DTN Research Group
Internet-Draft
Expires: December 10, 2005

S. Symington
The MITRE Corporation
S. Farrell
Trinity College Dublin
H. Weiss
SPARTA, Inc.
June 8, 2005

Bundle Security Protocol Specification
draft-irtf-dtnrg-bundle-security-00

Status of this Memo

Copyright Notice

Abstract

This document defines the bundle security protocol, which provides data integrity and confidentiality services. We also describe various bundle security considerations including policy options.

Table of Contents

1. Introduction

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [1].

This document defines security features for the bundle protocol [2] intended for use in delay tolerant networks, in order to provide the DTN security services as described in the DTN Security Overview and Motivations document [5].

The bundle protocol is used in DTNs which overlay on top of multiple networks, some of which are challenged by limitations such as intermittent and possibly unpredictable loss of connectivity, long or variable delay, asymmetric data rates, and high error rates. The purpose of the bundle protocol is to support interoperability across such stressed networks. The bundle protocol is layered on top of underlay-network-specific convergence layers, on top of network-specific lower layers, to enable an application in one network to communicate with an application in another network, both of which are spanned by the DTN.

Security will be important for the bundle protocol. The stressed environment of the underlying networks over which the bundle protocol will operate makes it important that the DTN be protected from unauthorized use, and this stressed environment poses unique challenges on the mechanisms needed to secure the bundle protocol. Furthermore, DTNs may very likely be deployed in environments where a portion of the network might become compromised, posing the usual security challenges related to confidentiality, integrity and availability.

The security features described in this document are RECOMMENDED for deployment with the Bundle Protocol. Bundle Protocol implementations SHOULD support these features, except where they cannot, e.g. due to processor or bandwidth constraints. [Comment.1]

Implementation of these security features is not mandatory because it is recognized that some bundle nodes will be so constrained that they are simply unable to use the mechanisms defined here. For example, if a communications subsystem is limited to emitting 40-byte segments, then we cannot use a 2048-bit RSA signature to secure such segments.

The implications of omitting or not using these security features however, need to be fully understood, especially with regard to the ability of an implementation to interact with bundle protocol implementations that do implement security and that may be configured

to require that incoming bundles include specific security headers. [Comment.2]

1.1 Related Documents

This document is best read and understood within the context of the following other DTN documents: [Comment.3]

The Delay-Tolerant Network Architecture [3] defines the architecture for delay-tolerant networks, but does not discuss security at any length.

The DTN Bundle Protocol [2] defines the format and processing of the headers used to implement the bundle protocol, excluding the security-specific headers defined here.

The Delay-Tolerant Network Security Overview and Motivations [5] document provides an informative overview and high-level description of DTN security; it defines a security architecture for DTNs by describing the aspects of the DTN architecture requiring protection and the security mechanisms that can be used to provide this protection, along with motivating rationale.

1.2 Terminology

We introduce the following terminology for purposes of clarity:

source - the bundle node (application, bundle protocol agent, etc.) from which the application data unit originates

destination - the bundle node (application, bundle protocol agent, etc.) to which the application data unit is ultimately destined

sender - the bundle protocol agent that forwarded the bundle on its most recent hop

recipient or "next hop" - the neighboring bundle protocol agent to which a sender forwards a bundle.

[Comment.4]

In the figure below, which is adapted from figure 1 in the Bundle Protocol Specification [2], four bundle protocol agents (denoted BPA 1, BPA 2, BPA 3, and BPA 4) that implement the bundle protocol reside above some transport layer(s). Three distinct transport and network protocols (denoted T1/N1, T2/N2, and T3/N3) are also shown. Bundle protocol agents BPA 1 and BPA 4 expose the bundle service interface

to BundleNodeX and BundleNodeY, respectively.

BPA = "Bundle Protocol Agent"--the implementation of the bundle protocol at the bundle layer

The protocol stacks on four DTN nodes.

Figure 1

Bundle node "BundleNodeX" originates data that is sent to its underlying bundle protocol agent BPA1. That data is formatted into a bundle and forwarded to BPA2, BPA3, and BPA4 and then transmitted up to bundle node "BundleNodeY". BundleNodeX is the source and BundleNodeY is the destination for this data. BPA1 is the first sender, and BPA2 is the first recipient; BPA2 then becomes the sender, and BPA3 the recipient; BPA3 then becomes the last sender, and BPA4 the last recipient.

If bundle protocol agent BPA1 originates data (for example, an administrative payload such as a bundle status report or custodial signal), which is then forwarded on to BPA2, BPA3, and finally BPA4, then BPA1 is the source of the bundle (as well as being the first sender of the bundle) and BPA4 is the destination of the bundle (as well as being the final recipient).

We introduce the following security-specific DTN terminology: [Comment.5][Comment.6]

security-source - a bundle node that adds an "end-to-end" security header

security-destination - a bundle node that processes an "end-to-end" security header

Referring to figure 1 again:

If the bundle that originates at BundleNodeX as source is given a Payload Security Header (PSH) by BPA1, then BPA1 is the security-source of this bundle, even though BundleNodeX is its source.

If the bundle that originates at BundleNodeX as source is given a Payload Security Header (PSH) by BPA2, then BPA2 is the security-source of this bundle, even though BundleNodeX is its source.

If the bundle that originates at BundleNodeX as source is given a Confidentiality Header (CH) by BPA1 protected using a key held by BPA2, then the BPA1 is the security-source of this bundle, and BPA-2 is the security destination.

2. Security-Specific Bundle Service Primitives and Parameters

2.1 The Delivery Options Parameter

[Comment.7][Comment.8]

The bundle service consumes a Send.request primitive that is used by the bundle node to request transmission of an application data unit from the source bundle endpoint to a destination bundle endpoint. One of the parameters of this primitive is "delivery options".

The bundle service uses the Data.indication primitive to deliver the content of a bundle to one or more bundle nodes indicated by the bundle's destination endpoint. One of the parameters of this primitive is also "delivery options".

The Delivery Options parameter indicates the optional security-specific procedures to be followed when transmitting and delivering the application data unit. Separate options MAY be specified for each of the confidentiality, authentication and error detection services. In each case, we use a ciphersuite ID to indicate how the service is to be provided. The ciphersuite ID can be omitted, leaving the decision up to the implementation. If the ciphersuite ID is provided, the additional ciphersuite parameters MAY be provided. The details of valid parameters are part of the definition of each of the ciphersuites.

