Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 28, 2016

K. Viswanathan
F. Templin
Boeing Research & Technology

August 27, 2015

Architecture for a Delay-and-Disruption Tolerant Public-Key Distribution Network (PKDN)

draft-viswanathan-dtnwg-pkdn-00.txt

Abstract

Delay/Disruption Tolerant Networking (DTN) introduces a network model in which communications can be subject to long delays and/or intermittent connectivity. DTN specifies the use of public-key cryptography to secure the confidentiality and integrity of messages in transit. The use of public-key cryptography posits the need for certification of public keys and revocation of certificates. This document formally defines the DTN key management problem and then provides a high-level design solution for delay and disruption tolerant distribution and revocation of public-key certificates along with relevant design options and recommendations for design choices.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 28, 2016.

Table of Contents

1. Introduction

Key management protocols, for distribution and revocation of public keys on the terrestrial Internet, have required on-demand interactive communications, which has been realized using TCP [RFC0793] connections. The interactions in a public-key management system are between: (a) the sender (or owner of the public key) and the receiver (or user of the public key); and, (b) the receiver and a trusted authority (Certificate Authority or CA). On-demand messaging is not feasible on DTN. Therefore, terrestrial key management protocols may not always function as intended on DTN.

The Online Certificate Status Protocol (OCSP) [RFC6960], for example, requires the receiver of a public key certificate to have on-demand interactions with a Certification Authority (CA) in order to get the current status information for the certificate. Three status responses may be received by the receiver from the CA, namely: good, revoked, and unknown. The receiver needs to accept good certificates and reject revoked certificates. The CA sends a response indicating the unknown state usually when it does not recognize the issuer of

the certificate. In this case, the receiver is expected to interact on-demand with other CAs for determining if the certificate was revoked. When the status in the response is good, since the CA does not remember the receiver's interest in the certificate, the receiver is required to periodically request the status before every use of the certificate.

OCSP is a resource intensive protocol. In order to reduce the round-trip costs for the temporal validation of the certificates, especially in constrained clients (receivers), a provision in TLS Extensions (see Section 8) [RFC6066] has been proposed so that the senders shall send what is called a "stapled Certificate Status" to the receivers. The stapled Certificate Status is a time-stamped certificate-status certificate obtained from a trusted authority by the sender. If the constrained receiver (client) accepts the stapled Certificate Status, then it need not interact with any CA to ascertain the temporal validity of the certificate -- thus reducing communication costs on the receiver side. Although such proposals are useful when dealing with constrained clients (or receivers of certificate), they only transfer the burden of certificate-status queries towards the senders and away from the receivers. Such mechanisms do not obviate the need for on-demand interactions.

The Secure/Multi-purpose Internet Mail Extensions (S/MIME) [RFC5751] allows a sender to encapsulate its certificate as a meta-data (in the message header) for processing an email message. The receiver is expected to consult with a Certificate Revocation List (CRL) or other certificate status verification mechanisms to validate the temporal validity of the certificate. Thus, S/MIME does not obviate the need for on-demand interactions with remote trusted authorities.

As mentioned earlier, on-demand interactions with any party, trusted or otherwise, is not feasible in the network model for DTN. Therefore, existing terrestrial key management protocols are not suitable for DTN. This proposal describes the high-level design choices for a mechanism, which can satisfy the requirements for DTN Key Management [I-D.templin-dtnskmreq], that does not require on-demand interactions with remote parties.

1.1. Related Documents

The following documents provide the necessary context for the high-level design described in this document.

RFC 4838 [RFC4838] describes the architecture for DTN and is titled, "Delay-Tolerant Networking Architecture." That document provides a high-level overview of DTN architecture and the decisions that underpin the DTN architecture.

RFC 5050 [RFC5050] describes the protocol and message formats for DTN and is titled, "Bundle Protocol Specification." That document provides details for the protocol message format for DTN, which is called as Bundle, along with the description of processes for generating, sending, forwarding, and receiving Bundles. It also specifies an encoding format called SDNV (Self-Delimiting Numeric Values) for use in DTN.

