Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 9, 2020

P. Hallam-Baker
July 8, 2019

Mathematical Mesh Part VII: Security Considerations
draft-hallambaker-mesh-security-01

Abstract

The Mathematical Mesh 'The Mesh' is an end-to-end secure infrastructure that facilitates the exchange of configuration and credential data between multiple user devices. The core protocols of the Mesh are described with examples of common use cases and reference data.

This document is also available online at http://mathmesh.com/Documents/draft-hallambaker-mesh-security.html [1] .

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 https://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 January 9, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

This section presents the related specifications and standard, the terms that are used as terms of art within the documents and the terms used as requirements language.

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] .

The terms of art used in this document are described in the Mesh Architecture Guide [draft-hallambaker-mesh-architecture] .

The architecture of the Mathematical Mesh is described in the Mesh Architecture Guide [draft-hallambaker-mesh-architecture] . The Mesh

documentation set and related specifications are described in this document.

The implementation status of the reference code base is described in the companion document [draft-hallambaker-mesh-developer] .

The following classes are used as common elements in Mesh profile specifications.

2.5.2. Structure: PublicKey

The PublicKey class is used to describe public key pairs and trust assertions associated with a public key.

UDF: String (Optional) UDF fingerprint of the public key parameters/

X509Certificate: Binary (Optional) List of X.509 Certificates

X509Chain: Binary [0..Many] X.509 Certificate chain.

X509CSR: Binary (Optional) X.509 Certificate Signing Request.

2.5.3. Structure: KeyComposite

Service: String (Optional) Service holding the additional contribution

2.5.4. Structure: KeyOverlay

UDF: String (Optional) Fingerprint of the resulting composite key (to allow verification)

BaseUDF: String (Optional) Fingerprint specifying the base key

2.5.5. Structure: EscrowedKeySet

A set of escrowed keys.

[No fields]

2.5.6. Structure: DeviceRecryptionKey

UDF: String (Optional) The fingerprint of the encryption key

RecryptionKey: PublicKey (Optional) The recryption key

EnvelopedRecryptionKeyDevice: DareEnvelope (Optional) The decryption key encrypted under the user's device key.

2.6. Assertion classes

Classes that are derived from an assertion.

2.6.1. Structure: Assertion

Parent class from which all assertion classes are derived

Names: String [0..Many] Fingerprints of index terms for profile retrieval. The use of the fingerprint of the name rather than the name itself is a precaution against enumeration attacks and other forms of abuse.

Updated: DateTime (Optional) The time instant the profile was last modified.

NotaryToken: String (Optional) A Uniform Notary Token providing evidence that a signature was performed after the notary token was created.

2.6.2. Structure: Condition

Parent class from which all condition classes are derived.

[No fields]

2.6.3. Profile Classes

Profiles are self signed assertions.

2.6.4. Structure: Profile

Inherits: Assertion

Parent class from which all profile classes are derived

KeySignature: PublicKey (Optional) The permanent signature key used to sign the profile itself. The UDF of the key is used as the permanent object identifier of the profile. Thus, by definition,

the KeySignature value of a Profile does not change under any

circumstance. The only case in which a

OnlineSignatureKeys: PublicKey [0..Many] A Personal profile contains at least one OSK which is used to sign device administration application profiles.

2.6.5. Structure: ProfileMaster

Inherits: Profile

Describes the long term parameters associated with a personal profile.

MasterEscrowKeys: PublicKey [0..Many] A Personal Profile MAY contain one or more PMEK keys to enable escrow of private keys used for stored data.

KeyEncryption: PublicKey (Optional) Key used to pass encrypted data to the device such as a DeviceUseEntry

2.6.6. Structure: ProfileDevice

Inherits: Profile

Describes a mesh device.

Description: String (Optional) Description of the device

KeyEncryption: PublicKey (Optional) Key used to pass encrypted data to the device such as a DeviceUseEntry

KeyAuthentication: PublicKey (Optional) Key used to authenticate requests made by the device.

