Workgroup: Network Working Group
Internet-Draft: draft-hallambaker-mesh-schema
Published: 23 October 2022
Intended Status: Informational
Expires: 26 April 2023
Authors: P. M. Hallam-Baker
ThresholdSecrets.com

Mathematical Mesh 3.0 Part IV: Schema Reference

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.

[Note to Readers]

Discussion of this draft takes place on the MATHMESH mailing list (mathmesh@ietf.org), which is archived at https://mailarchive.ietf.org/arch/search/?email_list=mathmesh.

This document is also available online at http://mathmesh.com/Documents/draft-hallambaker-mesh-schema.html.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes the data structures of the Mathematical Mesh with illustrative examples. For an overview of the Mesh objectives and architecture, consult the accompanying *Architecture Guide* [draft-hallambaker-mesh-architecture]. For information on the implementation of the Mesh Service protocol, consult the accompanying *Protocol Reference* [draft-hallambaker-mesh-protocol]

This document has two main sections. The first section presents examples of the Mesh assertions, catalog entries and messages and their use. The second section contains the schema reference. All the material in both sections is generated from the Mesh reference implementation [draft-hallambaker-mesh-developer].

Although some of the services described in this document could be used to replace existing Internet protocols including FTP and SMTP, the principal value of any communication protocol lies in the size of the audience it allows them to communicate with. Thus, while the Mesh Messaging service is designed to support efficient and reliable transfer of messages ranging in size from a few bytes to multiple terabytes, the near-term applications of these services will be to applications that are not adequately supported by existing protocols if at all.

2. Definitions

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.

2.1. 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].

2.2. Defined Terms

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

2.3. Related Specifications

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.

2.4. Implementation Status

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

3. Actors

The Mesh mediates interactions between three principal actors: **Accounts**, **Devices**, and **Services**.

Currently two account types are specified, **user accounts** which belong to an individual user and **group accounts** that are used to share access to confidential information between a group of users. It may prove useful to define new types of account over time or to eliminate the distinction entirely. When active a Mesh account is bound to a Mesh Service. The service to which an account is bound **MAY** be changed over time but an account can only be bound to a single service at a time.

A Mesh account is an abstract construct that (when active) is instantiated across one or more physical machines called a device. Each device that is connected to an account has a separate set of cryptographic keys that are used to interact with other devices connected to the account and **MAY** be provisioned with access to the account private keys which **MAY** or **MAY** NOT be mediated by the current Mesh Service. A user's Mesh accounts and the devices connected to them constitute that user's Personal Mesh.

A Mesh Service is an abstract construct that is provided by one or more physical machines called Hosts. A Mesh Host is a device that is attached to a service rather than an account.

3.1. Accounts

A Mesh Account is described by a Profile descended from Profile Account and contains a set of Mesh stores. Currently two account profiles are defined:

ProfileUser Describes a user account.

ProfileGroup Describes a group account used to share confidential information between a group of users.

Both types of profile specify the following fields:

ProfileSignature The public signature key used to authenticate the profile itself

AccountAddress The account name to which the account is currently bound. (e.g. alice@example.com, @alice).

ServiceUdf If the account is active, specifies the fingerprint of the service profile to which the account is currently bound.

AdministratorSignature The public signature key used to verify administrative actions on the account. In particular addition of devices to a user account or members to a group account.

AccountEncryption The public encryption key for the account. All messages sent to the account **MUST** be encrypted under this key. By definition, all data encrypted under this account is encrypted under this key.

User accounts specify two additional public keys, AccountSignature and AccountAuthentication which allow signature and authentication operations under the account context.

Every account contains a set of catalogs and spools that are managed by the service as directed by the contents of the associated Access catalog.

