DTN Research Group
Internet-Draft
Expires: September 22, 2005

S. Symington
The MITRE Corporation
A. Rest
Various Organizations
March 21, 2005

Delay-Tolerant Network Security Overview and Motivation
draft-irtf-dtnrg-sec-arch-01

Status of this Memo

Copyright Notice

Abstract

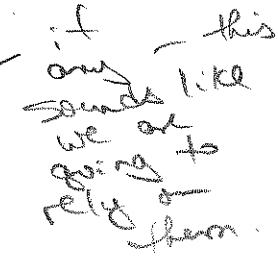
This document provides an overview of DTN security, defining a security architecture for Delay-Tolerant Networks (DTNs). It describes what aspects of the DTN architecture will be protected and the security mechanisms that will be used to provide this protection, along with design decisions and supporting rationale.

Table of Contents

The stressed environment of the underlying regional networks over which a DTN must operate poses unique challenges to the mechanisms needed to secure the DTN and thereby constrains available security solutions that can be applied to address DTN network security threats. Furthermore, it would not be unusual for DTNs to be deployed in environments where a portion of the network might become compromised, posing still further challenges to the already limited security solutions. For example,

The high round-trip times and frequent and unpredictable disconnections that may be characteristic of a DTN indicate that security solutions should not depend on frequent distribution of a large number of certificates and encryption keys end-to-end across the DTN. A solution that does not require each user's keys and credentials to be distributed throughout the network, but that requires them only at neighboring or nearby nodes, is more scalable.

Stressed regional networks also mean that there may be delayed or frequent loss of connectivity to a key or certificate server. Such disruption of connectivity to a certificate authority/key server indicates that there should be multiple such certificate authorities/key servers in the network. It also indicates that user credentials should expire periodically rather than depend on certificate revocation messages (which might never be able to be sent).



The long delays that may be inherent in DTNs mean that messages may be valid for days or weeks, so depending on message expiration

alone to rid the network of unwanted messages may not be as efficient or as feasible as in networks with different characteristics.

The stressed environment of the underlying regional networks over which a DTN must operate not only constrains the available security solutions. It also drives the design of DTN security. As discussed in [2], the DTN architecture is based on a number of design principles intended to address the challenges posed by the limitations of the underlying regional networks. In particular, the DTN security architecture is based on the fundamental design principle that security mechanisms that protect the already-limited DTN infrastructure from unauthorized use must be provided.

1.1 Scope

Given that a well-known precept of network security is that any network is only as secure as its weakest link, it follows that a DTN network is only as secure as its weakest regional network, and each of those regional networks, in turn, is only as secure as its weakest link.

Securing DTNs will be even more problematic than securing conventional wired networks or any of the other regional network types that a DTN may span. The DTN security problem space is larger

than the union of the security problem spaces of each of the regional subnetworks that the DTN overlays. DTNs are vulnerable to all of the same security threats as these regional networks, plus those threats that are specific to components of the DTN overlay, such as the Bundle Protocol, DTN routers and routing information exchange, convergence layers, and endpoint address translation. In particular, there are architecture, router, and protocol characteristics that are unique to DTNs that result in security vulnerabilities and concerns that are specific to DTNs.

Although the security of a DTN is bound up with and dependent on the security of each of the regional networks that it overlays, and anything that poses a security threat to one of these regional networks also poses a threat to the DTN, this document will not address the security of any particular regional network type. Instead it will limit itself to the definition and discussion of the security architecture of the DTN overlay, ignoring the native portions of the regional networks that must be passed through to get from one DTN node to the next.

1.2 Purpose

The purpose of this paper is to provide an overview of and

motivations for DTN security, by defining the DTN Security Architecture in terms of the security services, mechanisms, and concepts that must be integrated into the DTN architecture to make it secure. It presents the security threats to which DTNs are vulnerable and describes protection mechanisms for addressing those threats.

1.3 Related Documents

As an overview and motivations document, this paper provides a high level description of how DTN security works, introducing the security mechanisms and how they work together to provide the services needed to meet DTN security protection goals. Other documents provide more detailed definition of some of the DTN security mechanisms and protocols. These documents include:

DTN Security Protocols [8] -- a document defining the Bundle Authenticatin Header (BAH) and Payload Security Header (PSH) security headers for the DTN Bundle Protocol, as well as how those headers should be processed by bundle agents to provide security. (this document is not yet written)

Key Management for DTN -- a document defining methods of certificate and key distribution, credential assurance, and key validation and revocation for DTNs (also not yet written)

Secure DTN Multicast -- a document defining methods of supporting secure multicast in DTNs (also not yet written)

1.4 Organization and Terminology

This document, DTN Security Overview and Motivations, is defined

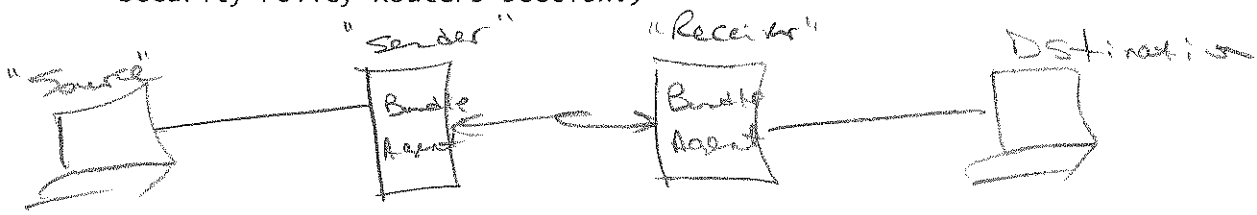
separately from the DTN Architecture [2] and the DTN Security Protocol [DTNsec--to be written] is defined separatly from the Bundle Protocol [3] because security is an optional service in DTN. A DTN does not necessarily have to incorporate the security services defined in the DTN Security Overview and Motivations document and the DTN Security Protocol. In order to provide a DTN service securely, however, a DTN node must support the DTN Security Architecture as defined in this Overview and Motivations docunent and must operate according to the DTN Security Protocol.

In the following sections we discuss the DTN security goals and the various DTN security services that have been defined to meet those goals. Discussion of each of these security services is organized according to whether the service is used to provide infrastructure protection or protection for DTN applications. We also provide a separate section for discussion of security policy routers, which may

be used to provide important access control services to DTNs.

Note that the distinction between security for infrastructure protection versus security for application protection necessitates the use of terminology that distinguishes between the terms "source" and "sender" and between the terms "destination" and "recipient". The distinctions between these terms must be properly understood in order to understand DTN security: a "source" is the entity from which the application data unit originates, and a "sender" is the DTN bundle agent that forwarded the bundle on its most-recent hop. A "destination" is the entity to which the bundle is ultimately destined, and a "receiver" or "next hop" is the neighboring bundle agent to which a sender forwards a bundle.

This DTN Security Overview and Motivations document refers to [3] and [8] insofar as it discusses the provision of various security services in the DTN Security Architecture in terms of the use of two optional security-related Bundle Protocol headers: the Bundle Authentication Header (BAH) and the Payload Security Header (PSH). Another way to understand the distinction between "source" and "sender" and the distinction between "destination" and "recipient" is that the source is the entity that encrypts the hash in the PSH and the sender is the entity that encrypts the hash in the BAH. The destination is the entity that verifies the value of the hash received in the PSH, and the receiver is the entity that verifies the value of the hash received in the BAH. (A receiver can also be allowed to optionally verify the value of the hash in the PSH for purposes of enforcing security policy, as will be discussed in the Security Policy Routers section.)



2. DTN Security Goals

Due to the resource-scarcity that characterizes DTNs, the focus of security within DTNs is on protection of the DTN infrastructure from unauthorized access and use. Specifically, the infrastructure protection goals include the capability to:

1. Prevent access by unauthorized applications,
2. Prevent unauthorized applications from asserting control over the DTN infrastructure,
3. Prevent authorized applications from sending bundles at a rate or class of service for which they lack permission,
4. Promptly detect and discard bundles that were not sent by authorized DTN nodes
5. Promptly detect and discard bundles whose headers have been modified since being sent,
6. Promptly detect and disable compromised entities.

It is important to note that the above DTN infrastructure protection goals do not include the following abilities at arbitrary DTN nodes:

1. The ability to detect bundles that have had their payloads modified,
2. The ability to detect bundles that are replays of previously-transmitted bundles, and
3. Traffic flow analysis protection.

Secondary emphasis is placed on providing optional end-to-end security services to protect DTN applications so that those applications can optionally have assurance about the data that they send and receive over the DTN, even if one or more DTN routers has become compromised. These secondary DTN security goals are to provide assurance to destination DTN applications regarding the:

1. Authenticity of the identity of a bundle's source,
2. Authenticity of the identity of the intended destination,
3. Integrity of the bundle (including of the bundle's payload) insofar as it has not been modified since being sent from the source,
4. Authenticity of the bundle insofar as it is not a playback of a previous transmission, and the
5. Confidentiality of the application data unit that is carried in the bundle, and the ability to decrypt this encrypted application data unit using the appropriate algorithm and key as identified in optional bundle fields used for this purpose.

3. DTN Security Services Overview

This section provides an overview of the security services that are and are not provided as part of the DTN security architecture. Each of the security services introduced here that is part of the DTN security architecture is discussed in detail in the following sections of the document. The security services discussed here are provided through the use of two optional security-related bundle protocol security headers and protocols related to each, combined with the use of cryptographic procedures [8].

3.1 Protect the DTN Infrastructure

As stated earlier, the emphasis of DTN security is on protection of the infrastructure from unauthorized use. Due to this emphasis on protection of the infrastructure as opposed to protection of DTN applications, the key security services within DTNs are:

1. access control, to ensure that only legitimate applications with appropriate authority and permissions are allowed to inject bundles into the network;
2. hop-by-hop sender authentication, to verify the identity of the previous-hop bundle agent that claims to have sent a bundle;
3. hop-by-hop bundle header integrity, to detect bundles that have had their headers modified since being sent from the previous-hop router; and
4. limited protection against denial of service, to ensure that some types of illegitimate traffic on the DTN are detected as soon as possible and dropped immediately upon detection. For example,:
    1. Bundles from legitimate applications but requesting an unauthorized class of service,
    2. Bundles from illegitimate bundle agents, and
    3. Legitimate bundles that have had their headers modified

By ensuring that all bundles injected into the network are from an authorized source, that the headers of these bundles remain uncorrupted, and that the amount of illegitimate bundle payload that can be injected is limited, these security services are designed to minimize the presence of illegitimate traffic in the DTN and thereby minimize the impact of such traffic on the legitimate traffic of legitimate users. The DTN Bundle Security Protocols [8] feature that is used to provide hop-by-hop sender authentication, hop-by-hop bundle header integrity, and limited protection against denial of service is the Bundle Authentication Header (BAH). The Bundle Authentication Header is computed at every sending bundle agent and checked at every receiving bundle agent on every hop along the way from the source to the destination.

3.1.1 Optionally Protect DTN Applications

Optional end-to-end security services for protecting bundle application data are also provided. These include:

1. source authentication, to enable the destination bundle agent to verify the identity of the source that claims to have originated the bundle,
2. destination authentication, to enable a destination bundle agent to verify that all bundles that it receives were in fact intended for it,
3. end-to-end bundle integrity, to enable the destination bundle agent to detect bundles (including bundle payload) that have been modified since being sent from the source or that are replays of previously-received bundles, and
4. support for application-provided data confidentiality, to protect the application data units transmitted within bundles from being disclosed to unauthorized third-parties while in transit from source to destination.

The Bundle Protocol feature that is used to provide end-to-end source and destination authentication, end-to-end bundle integrity, and support for application-provided data confidentiality is the Payload Security Header (PSH). The Payload Security Header is computed once at the source bundle agent, carried unchanged, and checked at the destination bundle agent (and possibly also at security policy router bundle agents).

3.2 Enable special nodes (security policy routers) to impose access control on bundles forwarded through them

Some DTN nodes may be optionally equipped to serve as security policy routers by having the ability to enforce their own access control policies on bundles that they receive from other DTN nodes. These security policy routers may enforce access control based on bundle source identity and permissions rather than be required to trust the access control decisions made by upstream nodes.