2.6.7. Structure: ProfileService

Inherits: Profile

Profile of a Mesh Service

AuthenticationKey: PublicKey (Optional) Key used to authenticate service connections.

2.6.8. Structure: ProfileAccount

Inherits: Profile

Account assertion. This is signed by the service hosting the account.

ServiceIDs: String [0..Many] Service address(es).

MeshProfileUDF: String (Optional) Master profile of the account being registered.

AccountEncryptionKey: PublicKey (Optional) Key used to encrypt data under this profile

2.6.9. Structure: ProfileGroup

Inherits: Profile

Describes a group. Note that while a group is created by one person who becomes its first administrator, control of the group may pass to other administrators over time.

[No fields]

2.6.10. Structure: ProfileHost

Inherits: Profile

Inherits: Profile

KeyAuthentication: PublicKey (Optional) Key used to authenticate service connections.

2.6.11. Connection Classes

2.6.12. Structure: Connection

Inherits: Assertion

Inherits: Assertion

SubjectUDF: String (Optional) UDF of the connection target.

AuthorityUDF: String (Optional) UDF of the connection source.

2.6.13. Structure: Permission

Name: String (Optional)

Name: String (Optional)

Role: String (Optional)

Role: String (Optional)

Capabilities: DareEnvelope (Optional) Keys or key contributions enabling the operation to be performed

2.6.14. Structure: ConnectionDevice

Inherits: Connection

Inherits: Connection

Permissions: Permission [0..Many] List of the permissions that the device has been granted.

KeySignature: PublicKey (Optional) The signature key for use of the device under the profile

KeyEncryption: PublicKey (Optional) The encryption key for use of the device under the profile

KeyAuthentication: PublicKey (Optional) The authentication key for use of the device under the profile

2.6.15. Structure: ConnectionAccount

Inherits: Connection

Inherits: Connection

Permissions: Permission [0..Many] List of the permissions that the device has been granted.

KeySignature: PublicKey (Optional) The signature key for use of the device under the profile

KeyEncryption: PublicKey (Optional) The encryption key for use of the device under the profile

KeyAuthentication: PublicKey (Optional) The authentication key for use of the device under the profile

2.6.16. Structure: ConnectionService

Inherits: Connection

[No fields]

2.6.17. Structure: ConnectionHost

Inherits: Connection

[No fields]

2.6.18. Structure: ConnectionApplication

Inherits: Connection

[No fields]

2.6.19. Activation Classes

2.6.20. Structure: Activation

Inherits: Assertion

Contains the private activation information for a Mesh application running on a specific device

[No fields]

2.6.21. Structure: ActivationDevice

Inherits: Assertion

Inherits: Assertion

EnvelopedAssertionDeviceConnection: DareEnvelope (Optional) The signed AssertionDeviceConnection.

KeySignature: KeyOverlay (Optional) The key overlay used to generate the account signature key from the device signature key

KeyEncryption: KeyOverlay (Optional) The key overlay used to generate the account encryption key from the device encryption key

KeyAuthentication: KeyOverlay (Optional) The key overlay used to generate the account authentication key from the device authentication key

2.6.22. Structure: ActivationAccount

Inherits: Activation

Inherits: Activation

AccountUDF: String (Optional) The UDF of the account

EnvelopedAssertionAccountConnection: DareEnvelope (Optional) The account connection assertion

KeyEncryption: KeyComposite (Optional) The key contribution for the decryption key for the device. NB this is NOT an overlay on the device signature key, it is an overlay on the corresponding recryption key.

KeyAuthentication: KeyOverlay (Optional) The key overlay used to generate the account authentication key from the device authentication key

KeySignature: KeyOverlay (Optional) The key overlay used to generate the account signature key from the device signature key

2.7. Cataloged items

2.7.1. Data Structures

Classes describing data used in cataloged data.