Even if the ciphersuite ID for a service is omitted, the implementation MAY decide to apply that service, based on some policy settings. However, if a ciphersuite ID is supplied, then the implementation MUST NOT send the bundle without successfully applying the required service.

The following service-specific parameters are defined:

Confidentiality - indicates that a CH header MUST be applied to the bundle, and at least the bundle payload (and possibly other bundle headers) MUST be encrypted appropriately[Comment.9]

Authentication - indicates that a PSH MUST be applied to the bundle, and the (relevant parts of the) bundle digitally signed or MACed appropriately

Error detection - indicates that a PSH MUST be applied to the bundle and the (relevant parts of the) bundle digested appropriately [Comment.10]

Notes:

1) The above means that a given bundle can only use one of the authentication or error detection services

2) Hop-by-hop authentication is not controlled using this interface

3) Throughout this specification we use the standard security usage of the acronym MAC - meaning "message authentication code"

2.2 Data indications

As with Send.request, Data.indication primitives SHOULD indicate the security services which were applied to the message. In each case the indication MUST contain a ciphersuite ID and MAY contain a ciphersuite-specified ciphersuite parameters item. Note that the ciphersuite parameters may differ between the Send.request and Data.indication primitives, e.g. for a confidentiality item in a Send.request, the security-destination will often be a parameter, whereas the security-source is what is of interest for the Data.indication (if the ciphersuite allows).

We define the following indications:

The presence of the Confidentiality item in the Data.indication primitive indicates that the bundle contained a CH and that its payload was encrypted. Note that it is possible that this bundle protocol agent is not the CH security-destination, i.e. the payload may have been decrypted by a bundle protocol agent earlier in the route.

The presence of the Authentication Item in the Data.indication primitive indicates that the bundle contained a PSH with an authentication ciphersuite. Note that it is possible that this bundle protocol agent is not the PSH security-destination, i.e. the PSH MAY have been verified by a bundle protocol agent earlier in the route.

The presence of the Error Detection item in the Data.indication primitive indicates that the received bundle included a PSH with an error detection ciphersuite.

2.3 The At-Most-Once-Delivery Option

The bundle service consumes a Register.request primitive that is used to notify the bundle protocol agent of a bundle node's interest in bundles destined for a specified bundle endpoint and of the action to

take on the bundle node's behalf with regard to those bundles. One of the parameters of this primitive is "at-most-once-delivery". [Comment.11] If this parameter is set, then the bundle node registering interest in receiving bundles is indicating that the receiving bundle protocol agent SHALL check for replayed bundles, discard duplicates, and deliver at most one copy of each received bundle to the registering bundle node. If this parameter is not set, the bundle node is indicating that the receiving bundle protocol agent SHALL deliver all received bundle copies to the registering bundle node. If this parameter is not set, the bundle protocol agent MAY (subject to policy) deliver all bundles or else behave as if this parameter had been set.[Comment.12]

A method that the receiving bundle protocol agent MAY [Comment.13] use to detect duplicates is to look at the (source, timestamp) pair value of each received bundle and determine if this pair value, which uniquely identifies a bundle, has been received before. If it has, then the bundle is a duplicate and SHALL be discarded. If it has not, then the bundle SHALL be delivered to the bundle node and the (source, timestamp) pair value SHALL be added to the list of pair values already received by that bundle protocol agent.

3. Security Headers

There are three security headers that MAY be included in a bundle as listed in the Bundle Protocol [2]. These are the Bundle Authentication Header (BAH), the Payload Security Header (PSH), and the Confidentiality Header (CH).

The BAH is used to assure the authenticity of the bundle along a single hop from sender to recipient.

The PSH is used to assure the authenticity of the bundle from the PSH security-source, which creates the PSH, to the PSH security-destination, which verifies the PSH authenticator. The authentication information in the PSH may (if the ciphersuite allows) be verified by any bundle protocol agent in between the PSH security-source and the PSH security-destination that has access to the cryptographic keys and revocation status information required to do so.

The CH indicates that the bundle payload (and possibly other bundle fields) has been encrypted while en route between the CH security-source and the CH security-destination.

3.1 Abstract Security Header

Since the three security headers have most fields in common, we can shorten the description of them if we first define an abstract security header (ASH) and then specify each of the real headers in terms of the fields which are present/absent in an ASH. Note that no bundle ever contains an ASH, which is simply a specification artifact

An ASH consists of the following mandatory and optional fields: [Comment.14]

Header-type - as in all bundle protocol headers except the primary bundle header

Flags - as in all bundle protocol headers[Comment.15]

Length - the overall length of this header encoded as an SDNV-8 (see below). This includes the bytes required for the header type and ciphersuite ID fields.

Ciphersuite ID - identifies the ciphersuite in use. This is one byte long, though the top three bytes are used to indicate the presence or absence of the three optional fields (see below).

(optional) Ciphersuite parameters - parameters to be used with the ciphersuite in use, e.g. a key identifier or initialization vector (IV). This is encoded as an SDNV-8.

(optional) Security-source - specifies the security source for the service. If this is omitted, then the source of the bundle is assumed to be the security-source. This is encoded using the normal bundle identity encoding scheme (see below).

(optional) Security-destination - specifies the security destination for the service. If this is omitted, then the destination of the bundle is assumed to be the security-destination. This is encoded using the normal bundle identity encoding scheme (see below).

Security result - contains the results of the appropriate ciphersuite-specific calculation (e.g. a signature, or MAC or encrypted session key). This is encoded as an SDNV-16.

The ciphersuite ID is an eight bit value with the top three bits indicating which of the optional fields are present (value = "1") or absent (value="0"). The remaining five bits indicate the ciphersuite.

Some ciphersuites are specified in Section 5, which also specifies the rules which MUST be satisfied by ciphersuite specifications. Additional ciphersuites MAY be defined in separate specifications.

The structure of the ciphersuite ID byte is shown in Figure 2

src - the most significant bit (bit 7) indicates whether the ASH contains an optional security source (value="1") or not (value="0")

dest - the 2nd most significant bit (bit 6) indicates whether the ASH contains an optional security destination (value="1") or not (value="0")

parm - the 3rd most significant bit (bit 5) indicates whether the ASH contains an optional ciphersuite parameters (value="1") or not (value="0")

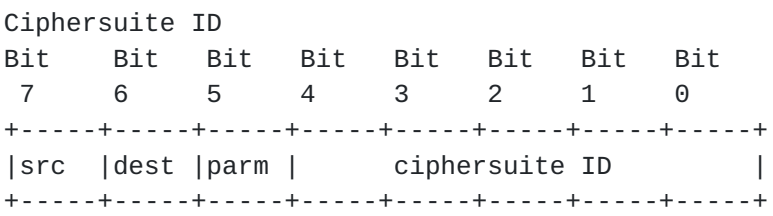

Figure 2

The Self-Delimiting Numeric Value (SDNV) scheme is a way of encoding variable length values and is described in the Licklider Transmission Protocol [4]. There are two variants described there: SDNV-8 and SDNV-16 and we use both.