By enabling some DTN nodes to optionally enforce their own access control policies based on source identity and permissions, such nodes can be used to proivide :

- Perimeter protection to control access of bundles sent from an insecure bundle agent to a secure portion of the DTN, or
- A higher granularity of protection for specific designated links or subregions within a secure DTN that may require the source and legitimacy of the traffic that is admitted to be policed with a higher level of scrutiny than that which can be provided by simply trusting upstream bundle agents to have enforced an access control

policy appropriate for those specific links or subregions.

3.3 Replay Detection at arbitrary nodes is not provided

DTNs do not include any security services for detecting replayed bundles at arbitrary nodes within the DTN infrastructure. The cost of protecting against denial of service attacks as executed via replay attacks has been judged not to be worth the benefit of that protection for two main reasons:

1. All replay detection approaches that have been devised so far require each DTN node to maintain an excessive amount of state, and
2. Some replays, such as retransmitted bundles and replicative routing information, are legitimate; additional bundle headers would be required to distinguish legitimate from illegitimate replays.

Instead of detecting and discarding replayed bundles, DTNs may rely on lower layer protocols to detect replays, they may require the use of routing protocols that are inherently loop-free, or they may employ mechanisms for detecting and discarding bundles that recirculate excessively through the DTN as discussed in Section 7. Even though protection of the DTN infrastructure against replay attacks is not provided, DTN does optionally support protecting applications from receiving replayed bundles. While bundle agents at arbitrary DTN nodes are not equipped with this capability, bundle agents at destination hosts and at security policy nodes do have the optional ability to keep a local list of already-received bundles, check all received bundles against this list, and discard duplicate bundles. Whether or not a bundle agent at a destination host enforces these "at most once" semantics is at the discretion of the DTN application that has registered an interest in the received bundles.

3.4 Traffic Flow Analysis Protection is not provided

DTNs do not include any security services for protecting against traffic flow analysis. Traffic flow analysis protection is probably not a realistic goal for DTNs, given their tendency to be resource-scarce.

3.5 Key Management and Distribution is required, but not yet defined

All DTN security services require the support of an appropriate cryptographic key management service to generate and distribute cryptographic keys among the necessary DTN users and DTN nodes. DTNs are architecturally flexible with regard to cryptographic algorithms that can be used on them-either public key or private key

cryptography is possible. However, this document does not yet adequately address the issue of cryptographic key management and distribution for DTNs. It assumes that DTN nodes have access to the necessary cryptographic keys (either public or private as appropriate) and that these keys could be revoked in the event that a node becomes compromised. It further assumes that the performance characteristics of DTNs will limit the scalability of key management and distribution across the DTN overlay.

3.6 Routing Protocols can be provided with security just like any other DTN application

Router and routing protocol security services will also be vital to DTN security because DTNs are extremely vulnerable to router and routing protocol, compromise. DTN routing protocols have not yet been defined, so it would be premature to discuss security features specific to any routing protocol. If routing is treated as a DTN application, however, so that routing information exchanged between two DTN routers would be treated as application data being transmitted between a source and a destination, DTN routing protocol information exchange may be provided with the same bundle integrity protection and endpoint authentication services that may optionally be provided to all DTN applications using the Bundle Protocol Payload Security Header (PSH) mechanism. Treating routing information exchanges between routers in this way would enable the integrity of this routing information to be verified, thus further protecting the infrastructure from the potential harm of corrupt routing information.

True depends but definition of optional as previously discussed.

This document does not address the issue of router security for DTNs. If a DTN router becomes compromised, there are two types of threats posed to DTN security. One threat is that the compromised router that is in the path of a bundle that has originated at another source will be able to modify the payload of this bundle that has originated from another source and forward this bundle on to its destination without the modification necessarily being detectable. The second threat is that the compromised router will be able to originate a bundle within the network, and that the bundle it originates could have content that actually damages the DTN infrastructure by providing damaging routing information. The first threat, to application data that originates at another source other than the compromised router, can be protected against using the end-to-end security information provided by the PSH, if this security header is used by the source. The second threat is far more troubling because it makes it difficult and probably impossible to achieve the goal of infrastructure protection until the compromised router is detected and disabled, and such detection can be difficult.

A compromised router will be in possession of ~~both~~ valid encryption keys for encrypting both BAH hashes on bundles that it is merely forwarding, and keys for encrypting PSH hashes on bundles of which it is the originator. When the compromised router forwards bundles to its adjacent routers, the adjacent routers will accept those bundles, authenticate them on a hop-by-hop basis using the BAH, and act upon them accordingly. If the compromised router was not the source of the bundle, the compromised bundle's adjacent router will forward the bundle on toward its destination. If the compromised router has modified the payload in these bundles and if the bundles have been protected using end-to-end authentication, the destination bundle agent will detect this modification and discard the bundle. In this way, the end-to-end security provided by the PSH can be used to protect application data even in the case in which one or more DTN routers has become compromised, as long as the compromised router was not the source of the bundle and as long as the bundle reaches its destination. That is, the PSH can be used to protect the integrity and authenticate the source of bundles transmitted over the DTN if

the bundles actually reach their destination, if there is one or more compromised routers. However, if there are one or more compromised routers, there is no guarantee that any bundles will actually be received at their destination, because of the second and more serious threat that a compromised router can itself originate bundles that damage the DTN infrastructure, for example, a bundle that contains false routing information, This is the ~~second and~~ far more troubling threat, which cannot be protected against by the PSH. A compromised router could ~~source~~ routing information that is false and damaging to the DTN infrastructure itself, and the adjacent router would authenticate the BAH of this bundle, the destination bundle agent would authenticate the PSH of this bundle, and the false routing information would thereby be fully authenticated. This document does not address the issue of router security, how to assure software integrity of the DTN router, or the network management and security issues of detecting that incorrect routing information is being promulgated and determining which router may be the source of this misinformation.

4. Infrastructure Protection

Infrastructure protection--preventing unauthorized use of the infrastructure and detecting and discarding unauthorized traffic as soon as possible--is the focus of DTN security. To this end, the DTN infrastructure is protected by access control, hop-by-hop data integrity, and sender authentication services. There is no support within the DTN security architecture for replay detection at an arbitrary DTN node. Nor is there any support for protection against traffic flow analysis. The following subsections will discuss the role played by various security services in protection of the DTN infrastructure.

4.1 Traffic Flow Analysis (no protection)

already in 3.4 — why state twice? If we want a detailed discussion here, then just use bullets in section 3.

Typically, the way that a network would protect against traffic flow analysis would be by generating random traffic for transmission among source and destination nodes that do not have a need to communicate, padding legitimate traffic to disguise the amount of traffic being transmitted, and encrypting source and destination addresses to prevent the disclosure of which endpoints are communicating with which other endpoints. All of these techniques are designed to disguise which traffic on the network is legitimate and which is not, so that a malicious third party that is monitoring activity on the network would not have legitimate traffic flow information on which

to perform an analysis. The Bundle Protocol specification does not attempt to provide protection against traffic flow analysis in any way. There is no provision for source or destination addresses to be encrypted. Furthermore, attempts to disguise the true traffic flow of a DTN would most likely be counterproductive. Generating additional traffic for the purpose of disguising legitimate traffic would further strain already scarce network resources and, as such, the benefit of doing so would probably not be worth the cost.

But this could be done at lower layers - eg on a link-by-link basis if needed.

4.2 Access Control Protection

DTN access control is provided by the bundle agent of the source host. When the bundle agent receives a Send.request primitive from a bundle application, the first step it may take is to check the permissions of the requesting application and limit or enforce access control based on local policy and source application permissions. In this way, the source host can play a crucial role in enforcing appropriate access control to DTN resources. This source host access control, in combination with the hop-by-hop security checks performed using the Bundle Authentication Header, is designed to ensure that DTN nodes discard illegitimate traffic as early as possible and thereby minimize the network resources that such traffic is permitted to waste. Using this combination of source host permissions checking

and hop-by-hop security checks, any given DTN router that receives a bundle can be assured that according to the local policy of its admitting host, the bundle had permission to use the DTN network at the specified Class of Service (COS).

This ability of the source host to enforce access control on application traffic injected into the DTN is an important security feature for screening out traffic that should not be admitted to the DTN. However, the access control policy enforced by the source host, while it may be appropriate for the local regional network on which the host is located, may not be restrictive enough for other regional networks located elsewhere within the DTN. If a DTN is comprised of various regional networks with widely differing characteristics, and therefore widely differing access control needs, then the access control decisions made at a given source host may not be restrictive enough to control access to a particular link or region in a distant part of the DTN. Portions of the DTN that are much more highly bandwidth restricted or more highly constrained in some other way than the source host's portion of the DTN may not be able to be properly protected by the source host's bundle agent's access control policy if that policy is based on the characteristics of the host's local regional network rather than on the more highly constrained portion of the DTN. For this reason, there may sometimes be a need for security policy routers to provide additional access control at designated locations within the DTN to protect more highly constrained portions of the network.

Bring up lots of issues, questions, problems

How is policy set?

How is policy updated?

How are ACLs created + updated?

End-to-end policy issues are really hard - like back in the early Internet days with different technologies + different rate (eg SATNET)

Furthermore, the bundle agent of the source host can only be relied upon to provide appropriate access control if the bundle agent is operating correctly and has not been compromised. There may be some portions of the DTN that are so sensitive to intrusion by unauthorized traffic that that portion of the network cannot afford

to take the risk of assuming that every source host is operating correctly and has not been compromised. Such portions of the DTN may use security policy routers, which are under their control and which can therefore be considered more trusted than source hosts, to provide access control at their perimeter. Security policy routers, therefore, enable both a more trusted form of access control and a higher granularity of access control to be enforced within the DTN than may have been available at the source host. Security Policy routers are discussed later in Section 6.

we really don't have data integrity - we have header integrity

4.3 Data Integrity Protection (hop-by-hop)

The Bundle Protocol supports a mandatory data integrity service along every hop in the DTN network, which may be provided in one of two ways:

1. The Bundle Protocol provides a mechanism to enable a receiving DTN bundle agent to verify that there have not been any modifications to the bundle (except possibly to the payload) since the bundle was sent by the previous bundle agent along the most recent hop.
2. The data integrity of each received bundle may be asserted by the convergence layer of the receiving host, most likely based on the use of security features available in the underlying regional network, such as IPsec or link encryption, that are in use along the most recent hop.

may not provide data integrity

The principle underlying DTN hop-by-hop data integrity is that each DTN node on the path from source to destination is able to verify that the bundle (except for possibly the payload) it receives has not been modified since being forwarded by the previous DTN node across the most-recent hop. This type of integrity is vulnerable to compromise of the private or secret key of any DTN nodes along the path from source to destination. If the key of a given node has not been compromised then a bundle sent from that node to an adjacent node can be verified by the recipient node not to have had any portion except possibly its payload modified while in-transit between the two nodes. Therefore, if the private keys of all of the nodes along the path from source to destination have not been compromised, then the bundle (except its payload) sent along this path from source to destination can be verified by the destination node not to have been modified while in-transit on the path from source to destination. However, if the private key of even just one of these nodes is compromised, then a bundle sent through that node cannot be provided with integrity, and a receiving destination node has no assurance that any portion of a bundle received from any path that includes that node is the same as when the bundle was originally sent.

where? Architecture? Bundle space? MAC? what algo? Keyed hash? Signature? DSS?

The mechanism that the Bundle Protocol uses to provide hop-by-hop integrity is the Bundle Authentication Header (BAH) combined with cryptography. As currently specified, each DTN node's bundle agent computes a hash over the entire bundle, excluding the payload, encrypts the hash, and then places the signed hash value in the Bundle Authentication Header. Upon receipt of this bundle at the

next hop node, the receiving next-hop node decrypts this received hash, computes its own hash over the bundle (excluding the payload) received, and compares the two values to ensure that they are equal. Using this process, the receiving node can determine whether any portion of the bundle, excluding the payload, has been modified while in transit along the most recent hop.