2.7.2. Structure: Contact

Inherits: Assertion

Inherits: Assertion

Identifier: String (Optional)

Identifier: String (Optional)

FullName: String (Optional)

FullName: String (Optional)

Title: String (Optional)

Title: String (Optional)

First: String (Optional)

First: String (Optional)

Middle: String (Optional)

Middle: String (Optional)

Last: String (Optional)

Last: String (Optional)

Suffix: String (Optional)

Suffix: String (Optional)

Labels: String [0..Many]

Labels: String [0..Many]

AssertionAccounts: ProfileAccount [0..Many]

AssertionAccounts: ProfileAccount [0..Many]

Addresses: Address [0..Many]

Addresses: Address [0..Many]

Locations: Location [0..Many]

Locations: Location [0..Many]

Roles: Role [0..Many]

2.7.3. Structure: Role

CompanyName: String (Optional)

CompanyName: String (Optional)

Addresses: Address [0..Many]

Addresses: Address [0..Many]

Locations: Location [0..Many]

2.7.4. Structure: Address

URI: String (Optional)

URI: String (Optional)

Labels: String [0..Many]

2.7.5. Structure: Location

Appartment: String (Optional)

Appartment: String (Optional)

Street: String (Optional)

Street: String (Optional)

District: String (Optional)

District: String (Optional)

Locality: String (Optional)

Locality: String (Optional)

County: String (Optional)

County: String (Optional)

Postcode: String (Optional)

Postcode: String (Optional)

Country: String (Optional)

2.7.6. Structure: Reference

MessageID: String (Optional) The received message to which this is a response

ResponseID: String (Optional) Message that was generated in response to the original (optional).

Relationship: String (Optional) The relationship type. This can be Read, Unread, Accept, Reject.

2.7.7. Structure: Task

Key: String (Optional) Unique key.

Start: DateTime (Optional)

Start: DateTime (Optional)

Finish: DateTime (Optional)

Finish: DateTime (Optional)

StartTravel: String (Optional)

StartTravel: String (Optional)

FinishTravel: String (Optional)

FinishTravel: String (Optional)

TimeZone: String (Optional)

TimeZone: String (Optional)

Title: String (Optional)

Title: String (Optional)

Description: String (Optional)

Description: String (Optional)

Location: String (Optional)

Location: String (Optional)

Trigger: String [0..Many]

Trigger: String [0..Many]

Conference: String [0..Many]

Conference: String [0..Many]

Repeat: String (Optional)

Repeat: String (Optional)

Busy: Boolean (Optional)

2.8. Catalog Entries

2.8.1. Structure: CatalogedEntry

Base class for cataloged Mesh data.

[No fields]

2.8.2. Structure: CatalogedDevice

Inherits: CatalogedEntry

Public device entry, indexed under the device ID

AccountIDs: String [0..Many] The accounts to which this device is bound.

UDF: String (Optional) UDF of the signature key of the device in the Mesh

DeviceUDF: String (Optional) UDF of the signature key of the device

EnvelopedProfileDevice: DareEnvelope (Optional) The device profile

EnvelopedDeviceConnection: DareEnvelope (Optional) The public assertion demonstrating connection of the Device to the Mesh

EnvelopedDevicePrivate: DareEnvelope (Optional) The device profile

2.8.3. Structure: CatalogedCredential

Inherits: CatalogedEntry

Inherits: CatalogedEntry

Protocol: String (Optional)

Protocol: String (Optional)

Service: String (Optional)

Service: String (Optional)

Username: String (Optional)

Username: String (Optional)

Password: String (Optional)

2.8.4. Structure: CatalogedNetwork

Inherits: CatalogedEntry

Inherits: CatalogedEntry

Protocol: String (Optional)

Protocol: String (Optional)

Service: String (Optional)

Service: String (Optional)

Username: String (Optional)

Username: String (Optional)

Password: String (Optional)

2.8.5. Structure: CatalogedContact

Inherits: CatalogedEntry

Inherits: CatalogedEntry

Self: Boolean (Optional) If true, this catalog entry is for the user who created the catalog. To be valid, such an entry MUST be signed by an administration key for the Mesh profile containing the account to which the catalog belongs.