For example, the personal account profile Alice created in

For example, Alice creates a personal account:

```
Alice> meshman account create alice@example.com
Account=alice@example.com
UDF=MDRR-5W72-3RJO-VZB3-VUVQ-IOEC-6UNA
```

The account profile created is:

```
{
  "ProfileUser":{
    "CommonSignature":{
      "Udf":"MCDG-TS7T-UPDD-V667-OXSX-QJ5G-FQRZ",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"hAe7iiCYnnu0jrTSau5WucO74Mj0ZA9DcSzTWyrNQUx7t
  5nJslfBzV0jbzZYjkooGjQlbvIrUTGA"}}},
    "AccountAddress":"alice@example.com",
    "ServiceUdf":"MBYH-BJ3I-EUWL-7QAI-NGIE-TPC6-X4KU",
    "EscrowEncryption":{
      "Udf":"MBMT-KJJW-FU7U-HRMR-K4OI-OKMY-XCYO",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"jMWm2oDjoAgIgNwJEwxi62FoFxk7M6GEL_QTpfrJhowi6
  yAI91GT8x_zEToMbuax09VJCEOPZzaA"}}},
    "AdministratorSignature":{
      "Udf":"MBFM-XW2H-CBLT-AMNQ-ZWVZ-USGI-KOGI",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"wIh4X_rzD3468TEZxKtfVwLRtteDPYPJjyaTQC0rIyo1N
  k6PNsdQvMkAO76Az9BG_ZLlU4NtOkgA"}}},
    "CommonEncryption":{
      "Udf":"MC7V-XVMJ-73OL-YWGL-5MIK-ROXQ-GL3Y",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"clDkQT4l0qWq8xRxJSl6jty_MuqlY39dMc9HaxQ0Ii96M
  4i8EUeQyoUOZQ3b1b40TW7yKAou9HyA"}}},
    "CommonAuthentication":{
      "Udf":"MAX3-E6WP-BMIS-IXPI-MYPR-M56C-OIU3",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"pjgcviHEOran2ZaLka9fegnaj7ut9NRwcS5FGZiF80oJe
  3FzUxvsxMqutI4Zq5nsmP0l8DkQOQIA"}}},
    "ProfileSignature":{
      "Udf":"MDRR-5W72-3RJO-VZB3-VUVQ-IOEC-6UNA",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"81swpm05T9olyqbMHO0daDTWR2i-PKFhHmBtGv5pNJ06h
  6kKE6NU0bCLv6Sy7pbnswWmFszKtSqA"}}}}}
```

3.2. Device

Every Mesh device has a set of private keys that are unique to that device. These keys **MAY** be installed during manufacture, installed from an external source after manufacture or generated on the device. If the platform capabilities allow, device private keys **SHOULD** be bound to the device so that they cannot be extracted or exported without substantial effort.

The public keys corresponding to the device private keys are specified in a ProfileDevice. This **MUST** contain at least the following fields:

ProfileSignature The public signature key used to authenticate the profile itself.

Encryption Public encryption key used as a share contribution to generation of device encryption keys to be used in the context of an account and to decrypt data during the process of connecting to an account.

Authentication Public authentication key used as a share contribution to generation of device authentication keys to be used in the context of an account and to authenticate the device to a service during the process of connecting to an account.

Signature Public signature key used as a share contribution to generation of device signature keys to be used in the context of an account.

For example, the device profile corresponding to one of the devices belonging to Alice is:

```
{
  "ProfileDevice":{
    "Encryption":{
      "Udf":"MA45-T6UD-ZGTI-CT4A-6ZVK-5QFN-CV4E",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"vC8YWlZOWss88PBimflpcecYHfQ59tYVYJhTjbEPABpkC
  SrsXG_GWhBtlKbeLL3t39VbVFajRw6A"}}},
    "Signature":{
      "Udf":"MAW3-J5NK-BZ7B-EBTD-UHUL-HB6L-ZNS2",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"4jrhVLSkJsGhYHhpVShm_m6KLlaxD0OmFJBuGVzWBqQoe
  5tIuNG2QYvO0cKGk0vqEaJRE2YCx82A"}}},
    "Authentication":{
      "Udf":"MCIB-UBQQ-RFSJ-HSYP-3KHU-7FFP-26ZS",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"iZIEfpYYz3rdJ-XLrh46PEpO2p3S9Blv62ZFKHKZlPsjX
  _YUQ8wRWzEJiOehAoTPinZDiOktsRuA"}}},
    "ProfileSignature":{
      "Udf":"MBYN-Q2AT-73EJ-2RO5-FZG3-CMIE-3YFA",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"hWEF49e2PsmfE2FQFQQjdLatqTfyDT35vziEL23AX7gsS
  Cn1q7grK7qTQAZ5EhNE4cSZBrtk1bmA"}}}}}
```

3.2.1. Activation

The device private keys are only used to perform cryptographic operations during the process of connecting a device to an account. During that connection process, a threshold key generation scheme is used to generate a second set of device keys bound to the account by combining the base key held by the device with a second device private key provided by the administration device approving the connection of the device to the account. The resulting key is referred to as the device key. The process of combining the base keys with the contributions to form the device keys is called Activation.