4.3.1 Including the payload in the hash calculation is incompatible with reactive fragmentation

By computing the signed hash in the BAH over the entire bundle except for the payload, the authentication information only enables the receiving DTN node to determine if the non-payload portion of the bundle has been modified while in transit over the most recent hop. Any modifications to the bundle payload would not be detected. While it may seem preferable to include the bundle payload in the signed hash calculation and thereby be able to detect modifications to the payload as well, excluding the payload from the signed hash calculation is required in order to be able to take advantage of a DTN feature known as reactive fragmentation.

Reactive fragmentation is the process of fragmenting a bundle at a receiving node rather than at a sending node. If a bundle is in the process of being transmitted when a link goes down, causing the bundle to be truncated, the process of reactive fragmentation can be used to enable all of the data that was able to traverse the link successfully to proceed on to its destination as a fragment without requiring that any of the data that had been transferred successfully be retransmitted. This enables DTN routers to forward bundles and bundle fragments as soon as they are received, effectively providing cut-through delivery of bundles. Given the bandwidth limitations and frequent disconnections inherent in DTNs, such cut-through delivery is essential for being able to efficiently take advantage of connectivity when it is available.

Enabling bundles to be fragmented reactively is an important feature of DTNs because it is crucial to the ability of DTNs to forward data efficiently despite unpredictable or frequent loss of connection. Unfortunately, reactive fragmentation is incompatible with calculating the hash value in the Bundle Authentication Header (BAH) over the bundle payload as well as the bundle header. If the authentication information is calculated over the entire bundle, then when transmission of a bundle is truncated before the entire payload has been received, the receiving node has no way of authenticating the portion of the bundle that was received because it needs the entire bundle in order to calculate the hash of that bundle and compare it with the received hash.

There is no known algorithm that enables a hash that has been computed on an entire bundle to be used to authenticate only an initial segment of that bundle that has been received. Nor does a node that receives only the initial part of a truncated bundle have the option of waiting for the rest of the bundle to be received so

that it can then authenticate the entire bundle and forward it. Having the receiving node wait for the rest of the bundle would

require the retransmitted portion of the bundle to follow the same route that the initial portion of the bundle followed, but constraining the routing of bundles in this way is not part of the bundle protocol and would be undesirable. Consider, for example, a plane flying over a node and receiving 90% of a bundle transmitted from that node during the flyover. There may not be any more flyovers scheduled for a long time; waiting for another flyover before the rest of the bundle can be received and authenticated would be unacceptable. Instead, the rest of the bundle would have to be transmitted over a different medium using a different route, such as that provided by a satellite transmission.

A method of enabling reactive fragmentation to be compatible with authentication of the entire bundle that was considered was an approach of arranging the payload as alternating segments of payload followed by encrypted hash values of those segments, so that if a bundle were truncated during transmission, all of those payload segments that got through with their encrypted hashes could be authenticated at the receiving node, and only the partial last payload segment that was truncated would have to be discarded at the receiver and retransmitted at the sender. This approach of segmenting the payload is depicted in [%%xref to Figure] below. [%%insert figure here] Note that the appeal of this solution of hashing segments of payload is that those segments from a truncated bundle that do reach the destination do not need to be retransmitted. However, the question of the optimal segment size to use is not easy to address, and the requirement that each segment have its own encrypted hash could substantially increase bundle overhead. If a small segment size is chosen, then many signed hashes would be necessary for each payload, thus increasing the size of the bundle. If a large segment size is chosen, then potentially large segments of a bundle would have to be retransmitted in the event of bundle truncation. Therefore, the benefit of calculating the signed hash over the payload is not worth the complication and cost that would be incurred to implement such a segmented payload solution. The protection of the infrastructure that is provided by calculating the signed hash over all other portions of the bundle except the payload is sufficient for providing the necessary infrastructure protection.

Calculating the hash over the entire bundle except for the payload addresses the incompatibility between reactive fragmentation and bundle authentication. By calculating the hash over the entire bundle minus the payload, when a bundle is truncated during transmission of its payload, the bundle header can be authenticated and the bundle header and portion of the payload that was received can be forwarded on without having to wait for the rest of the bundle and without having to wait for any data that has already been received to be retransmitted. Defining the calculation of the signed

hash to exclude the payload has made authentication compatible with reactive fragmentation.

Although the integrity protection provided by this way of calculating the BAH hash is not ideal insofar as it does not protect the bundle payload, it does still provide other important protections insofar as it enables the detection of modifications to all other portions of the bundle, the corruption of which could be detrimental. Protecting the bundle header with the signed hash enables the DTN infrastructure to be protected from a malicious person injecting a bogus bundle into the network at some arbitrary point and having that traffic carried beyond its next hop. It also ensures good routing, as protecting the integrity of the bundle header authenticates the bundle source and destination values. The CoS field is also protected, ensuring that any attempt to modify the class of service information will be detected. Granted, leaving the payload itself out of the calculation means that the payload can be modified, truncated, or spoofed altogether. However, the payload length field will be included in the hash calculation and so it will not be possible for a malicious third party to inject more payload than is allowed by the length field. The purpose of the BAH can be understood as protection of the bundle transmission infrastructure rather than provision of a security service to bundle applications. While it does not protect the infrastructure perfectly, the protection that it provides is adequate and its imperfections are deemed worth the fact that it is compatible with reactive fragmentation. Furthermore, additional, optional mechanisms may be used to protect the payload and thereby provide security services to bundle applications if needed, as will be discussed later in Section 5.

Therefore, to summarize, as long as none of the private keys of any of the DTN routers in the path from source to destination have been compromised, the end-to-end integrity of the bundle (excluding the payload) transmitted can be ensured as a concatenation of the hop-by-hop integrity that is provided by the BAH. Any modifications, intentional or otherwise, that have been made to a bundle (except modifications to the payload) while in transit will be detected at the next DTN node that receives the bundle and will cause the bundle to be discarded. Any replayed bundles, however, will not be detected; they will instead be forwarded on toward their destination.

4.4 Sender Authentication (hop-by-hop)

Hop-by-hop sender authentication is very closely associated with hop-by-hop data integrity because the same mechanism is used to provide both services. As with the hop-by-hop data integrity service, the hop-by-hop sender authentication service is mandatory, and may be provided in one of two ways:

1. A sender authentication service may be provided by the Bundle Protocol of each DTN node along the path from source to destination using the authentication information in the Bundle Authentication Header, or

2. The sender authentication of each bundle may be asserted by the convergence layer, most likely based on its use of security features available in the underlying regional network, such as IPsec or link encryption.

Because the same mechanism is used to provide both hop-by-hop data integrity and hop-by-hop sender authentication, it is not possible for one of these services to be provided without the other using the Bundle Protocol. Either both hop-by-hop data integrity and hop-by-hop sender authentication are provided, or neither is.

4.4.1 Hop-by-hop endpoint authentication

Cryptography, in combination with the Bundle Authentication Header, is the mechanism that is used to provide both hop-by-hop data integrity and hop-by-hop sender authentication. When a router receives a bundle, it authenticates the bundle by looking up the appropriate key (public or symmetric) that is associated with the sent bundle, using this key to decrypt the hash value received in the Bundle Authentication Header, generating its own hash for the bundle, and verifying that the decrypted hash value it received is equivalent to the hash value that it computed.

If the values are not equal, the bundle has failed to authenticate. If they are equal, the contents of the bundle (except the payload) have been verified not to have changed since being sent from the sender (as discussed above in Section 4.3). Furthermore, the endpoint ID of the sender of the bundle has also been authenticated, because this value was included in the information over which the hash was calculated. Also, given that cryptography was used to protect the hash, when the signed hash decrypts correctly, this means that the sender's key must have been used to sign the hash, and given that this is either a private key that is known only to the sender or a symmetric key that is known only to the sender and the receiver, the successful decryption of the hash implicitly authenticates the sender as having in fact encrypted the hash and sent the data. Therefore, assuming that the key of the sending router has not been compromised, the same process that assures the hop-by-hop data integrity of the bundle also ensures its hop-by-hop sender endpoint authentication. If the sender's key has been compromised, nothing is assured. If private key cryptography was used, the sender must have been either the sender or the recipient, since those are the only two bundle agents that are assumed to share the key that was used in the symmetric encryption of the hash value. If this key has been

compromised, nothing is assured.

Note that depending on the type of cryptography used and the extent to which keys are shared, the endpoint ID of the recipient is not necessarily authenticated in this process, because the endpoint ID of the recipient is not present in the bundle and therefore was not included in the information over which the hash was calculated. Therefore, the process of authenticating the bundle does not necessarily serve to verify that the bundle was intended for the DTN node that received it (as opposed to having been a replay of a bundle that had previously been legitimately sent to a different DTN node).

This threat is further discussed below in Section 4.5.3.1.2 .

4.5 Limited Protection Against Denial of Service (DOS) Attacks

Denial of Service (DoS) is impossible to completely guard against in any network. Anyone can cut a wire, cause a power outage, or perform other drastic acts to render a network inoperable.

In a network that requires the use of cryptographically based security service for the transmission of data, a successful denial of service attack can be as simple as targeting the automated key management and distribution service. Preventing users and nodes from receiving the keys that they will need to sign messages, encrypt messages, or decrypt messages can effectively deny service on the network. Without valid, current keys, a secure network can be brought to a standstill.

Despite how futile it is to attempt to protect against the above types of denial of service attacks, there are other types of attacks that depend upon the network being operable, and some of these can be prevented or limited if the right measures are taken. These types of denial of service attacks often involve flooding all or part of a network with large amounts of data to overwhelm specific nodes or links so that those nodes or links will spend so much of their cycles and bandwidth servicing the bogus traffic that they will not have any available resources remaining to service legitimate traffic. DTNs are especially susceptible to these types of denial of service attacks because of their limited resources, so DTNs include security mechanisms at every node that are intended to provide limited protection against these types of attacks.

The use of access control as enforced by the bundle agent of the source DTN host, the use of additional access control as enforced by security policy routers and the use of sender authentication and data integrity as enforced at every DTN router combine to provide protection against denial of service attacks for DTN networks. The following subsections describe how access control, sender

authentication, and data integrity services form an integrated but limited protection against DOS attacks in the DTN. One subsection also addresses the fact that the DTN security architecture has not been designed to protect against DOS attacks as executed by replay attacks.

4.5.1 Protection against DOS as executed by unauthorized access via legitimate bundle agents

One security service that the DTN bundle layer provides to protect against denial of service is access control at every DTN host. Every bundle agent is capable of enforcing policy to restrict the permissions of its local applications, so that any bundle agent that receives a bundle from a local bundle application may, depending on the permissions of the source user and other policy considerations, limit the rate at which the application is permitted to inject traffic into the network or limit the class of service options that the application is permitted to use. Hence, access control at the

source host bundle agent is the first line of defense against the injection of traffic that may be used to launch a denial of service attack.

4.5.2 Protection against DOS as executed by sending bundles from illegitimate bundle agents or by modifying the headers of bundles that were sent by legitimate bundle agents

Two other important security services that the DTN bundle layer uses to protect against denial of service attacks are the provision of hop-by hop data integrity and hop-by-hop sender authentication services. These two services combine to ensure that when a bundle is forwarded to any DTN node within the network, the receiving DTN node has the ability to verify the identity (and thereby the validity) of the sender and to detect whether or not the header of the bundle (but not the payload) has been modified since being forwarded by the sender. If the receiving DTN node determines that the bundle was not in fact forwarded by the node that claims to have forwarded it, or that the bundle header has been modified in transit, the receiving DTN node is obligated to discard the bundle. In this way, such illegitimate traffic that is injected into a DTN will be detected at the next hop router and discarded, so it will at most impact one link and the receiving next-hop node. Furthermore, in response to this detection of illegitimate traffic, the Bundle Protocol has been designed so that it is not required to generate any status report or error messages, the stimulation of which could themselves result in a denial of service attack.