Key: String (Optional) Unique key.

Permissions: Permission [0..Many] List of the permissions that the contact has been granted.

EnvelopedContact: DareEnvelope (Optional) The (signed) contact data.

2.8.6. Structure: CatalogedContactRecryption

Inherits: CatalogedContact

[No fields]

2.8.7. Structure: CatalogedBookmark

Inherits: CatalogedEntry

Inherits: CatalogedEntry

Uri: String (Optional)

Uri: String (Optional)

Title: String (Optional)

Title: String (Optional)

Path: String (Optional)

2.8.8. Structure: CatalogedTask

Inherits: CatalogedEntry

Inherits: CatalogedEntry

EnvelopedTask: DareEnvelope (Optional)

EnvelopedTask: DareEnvelope (Optional)

Key: String (Optional) Unique key.

2.8.9. Structure: CatalogedApplication

Inherits: CatalogedEntry

Inherits: CatalogedEntry

Key: String (Optional)

2.8.10. Structure: CatalogedApplicationAccount

Wrapper for a signed AccountAssertion

Inherits: CatalogedApplication

Inherits: CatalogedApplication

EnvelopedAccountAssertion: DareEnvelope (Optional) The account assertion

2.8.11. Structure: CatalogedMember

UDF: String (Optional)

UDF: String (Optional)

Inherits: CatalogedEntry

2.8.12. Structure: CatalogedGroup

Inherits: CatalogedApplication

[No fields]

2.8.13. Structure: CatalogedApplicationSSH

Inherits: CatalogedApplication

[No fields]

2.8.14. Structure: CatalogedApplicationMail

Inherits: CatalogedApplication

[No fields]

2.8.15. Structure: CatalogedApplicationNetwork

Inherits: CatalogedApplication

[No fields]

2.9. Messages

2.9.1. Structure: Message

MessageID: String (Optional)

MessageID: String (Optional)

Sender: String (Optional)

Sender: String (Optional)

Recipient: String (Optional)

Recipient: String (Optional)

References: Reference [0..Many]

2.9.2. Structure: MessageComplete

Inherits: Message

[No fields]

2.9.3. Structure: MessagePIN

Account: String (Optional)

Account: String (Optional)

Inherits: Message

Inherits: Message

Expires: DateTime (Optional)

Expires: DateTime (Optional)

PIN: String (Optional)

2.9.4. Structure: RequestConnection

Connection request message. This message contains the information

Inherits: Message

Inherits: Message

ServiceID: String (Optional)

ServiceID: String (Optional)

EnvelopedProfileDevice: DareEnvelope (Optional) Device profile of the device making the request.

ClientNonce: Binary (Optional)

ClientNonce: Binary (Optional)

PinUDF: String (Optional) Fingerprint of the PIN value used to authenticate the request.

2.9.5. Structure: AcknowledgeConnection

Connection request message generated by a service on receipt of a valid MessageConnectionRequestClient

Inherits: Message

Inherits: Message

EnvelopedMessageConnectionRequest: DareEnvelope (Optional) The client connection request.

ServerNonce: Binary (Optional)

ServerNonce: Binary (Optional)

Witness: String (Optional)

2.9.6. Structure: RequestContact

Inherits: Message

Inherits: Message

Reply: Boolean (Optional)

Reply: Boolean (Optional)

Self: DareEnvelope (Optional) The contact data.

2.9.7. Structure: RequestConfirmation

Inherits: Message

Inherits: Message

Text: String (Optional)

2.9.8. Structure: ResponseConfirmation

Inherits: Message

Inherits: Message

ResponseID: String (Optional)

ResponseID: String (Optional)

Accept: Boolean (Optional)

2.9.9. Structure: RequestTask

Inherits: Message

[No fields]

3. Mesh Portal Service Reference

HTTP Well Known Service Prefix: /.well-known/mmm

Every Mesh Portal Service transaction consists of exactly one request followed by exactly one response. Mesh Service transactions MAY cause modification of the data stored in the Mesh Portal or the Mesh itself but do not cause changes to the connection state. The protocol itself is thus idempotent. There is no set sequence in which operations are required to be performed. It is not necessary to perform a Hello transaction prior to a ValidateAccount, Publish or any other transaction.