RFC 6257 [RFC6257] is titled, "Bundle Security Protocol Specification." It specifies the message formats and processing rules for providing three types of security services to bundles, namely: confidentiality, integrity, and authentication. It does not specify mechanisms for key management. Rather, it assumes that cryptographic keys are somehow in place and then specifies how the keys shall be used to provide the security services. Additionally, it attempts to standardize the cipher suite in DTN.

The Internet Draft [I-D.birrane-dtn-sbsp] for DTN Key Management is titled, "Streamlined Bundle Security Protocol Specification (SBSP)." When compared with RFC 6257, it is silent on concepts such as Security Regions, at-most-once-delivery option, and cipher suite specification. It provides more detailed specification for bundle canonicalization and rules for processing bundles received from other nodes. Like RFC 6257, the draft does not describe any key management mechanisms for DTN but assumes that suitable key management mechanism shall be in place.

The Internet Draft for specifying requirements for DTN Key Management [I-D.templin-dtnskmreq] is titled, "DTN Security Key Management - Requirements and Design." It sketches nine requirements and four design criteria for DTN Key Management system. The last two requirements are the need to support revocation in a delay tolerant manner. It also specifies the requirements for avoiding single points of failure and opportunities for the presence of multiple key management authorities.

1.2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT","SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. Lower case uses of these words are not to be interpreted as carrying RFC2119 significance.

This draft introduces the following terminologies.

Public Key Distribution Network (PKDN)

is an overlay network that can operate on top of DTN. It is a network of trusted authorities that have information about temporal validity (revoked or otherwise) of public keys certificates. The objective of PKDN is the distribution of valid public-key certificates and revocation of invalidated public-key certificates in a secure, delay and disruption-tolerant manner.

PKDN Bundle
encapsulates a public-key certificate and can be transported in a DTN Bundle. PKDN bundle may optionally encapsulate one or more message payloads (or application data) that are authenticated using the public-key in the encapsulated certificate. The source of the PKDN bundle may provide confidentiality to the message payloads using the public-key of the intended receiver of the message payloads. The message payloads may be DTN Bundles.

PKDN Sender
is the source of a PKDN bundle. It generates PKDN bundles by encapsulating its public-key certificate and using the corresponding private key. It may optionally encapsulate authenticated message payloads in the PKDN bundle. It sends the PKDN bundle to a PKDN Router so that the bundle can be forwarded to the PKDN Receiver in designated the bundle.

PKDN Router
receives a PKDN Bundle from a PKDN Sender, validates it, and generates & forwards a Validated PKDN Bundle to the designated destination. Additionally, the PKDN Router records the destination's interest in the public-key certificate encapsulated in the PKDN Bundle so that it can send periodic status updates to the destination.

Validated PKDN Bundle
is generated by an authorized PKDN Router after receiving a PKDN Bundle that satisfies two conditions, namely: (a) it can be authenticated successfully using the encapsulated certificate; and, (b) revocation information for the encapsulated certificate is not available. A Validated PKDN Bundle includes the PKDN Bundle and a PKDN Bundle Validation Block (PBVB) generated by a PKDN Router. PBVB essentially includes the identity of PKDN Router and information for: (a) asserting the temporal validity of the public-key certificate encapsulated in the PKDN Bundle; (b) the time when the assertion was made; and, (c) message-origin authentication for the PKDN Bundle, the assertion of temporal validity, and the PKDN Router time.

PKDN Receiver

is the destination designated in the PKDN bundle and the node that shall consume the Validated PKDN Bundle. Upon validating the PKDN Bundle and verifying the Validated PKDN Bundle, the PKDN Receiver may store the encapsulated public-key certificate locally. Upon accepting the received Validated PKDN Bundle, it may optionally respond with an acknowledgement to the PKDN Sender via the PKDN Router, from which it received the Validated PKDN Bundle. The acknowledgement may include its own encapsulated public-key certificate and message payloads -- this would be the optional return path for the messaging.