Ironically, the use of data integrity and source authentication services with any network (DTN or not), opens that network up to a

more targeted and subtle type of denial of service attack that is made possible by virtue of the fact that the network will discard transmitted data that does not pass the sender authentication and data integrity checks. For example, within a DTN network, where each node verifies the authenticity of the sender and the integrity of the bundle, a malicious third-party could prevent a bundle from being received by its destination simply by modifying a single bit or two of that bundle while in transit on a link. This change to the bundle would cause the bundle's signed hash value to be incorrect, and the next-hop DTN node would be obligated to discard the bundle for having failed the authentication and integrity checks. Again, however, this is a more general problem regarding protection against denial of service attacks in many types of secure networks and is not specific to DTNs.

4.5.3 No protection against DOS as executed by replay attacks

As described earlier, the security services of access control, data integrity, and sender authentication combine in DTN to provide integrated but limited protection against DOS attacks. This protection is described as limited because the DTN architecture is susceptible to replay attacks. Ideally it would be desirable to protect the DTN infrastructure from DOS as executed by replay attacks. However, the cost of doing so is deemed to be too high relative to the perceived risk of such an attack and the benefits of

protecting against such an attack. The decision as to what security services to provide in any architecture requires weighing the benefits that the security will yield against the costs of providing that security. Such analysis involves considering the tradeoffs among various factors, such as the vulnerabilities of the architecture; the perceived threats to the architecture; the likelihood that attacks targeting those threats will be mounted, as determined by the difficulty and cost of mounting such attacks compared to comparable attacks; the severity of the harm that a successful attack could inflict; and the cost of providing the security services. In arriving at the decision not to protect the DTN infrastructure from a denial of service attack as executed by replaying bundles, the following considerations come into play:

1. Given the limited resources inherent in DTNs, DTNs are vulnerable to many kinds of denial of service attacks. Limited bandwidth links, frequent disconnection, and varying delays already serve to naturally limit the service that DTNs can provide, so any attack that increases the severity of these limitations could easily become a denial of service attack. Therefore, if practical, it makes sense to try to limit the ability of an enemy to mount a denial of service attack on DTNs.
2. There are numerous types of denial of service attacks and many of these are impossible to completely guard against. Anyone can cut

a wire, cause a power outage, or perform other possibly more drastic acts to render a network inoperable. Given that it is relatively easy to mount one of these types of denial of service attacks, it doesn't seem to be practical to spend an inordinate amount of effort or resources to try to guard against other types of denial of service attacks that may be more difficult or costly to mount. Mounting a replay attack, for example, requires a malicious third party to both eavesdrop on bundles and to inject replays of the eavesdropped bundles onto the network at an appropriate location, which is not simple or straightforward on many networks. Because replay attacks are expensive and difficult to mount compared to other types of DoS attacks, there is questionable practicality in incurring the cost and effort that would be required to protect against replay attack-based DoS attacks in particular.

3. Some types of denial of service attacks involve flooding a network with large amounts of traffic to overwhelm specific nodes or links. Because of their limited resources, DTNs are susceptible to these types of denial of service attacks. The Bundle Protocol does include protections that are designed to limit such denial of service attacks. For example, every bundle agent is capable of enforcing policy to restrict or deny the permissions of its local applications to inject traffic into the DTN. This access control at the source host is the first line of defense against denial of service attacks, because it prevents unauthorized users from injecting traffic into the network. A second mechanism that the Bundle Protocol uses to limit flooding denial of service attacks is the Bundle Authentication Header (BAH). The BAH includes a cryptographically-protected hash of the Bundle (minus the payload), which enables the receiving node at each hop to be sure that the received bundle was forwarded by an authorized sender and that the bundle header has not been

modified since being forwarded by that sender. Given that the bundle header includes the information regarding the class of service and length of the bundle, among other things, use of the BAH enables the receiving node to detect an attempt by a malicious third party to inject bogus bundles, to modify the class of service that had been requested for legitimate bundles, or to add unlimited bogus payload traffic to legitimate bundles. If any of these conditions are detected, the receiving node will discard the bundle, limiting its impact to only one link and the receiving node.

4. Despite the ability of the BAH to limit flooding-based denial of service attacks, it is not able to completely prevent them insofar as it is not designed to detect replayed bundles. Therefore, DTNs are susceptible to denial of service attacks as executed by the repeated transmission of replayed bundles. The replay attacks that can be mounted on a DTN, however, are limited

by the fact that replayed bundles can only be injected on the link from the original sender to the original receiver, or, if public key cryptography is used or if private key cryptography is used and multiple adjacent routers share the same private key, on a link to another receiver that is adjacent to the original sender. Replayed bundles cannot be injected at arbitrary locations within the DTN. Furthermore, the time period over which a bundle can be replayed is limited by the bundle's lifetime as specified by its Expiration Time field value. Thus, the amount of damage that replay attacks can do in DTNs is inherently limited by network topology and bundle lifetime. Unfortunately, however, given the inherent limitations of a DTN, bundle lifetime in a DTN may be measured in terms of days or weeks.

5. Fully defending against replays in a DTN context would be costly. As a first example, consider the type of replay in which a bundle that is sent from router X to router Y could be replayed on the link from router X to router Z. If public key cryptography were used to encrypt the BAH hash in this bundle, the bundle would be authenticated by router Z upon receipt because even though the bundle was not intended for Z, there would be no way for Z to detect this. Protecting against this type of replay would require the BAH's signed hash, which is used to authenticate the bundle, to be computed over the receiver's endpoint ID. Including this information in the signed hash computation, however, is not feasible in DTN networks because DTN senders do not always have a specific next hop to which they are forwarding a bundle. The sender may not have any way of knowing in advance which of several possible receivers will receive the bundle it transmits. Therefore, there is no easy way within the context of DTN to protect against such a replay attack. Fortunately, this type of replay attack is limited by network topology and bundle lifetime, as previously discussed.
6. As a second example, consider the type of replay in which a legitimate bundle is recorded and then later replayed over and over along its original transmission route. This type of replay attack is possible because DTN nodes do not check bundles for uniqueness. As long as a bundle has not expired, if its BAH encrypted hash information authenticates, it will be forwarded,

even if it has already been received and forwarded many times. To detect this type of replay attack, every DTN node could be designed to check bundles for uniqueness and discard duplicates. Each bundle could be uniquely denoted by a (source endpoint ID, time stamp, fragment offset) tuple because every bundle is uniquely denoted by its (source, time stamp) pair, and, if fragmented, further identified by its fragment offset value. If every DTN node were to maintain a local list of tuples corresponding to those bundles that it had received, it could

check every received bundle against this list to detect duplicates. This would not, however, enable a node to distinguish bundles that have legitimately been retransmitted by their custodian from bundles that have been maliciously replayed, and the ability to legitimately retransmit bundles when necessary is essential to the correct operation of the DTN. Two methods have been suggested for enabling a node to distinguish a legitimate retransmission from a legitimate replay:

The first method suggested for enabling legitimate retransmissions to be distinguished from replays is to, upon retransmission, have the retransmitting node add a field to the bundle header, possibly indicating the bundle's retransmission time, to enable this retransmitted bundle to be uniquely distinguished from the original bundle. Hence, the list of tuples that would have to be maintained at each node to detect replays would include (source, time stamp, offset, retransmission time) for each bundle. This method would work, but it has the undesirable affect of increasing the size of the bundle header by the number of bits that would be required for the new "retransmission time" field.

The second method suggested for enabling legitimate retransmissions to be distinguished from replays is to associate with every link of the network an assumption regarding what a "reasonable" transmission time is for that link. Then, when a duplicate bundle is received, the node would make a determination regarding whether the time that has passed since the original transmission of the bundle is "too long" for the duplicate bundle to be a retransmission, meaning that it must be a replay. This method has the appeal of not requiring the number of bits in the bundle header to increase, but it suffers from impreciseness inherent in the fact that there is no way to effectively determine how long is "too long" for each link, especially in a DTN in which frequent and unpredictable disconnection may be the norm.

A major problem with both of these methods, however, is that they require every DTN node to maintain a local list of received bundles (or at least received bundle tuples). Even though this list of tuples could be purged periodically according to the expiration time of each bundle, it is expected that this list would quickly become too large and unwieldy at each DTN node, making its maintenance and use infeasible.

In summary, replay attacks are expensive and difficult to mount versus other types of DoS attacks, and they would be costly to

protect against relative to limited DTN resources, so it does not make sense from a cost/benefit tradeoff analysis viewpoint to, as a

matter of course, incorporate mechanisms for defending against them. Furthermore, replay attacks are sufficiently limited by time and topology that, as far as protecting against Denial of Service attacks as executed by replay attacks, the Bundle Protocol is deemed to already provide "good enough" protection given the costs versus benefits of incorporating more specific replay protections.

It should be noted that by taking this approach and explicitly not building any mechanisms into the bundle protocol to detect replays, we are either placing a burden on our routing protocols that they be inherently loop-free, because we will not have a mechanism for detecting routing loops, which would manifest themselves as replays, or a mechanism for detecting and eliminating routing loops will be required.

4.5.3.1 Threats to which the DTN is Vulnerable

In this section we document specific types of replay attacks that can be mounted that are not currently protected against.

4.5.3.1.1 Replayed packets are not detected

Typically, replay attacks would be protected against by having each packet (or, in this case each bundle) be uniquely identifiable, and then having every node maintain a list of bundles that have been received. All bundles received would be checked against this list for uniqueness and discarded if they were found to have already been received. In the case of bundles, every bundle is uniquely identifiable by its (source ID, creation timestamp) pair value. If a bundle is fragmented, however, every fragment created from the original bundle has the same (source ID, timestamp) pair. So, to further differentiate fragments of the same bundle from each other, the bundle fragment offset value in the Bundle Fragment Header would have to be used. The (source ID, creation timestamp, and fragment offset) tuple values serve to uniquely identify bundles in a DTN. In the DTN protocol, there is no mechanism defined to automatically check bundles for uniqueness, nor is there any other mechanism that has been defined to identify replayed bundles. Several methods for possibly identifying replayed bundles were considered, but no option was selected because, as discussed above, the cost of implementing such an approach, in terms of the large amount of state that would have to be stored at each node, was deemed to be too high relative to the perceived benefits of detecting and discarding replays at arbitrary nodes within the infrastructure. Therefore, one of the threats to which the DTN protocol is still vulnerable is the threat of a given bundle being retransmitted repeatedly. Such replayed bundles will not be detected and if a compromised router were to inject a large amount of replayed bundles, or if a malicious third

party were able to record legitimate bundles and inject repeated replays of those bundles into the infrastructure toward their intended next hop, these bundles would be able to overwhelm the DTN architecture, at least until they expire.

4.5.3.1.2 Bundles sent from one DTN router could be resent to another DTN router without detection

Even if measures were incorporated into the DTN Bundle Protocol to enable a DTN router to detect bundles that are replays of bundles it has already received, the DTN infrastructure would still be susceptible to a particular type of replay attack in which a malicious third party could record a bundle that is transmitted from X to Y and inject this on the link between X and Z, and, if public key cryptography were being used or if private key cryptography were being used and several adjacent routers were sharing the same symmetric key, router Z would not be able to detect this injection. A hash value that was signed by node X and received at node Z would correctly decrypt using node X's public key, even if it was originally sent to node Y. A hash value that was encrypted by node X with a symmetric key that nodes X, Y, and Z all share would correctly decrypt at Z using that symmetric key. Node Z would therefore authenticate the bundle and forward it, and it would continue to be forwarded by subsequent nodes until it reached the DTN node at which the route from X to Y and the route from X to Z converge on their way to the given destination. Although this replay attack is using up valuable resources on at least one link of the network, its detrimental effect would be limited by the topology of the network because only redundant links between the point of injection and the point of link convergence before the destination could be affected. Furthermore, like the simpler replay attack discussed previously, these links can only be affected by these replayed bundles for as long as the bundles' timestamp remains valid.

The cost of being vulnerable to this type of replay attack needs to be weighed against the cost of protecting against it. How could it be protected against? Well, the reason this vulnerability exists is that the bundle content that is protected by hop-by-hop integrity does not include any information about the address to which the bundle was forwarded on its most recent hop. The bundle header includes a destination field, but that is the address of the final DTN application, not the next-hop router in the path from source to destination. In order to prevent this type of replay attack in which a bundle forwarded to one router can be duplicated and sent to another router, the address of the destination router must be included in the BAH's encrypted hash value.