3.1. Request Messages

A Mesh Portal Service request consists of a payload object that inherits from the MeshRequest class. When using the HTTP binding, the request MUST specify the portal DNS address in the HTTP Host field.

3.1.1. Message: MeshRequest

Base class for all request messages.

Portal: String (Optional) Name of the Mesh Portal Service to which the request is directed.

3.2. Response Messages

A Mesh Portal Service response consists of a payload object that inherits from the MeshResponse class. When using the HTTP binding, the response SHOULD report the Status response code in the HTTP response message. However the response code returned in the payload object MUST always be considered authoritative.

3.2.1. Message: MeshResponse

Base class for all response messages. Contains only the status code and status description fields.

[No fields]

3.3. Imported Objects

The Mesh Service protocol makes use of JSON objects defined in the JOSE Signatgure and Encryption specifications.

3.4. Common Structures

The following common structures are used in the protocol messages:

3.4.1. Structure: KeyValue

Describes a Key/Value structure used to make queries for records matching one or more selection criteria.

Key: String (Optional) The data retrieval key.

Value: String (Optional) The data value to match.

3.4.2. Structure: SearchConstraints

Specifies constraints to be applied to a search result. These allow a client to limit the number of records returned, the quantity of data returned, the earliest and latest data returned, etc.

NotBefore: DateTime (Optional) Only data published on or after the specified time instant is requested.

Before: DateTime (Optional) Only data published before the specified time instant is requested. This excludes data published at the specified time instant.

MaxEntries: Integer (Optional) Maximum number of data entries to return.

MaxBytes: Integer (Optional) Maximum number of data bytes to return.

PageKey: String (Optional) Specifies a page key returned in a previous search operation in which the number of responses exceeded the specified bounds.

When a page key is specified, all the other search parameters except for MaxEntries and MaxBytes are ignored and the service returns the next set of data responding to the earlier query.

3.5. Transaction: Hello

Request: HelloRequest

Request: HelloRequest

Response: HelloResponse

Report service and version information.

The Hello transaction provides a means of determining which protocol versions, message encodings and transport protocols are supported by the service.

3.6. Transaction: ValidateAccount

Request: ValidateRequest

Request: ValidateRequest

Response: ValidateResponse

Request validation of a proposed name for a new account.

For validation of a user's account name during profile creation.

3.6.1. Message: ValidateRequest

Inherits: MeshRequest

Describes the proposed account properties. Currently, these are limited to the account name but could be extended in future versions of the protocol.

Account: String (Optional) Account name requested

Reserve: Boolean (Optional) If true, request a reservation for the specified account name. Note that the service is not obliged to honor reservation requests.

Language: String [0..Many] List of ISO language codes in order of preference. For creating explanatory text.

3.6.2. Message: ValidateResponse

Inherits: MeshResponse

States whether the proposed account properties are acceptable and (optional) returns an indication of what properties are valid.

Note that receiving a 'Valid' responseto a Validate Request does not guarantee creation of the account. In addition to the possibility that the account namecould be requested by another user between the Validate and Create transactions, a portal service MAY perform more stringent validation criteria when an account is actually being created. For example, checking with the authoritative list of current accounts rather than a cached copy.

Valid: Boolean (Optional) If true, the specified account identifier is acceptable. If false, the account identifier is rejected.

Minimum: Integer (Optional) Specifies the minimum length of an account name.

Maximum: Integer (Optional) Specifies the maximum length of an account name.

InvalidCharacters: String (Optional) A list of characters that the service does not accept in account names. The list of characters MAY not be exhaustive but SHOULD include any illegal characters in the proposed account name.

Reason: String (Optional) Text explaining the reason an account name was rejected.