Certificate Revocation Manager (CRM)
is an operationally off-line DTN node that shall maintain the System's Certificate Revocation List (CRL) and publish any changes to the CRL as Delta-CRLs. The CRM shall be housed in a physically protected location that is easily accessible for authorized and trusted human operators, who shall inject CRL updates into the CRM. The CRM, in-turn, shall inject the Delta-CRLs to PKDN Routers in the PKDN administered by the human operators. It is important to note that the CRM only propagates revocation information but not certificates. Certificates are propagated by the owners of the certificates, namely PKDN Senders.

2. DTN Key Management

This section shall introduce the problem statement for DTN Key Management problem followed by an enumeration of communication-patterns that can be used for potential solutions and a proposed solution for the problem that is called a Public-Key Distribution Network.

2.1. The DTN-Key-Management Problem Statement

The problem of DTN Key Management can be visualized as shown in Figure 1. The Receiver receives a public key certificate from the Sender. Since the Sender is not trusted to share timely revocation information, the Receiver needs to receive timely revocation information from a Trusted Authority. A basic problem is: (a) how can the Trusted Authority know when the Receiver needs the revocation information for a Public-Key Certificate; and, (b) how can periodic and consistent revocation information be availability in timely and delay-and-disruption tolerant manner? The second question gains importance in DTN because the delay and disruption in the communication link between the Sender and Receiver may not be correlatable with that between the Receiver and the Trusted Authority. This makes the DTN Key Management problem different from terrestrial key management systems, where communication links are assumed to be uniform, interactive, on-demand, and similar.

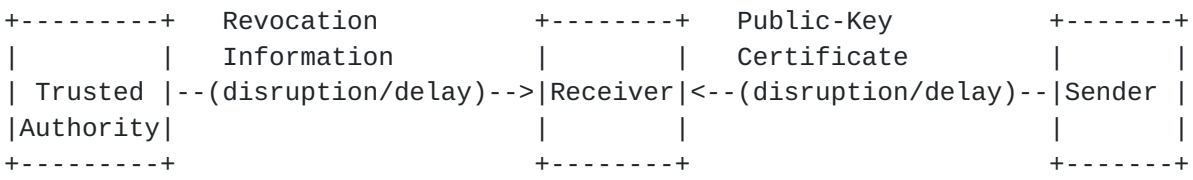

Figure 1: DTN Key Management Problem

An analysis of the above problem using CAP theorem [CAP] suggests that when network partition occurs, due to delay or disruption, the receiver needs to make a local decision in favour of either availability of its service for the received message or consistency of its operations in not accepting revoked certificate, which was used to provide integrity service to the received message. In other words, when the Receiver has received the public key certificate but has not received any revocation information as yet, it needs to vote in favour of either: (a) availability, by accepting the certificate without waiting for revocation information; or, (b) consistency, by waiting for the receipt of revocation information. If it votes in favour of availability, it risks the use of inconsistent information. If it votes in favour of consistency, it risks lack of availability of the public-key for some dependent information processing, which must be paused. Clearly, in the presence of delay and disruption, both consistency and availability cannot be achieved.

DTN Key Management solutions must be partition tolerant and provide trade-off options for their applications between availability and security consistency. Such a trade-off may be realized in an application-agnostic manner by aiming for eventual consistency instead of immediate consistency. Eventual consistency means that all DTN nodes will eventually reject revoked keys but until such an eventuality some DTN nodes are allowed to work with stale revocation information depending on their application security sensitivity. Immediate consistency is not possible in DTN but is possible in the terrestrial Internet. The time available for accepting or rejecting the certificate (and the message) will be decided by the application's security threshold.

2.2. Communication patterns for solving the DTN problem

As mentioned previously, the two-fold problem of DTN Key Management Problem is:(a) how can the Trusted Authority know when the Receiver needs the revocation information for a Public-Key Certificate; and, (b) how can periodic and consistent revocation information be made available in timely and delay-and-disruption tolerant manner?

Five communication patterns can provide solutions to the first question (Question a), namely:

Pattern 1: (Request-response) The Receiver informs the Trusted Authority every time when it needs fresh revocation information for a certificate by sending a request. The Trust Authority responds with a fresh status information for that certificate.