One way of including this destination address in the source

information would be to add a "next hop" field to the bundle's header somewhere and include the value of this field in the authentication information signed hash value. Adding additional fields to the Bundle Protocol, however, is undesirable, especially when the fields

have endpoint ID values that could consist of very long strings of bits. It is not clear that the cost of including this extra field would be worth the benefit of guarding against the limited type of replay attack discussed above. Another way of protecting against this type of replay attack would be to calculate the hash value in the BAH over the endpoint ID of the "next hop" router, but do not include the actual endpoint ID as part of the bundle header. In other words, the value of the BAH hash would be calculated over the entire bundle (excluding the payload) plus the endpoint ID of the next hop bundle agent (but this endpoint ID would not be sent as part of the bundle). This method would not add any additional bits to the Bundle Protocol, but it would serve to provide authentication of the next-hop endpoint ID and thereby eliminate the small vulnerability posed by the topology-limited replay attack as described earlier. Unfortunately, such a method would not be arbitrarily applicable in a DTN network, because there is no guarantee that a sending node always knows the identity of the next hop router to which it will be forwarding a bundle. In a situation in which a DTN node does not know which DTN node will be the next-hop receiver of a bundle, it would not be possible for the sending node to calculate the bundles BAH hash over the endpoint ID of the next hop. This is another reason why bundle protocol security has not been designed to be able to counter the type of replay attack in which a bundle that is sent to a given next-hop is replayed as having been sent from the original senderto a different next-hop recipient.

5. DTN Application Protection

The mandatory hop-by-hop security services that are used to provide infrastructure protection within DTNs also work together to provide some aspects of end-to-end security from source to destination, assuming that none of the bundle agents and none of the cryptographic keys of the bundle agents at each hop along the way have been compromised. However, there are legitimate reasons for wanting to be able to use optional end-to-end security services in addition to the mandatory hop-by-hop services, such as the ability to protect DTN applications from bogus or modified application data, to protect

This is really misleading. All that you have on an end-to-end basis is the knowledge that none of the bundle headers was messed with. You have nothing about the payload - and that's all that counts to the end-system.

against potential threats that could be posed by a compromised router, and to be able to provide a finer granularity of access control at certain locations within the network:

1. Protection from bogus or modified application data: The hop-by-hop data integrity service as provided by the BAH hash does not ensure the integrity of the bundle payload. In order to enable a destination bundle agent to detect payload that has been corrupted, modified, injected, or replayed, the encrypted hash in the PSH must be used. The security services offered by the PSH comprise an optional, second layer of defense for those DTN applications wishing to avail themselves of the PSH and ensure integrity of payload data received.
2. Protection from compromised routers: As discussed in [2] , the DTN security approach is known to be partially susceptible to compromised routers. In order to protect DTN applications in the event that a DTN router is compromised, an end-to-end security service provides a mechanism that enables a destination bundle agent to detect illegitimate, compromised, and even replayed bundles that have been injected into the network by a compromised router or other means.
3. Finer granularity of access control: End-to-end signatures may be checked at selected nodes in the DTN that wish to provide stricter security and do not want to have to rely on the access control decisions of all previous routers in the transmission path. For example, a last hop router before a very limited-resource link, such as an uplink to a spacecraft in Mars orbit, may use the original source signature as a basis for verifying the identity of the source and the authority of the source to use the link at the requested rate and class of service before forwarding the bundle on the limited-resource link. Without an end-to-end security mechanism, each DTN node would have no choice but to trust that all DTN nodes preceding it in a bundle transmission path have enforced the access control rules appropriate for its link.

Supporting the protection of DTN application data sent from one application to another from being disclosed to a third party,

enabling a receiving DTN application to detect whether the bundle it receives has been modified while in-transit from the sending application, and enabling a DTN application to detect replayed bundles that it receives are security service that are optionally provided or supported within DTN. These services will be discussed further in the following subsections.

5.1 DTN Application Data Confidentiality Protection

DTN users may use the DTN Bundle Protocol to provide end-to-end data confidentiality for application data, but the actual encryption of the user information to be transmitted as the application data unit must be accomplished by the application before being passed to its underlying bundle agent; similarly, decryption of the application data unit received at the destination DTN user node must be accomplished by the destination application rather than the destination bundle agent. There is no mechanism within the Bundle Protocol itself for a bundle agent to perform encryption or

decryption of the bundle payload or any portion of the bundle header.

When a bundle application wants to transmit an application data unit to a destination communications endpoint in encrypted form, the application data unit must be encrypted at the application layer before being transmitted to the source bundle agent. The source bundle agent itself does to perform encryption for the application data unit. However, it does support the transfer of encrypted application data units by providing the option for the source application to signal to the destination application ~~regarding~~ which encryption algorithm and encryption key has been used to encrypt the application data unit. The ability to identify the algorithm and key enables flexibility in the use of various algorithms and keys and in performing key rollover.

5.2 Bundle Integrity Protection (end-to-end)

The Bundle Protocol provides end-to-end data integrity for the entire bundle (including the payload), meaning that it provides a mechanism whereby the destination bundle agent can verify that the bundle received has not been modified in transit since being sent by the originating source bundle agent. The Payload Security Header is used to provide a cryptographically-based authentication service by having the source bundle agent calculate a hash over the entire bundle (including the payload), encrypting the hash, and placing it in the Payload Security Header. Upon receipt of the bundle, the destination decrypts the received PSH hash. The destination then calculates its own hash of the bundle and compares the hash value that it has calculated with the (now decrypted) PSH hash value received. If the two values are equivalent, then the destination

bundle agent can be assured that the contents of the bundle have not been modified (either intentionally or unintentionally) since being sent from the source. Furthermore, after the destination bundle agent compares the Payload Security Header hash value with the computed value and finds them to be equivalent, it may optionally make sure that the bundle is not a replay by comparing it with the bundles that it has already received. This optional replay detection capability is discussed further in Section 5.4 below.

5.2.1 The end-to-end bundle integrity service is vulnerable to source key compromise

This end-to-end data integrity security service is vulnerable to compromise of the source user key. As long as a source user's key has not been compromised, bundles originating from that source user can be provided with end-to-end data integrity, meaning that the destination can be assured that the contents of the bundle have not been modified while in-transit from the originating source. However, if the source user's key is compromised, then bundles claiming to have originated from that source cannot be provided with end-to-end data integrity and a destination has no assurance that the bundle received is the same as the bundle that was originally sent from the source. In fact, the recipient has no assurance that the bundle received was even sent from that source at all.

5.2.2 The end-to-end bundle integrity service is not vulnerable to router compromise

The encrypted hash value in the Security Payload Header provides data integrity protection for the entire bundle (except for the BAH and mutable fields), including source endpoint ID and timestamp, which determine the bundle's uniqueness; destination endpoint ID, class of service, and payload. Because all of these fields are included in the PSH hash calculation and the hash is encrypted by the sending application, the PSH truly provides an end-to-end data integrity service from source bundle agent to the destination bundle agent. The PSH enables a destination bundle agent to reliably determine whether or not a received bundle had been modified while in transit, even in the case in which one or more of the DTN routers along that path from source to destination may have been compromised.

A compromised router does not have access to a source application's private key, and access to this private key is what is required to undermine the end-to-end integrity service provided by the PSH. Although a compromised router could misroute a bundle, it would not be able to modify any portion of a bundle and forward that bundle without the modification being detectable by the destination. Therefore, the end-to-end bundle integrity service provided by the

Payload Security Header can be effective in the case in which one or more DTN routers may be compromised. In the event that both a DTN router's key has been compromised, and a source key has been compromised, however, the Bundle Protocol cannot be provided with any sort of end-to-end bundle integrity service.

5.3 Source and Destination Authentication (end-to-end)

5.3.1 End-to-end endpoint authentication

End-to-end endpoint authentication is very closely associated with end-to-end data integrity because the same mechanism, the optional use of the signed hash in the Payload Security Header, is used to provide both services. As with the hop-by-hop services, it is not possible for end-to-end data integrity to be provided without also providing end-to-end endpoint authentication. Either both end-to-end data integrity and end-to-end endpoint authentication are provided, or neither is.

Cryptography, in combination with the Payload Security Header, is the mechanism that is used to provide both services. When a destination host bundle agent receives a bundle for local delivery (once the bundle has been reassembled, if necessary), in order to perform end-to-end endpoint authentication, the destination host bundle agent must check the value of the encrypted hash value in the Payload Security Header for correctness. If the value is not correct, the bundle has failed to authenticate. If it is correct, the contents of the bundle have been verified not to have changed since being sent from the source (as already discussed in Section 5.2). Furthermore, the bundle has been verified to have been intended for the DTN destination that received it (because the signed hash value was calculated over the contents of the destination field in the Primary

Bundle Header). Lastly, the bundle has been verified to have originated from the endpoint ID listed in the source field (because the signed hash value was calculated over the contents of the source field in the Primary Bundle Header). Also, because the signed hash decrypted correctly, this means that the correct key must have been used to sign the hash, and given that this key is known only to the source (if it is a private, asymmetric key) or only to the source and destination (if it is a symmetric key), this authenticates the source as having in fact encrypted the PSH hash and as being the location from which the bundle originated. Therefore, assuming that the confidentiality of the source key has not been compromised, and even if a DTN router on the way from source to destination has been compromised, the same process that assures the end-to-end data integrity of the bundle also ensures its end-to-end source and destination endpoint ID authentication. If the source's key has been compromised, however, neither the integrity of the bundle nor the

authentication of any endpoints is assured. In this case, a destination would have no assurance that the bundle received is the same as the bundle that was originally sent, that the bundle received was actually sent from the claimed source, or that any bundle was even sent from the source at all.

5.3.2 Source application versus source bundle agent authentication

As distinguished from the hash value in the BAH, the hash value in the Payload Security Header, which is used to provide end-to-end security, is calculated only once, in the source host. It is calculated at the source host, using the source's key, and it is checked at the receiving host and possibly also at one or more DTN security policy routers while in transit. From a security standpoint, this hash should not be encrypted by the source bundle agent. In order for the bundle agent to encrypt the hash, the bundle agent would have to be privy to the source's private key, which is undesirable. The private keys of sources should not be divulged to bundle agents because making bundle agents privy to source private keys would make it impossible to distinguish between authenticating the source versus authenticating the source bundle agent as the originator of the bundle. Furthermore, each bundle agent may serve many different applications and therefore many different sources, which would further blur the determination as to what entity is being authenticated as the source.

From a practical standpoint, though, the hash value in the PSH is calculated over not only the payload (to which the application has access), but over the entire bundle, including many header fields to which the application does not have access, but to which the bundle agent does have access. So, while having the bundle agent encrypt the PSH hash is undesirable from a security standpoint, the bundle agent is in other ways the most practical location within the source host at which to encrypt the hash in the PSH. The bundle agent is the entity that has access to the many bundle field values needed to calculate the hash and it is also the entity that is responsible for assembling the bundle, including inserting the encrypted hash value into the bundle. Therefore, while the bundle agent should be responsible for calculating the (unencrypted) hash and inserting the

encrypted hash value into the security information field of the PSH once it has been encrypted, care must be taken to partition the source bundle agent from the actual process of using the appropriate confidential key to encrypt the hash.

While bundle agents should not encrypt the hash in the PSH, applications do not necessarily need to be burdened with having to encrypt this hash. Ideally, the hash value in the security information field of the PSH should be encrypted without the

application having to do anything special to make this happen, other than make the appropriate private key available. Private keys must at a minimum be stored in the client side of the application/bundle agent API to keep them separate from the source bundle agent. Perhaps these private keys can be kept in a library or server that is responsible for performing the encryption functionality, thereby keeping the keys separate from the host bundle agent while also freeing the application from direct involvement in the process of encrypting each PSH hash.