3.7. Transaction: CreateAccount

Request: CreateRequest

Request: CreateRequest

Response: CreateResponse

Request creation of a new portal account.

Unlike a profile, a mesh account is specific to a particular Mesh portal. A mesh account must be created and accepted before a profile can be published.

3.7.1. Message: CreateRequest

Request creation of a new portal account. The request specifies the requested account identifier and the Mesh profile to be associated with the account.

Inherits: MeshRequest

Inherits: MeshRequest

Account: String (Optional) Account identifier requested.

3.7.2. Message: CreateResponse

Inherits: MeshResponse

Reports the success or failure of a Create transaction.

[No fields]

3.8. Transaction: DeleteAccount

Request: DeleteRequest

Request: DeleteRequest

Response: DeleteResponse

Request deletion of a portal account.

Deletes a portal account but not the underlying profile. Once registered, profiles are permanent.

3.8.1. Message: DeleteRequest

Request deletion of a new portal account. The request specifies the requested account identifier.

Inherits: MeshRequest

Inherits: MeshRequest

Account: String (Optional) Account identifier to be deleted.

3.8.2. Message: DeleteResponse

Inherits: MeshResponse

Reports the success or failure of a Delete transaction.

[No fields]

3.9. Transaction: Get

Request: GetRequest

Request: GetRequest

Response: GetResponse

Search for data in the mesh that matches a set of properties described by a sequence of key/value pairs.

3.9.1. Message: GetRequest

Describes the Portal or Mesh data to be retreived.

Inherits: MeshRequest

Inherits: MeshRequest

Identifier: String (Optional) Lookup by profile ID

Account: String (Optional) Lookup by Account ID

KeyValues: KeyValue [0..Many] List of KeyValue pairs specifying the conditions to be met

SearchConstraints: SearchConstraints (Optional) Constrain the search to a specific time interval and/or limit the number and/or total size of data records returned.

Multiple: Boolean (Optional) If true return multiple responses if available

Full: Boolean (Optional) If true, the client requests that the full Mesh data record be returned containing both the Mesh entry itself and the Mesh metadata that allows the date and time of the publication of the Mesh entry to be verified.

3.9.2. Message: GetResponse

Reports the success or failure of a Get transaction. If a Mesh entry matching the specified profile is found, containsthe list of entries matching the request.

Inherits: MeshResponse

Inherits: MeshResponse

DataItems: DataItem [0..Many] List of mesh data records matching the request.

PageKey: String (Optional) If non-null, indicates that the number and/or size of the data records returned exceeds either the SearchConstraints specified in the request or internal server limits.

3.10. Transaction: Publish

Request: PublishRequest

Request: PublishRequest

Response: PublishResponse

Publish a profile or key escrow entry to the mesh.

3.10.1. Message: PublishRequest

Requests publication of the specified Mesh entry.

Inherits: MeshRequest

[No fields]

3.10.2. Message: PublishResponse

Reports the success or failure of a Publish transaction.

Inherits: MeshResponse

[No fields]

3.11. Transaction: Status

Request: StatusRequest

Request: StatusRequest

Response: StatusResponse

Request the current status of the mesh as seen by the portal to which it is directed.

The response to the status request contains the last signed checkpoint and proof chains for each of the peer portals that have been checkpointed.

[Not currently implemented]

3.11.1. Message: StatusRequest

Inherits: MeshRequest

Initiates a status transaction.

[No fields]

3.11.2. Message: StatusResponse

Reports the success or failure of a Status transaction.

Inherits: MeshResponse

Inherits: MeshResponse

LastWriteTime: DateTime (Optional) Time that the last write update was made to the Mesh

LastCheckpointTime: DateTime (Optional) Time that the last Mesh checkpoint was calculated.

NextCheckpointTime: DateTime (Optional) Time at which the next Mesh checkpoint should be calculated.

CheckpointValue: String (Optional) Last checkpoint value.

3.12. Transaction: ConnectStart

Request: ConnectStartRequest

Request: ConnectStartRequest

Response: ConnectStartResponse

Request connection of a new device to a mesh profile

3.12.1. Message: ConnectStartRequest

Inherits: MeshRequest

Initial device connection request.