The security source and destination fields are encoded as are all other bundle endpoint IDs using the schema/mark scheme specified in the base Bundle Protocol Specification. [2]

A little bit more terminology: when the header is a PSH then we refer to the PSH-source when we mean the security source field in the PSH. Similarly we may refer to the CH-dest, meaning the security-destination field of the CH.

3.2 Bundle Authentication Header

A BAH is an ASH with the following additional restrictions:

the header-type value MUST be 0x07 as defined in the Bundle Protocol Specification [2]

the flags field MUST be present as defined in the Bundle Protocol Specification

the ciphersuite ID MUST be documented as a hop-by-hop authentication-ciphersuite

the ciphersuite parameters field MAY be present

the security-source field MUST NOT be present[Comment.16]

the security-destination field MUST NOT be present

the security result is effectively the "output" from the ciphersuite calculation (e.g. the MAC or signature) applied to the (relevant parts of) the bundle.

3.3 Payload Security Header

A PSH is an ASH with the following additional restrictions:

the header-type value MUST be 0x08 as defined in the Bundle Protocol Specification [2]

the flags field MUST be present as defined in the Bundle Protocol Specification. It SHOULD indicate that the header does not need to be duplicated on all fragments.

the ciphersuite ID MUST be documented as an end-to-end authentication-ciphersuite or as an end-to-end error-detection-ciphersuite; the type of protection provided by the PSH (authentication versus error-detection) is indicated by the type of ciphersuite used

the ciphersuite parameters field MAY be present

the security-source field MAY be present

the security-destination field MAY be present

the security result is effectively the "output" from the ciphersuite calculation (e.g. the MAC or signature) applied to the (relevant parts of) the bundle.

For some ciphersuites, (e.g. those using asymmetric keying to produce signatures or those using symmetric keying with a group key), the security information can be checked at any hop on the way to the destination that has access to the required keying information.

Most symmetric PSH-ciphersuites will not require a PSH-source but simply a PSH-destination to indicate which node has access to the key needed to verify the PSH authenticator. Most asymmetric PSH-ciphersuites will use the PSH-source to indicate the signer and will not require the PSH-dest field because the key needed to verify the PSH authenticator will be a public key associated with the PSH-source.

3.4 Confidentiality Header

[Comment.17]

A CH is an ASH with the following additional restrictions:

the header-type value MUST be 0xTBD [Comment.18]

the flags field MUST be present as defined in the Bundle Protocol Specification. It SHOULD indicate that the header does not need to be duplicated on all fragments.

the ciphersuite ID MUST be documented as a confidentiality-ciphersuite

the ciphersuite parameters field MAY be present

the security-source field MAY be present

the security-destination field MAY be present

the security result normally represents an encrypted bundle encryption key

A typical confidentiality ciphersuite will encrypt the payload using a randomly generated payload encrypting key (PEK) and then use the security result to carry the PEK encrypted with some long term key encryption key (KEK). The CH-dest field will typically indicate the decryption key with which the PEK can be recovered.[Comment.19]

3.5 PSH and CH combinations

[Comment.20]

4. Security Processing

This section describes the security aspects of bundle processing. The subsections here parallel the subsections of section 4, Bundle Processing, in the Bundle Protocol Specification [2] to make it easier for the reader to understand how the security steps described herein fit into the Bundle Protocol. [Comment.21]

All Bundle Protocol Agents are required to have and enforce their own configurable security policies, whether these policies be explicit or default, as defined in Section 6.

4.1 Bundle Transmission Requests

As described in Section 4.1 of the Bundle protocol [2], which enumerates the steps in processing a Send.request primitive, all bundle protocol agents serve as Policy Enforcement Points (PEP) insofar as they enforce polices that may restrict the permissions of bundle nodes to inject traffic into the network. If a particular Send.request satisfies the bundle protocol agent's policy and is therefore accepted, then an outbound bundle SHALL be created and dispatched. The security-specific steps in processing a Send.request primitive to create the outbound bundle are as follows:

For clarity, the text below ignores the fact that the policy enforcing code MAY override all of the processing steps described. For example, it is valid to implement a bundle protocol agent which always attempts to attach a PSH. Similarly it is also valid to implement a bundle protocol agent which always rejects all requests which imply the use of a PSH. Basically, the policy code is allowed to override the algorithm described below, except in the case in which the Send.request asks for some service that is not allowed by the policy code. In this case, the bundle protocol agent MUST NOT send a bundle because to do so by sending a bundle without the requested service would mean that the bundle protocol agent would be transmitting data in a manner that is not as secure as was requested, but without the bundle node's knowledge.

If the Send.request primitive includes a Delivery Options parameter and that Delivery Options parameter includes a Payload Security Requirements item, the following steps SHALL be taken, in order, depending on the values of the Confidentiality, Authentication, and Error-detection elements of the Payload Security Requirements item. If the Send.request primitive does not include a Delivery Options parameter with a Payload Security Requirements item, processing SHALL proceed beginning at Section 4.5 below.

Confidentiality

If the Confidentiality element is present in the Delivery Options parameter of the Send.request primitive, this indicates that the payload MUST be encrypted and a CH header MUST be added to the bundle.

If the Confidentiality element is not present, a CH header SHALL NOT be created for this bundle, unless indicated by local policy.

Authentication

If the authentication element is present in the Delivery Options parameter of the Send.request primitive a PSH SHALL be created. (If both the authentication and error-detection options are indicated in the Send.request primitive, the authentication option takes precedence and authentication is provided, because authentication encompasses both authentication and error-detection.)

Error-detection

If the Error-detection element is present in the Delivery Options parameter of the Send.request primitive but the Authentication element is not present, a PSH SHALL be created.

If neither the Authentication nor the Error-detection elements of the Delivery Options parameter of the Send.request primitive are present, a Payload Security Header SHALL NOT be created for this bundle, unless indicated by local policy.