At the destination host (and possibly at one or more intermediate nodes that wish to enforce a finer granularity of access control), the encrypted hash in the Payload Security Field will have to be decrypted and checked for accuracy. From a security standpoint, there does not seem to be any advantage to decrypting the hash at the application layer rather than at the bundle layer, of the destination host if public key cryptography has been used to encrypt the hash. In this case, decryption of the hash signature at the destination host is distinct from signing the hash at the source host, because in a public key cryptography scheme, the private key of the source, which is used to sign the hash at the sender, must be kept private, but the public key of the source, which is used to decrypt the signed hash at the destination, need not be kept private. If private key cryptography is used to encrypt the hash, then the same care would need to be taken at the destination or intermediate nodes to ensure that the entity that decrypts the hash is partitioned from the bundle agent at that node.

5.4 Optional Replay Detection at Destination Hosts or Policy Routers

Replayed packets can pose two different types of threats: replayed packets can be injected into a network to overwhelm resources, thus resulting in a denial of service attack; and replayed packets can be injected into the network to fool the destination host into receiving multiple copies of the same information, thus resulting in a "content" attack. While DTN security does not have as a goal protection of the infrastructure against the first type of threat, optional mechanisms have been defined to enable destination hosts and security policy nodes to protect against the second type of threat. For example, suppose that the message in a legitimate bundle were "position the telescope 10 degrees to the right". If this message were to be recorded and replayed by a malicious third party, we have already seen that the Bundle Protocol will not detect or discard these duplicate messages at arbitrary intermediate hops within the network. Rather, the bundles will continue to be forwarded until they either reach their destination or expire. Therefore, it is

possible that the destination application may receive many (two, three, fifty) such messages, depending on the original lifetime of

the bundle and the distance that the bundles had to travel from the point at which they were injected into the network to reach the destination host. Requiring the destination bundle agent to pass these messages up to the destination application without detecting the fact that they are duplicates could be quite harmful if we consider the case in which positioning the telescope 20 or 30 degrees to the right could mean aiming it directly at the sun and thereby destroying its optics. Although there is no requirement for duplicate bundles to be detected at arbitrary nodes within the DTN, there is a need for destination bundle agents to be able to optionally detect and discard duplicate bundles received. The same optional ability should be available at DTN security policy nodes that may want to enforce their own access control policy before forwarding a bundle on a certain link. The optional ability to detect and discard duplicates could be crucial to the ability of the security policy node to protect the link from having its resources wasted by transmitting replayed bundles.

Note that technically, a receiving bundle agent is only able to reliably discard replayed bundles and enforce the at-most-once-delivery semantics if the bundle that is received is protected with a Payload Security Header (PSH) containing an encrypted hash, because, assuming that no routers have been compromised, a PSH with an encrypted hash guarantees that the bundle (including the source, timestamp, and payload fields) has not been modified in transit. In practicality though, if the at-most-once-delivery option has been used, even a bundle that is not protected with a PSH containing an encrypted hash should be checked to determine if it is a replay and, if so, it should be discarded. This is because if a bundle that has not been protected with a PSH is received, it is possible that the bundle payload may have been modified in transit without detection. However, assuming that no routers have been compromised, the source and timestamp fields, which are used to uniquely identify the bundle, are guaranteed to be unmodified from their original values by the concatenation of the DTN's hop-by-hop security. This means that when a bundle that does not contain an encrypted hash is received, if the (source, timestamp) pair value of the bundle has been received in a prior bundle, then either the received bundle's payload is the same as the payload in that prior bundle (in which case it is a replay and should be discarded) or the payload is different from the payload in that prior bundle (indicating that the bundle has been modified in transit, in which case it should be discarded anyway).

A receiving bundle agent has no control over whether a received bundle does or does not include a PSH. Nevertheless, if the at-most-once-delivery option has been requested by the receiving application,the actions that the receiving bundle agent takes are the

same regardless of the presence or absence of a PSH with an encrypted hash: the receiving bundle agent will check the received (source, timestamp) pair value against previously-received pair values and discard the bundle if there is a match.

The security-related Delivery Options parameter of the Data.Indication primitive will alert the receiving application to the fact that the bundle is not protected by an encrypted hash value in the PSH because the authentication Delivery Option will not be set [3]. If an application that has registered for at-most-once-delivery receives a bundle without the authentication Delivery Option set, therefore, the application can be assured that the bundle is not a replay of a previously-received bundle. It has no assurance about the content of the payload of that bundle, however. The payload could have been completely modified; it could have in fact been changed to be a replay of previously-received payload. This payload modification would not be detectable because of the absence of a PSH containing an encrypted hash.

6. Security Policy Routers

As previously explained, every bundle agent that receives bundles from a DTN application has the ability to perform access control on

those bundles by limiting the rate at which particular DTN applications may inject bundles or the COS options that particular DTN applications are allowed to use. Generic DTN routers, however--both the next-hop adjacent DTN router as well as those routers that may be very distant from the source host and possibly on regional networks that have very different characteristics than that source host--are dependent on the source host's bundle agent and its access control policy to determine what traffic the router should forward. There is no provision for the bundle agent at a generic DTN router to make an access control decision to discard or limit the access of an authentic, unexpired bundle that it receives from another bundle agent based on the identity or permissions of the source or on local access control policies.

While the ability of the source host to enforce access control on application traffic is an important security feature of the Bundle Protocol, this mechanism operates solely within the bundle agent of the source host, which may be out of the control of the DTN network provider. For this mechanism to properly protect the DTN, the source host's bundle agent must be configured correctly to enforce the desired access control policy and it cannot have been compromised. Furthermore, if this were the only available access control mechanism, every DTN router--both the next-hop adjacent DTN router as well as those routers that may be very distant from the source host and possibly on regional networks that have very different characteristics than that of the source host--would be dependent on the source host's bundle agent and its access control policy to determine what traffic the router should forward.

Instead, the security architecture defines special DTN security policy routers, which are DTN routers that have the ability to check the value of the signed hash in a bundle's PSH and make an access control decision to limit or discard an authentic, unexpired bundle received from another bundle agent based on the identity and permissions of the source in combination with local access control. Use of the PSH signed hash enables a security policy router to authenticate the source of the bundle and verify the integrity of not only the bundle payload, timestamp, and source and destination addresses, but also of the COS field value, which it can use to verify that the source has the appropriate privileges to use the resources at the requested class of service.

I contend that we really don't want to have lots of different flavors of bundle agents. Any/all BA's should have this optional capability.

Security policy routers enable access control to be enforced at the first possible location that is under control of the DTN service provider (for example, the next-hop router that is adjacent to the source host), and that is therefore more likely to be configured correctly and well-secured. They also enable access control to be enforced at other specific routers that may be the last hop before a link that has very limited bandwidth or that is subject to other challenges that require that it be provided with more protection than upstream parts of the DTN. Such a router may well not want to have to rely on the access control decisions of upstream routers, but may wish instead to apply its own local policy to bundles received depending on the identity and permissions of each bundle's source. This possibility is alluded to in [2] when it discusses the potential

Or do we want to develop the concept of a bundle-aware firewall/RADIUS server?

establishment of user-specific security boundaries within the network so that a decision by a DTN router on a university campus, for example, would not determine whether a bundle originating from that campus is radiated to a spacecraft in Mars orbit. Rather, a separate security boundary could be established at the last-hop router before the spacecraft and that last-hop router could use the original source-specific signed hash and source-specific credential information on which to base access control into the Mars orbit portion of the DTN.

Security policy routers, in addition to being able to decrypt the signed hash in the PSH to authenticate the identity of the source of the bundle and verify that that source has permissions to send the bundle at the specified COS, also have the optional capability to detect and discard replayed bundles if so configured. A security policy router that is configured to detect and discard replayed bundles can provide stricter access control that may be warranted on some limited-resource links, but at the cost of having to maintain state information regarding previous bundles received.

It must be noted that security policy routers, if not deployed appropriately, may inadvertently become bundle sinkholes. Consider the case in which a bundle is fragmented and at least one of those fragments follows a path through the network that requires it to be forwarded through a security policy router. If one fragment of the bundle reaches the security policy router, than all fragments of that bundle must. All fragments of a given bundle must be received and their payload extracted and reassembled into the original larger payload before the signed hash value in the PSH can be verified to be correct. Therefore, the security policy router may have to wait for the receipt of multiple bundles before the end-to-end signature can be authenticated. If all fragments of a given bundle are not received, the security policy router may be left waiting for the outstanding fragments indefinitely, in effect resulting in the security policy router becoming a sink hole for that bundle. All of the fragments that do reach the security policy router will be stranded at that security policy router until they time out and none

of the fragments will be forwarded.

Unless mechanisms are used to either guarantee that all fragments of a given bundle will pass through a given security policy router or to enable multiple security policy routers to work together to calculate the combined hash of various bundle fragments, the deployment of a security policy router becomes very risky. The differenc between the router being able to provide additional important access control security services versus the router acting as a barrier to bundle forwarding becomes solely a matter of network topology, so that security policy routers could only be placed within the DTN toppology at locations in which all fragments of a bundle would be guaranteed to pass through it.

One mechanism that has been defined in the Bundle Protocol that can be used to reduce the risk involved in deploying security policy routers without depending on DTN topology to guarantee that all fragments will pass through the security policy routers is that of a

Source Routing Header. A source may use the Source Routing Header to ensure that a given bundle, and therefore all fragments that may result from that bundle, will pass through one or more specified security policy routers en route to its destination.

It may still be desirable to define a mechanism whereby multiple security policy routers could work together to jointly calculate the combined hash of a bundle, the fragments of which are received at the various security policy routers. Such an ability to collaborate would mean that multiple securityi policy routers could be used to control access of bundles to a particular portion of the DTN without a single security policy router becoming a bottleneck. Definition of such a collaboration mechanism, however, is beyond the scope of this document.

The requirement that an arbitrary DTN router have general access to all keys and credentials to which a general security policy router must have access would constitute a scalability problem for DTNs, and be especially crippling given the DTN's limited resources. However, if access control is to be enforced only by the first-hop router that is adjacent to the source host, meaning that this first-hop router would be a security policy router that would requre access to only the keys needed to authenticate the PSH hash in bundles received from adjacent host sources, then only these first-hop security policy routers would be required to maintain user-specified key and capabilities information. On the other hand, the distribution of public keys and credentials to more distant DTN nodes at special security boundaries could be more problematic, but scenarios in which these boundary nodes could be given access to a limited number of source public keys and credentials can be envisioned. The benefits

of the finer granularity of access control that could be achieved on certain critical links by doing so could be worth sacrificing some scalability.

Another alternative for reducing the scalability problem inherent in distributing numerous public keys and credentials to arbitrary DTN nodes is to use an Identity Based Encryption scheme whereby public keys are a priori known to all nodes and need not be distributed. This scheme, and a key naming method that encodes a sender's credentials and the validity period of key into the key's identifier, are discussed in [6] .

Where is this & can I see it?

IBE is not a miracle cure because of the need to contact a server to obtain private keys - not to mention that the private key is now escrowed at the server.

More on replays? Way too much attention being applied to replays.

7. Excessive Replay Detection

As was noted in section Section 4.5.3 protecting the network from denial of service attacks as executed by replay attacks is not a goal of DTN security because all approaches that have so far been devised to do so seem to require each DTN node to maintain an excessive amount of state. By explicitly not building any mechanisms into the bundle protocol to detect replays, we are placing a burden on our routing protocols that they be inherently loop-free. If a routing loop were to make its way into the DTN either on purpose or inadvertently, the results could be devastating. In a DTN context, bundles may have lifetimes of days or weeks, so one cannot necessarily rely on bundle expiration to protect the DTN from the detrimental effects of routing loops.

In order to counter this threat, a "forwarding count" field of the bundle header has been defined to hold as its value the number of times the bundle has been forwarded. This field enables routers to discard bundles received that have a forwarding count value that exceeds a certain maximum value, thereby providing a mechanism whereby a router can discard bundles that are recirculating excessively (possibly due to routing loops). The router may also alert network managers to the possible presence of a routing loop so that it can be looked into and fixed, if necessary.

8. Secure Multipoint Delivery

There are currently no provisions in the Bundle Protocol for multipoint delivery of bundle payloads. If such provisions were to be built into the Bundle Protocol, however, so that a source could send a single bundle and have its payload received at multiple designated destinations, several areas of the Bundle Protocol would need to be addressed to reconcile their operation with multipoint delivery, including security. Discussing the security ramifications of multipoint delivery in this document, however, would be premature, given that no multipoint delivery mechanism has yet been defined for DTN.