SignedRequest: SignedConnectionRequest (Optional) Device connection request signed by thesignature key of the device requesting connection.

AccountID: String (Optional) Account identifier of account to which the device is requesting connection.

3.12.2. Message: ConnectStartResponse

Reports the success or failure of a ConnectStart transaction.

Inherits: MeshRequest

[No fields]

3.13. Transaction: ConnectStatus

Request: ConnectStatusRequest

Request: ConnectStatusRequest

Response: ConnectStatusResponse

Request status of pending connection request of a new device to a mesh profile

3.13.1. Message: ConnectStatusRequest

Inherits: MeshRequest

Request status information for a pending request posted previously.

AccountID: String (Optional) Account identifier for which pending connection information is requested.

DeviceID: String (Optional) Device identifier of device requesting status information.

3.13.2. Message: ConnectStatusResponse

Reports the success or failure of a ConnectStatus transaction.

Inherits: MeshRequest

Inherits: MeshRequest

Result: SignedConnectionResult (Optional) The signed ConnectionResult object.

3.14. Transaction: ConnectPending

Request: ConnectPendingRequest

Request: ConnectPendingRequest

Response: ConnectPendingResponse

Request a list of pending requests for an administration profile.

3.14.1. Message: ConnectPendingRequest

Inherits: MeshRequest

Specify the criteria for pending requests.

AccountID: String (Optional) The account identifier of the account for which pending connection requests are requested.

SearchConstraints: SearchConstraints (Optional) Constrain the search to a specific time interval and/or limit the number and/or total size of data records returned.

3.14.2. Message: ConnectPendingResponse

Reports the success or failure of a ConnectPending transaction.

Inherits: MeshRequest

Inherits: MeshRequest

Pending: SignedConnectionRequest [0..Many] A list of pending requests satisfying the criteria set out in the request.

PageKey: String (Optional) If non-null, indicates that the number and/or size of the data records returned exceeds either the SearchConstraints specified in the request or internal server limits.

3.15. Transaction: ConnectComplete

Request: ConnectCompleteRequest

Request: ConnectCompleteRequest

Response: ConnectCompleteResponse

Post response to a pending connection request.

3.15.1. Message: ConnectCompleteRequest

Reports the success or failure of a ConnectComplete transaction.

Inherits: MeshRequest

Inherits: MeshRequest

Result: SignedConnectionResult (Optional) The connection result to be posted to the portal. The result MUST be signed by a valid administration key for the Mesh profile.

AccountID: String (Optional) The account identifier to which the connection result is posted.

3.15.2. Message: ConnectCompleteResponse

Inherits: MeshRequest

Reports the success or failure of a ConnectComplete transaction.

[No fields]

3.16. Transaction: Transfer

Request: TransferRequest

Request: TransferRequest

Response: TransferResponse

Perform a bulk transfer of the log between the specified transaction identifiers. Requires appropriate authorization

[Not currently implemented]

3.16.1. Message: TransferRequest

Request a bulk transfer of the log between the specified transaction identifiers. Requires appropriate authorization

Inherits: MeshRequest

Inherits: MeshRequest

SearchConstraints: SearchConstraints (Optional) Constrain the search to a specific time interval and/or limit the number and/or total size of data records returned.

3.16.2. Message: TransferResponse

Inherits: MeshResponse

Reports the success or failure of a Transfer transaction. If successful, contains the list of Mesh records to be transferred.

DataItems: DataItem [0..Many] List of mesh data records matching the request.

PageKey: String (Optional) If non-null, indicates that the number and/or size of the data records returned exceeds either the SearchConstraints specified in the request or internal server limits.