For example, Alice connects the device whose profile is shown above to her account:

```
Alice2> meshman device complete
   Device UDF = MBYN-Q2AT-73EJ-2RO5-FZG3-CMIE-3YFA
   Account = alice@example.com
   Account UDF = MDRR-5W72-3RJO-VZB3-VUVQ-IOEC-6UNA
```

The activation record granting the device rights to operate as a part of the account is:

```
{
  "ActivationAccount":{
    "AccountUdf":"MBYN-Q2AT-73EJ-2RO5-FZG3-CMIE-3YFA",
    "ActivationKey":"ZAAQ-GRK7-IWMF-UM7Z-U5ZF-EI57-I7ZR-S5AI-77NB-H
N4P-K3HJ-JJEL-22JC-JHER"}}
```

And:

```
{
  "ActivationCommon":{
    "Encryption":{
      "Udf":"MC7V-XVMJ-73OL-YWGL-5MIK-ROXQ-GL3Y",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"clDkQT4l0qWq8xRxJSl6jty_MuqlY39dMc9HaxQ0Ii96M
  4i8EUeQyoUOZQ3b1b40TW7yKAou9HyA"}},
      "PrivateParameters":{
        "PrivateKeyECDH":{
          "Private":"-oBCXDopJubjHoHW2ViJJYD58JKQnEfUSNuCV_qvnGgE
  T1GXLOMxyyH_7LI2YVhxhF2i-10Hc_U",
          "crv":"X448"}}},
    "Authentication":{
      "Udf":"MAX3-E6WP-BMIS-IXPI-MYPR-M56C-OIU3",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"pjgcviHEOran2ZaLka9fegnaj7ut9NRwcS5FGZiF80oJe
  3FzUxvsxMqutI4Zq5nsmP0l8DkQOQIA"}},
      "PrivateParameters":{
        "PrivateKeyECDH":{
          "Private":"IauToFkwEzrAJZFext0A_MR2Vb-kBM7WHUoaaDzIQoP4
  w3JPwhceR1dmbrFpp9SAF3QJZ7TFHs4",
          "crv":"X448"}}},
    "Signature":{
      "Udf":"MCDG-TS7T-UPDD-V667-OXSX-QJ5G-FQRZ",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"hAe7iiCYnnu0jrTSau5WucO74Mj0ZA9DcSzTWyrNQUx7t
  5nJslfBzV0jbzZYjkooGjQlbvIrUTGA"}},
      "PrivateParameters":{
        "PrivateKeyECDH":{
          "Private":"2hSCscKw_CNBDyIzF6UX4DwM-t5Yv6-siz8EwD0QSl9a
  sh-da_ZxWquvCR8K4QVrqC2n9dwUS18",
          "crv":"Ed448"}}},
    "Entries":[{
        "Resource":"MMM_Contact",
        "Key":{
          "Udf":"MBBM-SBIP-VJED-CLGV-LMJK-DE5A-6F2T",
          "PublicParameters":{
            "PublicKeyECDH":{
              "crv":"X448",
              "Public":"7cz1xJ_DrPpTHFg8-QGd0JrQSdWRnjvUA7S4g5kFe
  TFRmk8O6eKMMj8JZS5eYfdhyaX8tT7E8rOA"}},
          "PrivateParameters":{
            "PrivateKeyECDH":{
```

```
            "Private":"O-jERsFaLJ2M99OIx43a_OjKahOveiE8xHiD13cC
NFpSSOOxvcphAEwrsQoJttbNgYH70xUXf-k",
            "crv":"X448"}}}},
    {
      "Resource":"MMM_Publication",
      "Key":{
        "Udf":"MAIC-H6BN-KIBL-RAHI-3JN2-V5J5-MG2I",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"WNZc4o1mblQxI3NMbwr3iVXpx5II4RpxRGCo-TYT2
M1moPpi0MTGn4AigJC6WPEUvy66LPqO1JSA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "Private":"i4cjkk1yjAp2vayEcHDQrHWBBp13udP3fIm4CSEl
iub-CchOqje27JoRwOOfedMfGmeKfUNhp3U",
            "crv":"X448"}}}},
    {
      "Resource":"MMM_Inbound",
      "Key":{
        "Udf":"MA5N-2NVE-BY52-W5MJ-TA6Y-3QNH-ZJOY",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"VjW8AECOUULW24znWqE0RV3eYOldnh3DcFF2tgjC7