Could discuss some of the ramifications of general Internet multicast wrt security, e.g., key dist to groups, key revocation, re-key, scalability, etc

9. Key Management

This document does not yet address the issue of cryptographic key management and distribution for DTNs. The bandwidth, connectivity, and other limitations that are inherent in DTNs serve to make the distribution of cryptographic keys across the DTN very challenging. One solution that is being investigated for supporting DTNs is that of Identity Based Encryption (IBE), which can be used to obviate the need to distribute public keys because in IBE, any string--even the ID of a private key--can be a valid public key. See [6] for a proposal to use a combination of Hierarchical Identity Based Encryption and signatures in support of DTN security.

9.1 Key Distribution

9.2 Credential Assurance

9.3 Key Validity and Revocation

But you're gotta distribute private keys which could be as much of a burden (unless an entity is born w/ private - but they could also be born w/ publics - IBE may solve a cacheing problem - maybe.)

10. DTN Security Features

10.1 Crypto-algorithm Independence

The security services used in the DTN are algorithm-independent, permitting the use of different sets of algorithms for different purposes, as required. A standard set of default algorithms may be specified to facilitate interoperability, but they are not mandatory. In order to provide this algorithm independence, there are optional mechanisms within the Bundle Protocol to identify which algorithms are to be used. In any given bundle, up to five different cryptographic algorithms may be used:

1. an encryption algorithm for providing application data unit confidentiality,
2. a hash algorithm for computing the hash in the PSH,
3. an encryption algorithm for encrypting the hash in the PSH,
4. a hash algorithm for computing the hash in the BAH, and
5. an encryption algorithm for encrypting the hash in the BAH,

The security protocol for DTN enables the source to specify the algorithms and key IDs to be used for providing application data unit confidentiality and for computing and encrypting the PSH hash, but the choice of algorithms to be used for computing and encrypting the BAH hash is not within the user's control. In fact, these algorithm choices, and whether or not a BAH is even used at all, may change from hop to hop.

The Bundle Protocol is also flexible with regard to the style of encryption (public key versus symmetric key) used. Either public key cryptography may be used to sign the hashes in the PSH or BAH, creating digital signatures, or the hashes may be encrypted with a symmetric key, creating message authentication codes. The type of cryptography used will be implicitly indicated within the Bundle Protocol as part of the optional identification of the cryptographic algorithms being used.

10.2 Taking advantage of native regional network security

DTNs have the ability to make use of the security mechanisms available in the native protocols of the regions they overlay. A DTN that spans a regional network with trusted security services, such as an IPsec-capable network or a U.S. DoD link-encrypted network, has the ability to make use of and rely on the security services provided by these underlying regional networks without having to provide redundant services in the DTN overlay. The mechanism that enables the DTN overlay to take advantage of security services available in the underlying regional networks is the Bundle Authenticity parameter

of the convergence layer Bundle.Indication primitive. Using this bundle authenticity parameter, the convergence layer may, based on the regional subnetwork over which the bundle was received, assert to its bundle agent the integrity and authenticity of the bundle. This mechanism enables the bundle agent to avoid having to use the BAH to perform its own hop-by-hop data integrity and sender authentication checks on received data that has been sent using a trusted security mechanism available in the regional subnetwork.

In order to take advantage of native regional network security mechanisms, a DTN bundle agent must insert a BAH into bundles that it forwards over regional subnetworks that are not trusted, but it need not insert a BAH into bundles that it forwards over regional subnetworks that are trusted. Upon receipt of a bundle, if the underlying convergence layer does not assert the authenticity of that bundle (which would be the case if the bundle was not received over a trusted regional subnetwork), then if the bundle does not include a BAH, the bundle agent must discard the bundle.

Suppose, for example, that a bundle arrives at an intermediate DTN node and the convergence layer of that node, based on how it has been configured to treat bundles arriving on transport connections of the type on which the bundle was received, sends the bundle up to the bundle agent with an assertion that the bundle is authentic. In this case, the bundle does not need to have a BAH and, if it does, the bundle agent does not need to process it. Similarly, before forwarding this bundle, the bundle agent will need to either create a BAH to protect the integrity and authentication of the bundle along its next hop, or schedule the bundle for transmission via a convergence layer adapter that is trusted to provide the necessary level of security.

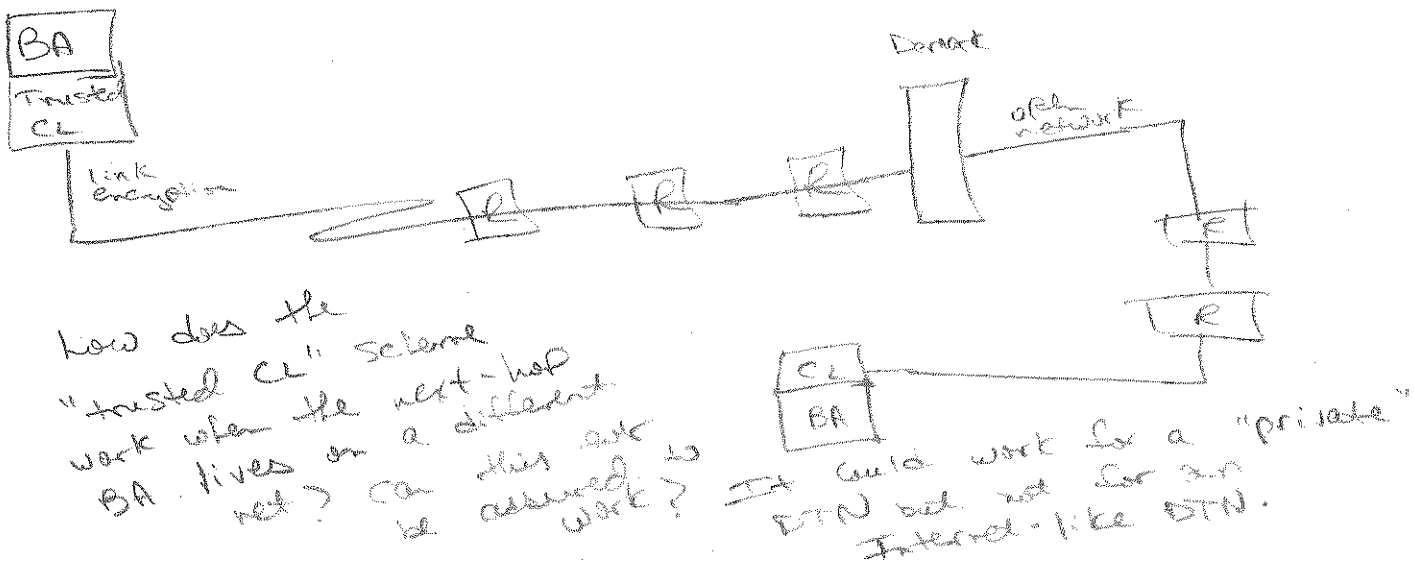
Note that in order to be considered to support the DTN Security Architecture as defined in the Overview and Motivations document and operate according to the DTN BP Security Protocol, a bundle agent must be able to take advantage of available native regional network security by having the ability to accept the convergence layer's assertion of the integrity and authenticity of data received over a secure regional network. In addition, that bundle agent must be able to correctly send, receive, and process bundles that have BAHs. A bundle agent that can only process bundles according to one of these methods, but not the other, is not conformant. In particular, a bundle agent that is only able to process bundles that are received with the convergence layer's assertion of the integrity and authtenticity of the bundle but that is not able to correctly process bundles that are received without such an assertion from the convergence layer is not conformant. Bundle agents must be able to send and receive both types of bundles: both those with and those

without BAHs.

Note the importance of configuration to achieving the desired level of protection in the case in which a BAH will not be used to

authenticate a bundle on a given hop because the subnetwork on that hop already provides sufficient security. In order to achieve the desired level of protection, the convergece layer must be configured correctly so that it does not erroneously assert the authenticity of bundles received on subnetworks that are not secure. Similarly, the bundle agent must be configured correctly so that it does not send bundles without BAHs using convergence layer adapters that should not be trusted. The importance of correct configuration of the bundle agent and of the convergence layers in with respect to what regional subnetworks to trust or not cannot be overstated. It is clearly a potential point of complexity, and therefore of vulnerability, as misconfiguration could result in a total breakdown of security that would not necessarily be detectable.

According to [7], we expect a small number of region types (e.g. Internet-like, ad-hoc mobile, periodic disconnected, etc.) may evolve and each instance of the same type will implement a similar stack of underlying protocols. Each of these region types would have its own set of available security mechanisms and its corresponding convergence layer would need to be configured to use these security mechanisms (or not) when forwarding bundles and to trust (or not) received bundles that have been protected by these security mechanisms. For simplification of configuration, such a limited number of region types would be preferable.



11. Secure DTN Topology and Configuration

11.1 Secure DTN Topology

This section is still not fully formed. It is written with my initial view of how security being optional would work in reality. That is, it is written with the idea that in a secure DTN all nodes must be secure nodes, meaning they all must implement the optional secuirty specs. Lately I have been thinking that this may be too rigid, however. Perhaps we can have the notion of a secure overlay on a DTN, so that two secure DTN nodes could be separated by one or

more insecure DTN nodes. The BAH would originate at one DTN node, be ignorded at the intervening insecure DTN node, and be checked at the following secure DTN node. Can we explore this overlay idea to see if it is something that would be useful? In the meantime, this section is written without this overlay concept...

Security is an optional service in DTN, so not all DTN nodes will incorporate it. In order to provide a DTN service securely, however, a DTN node must support the DTN Security Architecture as defined in this Overview and Motivations document and must operate according to the DTN Security Protocol. Furthermore, that node must interoperate with an adjacent node that also provide secure DTN. A node that implements secure DTN but forwards messages to or receives messages from a node that does not implement secure DTN is not providing a secure DTN service, because no infrastructure protection can be provided on the link between these two nodes.

In order for a given DTN to be considered secure, all of its nodes must support the DTN Security Architecture as defined in this Overview and Motivations document and must operate according to the DTN Security Protocol. A node that does not provide security as required in the DTN security architecture and the DTN Security Protocol cannot be considered to be part of a secure DTN. A network topology that intersperses insecure DTN nodes with secure DTN nodes cannot be considered to be secure, because all nodes must support the BAH in order to provide the infrastructure protection services inherent in DTN security.

In practice, a DTN network may consist of secure DTN nodes interspersed with insecure DTN nodes. Unlike other security architectures for network protocols, however, the secure DTN nodes cannot be conceptually connected to each other to create a secure overlay over the underlying DTN network. This is because infrastructure protection, which is fundamental to DTN security requires every DTN node at every hop in the network to support the BAH. Placing an insecure node adjacent to a secure node in a DTN topology, therefore, effectively limits the extent of the secure DTN,

which ends at the secure node and does not extend across the interface onto the link connecting that secure node with an adjacent insecure node.

If a DTN network consists of a combination of secure and insecure nodes, then the subsets of secure nodes that are interconnected with each other can be understood as secure portions of an otherwise insecure DTN. DTN security can only be provided to the bundles being generated by a DTN application if that application is hosted in a secure DTN node and its bundles travel a path through the DTN that consists only of secure DTN nodes to a destination application that is also hosted on a DTN node. If even a single insecure DTN node must be traversed, then the secure DTN node on which the source resides can be understood to terminate the hop before the insecure node and the secure DTN node on which the destination resides can be understood to terminate at the hop after the insecure node on the path from source to destination. There is no way for the two DTN nodes that are adjacent on either side of the insecure node to

authenticate themselves to each other using the BAH because they don't exchange a BAH.