4. Assets

4.1. Data

4.2. Credentials

4.3. Reputation

4.3.1. Outbound Messaging Abuse ()

5. Risks

5.1. Confidentiality

Is a regulatory requirement GDPR/HIPPA

5.1.1. Privacy

Stronger requirement, given data but with restrictions on use

Unintended use within an organization may put it in default

GDPR

HIPPA

5.2. Integrity

Modification of data enables control breaches

5.3. Availability

5.3.1. Data loss

Loss of the pictures of the kids at 5

5.3.2. Partial data survivability

Where they buried Aunt Agatha's jewelry but not where they buried Aunt Agatha.

5.4. Inbound Messaging Abuse (Spam)

6. Threats

6.1. End point Compromise

6.2.

7. Controls

7.1. Cryptographic

7.1.1. Triple lock

7.1.1.1. Transport Security

Traffic analysis protection

7.1.1.2. Message Security

Access control

Authentication / Integrity

7.1.1.3. Data Level Security

Data Confidentiality

Non-Repudiation

7.1.2. Key Protection

Use of platform provided facilities to bind private keys in the Device profile to the device is highly desirable. Ideally, private keys should be protected against extraction by hardware techniques presenting a high degree of resistance.

7.1.2.1. Windows

Use encrypted key store

Preferably use BitLocker

7.1.2.2. OSX

Use Key Ring

7.1.2.3. iOS

Use ???

7.1.2.4. Linux

Use the DBUS mechanism

7.1.2.5. Android

Hope and prayers.

7.1.3. Key and Nonce Generation

Use strong mechanisms as described in RFC???

Use of key co-generation as described in part 8 is advised

7.1.4. Key Escrow and Recovery

Master profile keys should be escrowed

Escrow strategies for DARE should take account of the fact that users may want some but not all their data assets to survive them.

7.1.5. Profile Verification

Check that the device credential has been signed by an administration device and that the administration device was properly authorized by the master profile.

Device catalog MUST be signed by the admin device.

Future ? provide protection against rollback attacks.

7.1.6. Identity Validation

See the separate document on the trust model

7.1.7. Trust Broker Accountability

Cert transparency type techniques

7.2. Mesh Messaging

7.2.1. Ingress Control

Every message is subject to access control

Mesh Services should perform abuse filtering on inbound mail

Mesh Services MUST apply user specified ingress control as specified in their contacts catalog.

7.2.2. Egress Control

Some applications may require egress control

For example, classified environments

Mail too stupid to send

7.2.3. Security Signal

Confirmation messages requiring payments

Need Accountability

Need to know the source of the accountability assertions

Should be distinguished from sender controlled part of a message

7.2.3.1. Brand

If messages are being sent on behalf of a corporate entity, this should be signaled to both sender and receiver

Sender ? remind them that they are speaking on behalf of another party

Receiver ? establish who is speaking by the familiar technique.

7.2.4. Accountability

Authentication and consequences

8. Security Considerations

This document comprises the security considerations for the use and implementation of the Mathematical Mesh.

8.1. Integrity

8.1.1. DNS Spoofing

8.1.2. TLS Downgrade

8.1.3. TLS Service Impersonation

8.1.4. Request Replay Attack

8.1.5. Response Replay Attack

8.2. Confidentiality

8.2.1. Side Channel Attack

8.2.2. Session Key Leakage

9. IANA Considerations

All the IANA considerations for the Mesh documents are specified in this document

10. Acknowledgements

A list of people who have contributed to the design of the Mesh is presented in [draft-hallambaker-mesh-architecture] .

11. References

11.1. Normative References

[draft-hallambaker-mesh-architecture]
Hallam-Baker, P., "Mathematical Mesh 3.0 Part I: Architecture Guide", draft-hallambaker-mesh-architecture-08 (work in progress), July 2019.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997.

11.2. Informative References

[draft-hallambaker-mesh-developer] Hallam-Baker, P., "Mathematical Mesh: Reference Implementation", draft-hallambaker-mesh-developer-08 (work in progress), April 2019.

11.3. URIs

[1] http://mathmesh.com/Documents/draft-hallambaker-mesh-security.html

Author's Address

Phillip Hallam-Baker

Email: phill@hallambaker.com