C5mdpPGe-8wEuEdPcFm7H5u3CcD8imr87wA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "Private":"T2qgwOakAuWWcyCzxLOk1j0DHnQIgoWIfwLxSst6
ezzeUpyQiWjfe1UOhsP4M3WiJC_q-Aw0bP0",
            "crv":"X448"}}}},
    {
      "Resource":"MMM_Outbound",
      "Key":{
        "Udf":"MDU4-UCFY-V3U4-GFZ4-WL2X-QALX-QAJI",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"hQ4CX_PjKhU817jJGW7fTQMxvPFrlUm7B7WHVou-C
387n6WPxGNEGwSXsuML_hugXL_zRj1KACmA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "Private":"IKn2nVh1qZXCZwHuH9Hs4q-2UYIWuC9VcVGY1Si6
LY_TJigbnVljACZTAvAOAZZXyS88hnnpmoE",
            "crv":"X448"}}}},
    {
      "Resource":"MMM_Network",
      "Key":{
        "Udf":"MBCW-EWLR-UEID-3E7U-RHXW-PXAC-OBQT",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"8JIcamPSmHwYvf3Bqroc_CdW2RUhAkbJQku0ThNbt
1fjGEFfM9BbaEg_Qa688VouIOwIrLBMwpKA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "Private":"9mxjYE6Z793CpcCVlIef_pkcpPmi7l6j71cU9tOt
gqPlEExLL2yCOBZrtjsiVuwYTch8riwC5Ns",
            "crv":"X448"}}}},
    {
      "Resource":"MMM_Application",
      "Key":{
        "Udf":"MCCG-5FMB-UMTQ-J7DY-3IZX-G6OT-R7A5",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"KqCkGl0tyD2q2JmgOiiN6ljRK4317oextGfDpuvtZ
mMBDHwmJZ2Wr0TEImHL9NZ6MgK8lTZfgF4A"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "Private":"kysyCj6AAXigg4F9EbABAVX7XvkP5HC0fuEMOWn7
P4_X0Scpf2oyleunMiKQ8Qszq2sN--LksHE",
            "crv":"X448"}}}},
    {
      "Resource":"MMM_Credential",
      "Key":{
        "Udf":"MD3C-QNUT-ZU52-7ZYI-KKLE-634C-FX46",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"8FqZ9krn7yIVRZs2KLjFxGCFIrYzmu2ON4eTDiAOc
ioWUWnQbDOc6hUqOYgK9Mmn4uCE6kXgFruA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "Private":"iwGM49demBp1p9r1MfabZo9Zclak7NtBN4wEy-hv
l4W0_sjxX29Jv4u5xO-5Nz_Gjwj61h1Bjfk",
            "crv":"X448"}}}},
    {
      "Resource":"MMM_Task",
      "Key":{
        "Udf":"MCTJ-433V-5CTJ-YTN3-H54E-6PHW-7ZTQ",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"5UvOZAmTAIA_AJUcZmsiAmQuVZojDVvhSqjEJUN05
clA5Yh0w3wOqOI6KwQLWG1yJCXgzjyGucmA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "Private":"Bq2ZEyMaopaBP0d1H50XJDi5Vj59l4qKcltfcLLw
aYsTCOP1GMUvTNUvwr3siB6mBIZ5Nh4uEg8",
            "crv":"X448"}}}},
    {
      "Resource":"MMM_Bookmark",
      "Key":{
        "Udf":"MDRE-KH6V-XRHQ-XC3I-RRCG-MO2J-N2EW",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"SyzyBK_wToIpD9MJI2HTW3_guC_LJaw2CNwUmaCfN
DaTTgQ_EtZJ01YnDQl376wcL6QHCNHfWskA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "Private":"M7f25o0PIARp9i75IV3x1VSjuDqyk1iI2VantjyI
sxuMH7J6dcoap20fmn1DG8kX4oe8foOkB0M",
            "crv":"X448"}}}}
    ]}}
```

The Mesh protocols are designed so that there is never a need to export or escrow private keys of any type associated with a device, neither the base key, nor the device key nor the contribution from the administration device.