Even though infrastructure protection cannot be provided over links connecting insecure nodes to secure ones, there is still a need for the secure DTN node at the perimeter of the secure DTN network to be able to scrutinize the bundles that are passed to it from insecure nodes to determine whether those bundles should be admitted to the DTN. A perimeter node wishing to apply such scrutiny to incoming bundles, therefore, should be a security policy router, which is a secure DTN node that has the additional capability to decrypt the PSH hash value in incoming bundles, calculate its own hash value for the incoming bundle, compare these two values and discard the bundle if the values do not match. Placed at the perimeter of the secure DTN, adjacent to insecure nodes, the security policy router is able to apply access control to determine which bundles to admit to the secure DTN based on the identity of and permissions associated with the source of the bundle, as well as on the ability of the bundle to authenticate as not having been modified since being sent from the source.

Topologically speaking, a secure DTN must consist only of interconnected secure DTN nodes. To provide appropriate access control on bundles being forwarded into a secure portion of a DTN from an insecure portion of the DTN, security policy routers should be placed at the perimeter of the DTN. The ability of these security policy routers to discriminate among bundles will depend on the security policy routers

- Being properly equipped with the keys required to decrypt hashes that may be encrypted by any source that may forward data through that security policy router into the secure DTN
- Being appropriately configured as required by DTN policy
  - To reject bundles that have PSH hash values that do not authenticate
  - Perhaps to reject bundles that do not have PSHs (policy-dependent)
  - Perhaps to reject bundles for which the security policy router does not have access to the key required to decrypt the PSH hash

11.2 Configuration Options

How a particular bundle agent should be configured in a given situation is dependent on the goals and requirements of the particular network and the particular bundle agent's role in that network. For example, if a bundle agent forms an integral part of a secure DTN, meaning that all of its neighboring DTN nodes are part of the secure DTN, then that bundle agent must be configured such that whenever the bundle agent forwards a bundle, the bundle must either be protected with a BAH or it must be sent over a trusted convergence layer, or both. It would be incorrect to configure the bundle agent to send the bundle over an untrusted convergence layer without attaching a BAH to the bundle, and doing so would result in a network vulnerability. Similarly, a bundle agent that forms an integral part of a secure DTN must be configured such that whenever it receives a

bundle, the bundle must either be protected with a BAH or the authenticity of the bundle must be asserted by the underlying convergence layer, or both. If the bundle agent were to receive a bundle that did not have a BAH and for which the convergence layer does not assert the authenticity, then the bundle agent must discard the bundle and may generate a bundle status report indicating the authentication failure.

On the other hand, if the bundle agent is not an integral part of a secure DTN, meaning that at least one of its neighboring DTN nodes does not support the DTN Security Architecture and/or does not operate according to the DTN BP Security Protocol, then that bundle agent may be configured differently, in such a way that it can form a secure perimeter to the secure part of the DTN, yet also send and receive bundles from insecure portions of the DTN. In this case, the bundle agent would most likely be a security policy router that would be capable of validating the PSH hash value of incoming bundles, thereby enabling it to impose source-information-based access control on bundles being forwarded from an insecure portion of the DTN to a secure portion of the DTN. This security policy router would have the same role as any other DTN node that is integral to the DTN network when it comes to receiving or forwarding bundles on links with other parts of the secure network. However, this security

policy router may also be configured to treat bundles on links with insecure parts of the network differently, in such a way that the security policy router could provide a perimiter protection capability for the secure DTN,enforcing access control on bundles sent from insecure portions of the DTN before they are admitted into the secure portion of the DTN.

As discussed in Section 10.2, a bundle agent that claims to implement secure DTN must be able to send and receive both bundles that are authenticated using a BAH and bundles for which the convergence layer asserts the authenticity of the bundle. Furthermore, such a bundle agent must insert a BAH into a bundle that it forwards onto a regional subnetwork that is not trusted and it must discard a bundle that is received without a BAH if the underlying convergence layer does not assert the authenticity of that bundle. These configuration requirements, however, apply only to bundle agents that claim to implement security. The requirements of security policy routers, which can be used to control access to a secure portion of a DTN, are a little more flexible, reflecting the unique position that these security policy routers may play as transition points that provide perimiter protection between insecure and secure portions of the DTN.

A security policy router has one or more interfaces into secure portions of the DTN, at which its adjacent DTN node is a bundle agent or security policy router that implements secure DTN; it may also have one or more interfaces into insecure portions of the DTN, at which its adjacent DTN node is a bundle agent that does not implement secure DTN. On all of its interfaces into secure portions of the DTN, the security policy router must conform to the configuration requirements defined in the preceding paragraph for secure bundle agents. On these interfaces, the security policy router must insert a BAH into a bundle that it forwards onto one of these interfaces if

the underlying regional subnetwork is not trusted, and it must discard a bundle that it receives without a BAH on one of these interfaces if the underlying convergence layer does not assert the authenticity of that bundle.

On those interfaces into insecure portions of the DTN, however, the security policy router's configuration requirments are more flexible, reflecting the special perimeter protection role that the security policy router can take on as the locus of the transition between an insecure and a secure portion of the DTN. A security policy router on the perimeter of a secure network may be configured to validate incoming bundles using the PSH and, if the bundles authenticate according to their PSH, forward them into the secure portion of the DTN. How such a perimeter security policy router should treat incoming bundles that do not have PSHs would be a matter of local security policy. The point is that the security policy router may

have the ability and be configured to enforce this policy.

12. Performance Issues

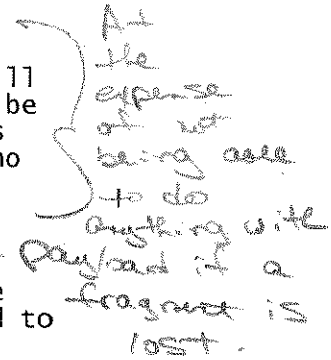
Provision of security within a DTN imposes both bandwidth utilization costs on the DTN links and computational costs on the DTN bundle agents.

12.1 Security-related Bandwidth Utilization Costs

The provision of DTN security imposes bandwidth utilization costs on the links of the DTN. If hop-by-hop security between adjacent bundle agents is provided using the BAH rather than simply asserted by the convergence layer based on the use of an underlying secure link, the presence of the BAH will increase the size of each bundle. The size of the BAH will depend on the hash and encryption algorithms used to compute and encrypt the BAH hash. Furthermore, with some encryption algorithms, the size of the encrypted hash will depend on the size of the key that was used with the encryption. If a bundle becomes fragmented, each fragment of that bundle will contain a BAH, further increasing security-related bandwidth utilization caused by the presence of the BAH.

If the bundle agents are configured to use one or more of the optional BAH fields for identifying the hash algorithm, the encryption algorithm, and/or the key used to compute the BAH hash value, the size of the BAH will be further increased by the size of the optional fields used.

If the optional PSH is used to provide application data security end-to-end from source to destination, the presence of the PSH will further increase the size of each bundle. Only one PSH needs to be present for each original bundle, however, so if a bundle becomes fragmented, the PSH need not be replicated in each fragment and no further security-related increase in bandwidth utilization would result from the fragmentation.



As with the BAH, the size of the PSH will depend on the hash and encryption algorithms, and possibly the size of the key, that are used to compute the encrypted hash in the PSH, if the PSH is used to provide bundle integrity and endpoint authentication.

If the source elects to use one or more of the optional PSH fields for identifying the hash algorithm, the encryption algorithm, and/or the key used to compute the PSH hash value, the size of the PSH will be further increased by the size of the optional fields used.

If the source elects to encrypt the application data unit, there is

not necessarily any increase in bandwidth utilization. However, if the source elects to use one or more of the optional PSH fields for

identifying the encryption algorithm or the key used to encrypt the application data, the size of the PSH will be further increased by the size of the optional fields used.

12.2 Security-related Computational Costs

The provision of DTN security also imposes computational performance costs on the DTN bundle agents that implement the DTN security protocols. If hop-by-hop security between adjacent bundle agents is provided using the BAH rather than simply asserted by the convergence layer based on the use of an underlying secure link, then the sending bundle agent will be required to compute the BAH hash, encrypt the BAH hash, possibly remove the existing BAH from the bundle, and insert the new BAh into the bundle. The receiving bundle agent will be required to decrypt the received BAH hash, compute its own hash on the received bundle, and compare these two values. All of this computation and comparison will increase the computational costs associated with forwarding and receiving each bundle, and because the BAH must be replicated in every fragment, fragmentation will further increase security-related computational costs caused by processing the BAH.

If the optional PSH is used to provide application data security end-to-end from source to destination, the need to create the PSH at the source and process the PSH at the destination and possibly at one or more intermediate policy routers will impose additional computational requirements on the source, destination, and security policy router bundle agents. Only one PSH needs to be present for each original bundle, however, so if a bundle becomes fragmented, the PSH need not be replicated in each fragment and no further security-related increase in computational would result from the fragmentation. If the PSH is used only to provide the optional confidentiality-related header fields, processing of the PSH will be minor, involving only inserting and retrieving algorithm and/or key identifiers. If the PSH is used to provide end-to-end source-to-destination authentication, however, processing would involve computation and encryption of the PSH hash at the source before inserting the PSH into the bundle. At the destination, computation would involve decrypting the received PSH hash, compute the destination's own hash on the received bundle, and comparing these two values. This increased computational cost would only be incurred at the involved bundle agents (source, destination, and possibly intermediate security policy routers), rather than at all intervening bundle agents.

Additionally, at any security policy routers or destination bundle agents that are optionally detecting and discarding duplicate bundles, additional computational performance costs are imposed by

the requirement to store previously-received bundles for comparison with new bundles received, and for the comparison itself.

13. Conformance Requirements

Security is an optional service in DTN; a DTN does not necessarily have to incorporate the security services defined in the DTN Security

Overview and Motivations document and the DTN Security Protocol. In order to provide a DTN service securely, however, a DTN node must support the DTN Security Architecture as articulated in the DTN Security Overview and Motivations document and must operate according to the DTN Security Protocol. All DTN bundle agents that claim to implement secure DTN must comply with all requirements of the DTN Security Overview and Motivations document except for those defined in Section 6 and those requirements listed in Section 11.2 as being specific to security policy routers.

All DTN bundle agents that claim to be security policy routers must comply with all requirements of the DTN Security Overview and Motivations document, including all requirements listed in Section 11.2 as being specific to security policy routers.

All DTN bundle agents that claim to provide optional DTN application protection must comply with all requirements of Section 5.

Question of optional (as stated earlier)
- optional to implement, or
- optional to use (but mandatory to implement).

14. Security Considerations

The subject of this document is security. Therefore, security considerations can be found throughout it.

Another not-fully-formed-idea: If we have the notion of a DTN security overlay over a DTN network, then we can require that all secure nodes implement both the BAH and the PSH, but not necessarily the security policy router operations. If we don't have the notion of a DTN security overlay, however, then some DTN nodes may want to provide only end-to-end security, but not hop-by-hop security. These

nodes would source bundles that have PSHs, but that are not protected by BAHs. The bundles would traverse the DTN network and when they reach a secuirty policy router on the boundary of the secure portion of the DTN network, the security policy router would verify their PSH and admit them based on source identity and permissions. Once in the secure portion of the network, the bundle would be protected hop-by-hop until it either reached its destination or exited the secure portion of the DTN en route to its destination. It's destination node could either be a secure node or a node that, like the source, only implements end-to-end, but not hop-by-hop, security. I think that enabling the concept of a security overlay is probably preferable to having the concept that some DTN nodes implement only the PSH but not the BAH. Thoughts?

15. References

[1] Bradner, S. and J. Reynolds, "Key workds for use in RFCs to Indicate Requirement Levels", RFC 2223, October 1997.

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

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

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

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

[6] Patra, R., "Using Identity based Cryptography for Security in DTNs", , December 2004.

[7] Fall, K., "A Delay-Tolerant Network Architecture for Challenged Internets", , February 2003.

[8] Symington, S., "DTN Security Protocols", draft-irtf-dtnrg-bpsecurity-spec-01.txt , April 2005.

Authors' Addresses

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

Phone: 703 883 7209
Email: susan@mitre.org
URI: http://mitre.org/

All The Rest

Various Organizations
7515 Colshire Drive
McLean, VA 22102
US

Phone: 703 883 7209
Email: susan@mitre.org
URI: http://mitre.org/

Intellectual Property Statement

Full Copyright Statement

Acknowledgment

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