Ciphersuites and Ciphersuite Parameters

The following rules hold for each of the confidentiality, authentication, and error-detection elements:

If any element (confidentiality, authentication, or error-detection) includes a ciphersuite ID then the corresponding header (CH or PSH) and calculation MUST use this ciphersuite or else the bundle MUST NOT be transmitted/forwarded. All rules specified in the ciphersuite definition MUST be followed.

If any element does not include a ciphersuite ID then the corresponding header and calculation MUST use the default ciphersuite or else the bundle MUST NOT be transmitted/ forwarded. All rules specified in the default ciphersuite definition MUST be followed.

If ciphersuite parameters are provided they MUST be honored. It is an error to emit a bundle using some other parameters. If ciphersuite parameters are not provided then the default parameters for that ciphersuite MUST be used.

4.2 Canonicalisation of bundles

In order to verify a signature or MAC on a bundle the exact same bits must be input to the calculation upon verification as were input upon initial computation of the original signature or MAC value. Since bundles may be modified while in transit (either correctly or due to implementation errors), a canonical form of any given bundle (that contains a PSH) must be defined.

This section defines the default bundle canonicalisation algorithm.[Comment.22]

In order for the PSH to be useful, all bundle information over which the PSH is calculated at the source must be the same, and in the same order, as all bundle information over which the PSH is calculated at the point at which the PSH is being validated. Therefore, the following transmission requirements SHALL hold:

Once any of the following headers are placed in a bundle, they MUST both remain in the bundle and continue to be positioned in the same order relative to one another in the bundle as the bundle travels through the DTN to its destination. Neither the presence nor the position of these headers relative to one another is allowed to change at any time. If the bundle becomes fragmented, each of these headers MUST be present in some fragment and they must continue to be positioned in the same order following bundle re-assembly:

- Primary Bundle Header
- Dictionary Header
- Override Report-to Header
- Source Routing Header
- Payload Security Header
- Confidentiality Header
- Bundle Payload Header

If a Payload Security Header is placed in a bundle, it MUST remain in

the bundle and continue to be positioned in the same order relative to the other bundle headers as the bundle travels throughout the DTN. However, if the bundle becomes fragmented, at least the fragment containing the initial portion of the original payload (that is, the fragment that has a Fragment Offset field value of zero) MUST contain the PSH and at least some fragment MUST contain the CH.

The following additional headers MAY be present in the bundle when it arrives at a bundle protocol agent or at the destination, and if they are present, they must be (logically) removed from the bundle before the PSH is verified:

Custody related headers

Bundle Authentication Header

Bundle Fragment Header

Before the PSH is initially calculated and before it is verified by applying the appropriate ciphersuite to the appropriate parts of the bundle, the fields of the bundle that are mutable as the bundle travels from source to destination must be temporarily made irrelevant to the calculation by performing the following:

the BAH, if present, MUST be removed from the bundle

the offset values in the source, destination, reply-to endpoint, sender, CH security-source (if present), CH security-destination (if present), PSH security-source (if present), and PSH security-destination (if present) fields must be used to obtain the corresponding scheme/mark format of the endpoint IDs from the dictionary header and these endpoint IDs MUST be substituted for their corresponding fields in the concatenated bundle header for purposes of calculating the PSH authenticator over the bundle. Note that if additional bundle headers that contain offsets into the dictionary header are defined (e.g., a source route header), then the definition of the new header SHOULD specify whether or not the referenced endpoint IDs should be substituted for the corresponding offset fields in the new header when constructing the canonical form of the bundle for purposes of computing the PSH. If a new header is defined and its definition does not specify whether or not the referenced endpoint IDs should be substituted for the corresponding fields in the header, then by default they SHALL be.

the portion of the dictionary header corresponding to the custodian field (as indicated by the offset value in the custodian field of the Primary Bundle Header) MUST be zeroed out

the portion of the dictionary header corresponding to the sender field (as indicated by the offset value in the sender field of the Primary Bundle Header) MUST be zeroed out

the security information field of the PSH MUST be zeroed out[Comment.23]

4.3 Creating a Confidentiality Header

This subsection describes the steps taken to create the CH and populate it with the appropriate values as well as modify the payload field of the Bundle Payload Header.

If the CH is being created by some entity other than the bundle source, then a CH-source field MUST be placed in the CH and populated with the endpoint ID of the entity creating the CH. [Comment.24]

The values of the CH MUST be populated as described in Section 3 and according to the rules of the relevant ciphersuite described in Section 5. All ciphersuites will require the input to the security result calculation to include at least the bundle payload. Some ciphersuites will require the input to include the bundle payload concatenated with some canonical sequence of bundle header fields (e.g., source, destination, portions of the dictionary header) for which confidentiality will also be provided. Note that the security result calculated will be placed in the payload field of the bundle payload header. The CH does not include a field for the security result.[Comment.25]

4.4 Creating the Payload Security Header

This subsection describes the steps taken to create the PSH and populate it with the appropriate values.

A PSH MUST be created after all other bundle headers except for the BAH have been added to the bundle and populated with their correct values.

If the PSH is being created by some entity other than the bundle source, then a PSH-source field MUST be placed in the PSH and populated with the endpoint ID of the entity creating the PSH.

The values of the PSH MUST be populated as described in Section 3 and according to the rules of the relevant ciphersuite described in Section 5. Almost all ciphersuites will require the input to the security result calculation to be the canonical form (see Section 4.2) of the bundle.

4.5 Creating the Bundle Authentication Header

Regardless of whether or not the Send.request primitive includes a delivery options parameter, the bundle protocol agent SHALL consult its security policy to determine whether or not the bundle MUST be provided with a Bundle Authentication Header before being forwarded. If the bundle does not need to be provided with a BAH, no additional security-related processing is required.

The optional ciphersuite parameters field of the BAH SHOULD (if bandwidth permits) be included in the BAH.

The authentication information field of the PSH SHALL be zeroed out before the ciphersuite is applied to the (relevant parts of) the bundle.

4.6 Bundles received from other bundle protocol agents

The security-specific steps in processing a bundle received from another bundle protocol agent are as follows:

The receiving bundle protocol agent SHALL consult its security policy to determine whether or not the bundle is required to include a BAH. If the bundle is not required to have a BAH, then security processing on the received bundle is complete and the bundle is ready to be processed for either local delivery or forwarding.

If the bundle is required to have a BAH but it does not, then the bundle MUST be discarded and processed no further; in this case a bundle status report indicating the authentication failure MAY be generated, destined for the receiving bundle protocol agent's own endpoint.

