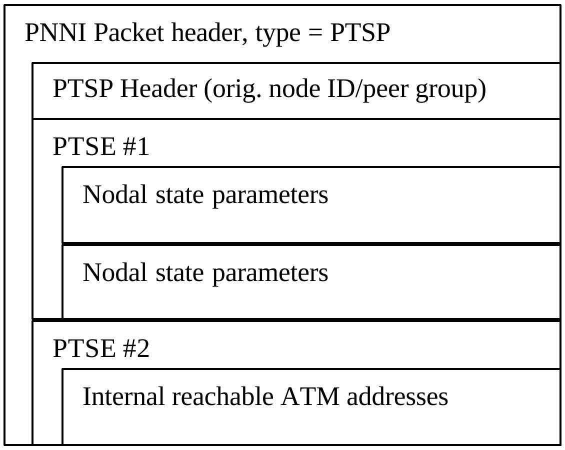
1. encapsulated within a Security IG using the TLV nested coding technique described in Section 5.14, or
2. part of the higher level IG in which the Security IG is contained.

This provides flexibility in identifying both secure and insecure information within the same packets or IGs by the repetition of the Security IG. It also ensures that a node unaware of security will ignore the information if the security-aware sending node so desires.

This appendix provides examples to show how these things may be done.

A.2. Conventions

For the sake of this appendix, the nesting of IGs in a packet is depicted as follows:


PNNI Packet header, type = PTSP
PTSP Header (orig. node ID/peer group)
PTSE #1
Nodal state parameters
Nodal state parameters
PTSE #2
Internal reachable ATM addresses

This shows a PNNI packet of type PTSP containing two PTSEs (after the PTSP Header). The first PTSE contains two Nodal state parameters IGs, the second contains an Internal reachable ATM addresses IG.

A.3. Examples

The following examples depict various uses of the Security IG and show which part of the packet is labeled by the Security IG (shaded). With each example, a description of the action taken by a node unaware of security (i.e., one that does not recognize the Security IG) is provided. The actions described for the node unaware of security are fully defined in [3] without addition of any of the capabilities described in this security addendum.

A.3.1. Example 1 - All Data in Packet are Tagged with the Same Security Status

In this example, the entire PTSP has the same security status. This may be done in one of two ways:

A. A Security IG may be added independently from the PTSEs and marked as scope = higher level as shown in Example 1A.

B. A Security IG may be added "above" the list of PTSEs and marked as scope = included IGs as shown in Example 1B.

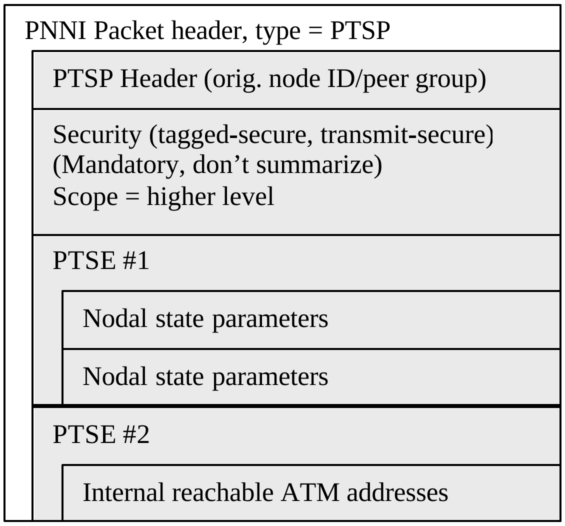

PNNI Packet header, type = PTSP
PTSP Header (orig. node ID/peer group)
Security (tagged-secure, transmit-secure)
(Mandatory, don't summarize)
Scope = higher level
PTSE #1
Nodal state parameters
Nodal state parameters
PTSE #2
Internal reachable ATM addresses

Figure A-1: Example 1A.

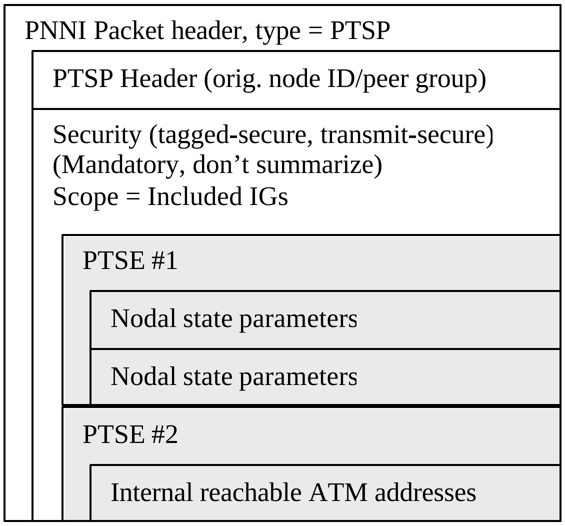

PNNI Packet header, type = PTSP
PTSP Header (orig. node ID/peer group)
Security (tagged-secure, transmit-secure)
(Mandatory, don't summarize)
Scope = Included IGs
PTSE #1
Nodal state parameters
Nodal state parameters
PTSE #2
Internal reachable ATM addresses

Figure A-2: Example 1B.