Pattern 2: (Publish-subscribe) The Receiver informs the Trusted Authority about its interest in a certificate only once, which is the first time when it needs the revocation information, by sending a subscription request. The Trusted Authority responds to the subscription request with a fresh status information for that certificate and remembers the subscription request. Whenever there is a change in status information, the Trusted Authority sends the updates to the Receiver without having to receive a request for the same.

Pattern 3: (Blacklist broadcast) The Trusted Authority does not receive any certificate-specific request from any Receiver. It periodically broadcasts Certificate Revocation Lists (CRLs)to all DTN nodes including the Receiver. If the broadcast mechanism were to be replaced with a multicast mechanism, then the Receiver will be expected to register its address with the Trusted Authority exactly once as a registration process. Note that the registration process does not reference any certificate unlike the subscription process in the previous pattern.

Pattern 4: (White-list broadcast) This communication pattern is similar to the previous communication pattern except that the Trusted Authority periodically broadcasts a list of valid certificates instead of broadcasting a list of invalidated certificates. This communication pattern is useful when the number of certified public-keys are less.

Pattern 5: (Publish with proxy subscribe) The Sender routes its certificate through the Trusted Authority to the Receiver, who shall accept certificates only from the Trusted Authority. The Trusted Authority validates the certificate before forwarding it to the Receiver. The Trusted Authority subscribes the Receiver for interest in the Sender's certificate so that periodic updates can be sent in the future for the certificate. Thus, the Sender acts as a proxy for the Receiver and subscribes the Receiver for future updates from the Trusted Authority.

Pattern 1 describes the communication style used by terrestrial key management solutions such as OCSP. The Receiver may receive the certificate from the Sender every time a security session is established as is the case in TLS [RFC5246]. Thus, the Receiver may need to send a request to the Trusted Authority every time a security session is established. Section 1 discussed why this communication style is not suitable for DTN.

Pattern 2 has a similar complexity as Pattern 1 for the first round of communication for a certificate between the Receiver and the Trusted Authority. The communication complexity greatly eases from the second round onwards when the Trusted Authority can send updates to the Receiver without requiring a request. Although this pattern improves the communication complexity from the second round onwards, it does not improve communication complexity of the first round of communications, which is a bottleneck in the DTN settings as described for Pattern 1 in Section 1.

Patterns 3 and 4 require periodical broadcast/multicast of a list data structure (CRL or list of valid public keys). The efficiency of such patterns depend on three factors, namely: the size of the list of revoked certificates, the number of communication recipients, and the frequency of communication. If any one of these factor were to increase, bandwidth utilization will be inefficient because not all recipients of the communication may be interested in all elements of the list that they receive. Thus, most recipients will end up discarding many communications that they receive from the Trusted Authority. When two or more of the factors were to increase simultaneously, the communication system may be overloaded and normal application communications may be affected. Clearly, this solution is not scalable with the increase in number of recipients. Additionally, since Pattern 4 uses white-lists and, in public key management, white-lists grow more frequently than black-lists, the frequency of communications between the Trusted Authority and the Receivers will be higher than in Pattern 3. Also, since the Receivers depend on the Trusted Authority for timely delivery of white-listed keys, the first communication from the Sender to the Receiver must strictly happen after the Trusted Authority has sent the Sender's public key to the Receiver in a white-list communication. Otherwise, the Sender's communication will have to be rejected by the Receiver even though the Sender may be in possession of a registered (or authorized) public key. This calls for increased out-of-band delay-tolerant synchronization between the Sender and the Receiver. For reasons mentioned above, this document shall not pursue Patterns 3 and 4.

Pattern 5 requires every Sender to route their public-key certificates through the Trusted Authority to the Receiver. The

Trusted Authority can be a PKDN Router, which is allowed to filter communications with revoked public-key certificates. Additionally, the PKDN Router remembers the Receiver's interest in order to send periodic revocation updates for the forwarded public-key certificates. The rest of this document shall employ this communication pattern.