Otherwise, if the bundle does have a BAH, then the value in the authentication information field of the BAH of the received bundle MUST be verified according to the ciphersuite specification. If the check fails, the bundle has failed to authenticate and the bundle SHALL be discarded and processed no further; in this case, a bundle status report indicating the authentication failure MAY be generated, destined for the receiving bundle protocol agent's own endpoint. Otherwise, if the BAH verifies, it is ready to be processed for either local delivery or forwarding.

4.7 Local Bundle Delivery

The security-specific steps in processing a bundle if one or more bundle nodes have registered with the bundle protocol agent to receive bundles sent to the bundle's destination endpoint ID are as

follows:

These steps SHALL be performed after the payload of the original bundle has been reassembled from the payloads of received fragments, if necessary, to reconstruct the original bundle in its entirety.

If the bundle does not include a Payload Security Header, then security processing on the received bundle should proceed from Section 4.8 below. Otherwise, if the bundle does include a PSH, then the value in the security results field of the PSH of the received bundle MUST be verified according to the rules of the ciphersuite.

If the received value fails verification, the bundle has failed to authenticate and the bundle SHALL be discarded and processed no further; in this case, a bundle status report indicating the authentication failure MAY be generated, destined for the receiving bundle protocol agent's own endpoint. Otherwise, if the check passes, the bundle has been authenticated as having been sent from the claimed source and as not having been modified since being sent.

4.8 Optionally checking for Replays

If one or more bundle nodes have indicated that they want to use the "at most once" delivery option when they registered with the bundle protocol agent using the destination endpoint ID that is in the bundle, then the bundle protocol agent MUST compare the (source, timestamp) pair values of the bundle with the local list of such values of already-received bundles.

If the pair value is a duplicate, the bundle MUST not be delivered to any node that is registered using the "at most once" delivery option.

If the pair value is unique, the pair MUST be added to the local list.

4.9 Bundle Forwarding

The security-specific steps in processing a bundle if no bundle nodes have registered with the bundle protocol agent to receive bundles sent to the bundle's destination endpoint ID are as follows:

The bundle protocol agent SHALL determine the next hop for the bundle.

The bundle protocol agent SHALL consult its security policy to determine the criteria that the received bundle ought to meet before

it will be forwarded. These criteria MAY include a determination of whether or not the received bundle must include a valid BAH, PSH or CH. If the bundle does not meet the bundle protocol agent's policy criteria, then the bundle MUST be discarded and processed no further; in this case, a bundle status report indicating the failure MAY be generated, destined for the forwarding bundle protocol agent's own endpoint. [Comment.26][Comment.27]

If the bundle does meet the bundle protocol agent's policy criteria, then the bundle protocol agent's endpoint ID SHALL be inserted into the Data Dictionary, if it is not already there, and a pointer to this string SHALL be placed in the sender ID field of the Primary Bundle Header.

If the bundle is not a bundle fragment, and if the bundle does not already have a PSH, the bundle protocol agent SHALL consult its security policy to determine whether or not it should add a PSH to the bundle. If so, the PSH SHALL be created and have its fields populated with all of the appropriate values as described in "Creating the Payload Security Header" (see Section 4.4). In particular, the forwarding bundle protocol agent MUST populate the Security Source field of the PSH with its own endpoint ID.

Next, the bundle protocol agent SHALL consult its security policy to determine whether or not the bundle needs to be given a BAH. If so, the BAH SHALL be created and have its fields populated with all of the appropriate values as described in "Creating the Bundle Authentication Header" (Section 4.5).

If the bundle is to be fragmented before being forwarded, each of the fragments MUST be given its own BAH that is specific to that fragment.

At this point, the bundle is ready for forwarding.

4.10 Bundle Fragmentation and Reassembly

If it is necessary for a bundle protocol agent to fragment a bundle and security is being used on that bundle, the following security-specific processing is REQUIRED:

If the bundle is required by the security policy to have a BAH before being forwarded, all fragments resulting from that bundle MUST contain individual BAH values.

If the original bundle had a PSH, then the fragment that is given a Fragment Offset field value of zero MUST include an exact copy of the PSH that was in the original bundle.

If the original bundle had a CH, then some fragment MUST include an exact copy of the CH that was in the original bundle.

When original bundle transmission is terminated before the entire payload has been transmitted, the receiving bundle protocol agent SHALL consult its security policy to determine whether it is permitted to transform the received portion of the bundle into a bundle fragment for further forwarding. Whether or not such reactive fragmentation is permitted SHALL be dependent on the security policy in combination with the ciphersuite used to calculate the BAH authentication information. [Comment.28]

In such cases, if the original bundle is fragmented by the recipient (reactively), and the BAH-ciphersuite is a TP-ciphersuite (see Section 5), the bundle MUST be fragmented immediately after the last hash value in the partial payload that is received. Any data received after the last hash value MUST be dropped.

If an original bundle transmission is terminated before the entire payload has been transmitted, if the truncated bundle arriving at the receiving bundle protocol agent is reactively fragmented and forwarded, only the part of the bundle that was not received must be retransmitted at the source (providing it has not yet expired). Before retransmitting this portion of the bundle, it SHALL be changed into a fragment and, if the original bundle included a BAH, the fragmented bundle MUST also, and its BAH SHALL be recalculated. If a TP-type ciphersuite is in use for this bundle, all of the hashes interspersed in its payload shall be recalculated.

4.11 Custodial Retransmission and Release

No security-specific steps relating to the custodial retransmission and release phase of bundle processing have yet been identified.

4.12 Bundle Status Report Status Flags

The following security-specific values for Bundle Status Report Status Flags have been defined in addition to those already present in Table 5 of the Bundle Protocol Specification [2]:

A source was found to lack the appropriate permissions to use the requested COS

The security policy does not permit acceptance of a bundle is received without a BAH.

A bundle failed the hop-by-hop bundle authentication check (using the Bundle Authentication Header).

A bundle failed the end-to-end authentication check (using the Payload Security Header) at an intermediate DTN security policy node.

A bundle was found to be an illegitimate replay at an intermediate node. (Note that no method of detecting such replays has yet been defined.)

A bundle failed the end-to-end authentication check (using the Payload Security Header) at the destination host.

A bundle was found to be an illegitimate replay at the destination host.

Received a payload longer than that defined by the preceding bundle header.

The indicated BAH ciphersuite is not implemented.

The indicated PSH ciphersuite is not implemented.

the indicated CH ciphersuite is not implemented.