In Example 1A, a node unaware of security would not recognize the Security IG and would process the remainder of the PTSP based on the contents of the Information Group Tags in the Security IG. This construction could, for example, instruct it to not use any of the PTSEs in this PTSP in summarization.

In Example 1B, a node unaware of security would see the above as a PTSP containing no PTSEs, since its only contents is an IG it does not understand.

A.3.2. Example 2 - All Data in Packet are Tagged with Different Security Status

In this example, the two PTSEs are marked with different security information. This may be done in one of two ways:

A. Each PTSE is encapsulated in a separate Security IG, which is marked as scope =included IGs as shown in Example 2A.

B. A Security IG is added to both PTSEs and marked as scope = higher level as shown in Example 2B.

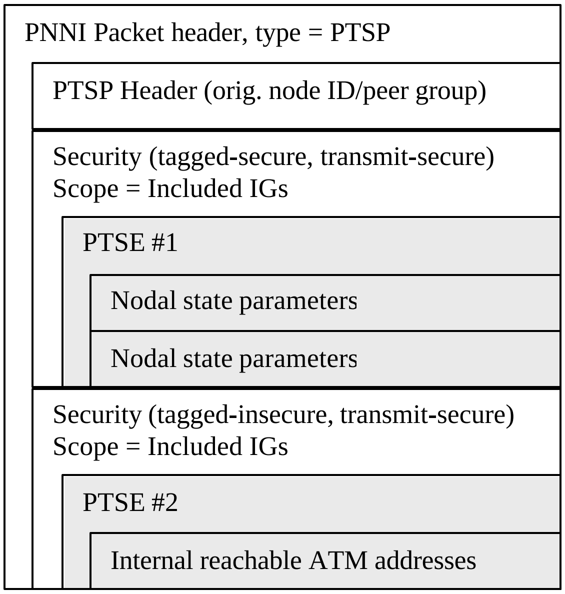

PNNI Packet header, type = PTSP
PTSP Header (orig. node ID/peer group)
Security (tagged-secure, transmit-secure)
Scope = Included IGs
PTSE #1
Nodal state parameters
Nodal state parameters
Security (tagged-insecure, transmit-secure)
Scope = Included IGs
PTSE #2
Internal reachable ATM addresses

Figure A-3: Example 2A.

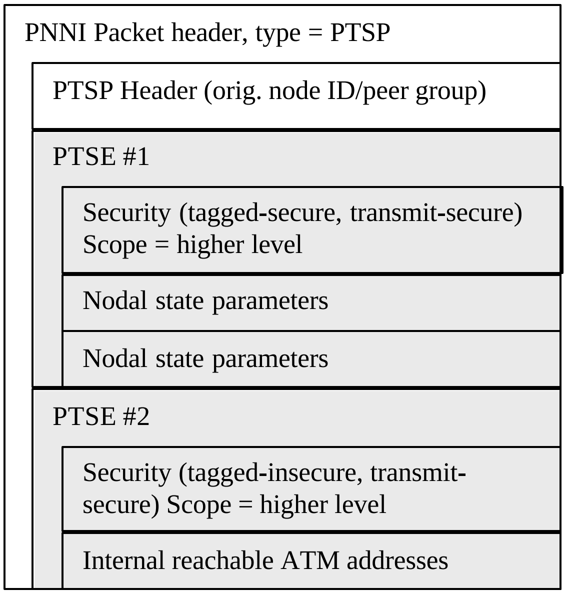

PNNI Packet header, type = PTSP
PTSP Header (orig. node ID/peer group)
PTSE #1
Security (tagged-secure, transmit-secure)
Scope = higher level
Nodal state parameters
Nodal state parameters
PTSE #2
Security (tagged-insecure, transmit-secure) Scope = higher level
Internal reachable ATM addresses

Figure A-4: Example 2B.

In Example 2A, a node unaware of security interprets this as a PTSP containing nothing, since it does not recognize the new IGs. If it considers an empty PTSP as an error, it would take the appropriate action; however, PNNI does not specify that an empty PTSP constitutes an error condition.

In Example 2B, a node unaware of security ignores the IGs it does not understand (the Security IGs) and processes the remaining IGs (the PTSEs) in the normal way.