This approach to device configuration ensures that the keys that are used by the device when operating within the context of the account are entirely separate from those originally provided by the device manufacturer or generated on the device, provided only that the key contributions from the administration device are sufficiently random and unguessable.

3.2.2. Connection Assertion

The administration device combines the public keys specified in the device profile with the public components of the keys specified in the activation record to calculate the public keys of the device operating in the context of the account. These public keys are then used to create at a ConnectionDevice and a ConnectionService assertion signed by the account administration signature key.

The ConnectionDevice assertion is used by the device to authenticate it to other devices connected to the account. This connection assertion specifies the Encryption, Authentication, and Signature keys the device is to use in the context of the account and the list of roles that have been authorized for the device..

```
{
  "ConnectionDevice":{
    "Roles":["message",
      "web"
      ],
    "Signature":{
      "Udf":"MA56-V5KL-YMCF-GI3D-PI2F-4OWT-73K6",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"297PWEg-l0jLJzknMVhNY9OGAZZNYHc_leI4Nq72_XRQa
  8LZSajlhJBKOtEjVGyUITQRLj0aYO8A"}}},
    "Encryption":{
      "Udf":"MA6D-RU2J-LL73-LAW6-7JO6-IFCU-WRNI",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"D-HnzU7WQrAjSfiQYLRxSiIK-PBqBHXKSR-1oX1CO5Gb6
  1L31-IV13stjhnXipqeNmuYfpovg0EA"}}},
    "ProfileUdf":"MDRR-5W72-3RJO-VZB3-VUVQ-IOEC-6UNA",
    "Authentication":{
      "Udf":"MBYN-SC4W-IU4X-LIVF-PSC6-6ADO-ZJOF",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"e1nZiuxVRE20PCUKSfqC-ee5yRis7TaKZrlwmEI9RpacG
  f0vc7n3i8l7D_BaryByAUmpFyfKUs0A"}}}}}
```

The ConnectionService assertion is used to authenticate the device to the Mesh service. In order to allow the assertion to fit in a single packet, it is important that this assertion be as small as possible. Only the Authentication key is specified.

The corresponding ConnectionService assertion is:

```
{
  "ConnectionService":{
    "ProfileUdf":"MDRR-5W72-3RJO-VZB3-VUVQ-IOEC-6UNA",
    "Authentication":{
      "Udf":"MBYN-SC4W-IU4X-LIVF-PSC6-6ADO-ZJOF",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"e1nZiuxVRE20PCUKSfqC-ee5yRis7TaKZrlwmEI9RpacG
  f0vc7n3i8l7D_BaryByAUmpFyfKUs0A"}}}}}
```

The ConnectionDevice assertion **MAY** be used in the same fashion as an X.509v3/PKIX certificate to mediate interactions between devices connected to the same account without the need for interaction with the Mesh service. Thus, a coffee pot device connected to the account can receive and authenticate instructions issued by a voice recognition device connected to that account.

While the ConnectionDevice assertion **MAY** be used to mediate external interactions, this approach is typically undesirable as it provides the external parties with visibility to the internal configuration of the account, in particular which connected devices are being used on which occasions. Furthermore, the lack of the need to interact with the service means that the service is necessarily unable to mediate the exchange and enforce authorization policy on the interactions.

Device keys are intended to be used to secure communications between devices connected to the same account. All communication between Mesh accounts **SHOULD** be mediated by a Mesh service. This enables abuse mitigation by applying access control to every outbound and every inbound message.