4.13 Convergence Layer Security Services

As discussed in the Bundle Protocol Specification [2], the convergence layer SHALL deliver the Bundle.indication primitive to the bundle protocol, and one of the parameters of this primitive, Bundle Authenticity, is security-specific. The meaning of this parameter and its possible values are defined in the Bundle Protocol Specification [2].

5. Ciphersuites

Four types of ciphersuites are defined:

hop-by-hop authentication ciphersuites for use in the BAH

end-to-end authentication ciphersuites for use in the PSH

end-to-end error-detection ciphersuites for use in the PSH

confidentiality ciphersuites

The ciphersuite field of the BAH, PSH and CH are 5-bit integer values as defined below: [Comment.29]

EntireBundleHMAC - value 000

HeadOfBundleHMAC - value 001

EntireBundleSig - value 011

HeadOfBundleSig - value 010

CombinationSig - value 100

EntireBundleMAC - value 101

TP-HMAC - value 1111

TP-HSIG - value 1110

TP-DSS - value 1101[Comment.30]

5.1 EntireBundleHMAC Ciphersuite

This ciphersuite involves computing a MAC over the entire bundle using HMAC-SHA1. It is both a hop-by-hop authentication ciphersuite and an end-to-end authentication ciphersuite, so it may be used to calculate the BAH security result and/or the PSH security result.

If used to calculate the BAH security result, the authentication information field of the BAH MUST be zeroed out before the ciphersuite is applied, and the ciphersuite MUST be applied to the entire bundle.

If used to calculate the PSH security result, the bundle must be put into canonical form as described in Section 4.2 before the

ciphersuite is applied, and the ciphersuite MUST be applied to the entire bundle.

5.1.1 Ciphersuite Parameters

This ciphersuite has one associated parameter, which identifies the cryptographic key used in the HMAC computation. This parameter is encoded in SDNV-8 format.

5.2 HeadOfBundleHMAC Ciphersuite

This ciphersuite involves computing a MAC over the entire bundle except for the payload field of the Bundle Payload Header using HMAC-SHA1. It is both a hop-by-hop authentication ciphersuite and an end-to-end authentication ciphersuite, so it may be used to calculate the BAH security result and/or the PSH security result.

If used to calculate the BAH security result, the authentication information field of the BAH MUST be zeroed out before the ciphersuite is applied, and the ciphersuite MUST be applied to the entire bundle except for the payload field.

If used to calculate the PSH security result, the bundle must be put into canonical form as described in Section 4.2 before the ciphersuite is applied, and the ciphersuite MUST be applied to the entire bundle except for the payload field.

5.2.1 Ciphersuite Parameters

This ciphersuite has one associated parameter, which identifies the cryptographic key used in the HMAC computation. This parameter is encoded in SDNV-8 format.

5.3 EntireBundleSig Ciphersuite

This ciphersuite involves using SHA-1 to compute a hash over the entire bundle and then signing this hash by generating a Digital Signature Standard (DSS) signature. It is both a hop-by-hop authentication ciphersuite and an end-to-end authentication ciphersuite, so it may be used to calculate the BAH security result and/or the PSH security result.

If used to calculate the BAH security result, the authentication information field of the BAH MUST be zeroed out before the ciphersuite is applied, and the ciphersuite MUST be applied to the entire bundle.

If used to calculate the PSH security result, the bundle must be put

into canonical form as described in Section 4.2 before the ciphersuite is applied, and the ciphersuite MUST be applied to the entire bundle.

5.3.1 Ciphersuite Parameters

This ciphersuite has one associated parameter, which identifies the cryptographic key used to create the signature. This parameter is encoded in SDNV-8 format.

5.4 HeadOfBundleSig Ciphersuite

This ciphersuite involves using SHA-1 to compute a hash over the entire bundle except for the payload field of the Bundle Payload Header and then signing this hash by generating a DSS signature. It is both a hop-by-hop authentication ciphersuite and an end-to-end authentication ciphersuite, so it may be used to calculate the BAH security result and/or the PSH security result.

If used to calculate the BAH security result, the authentication information field of the BAH MUST be zeroed out before the ciphersuite is applied, and the ciphersuite MUST be applied to the entire bundle except for the payload field.

If used to calculate the PSH security result, the bundle must be put into canonical form as described in Section 4.2 before the ciphersuite is applied, and the ciphersuite MUST be applied to the entire bundle except for the payload field.

5.4.1 Ciphersuite Parameters

This ciphersuite has one associated parameter, which identifies the cryptographic key used to create the signature. This parameter is encoded in SDNV-8 format.

5.5 CombinationSig Ciphersuite

[Comment.31]

This ciphersuite involves using SHA-1 to compute two hash values: a hash over just the head of the bundle (the entire bundle except for the payload field of the Bundle Payload Header), and a hash over the entire bundle. These two hash values are concatenated (the hash of the head of the bundle preceding the hash of the entire bundle) and the concatenated result is signed by generating a DSS signature. It is only a hop-by-hop authentication ciphersuite, so it may only be used to calculate the BAH security result.

The authentication information field of the BAH MUST be zeroed out

before this ciphersuite is applied.

5.5.1 Ciphersuite Parameters

This ciphersuite has one associated parameter, which identifies the cryptographic key used to create the signature. This parameter is encoded in SDNV-8 format.

5.6 EntireBundleMAC

This ciphersuite involves using SHA-1 to compute a hash value over the entire bundle. No keys are involved. It is only an end-to-end error-detection ciphersuite, so it may only be used to calculate the PSH security result.

The bundle must be put into canonical form as described in Section 4.2 before this ciphersuite is applied, and the ciphersuite MUST be applied to the entire bundle.

5.6.1 Ciphersuite Parameters

This ciphersuite has no associated parameters, so no associated parameters fields are included in the PSH when this ciphersuite is used.

5.7 TP-type Ciphersuites

[Comment.32]

All TP-type ciphersuites involve breaking the bundle payload into segments of a given size, computing authentication information for the bundle header concatenated with that payload segment, and inserting the authentication information for a given segment into the payload itself right after the given segment. Therefore, the payload of a bundle on which the TP Ciphersuite was used to calculate the authentication information would be formatted as a sequence of payload segments of a given size interspersed with authentication information of a given size. [Comment.33]

There are several TP-type ciphersuites that have been defined. All involve formatting the payload as fixed-length segments of payload followed by fixed-length authentication information calculated over the bundle header concatenated with the previous payload segment. The only differences between the various TP-type ciphersuites is the method used to calculate the authentication information.
[Comment.34]

We may define the following TP-ciphersuites:

TP-HMAC Ciphersuites

TP-HSIG Ciphersuites

TP-DSS Ciphersuites

[Comment.35]

5.7.1 Ciphersuite Parameters

The following two parameters must be included with TP-type ciphersuites that use fixed-length payload and segment sizes: segment-length, authenticator-length. These two fields MUST be included in the BAH and PSH headers that use such TP-type ciphersuites and they MUST be encoded in SDNV-8 format.