3. Architecture for Public Key Distribution Network (PKDN)

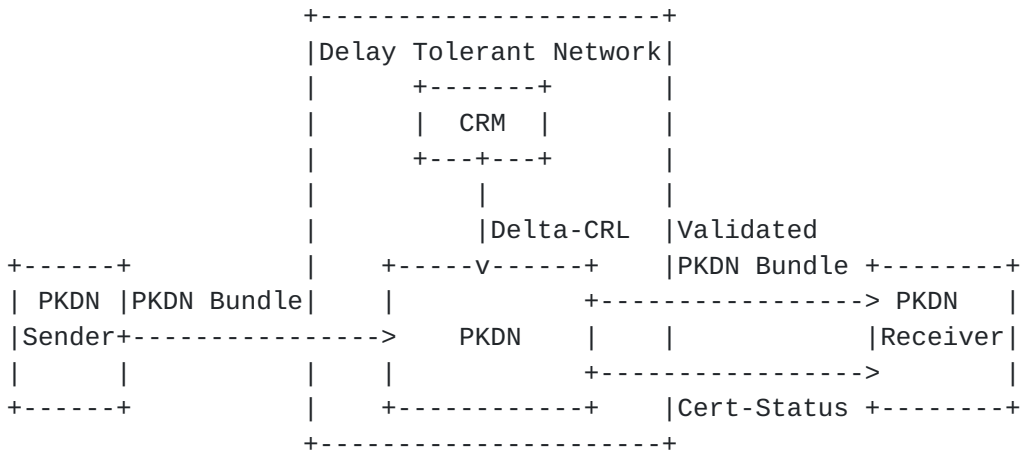

Figure 2: Architecture of Public Key Distribution Network

As mentioned in the previous section, this proposal adopts Communication Pattern 5 for designing Public Key Distribution Network (PKDN). The elements of PKDN are shown in Figure 2.

a. An operationally off-line Certificate Revocation Manager (CRM) periodically injects timestamped updates to the System's Certificate Revocation Lists, called Delta-CRLs, into a few routers in the PKDN, which, in turn, shall propagate the updates to other routers in the PKDN. The information in the CRL needs to reach PKDN Receivers in part or full.

b. PKDN is an overlay network on a Delay Tolerant Network (DTN) that is composed of a logical interconnection of PKDN Routers.

c. PKDN Senders send PKDN Bundles to PKDN (PKDN Routers) so that the PKDN Bundles can be forwarded to PKDN Receivers.

d. PKDN Receivers received Validated PKDN Bundles from PKDN and install the public key certificates in the PKDN Bundles locally. They also received PKDN Status messages from the PKDN

Within this architectural setting, loosely synchronized PKDN Routers perform three basic functions as described below.

1. (Routing) Receive and validate PKDN Bundles from Senders and forward Validated PKDN Bundles to Receivers designated in the PKDN Bundles.

2. (Cache synchronization) Update local CRL cache using authenticated and time-stamped CRL updates from authorized PKDN nodes. Such authenticated updates to certificate revocation lists shall be called delta-CRLs. Forward delta-CRLs to other PKDN Routers.

3. (Revocation updates) PKDN Routers construct and send periodic certificate status updates to PKDN Receivers using the local CRL cache.

The above three basic functions can now be used to enumerate the design choices for PKDN.

3.1. Design Choices for Routing Function

It was discussed that PKDN is an overlay network of PKDN routers. Also, a PKDN Bundle from a PKDN Sender to a PKDN Receiver needs to go through at least one PKDN Router. The following questions lead to the design choices for PKDN.

1. How many PKDN Routers must there be between any given PKDN Sender and PKDN Receiver? The answers can be one, two, or more. The higher the number of PKDN Routers, the higher will be the routing delay. In order to reduce delay, having only one PKDN Router in any given path for a PKDN Bundle is the best design choice.