A.3.3. Example 3 - Several Lists, Each with a Different Security Status

In this example, a Database Summary Packet contains three different groups of PTSE summaries, some tagged-secure and transmit-secure, some not tagged-secure but transmit-secure, and some not tagged-secure but transmit-secure.

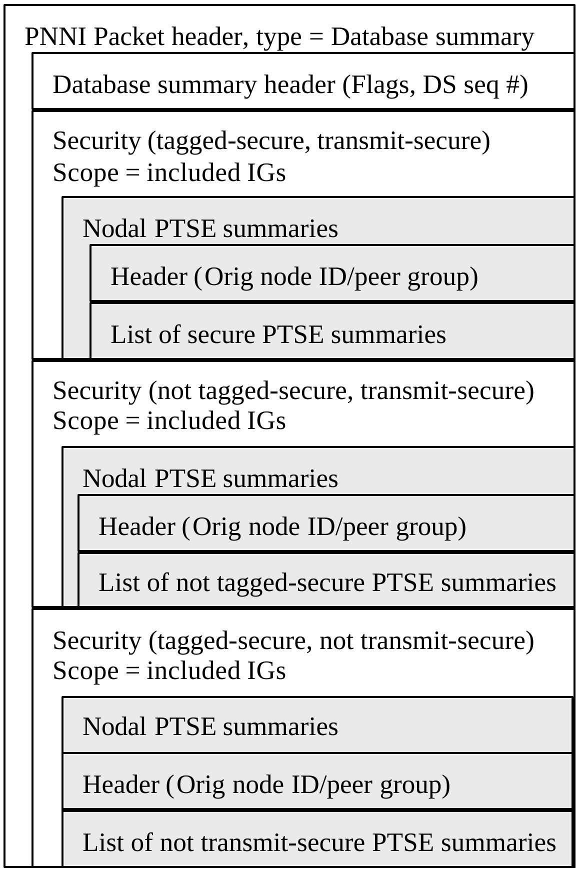

PNNI Packet header, type = Database summary
Database summary header (Flags, DS seq #)
Security (tagged-secure, transmit-secure)
Scope = included IGs
Nodal PTSE summaries
Header (Orig node ID/peer group)
List of secure PTSE summaries
Security (not tagged-secure, transmit-secure)
Scope = included IGs
Nodal PTSE summaries
Header (Orig node ID/peer group)
List of not tagged-secure PTSE summaries
Security (tagged-secure, not transmit-secure)
Scope = included IGs
Nodal PTSE summaries
Header (Orig node ID/peer group)
List of not transmit-secure PTSE summaries

Figure A-5: Example 3.

In Example 3, a node unaware of security ignores all of these database summaries and sees an empty PTSP. If it considers an empty PTSP as an error, it would take the appropriate action; however, PNNI does not specify that an empty PTSP constitutes an error condition.

A.3.4. Example 4 - Multiple Security IGs in One Packet

In this example, multiple Security IGs are included at different levels.

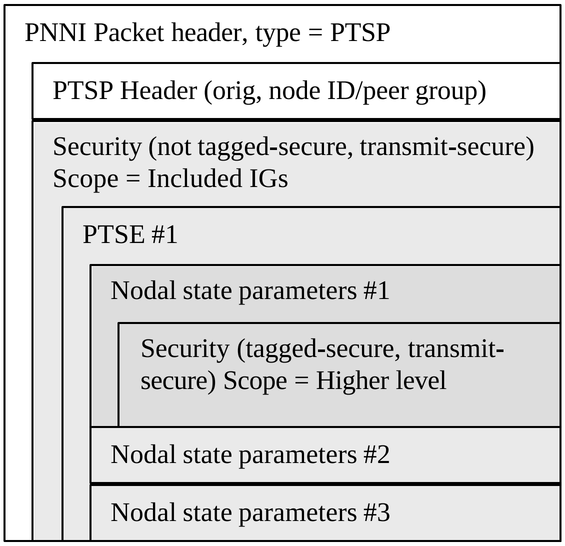

PNNI Packet header, type = PTSP
PTSP Header (orig, node ID/peer group)
Security (not tagged-secure, transmit-secure) Scope = Included IGs
PTSE #1
Nodal state parameters #1
Security (tagged-secure, transmit-secure) Scope = Higher level
Nodal state parameters #2
Nodal state parameters #3

Figure A-6: Example 4.

Two Security IGs could apply to the Nodal State Parameters #1. However, the lower-level one applies. That means that, while Nodal State Parameters #2 and #3 are not tagged-secure, the Nodal State Parameters #1 are tagged secure. All are transmit secure.

A node unaware of security will ignore all data, since it does not understand the top-level Security IG, and it will see an empty PTSP.