6. Default Values

Several of the security headers include optional fields that MAY contain ciphersuite parameter information. If a bundle includes a security header that does not contain one of these optional fields, then the default value for that field SHALL be used. The default values are as follows:

BAH Ciphersuite ID: EntireBundleSig [Comment.36]

BAH EntireBundleSig parameters: The key associated with the endpoint indicated in the sender field of the Primary Bundle Header. If this value is not valid, use the key associated with the endpoint ID that was passed up as a parameter of the Bundle.Indication primitive.

PSH Ciphersuite ID when Authentication is being used: TBD

PSH Ciphersuite ID when Error Detection is being used: TBD

PSH ciphersuite parameters: If the PSH includes an optional PSH Security-Source field, use the key associated with the endpoint indicated in this field. Otherwise, use the key associated with the endpoint indicated by the source field of the primary bundle header.

CH ciphersuite ID: TBD

CH ciphersuite parameter: If the CH includes an optional CH Security-Source field, use the key associated with the endpoint indicated in this field. Otherwise, use the key associated with the endpoint indicated by the source field of the primary bundle header.

Default Security Policy. Every bundle protocol agent serves as a Policy Enforcement Point (PEP) insofar as it enforces some policy that controls the forwarding and delivery of bundles via one or more interfaces. Consequently, every bundle protocol agent SHALL have and operate according to its own configurable security policy, whether the policy be explicit or default. The policy SHALL specify:

- Under what conditions received bundles SHALL be forwarded.
- Under what conditions received bundles SHALL be required to include valid BAHs.
- Under what conditions the authentication information provided in a bundle's BAH SHALL be deemed adequate to authenticate the bundle.
- Under what conditions received bundles SHALL be required to have valid PSHs.

Under what conditions the authentication information provided in a bundle's PSH SHALL be deemed adequate to authenticate the bundle.

Under what conditions a BAH SHALL be added to a received bundle before that bundle is forwarded.

Under what conditions a PSH SHALL be added to a received bundle before that bundle is forwarded.

The actions that SHALL be taken in the event that a received bundle does not meet the receiving bundle protocol agent's security policy criteria.

This specification does not address how security policies get distributed to bundle protocol agents. It only REQUIRES that bundle protocol agents have and enforce security policies. [Comment.37]

If no security policy is specified at a given bundle protocol agent, or if a security policy is only partially specified, that bundle protocol agent's default policy regarding unspecified criteria SHALL consist of the following:

Bundles that are not well-formed do not meet the security policy criteria.

All bundles received MUST either have their authenticity asserted by the convergence layer or by valid authentication information present in a BAH. If the value of the Bundle Authenticity parameter of the convergence layer Bundle.indication primitive is "Bundle authenticity is asserted", then bundle authentication is complete. If the value of the Bundle Authenticity parameter is "Bundle authenticity is denied", or "Bundle authenticity is unknown", then the bundle MUST have a BAH and the value of the authentication information field of the BAH must be verified. In this case, if the bundle does not have a BAH, then the bundle must be discarded and processed no further; a bundle status report indicating the authentication failure may be generated, destined for the receiving bundle protocol agent's own endpoint. If the bundle does have a BAH, then the value in its authentication information field SHALL be verified.

The authentication information value in the BAH MUST be calculated using the YADA-YADA-YADA ciphersuite; that is, the value MUST be calculated over the entire/head/specific segmented portion of the bundle, using the yada-yada-yada algorithms.

No received bundles SHALL be required to have a PSH; if a received bundle does have a PSH, however, that can be ignored unless the

receiving protocol agent is the destination bundle protocol agent, in which case the PSH MUST be verified.

The security information value in the PSH MUST be calculated using the YADA-YADA-YADA ciphersuite; that is, the value MUST be calculated over the entire/head of the bundle, using the yada-yada-yada algorithms.

A PSH SHALL NOT be added to a bundle before sourcing or forwarding it.

A BAH MUST always be added to a bundle before that bundle is forwarded.

If a received bundle does not satisfy the bundle protocol agent's security policy for any reason, then the bundle MUST be discarded and processed no further; in this case, a bundle status report indicating the failure SHOULD be generated, destined for the receiving bundle protocol agent's own endpoint.

7. Security Considerations

TBD. We have to add text here describing the downside of using these mechanisms (e.g. new DoS opportunities), issues with key mgmt, implementation hints, ciphersuite strength considerations etc. etc. Ok to do it a bit later though, but we should start collecting items now.

8. IANA Considerations

TBD. We may want to have IANA register ciphersuites.

9. References

9.1 Normative References

[1] Bradner, S. and J. Reynolds, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, October 1997.

[2] Scott, K., "Bundle Protocol Specification", draft-irtf-dtnrg-bundle-spec-02.txt , September 2004.

[3] Cerf, V., "Delay-Tolerant Network Architecture", draft-irtf-dtnrg-arch-02.txt , July 2004, <draft-irtf-dtnrg-arch-02.txt>.

[4] Ramadas, M., Burleigh, S., and S. Farrell, "Licklider Transmission Protocol", draft-irtf-dtnrg-ltp-02.txt , December 2004.

9.2 Informative References

[5] Symington, S., "Delay-Tolerant Network Security Overview and Motivation", draft-irtf-dtnrg-securityOverview-spec-01.txt , March 2005.

[6] Warthman, F., "Delay-Tolerant Networks (DTNs) A Tutorial", http://www.dtnrg.org/ , March 2003.

[7] Durst, R., "An Infrastructure Security Model for Delay Tolerant Networks", http://www.dtnrg.org/ , July 2002.

Editorial Comments

[Comment.1] Editors: Actually, the "bundle security is RECOMMENDED" statement needs to be in the core Bundle Protocol spec and not here. But for now, let's put it here until we get agreement with the larger group.

[Comment.2] Editors: We're using the xml2rfc tools so embedded comments are marked like this.

[Comment.3] Editors: This is the first informative reference. We should, if we want to look like a standards-track RFC, distinguish between normative and informative but I'm not sure how to with this tool yet.