2. How to determine the designated PKDN Router between a given PKDN Sender and PKDN Receiver? Naming of routers is fundamental to determining designated PKDN Router between two given communication endpoints. Two types of naming options have been considered for DTN [EPDTN], namely: (i) addresses with topological information; and, (ii) identifiers without topological information. The design choices for determining the designated PKDN Router are: (a) the PKDN Router name for a given PKDN Sender is manually configured for every PKDN Sender; (b) the PKDN Sender discovers the name of its nearest PKDN Router using a broadcast-based discovery protocol; (c) the PKDN Sender uses its DTN address to derive the DTN address of its PKDN Router; or, (d) the PKDN Sender uses the PKDN Receiver's DTN address to derive the DTN address of the PKDN Receiver's PKDN Router. When DTN node addresses with topological encoding are available, Options (c) and (d) provide non-interactive PKDN Router determination, which will be well suited for delay-and-disruption tolerance. To see how Options (c) and (d) may be designed, lets assume that the

address of PKDN Sender has the following encoded region and entity information: {region, entity:port} = {earth.sol.int, rover_monitor.nasa.gov:23}. Let the address of the PKDN Receiver be: {mars.sol.int, rover1.rovernet.nasa.gov:23}. The PKDN Router for the PKDN Sender could be derived as: {earth.sol.int, pkdn.nasa.gov:2} and the PKDN Router for the PKDN Receiver could be derived as: {mars.sol.int, pkdn.nasa.gov:2}. Thus, if such a topological encoding were available, network service discovery can simply be address-based host discovery. But, when only DTN identifiers (without topological information) are available, only design choices (a) and (b) are feasible. Furthermore, since Option (a) is non-interactive while Option (b) is not, Option (a) may be better suited when only DTN identifiers are available.

3.2. Design Choices for Cache Synchronization

It was specified that the Certificate Revocation Manager (CRM) needs to publish Delta-CRLs into the PKDN. It was also specified that PKDN is a network of PKDN Routers (please refer to Figure 2). Thus, the CRM needs to publish its Delta-CRLs to one or more PKDN Routers. The problem is that of synchronization of a distributed cache of CRL information, which is a distributed aggregation problem. A survey of decentralized aggregation protocols has been published by Makhloufi et. al. [DAgg]. They identify gossip based, tree based, and hybrid aggregation protocols. Although decentralized aggregation is best suited for decentralized DTN, an additional centralized aggregation choice (as hub-and-spoke propagation) is identified as a choice. Note that the PKDN Senders and Receivers are assumed, without loss of generality, to be agnostic of these design choices. The design choices for such a network propagation of Delta-CRLs are as follows.

1. (Hub-and-spoke propagation) Every authorized PKDN Router is registered with the CRM and the CRM (as the hub) periodically sends updates to all registered PKDN Routers (as the spokes) using DTN. The hub-and-spoke propagation is deterministic and simple but the load on the CRM is high and the same information (Delta-CRL) is carried by multiple DTN Bundles along similar DTN paths. In other words, the hub-and-spoke arrangement is not efficient use of the network but simple and deterministic.

2. (Depender graph propagation) This model of propagation is described by Wright et. al. [FTCR]. The basic idea is to let a few first-level PKDN Routers receive Delta-CRLs from CRM, which shall propagate the same to second-level PKDN Routers. The second-level PKDN Routers shall propagate the same to third-level PKDN Routers and so on and so forth. Thus a hierarchy of PKDN Routers shall be organized as an Rooted, Directed Acyclic Graph (ADAG), with the CRM functioning as the root of the graph. The

number DTN bundles with the same information (Delta-CRL) along the same DTN path can be reduced. Additionally, unlike the hub-and-spoke propagation, k-path-redundancy can be realized in the PKDN by requiring every PKDN Router to receive Delta-CRL updates from k sources (one of the k sources for the first-level PKDN Routers must be the CRM.) The disadvantage of this design choice is its design and implementation complexity when compared with the hub-and-spoke design choice.

3. (Gossip propagation) No security literature has been found, as yet, for propagation of CRL information in a dependable manner using gossip protocols. The security property expected out of such gossip-based CRL propagation protocols is only a theoretical feasibility that each Delta-CRL shall eventually reach all PKDN Routers in the PDKN.

4. (Hybrid propagation) Uses a combination of tree-based and gossip-based propagation. No security literature has been found, as yet, for propagation of CRL information in a dependable manner. The security property for such protocols is the same as that stated for the Gossip propagation.