3.3. Service

Mesh services are described by a ProfileService. This specifies the encryption, and signature authentication keys used to interact with the abstract service.

```
{
  "ProfileService":{
    "ServiceAuthentication":{
      "Udf":"MB6K-DWNX-DYI7-SN2G-HES2-HVCS-LOH4",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"fsMd9IFsWrnLPrjW47RVhzRzqtspSBOr1KbzyskRFhuuI
  wXgJ_xL9Cog9oDS9pPzn9kz8q4RsQMA"}}},
    "ServiceEncryption":{
      "Udf":"MBQA-LJKA-Y7AX-5UZL-HGVL-CJEA-4EI6",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"6qmKuNf5OkUKGfstZKs2HRb-OE8Hh8DQ_74yIoYM5MthZ
  yXkfz7u-SM1qppNXxCowQIuYgGJ_HkA"}}},
    "ServiceSignature":{
      "Udf":"MCDV-VNDH-GUQV-7FEZ-GGXJ-ZL7Y-TN2L",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"hUhRrGH3uf79S6mkpjLDGmU5dBFlkaeM-hO_9TgG1r47f
  aKT8ngURL8dJaZ4Ac0NffnK1zrebIaA"}}},
    "ProfileSignature":{
      "Udf":"MBYH-BJ3I-EUWL-7QAI-NGIE-TPC6-X4KU",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"GX0RRoBCMcp44F0Y9WzuYtv7UjLFnQyN54OPfOzuDaMM4
  cwPKLyA0BbpGqS81xGpVBO88nFCrMyA"}}}}}
```

Since Mesh accounts and services are both abstract constructs, they cannot interact directly. A device connected to an account can only interact with a service by interacted with a device authorized to provide services on behalf of one or more accounts connected to the service. Such a device is called a Mesh Host.

Mesh hosts **MAY** be managed using the same ProfileDevice and device connection mechanism provided for management of user devices or by whatever other management protocols prove convenient. The only part of the Service/Host interaction that is visible to devices connected to a profile and to hosts connected to other services is the ConnectionHost structure that describes the set of device keys to use in interactions with that specific host.

```
{
  "ConnectionService":{
    "ProfileUdf":"MBAW-CPS4-3HUA-XAPW-P2KQ-3FKK-MIJ4",
    "Subject":"MA3K-NG43-GM33-UEAA-TRU7-6C3A-DSBB",
    "Authority":"MBYH-BJ3I-EUWL-7QAI-NGIE-TPC6-X4KU",
    "Authentication":{
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"BpL7zgGXVRv8uYtCsvSfuKyubG3TW7VXEGwe9Mib3mgWc
  nd7HGZGNPJ5q4nqaHN34EPWt_o0S_CA"}}}}}
```

Mesh Services **MAY** make use of the profile and activation mechanism used to connect devices to accounts to manage the connection of hosts to services. But this is optional. It is never necessary for a device to publish a ProfileHost assertion.

4. Catalogs

Catalogs track sets of persistent objects associated with a Mesh Service Account. The Mesh Service has no access to the entries in any Mesh catalog except for the Device and Contacts catalog which are used in device authentication and authorization of inbound messages.

Each Mesh Catalog managed by a Mesh Account has a name of the form:

<prefix>_<name>

Where <prefix> is the IANA assigned service name. The assigned service name for the Mathematical Mesh is mmm. Thus, all catalogs specified by the Mesh schema have names prefixed with the sequence mmm_.

The following catalogs are currently specified within the Mathematical Mesh.

Access: mmm_Access Describes access control policy for performing operations on the account. The Access catalog is the only Mesh catalog whose contents are readable by the Mesh Service under normal circumstances.

Application: mmm_Application Describes configuration information for applications including mail (SMTP, IMAP, OpenPGP, S/MIME, etc) and SSH and for the MeshAccount application itself.

Bookmark: mmm_Bookmark Describes Web bookmarks and other citations allowing them to be shared between devices connected to the profile.

Contact: mmm_Contact Describes logical and physical contact information for people and organizations.

Credential: mmm_Credential Describes credentials used to access network resources.

Device: mmm_Device Describes the set of devices connected to the account and the permissions assigned to them

Network: mmm_Network Describes network settings such as WiFi access points, IPSEC and TLS VPN configurations, etc.

Member: mmm_Member Describes the set of members connected to a group account.

Publication: mmm_Publication Describes data published under the account context. The data **MAY** be stored in the publication catalog itself or on a separate service (e.g. a Web server).

Task: mmm_CatalogTask Describes tasks assigned to the user including calendar entries and to do lists.

The Access, and Publication catalogs are used by the service in certain Mesh Service Protocol interactions. The Device and Member catalogs are used to track the connection of devices to a user account and members to a group for administrative purposes. These interactions are further described below.