[Comment.4] Editors: There may be some (valid) question marks over this terminology. We weren't entirely sure ourselves.

[Comment.5] Editors: There is still work to be done on how to handle potential combinations of security services (e.g., confidentiality with authenticity).

[Comment.6] Editors: We discussed whether to model e.g. the "signer" as the bundle node or the bundle protocol agent that provides the bundle service to the bundle node. In the end, we didn't care too much and are happy to allow either term. The main reasons we didn't care were a) due to the fact that there's mainly no real security difference since a suitable API that properly isolates which layer is trusting which other layer for what is at least very hard, and b) the destination mostly doesn't care which layer of the cake signed in any case.

[Comment.7] Editors: This section needs to be coordinated with the Bundle Protocol Spec.

[Comment.8] Editors: The extent to which the interfaces defined in this section should reflect a 1:1 correspondence with the protocol data units is to be determined. The issue is that we want to allow policy code below the interface to influence matters, so, e.g. if the interface omits to specify, or defaults, an algorithm parameter then the implementation could select anything, presumably after running some policy code. The net effect is that the PDUs reflect some combination of the interface inputs and the policy settings.

[Comment.9] Editors: Basic confidentiality requires that the payload be encrypted; we may also want to define a confidentiality service that also keeps certain fields like source and destination confidential. It is not clear if we should define specific ciphersuites to provide specific types of confidentiality or if we should specify that a tunnel be created and used to provide confidentiality of the entire bundle. Note that encrypting certain fields like source and destination would not be sufficient, given that these fields only contain offsets into the Dictionary Header. Appropriate portions of the dictionary header would have to be encrypted as well.

[Comment.10] Editors: This might be done this way or else changed to not use the PSH in order to make it clearer that we're not getting data integrity from this service (alone). The reason to do it this way is that it provides a nice debugging option to verify much of the non-key management code in a bundle-security stack. That may, or may not be considered sufficient benefit for the potential confusion caused.

[Comment.11] Editors: Note, this parameter still has to be added to the Bundle Spec.

[Comment.12] Editors: there's definitely more to be said about replay, so consider this as a placeholder until decisions on replay get made.

[Comment.13] Editors: This can't be a MAY. There has to be some MUST somewhere.

[Comment.14] Editors: Details of this encoding scheme will change. The next-header mightn't become header-type; depending on how the sources are encoded (a moving target elsewhere) we may need to include an offset to point to the start of those and 32 is really a factor of ~3 too small for specifying all standard ciphersuites ever.

[Comment.15] This flags field is brand new; we still have to discuss how the sittings of its various bits get selected (Send.Request interface parameters, possibly overridden by security policy) as well as how it is treated. Security policy will take precedence. Will the flags be obeyed if the policy is silent?

[Comment.16] Editors: This is assuming that the primary bundle

header has a "sender" field, as has been discussed. If not, the security source field MUST be present. Requires coordination with the Bundle Protocol Spec.

[Comment.17] Editors: Note that we are defining a new Confidentiality Header and its type has to be added to Table 1 of the Bundle Protocol.

[Comment.18] Editors: Note that we are defining a new Confidentiality Header and its type has to be added to Table 1 of the Bundle Protocol.

[Comment.19] Howie: It may be nice to be able to signal both confidentiality and authentication using only one header instead of using two separate headers (PSH and CH).

[Comment.20] Editors: As you can see - this is TBD!

[Comment.21] Editors: Not sure if paralleling the bundle spec. will be a good or bad idea - it might be simpler to just do our own thing here. I guess we'll see.

[Comment.22] Editors: ...but is still mainly tbd

[Comment.23] Editors: These instruction for canonicalisation may not be exactly correct, but we think they're on the right track.

[Comment.24] Susan: What drives the inclusion of a CH (or PSH) security destination in the CH (or PSH), and where does this value come from? Is it security-policy driven? Should there be a parameter in the Send.Request service primitive to enable the bundle node to specify these security destinations, if desired?

[Comment.25] Susan: We will eventually want to say more about possibly providing confidentiality for some bundle header fields in addition to simply providing payload confidentiality.

[Comment.26] Howie: The security policy database will need to be discussed somewhere. Does it belong in this document, the bundle protocol spec., both, some other document?

[Comment.27] Editors: The paragraph below should be elsewhere; unfortunately it isn't anywhere else yet, because it was a security discussion that brought up the need for

a "sender" field. This field has not yet made it into the Bundle protocol. We must coordinate with the Bundle Protocol spec authors to insert this text into the Bundle Protocol. I have it here for now so it is at least somewhere.

[Comment.28] Editors: These are initial thoughts and may be wrong, or changed!

[Comment.29] Editors: These are really placeholders; very sketchy at this point.

[Comment.30] Add DSS, with variations for entire bundle, head of bundle... (should use of DSS be the default? Over entire bundle or just front?)

[Comment.31] Editors: this one is more efficient if we're not pkcs1v1.5 compliant.

[Comment.32] Editors: we need another name, or else have to explain what TP expands to...:-)

[Comment.33] This explanation needs to be more detailed, saying exactly what parts of the canonical form of the bundle each authenticator is calculated over.

[Comment.34] Editors: It isn't clear what entity decides the size of the payload segments and breaks the payload into these segments--the source bundle node or the source bundle protocol agent? If the source bundle node, why doesn't it just send each of these segments as separate bundles? if the source bundle node, then wouldn't the send.Request primitive need to be modified to enable the segment size to be specified? If the bundle protocol agent decides the segment size, on what does it base this decision?

[Comment.35] Howie: Should the TP segments sizes be fixed or random (smaller than some maximum)? If they are fixed, what compelling argument is there that any of them will get through? If they are random in size (and we define a TLV format for conveying the appropriate information) then maybe something will get through.

[Comment.36] Editors: this still needs to be defined in the Bundle Protocol.

[Comment.37] Howie: Eventually we will need to state where the

security policy information/DB does get discussed/
specified.

Authors' Addresses

Susan Flynn Symington
The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102
US

Phone: +1 (703) 983-7209
Email: susan@mitre.org
URI: http://mitre.org/

Stephen Farrell
Trinity College Dublin
Distributed Systems Group
Department of Computer Science
Trinity College
Dublin 2
Ireland

Phone: +353-1-608-1539
Email: stephen.farrell@cs.tcd.ie

Howard Weiss
SPARTA, Inc.
7075 Samuel Morse Drive
Columbia, MD 21046
US

Phone: +1-410-872-1515 x201
Email: hsw@sparta.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.