The current recommendation for PKDN Cache Synchronization protocols is either: (a) to develop depender-graph propagation mechanisms; (b) to design and develop gossip-based propagation mechanisms; or, (c) to design and develop hybrid propagation mechanism. The lead-time for developing depender-graph propagation mechanisms may be least among the recommendations. The hub-and-spoke model of propagation is not recommended as it is a special case and not useful for a highly decentralized application of DTN.

3.3. Design Choices for Revocation Updates

The design choice for revocation updates is centered around the following questions. Which PKDN Router needs to send updates to a specific PKDN Receiver? The answers to this question provides the following choices:

1. (Updates from distributed PKDN Routers) Every PKDN Router that generated a Validated PKDN Bundle designated for the specified PKDN Receiver. in this case two sub-choices exist as follows: either (a) every PKDN Router sends the entire Delta-CRL to the PKDN Receiver; or (b) each PKDN Router sends authenticated updates only for those certificates that were forwarded to the PKDN Receiver by that PKDN Router. Option (a) generates redundant traffic in the DTN as a PKDN Receiver will receive the same information from multiple PKDN Routers. Therefore, it is recommended that Option (a) be avoided. Option (b) conserves

bandwidth while avoiding single points of failures in PKDN revocation update functionality. But, Option (b) implies increased complexity of design when dealing with PKDN-Receiver crash recovery or de-registering a PKDN Receiver's interest in a given certificate.

2. (Updates from a designated PKDN Router) Every PKDN Receiver registers with a designated PKDN Router for receiving updates or Delta-CRLs. This option requires a one-time idempotent registration from a PKDN Receiver with a PKDN Router during bootstrap. A local copy of CRL need not be saved by the PKDN Receiver. Whenever the PKDN receiver receives a Delta-CRL from the network, it only needs to determine which of the PKDN Sender Certificates in its local database have been revoked due to a Delta-CRL. Crash recovery in this option is naturally available because PKDN Receivers need not store CRLs and no state needs to be stored in the PKDN Routers. To avoid single point of failures in receiving revocation updates, a given PKDN Receiver may subscribe to more than one PKDN Router.

Having a designated PKDN Router for each PKDN Receiver results in a stateless system, which will be scalable. Typically, the design choice for designated PKDN Router is valid when the size of Delta-CRLs are small enough for resource-constrained PKDN Receivers, such as Mars Rovers, to handle. The maximum reported size of CRLs [SizeCRL] on the terrestrial Internet is about 27 Mega Bytes. The size of the Delta-CRLs will be much smaller because the CRLs are partitioned into sub-sets using suitably sized windows of time. In a given 24 hour period, the reported [SizeCRLGrowth] maximum number of certificates issued by VeriSign Inc. [verisign], during a given year, is about 200 certificates or 1% of total number of certified public keys for use on the terrestrial Internet. The Delta-CRL for 200 certificates will be a maximum of few tens of Kilo Bytes. Assuming that the statistics of certificate revocation is going to be similar for DTNs, having a designated PKDN Router for each PKDN Receiver will be a good design choice.

4. Summary of Recommended Design Choices

The following are the recommended design choices for each function of PKDN.

1. (Routing) Since the state-of-art of DTN only includes endpoint identifiers instead of addresses, Option (a) is recommended for designating the PKDN Router between a given PDKN Sender and PKDN Receiver. The PKDN Sender shall route all its PKDN Bundles through its PKDN Router. A (certificate-based) chain of trust must be in place so that the PKDN Receiver can authenticate the

origin of Validated PKDN Bundles. The design for the key management structures for establishing the trust relationship between the sender's PKDN Routers and PKDN Receivers shall be described in a follow-up Internet Draft.

2. (Cache synchronization) The use of depender-graph propagation is recommended because the eventual availability of Delta-CRLs at all PKDN Routers has been proved [FTCR]. If a gossip or hybrid propagation were to be available with similar proof, they will be preferred over depender-graph propagation. This is because gossip and hybrid propagation can allow the existence of an unplanned PKDN while depender-graph propagation requires a planned PKDN.

3. (Revocation updates) Assuming small sized Delta-CRLs, which is evinced [SizeCRLGrowth] in the terrestrial Internet, a designated PKDN Router for every PKDN Receiver is recommended. The PKDN Receiver's PKDN Router shall be designated using the same mechanism as the PKDN Sender's PKDN Router was designated -- Option (a) was recommended above for the Routing function.

5. Future work

The feedbacks to this document shall be used to finalize the design of PKDN as a key management protocol suite for DTN in a subsequent Internet Draft. Additionally, the detailed protocol, data structure, and key hierarchy for PKDN shall be described in the subsequent Internet Draft.

6. IANA Considerations

This document potentially contains IANA considerations depending on the design choices adopted for future work. But, in its present form, there are no immediate IANA considerations.

7. Security Considerations

Security issues and considerations are discussed through out this document.

8. References

8.1. Normative References

[I-D.birrane-dtn-sbsp]
Birrane, E., "Streamlined Bundle Security Protocol Specification", draft-birrane-dtn-sbsp-00 (work in progress), December 2014.

[I-D.templin-dtnskmreq]
Templin, F. and S. Burleigh, "DTN Security Key Management - Requirements and Design", draft-templin-dtnskmreq-00 (work in progress), February 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant Networking Architecture", RFC 4838, DOI 10.17487/RFC4838, April 2007, <http://www.rfc-editor.org/info/rfc4838>.

[RFC5050] Scott, K. and S. Burleigh, "Bundle Protocol Specification", RFC 5050, DOI 10.17487/RFC5050, November 2007, <http://www.rfc-editor.org/info/rfc5050>.

[RFC6257] Symington, S., Farrell, S., Weiss, H., and P. Lovell, "Bundle Security Protocol Specification", RFC 6257, DOI 10.17487/RFC6257, May 2011, <http://www.rfc-editor.org/info/rfc6257>.

8.2. Informative References

[CAP] Brewer, E., "CAP twelve years later: How the "rules" have changed", Feb 2012, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6133253>.

[DAgg] Makhloufi, R., Bonnet, G., Doyen, G., and D. Gaiti, "Decentralized Aggregation Protocols in Peer-to-Peer Networks: A Survey", March 2010, <http://dl.acm.org/citation.cfm?id=1692756>.

[EPDTN] Clare, L., Burleigh, S., and K. Scott, "Endpoint naming for space delay / Disruption Tolerant Networking", March 2010, <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5446949>.

[FTCR] Rebecca, N., Lincoln, P., and J. Millen, "Efficient Fault-Tolerant Certificate Revocation", Jun 2000, <http://www.csl.sri.com/papers/dependers/>.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <http://www.rfc-editor.org/info/rfc793>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, DOI 10.17487/RFC5751, January 2010, <http://www.rfc-editor.org/info/rfc5751>.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <http://www.rfc-editor.org/info/rfc6066>.

[RFC6960] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 6960, DOI 10.17487/RFC6960, June 2013, <http://www.rfc-editor.org/info/rfc6960>.

[SizeCRL] Raytheon Websense, "Digging Into Certificate Revocation Lists", July 2013, <http://community.websense.com/blogs/securitylabs/archive/2013/07/11/digging-into-certificate-revocation-lists.aspx>.

[SizeCRLGrowth] Walleck, D., Li, Y., and S. Xu, "Empirical Analysis of Certificate Revocation Lists", 2008, <http://rd.springer.com/chapter/10.1007%2F978-3-540-70567-3_13>.

[verisign] Wikipedia Inc, "Wikipedia entry for Verisign Inc", August 2015, <https://en.wikipedia.org/wiki/Verisign>.

Authors' Addresses

Kapali Viswanathan
Boeing Research & Technology
Unit 501, 5th Floor, Tower D, RMZ Infinity
No 3, Old Madras Rd
Bangalore, KA 560016
IN

Email: kapaleeswaran.viswanathan@boeing.com

Fred L. Templin
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
USA

Email: fltemplin@acm.org