In many cases, the Mesh Catalog offers capabilities that represent a superset of the capabilities of an existing application. For example, the task catalog supports the appointment tracking functions of a traditional calendar application and the task tracking function of the traditional 'to do list' application. Combining these functions allows tasks to be triggered by other events other than the passage of time such as completion of other tasks, geographical presence, etc.

In such cases, the Mesh Catalog entries are designed to provide a superset of the data representation capabilities of the legacy formats and (where available) recent extensions. Where a catalog entry is derived from input presented in a legacy format, the original data representation **MAY** be attached verbatim to facilitate interoperability.

4.1. Access

The access catalog mmm_Access contains a list of access control entries providing authorization to devices authenticated by a particular credential. The access catalog provides information that is necessary for the Mesh Service to act on behalf of the user. It is therefore necessary for the service to be able to decrypt entries in the catalog.

The entries in the catalog have type CatalogedAccess and specify a capability. The following capabilities are defined:

NullCapability A capability granting no access rights. May be used to establish a positive statement denying all access.

AccessCapability Authorizes a device authenticated by specified means to request privileged account operations. For example, requesting the status of an account catalog. Also used to provision devices with a copy of their CatalogedDevice entry encrypted under a key held by the device.

CryptographicCapability Specifies a private key encrypted under the encryption key of the service and criteria specifying the parties authorized to request use of the key.

PublicationCapability Authorizes a device authenticated by specified means to obtain a data item.

The Access catalog plays a central role in all operations performed by the service on behalf of the user.

Every access capability is gated by a specified set of authentication criteria. The following authentication criteria are currently defined:

Profile Authentication Key The account profile authentication key authorizes any account action without the need for an access catalog entry. This capability is normally only used during account binding. Administration devices **SHOULD NOT** have access to the account profile authentication key after binding is completed.

Device Authentication Key The service will only perform the operation if the device making the request presents the specified authentication key.

This form of authentication is necessary to restrict access to account operations so that only connected devices can interact with stores, etc.

Account Profile Identifier The service will only perform the operation if the device making the request presents an authentication key that is credentialed by a connection assertion to the specified account profile.

This form of authentication is necessary to perform administration operations on a group account since it is the account rather than the device that is authorized to perform the operation.

Proof of Knowledge The service will only perform the operation if proof of knowledge of the identified shared secret is provided.

This form of authentication criteria is used to allow device connection and contact exchange by means of static (i.e. printed) QR codes.

Future: Currently, the set of authentication criteria is limited to direct grants of a single capability to a single specified device or account. This approach may prove to be unnecessarily verbose requiring the same information to be repeated multiple times.

4.1.1. Access Capability

The access capability permits a specified service operation on the account. Optionally, an access capability **MAY** specify a Data entry encrypted to a key held by the device.

The access capability specifies the set of rights granted to the requester and optionally specifies an EnvelopedCatalogedDevice entry containing the CatalogedDevice entry for the device encrypted under the base encryption key or account encryption key of the device.

The CatalogedDeviceDigest value serves as a tag for the cached data.

4.1.1.1. Operation Rights

The reference code does not currently implement operation rights beyond denying all operations to devices that do not have an access capability entry.

Expansion of the rights handling is planned to permit granular expression of access rights.

mmm_o_UnbindAccount UnbindAccount

mmm_o_Connect
Connect

mmm_o_Complete Complete

mmm_o_Status Status (of specified catalogs or all catalogs)

mmm_o_Download Download (of specified catalogs or all catalogs)

mmm_o_Transact Transact (of specified catalogs or all catalogs)

mmm_o_Post Post outbound message

4.1.1.2. Messaging

The reference code has limited messaging capabilities at present and messaging rights are not specified. The following is a list of possible rights:

mmm_m_Contact Contact messages from the specified subject.

mmm_m_Confirmation Confirmation messages from the specified subject.

mmm_m_Async Asynchronous delivery messages (e.g. mail)

mmm_m_Sync Synchronous delivery messages (e.g. chat)

mmm_m_Presence Forward presence request.

The following media are defined

mmm_c_Text Text that **MUST NOT** contain links or external references

mmm_c_Linked Text that **MAY** contain links or external reference

mmm_c_Audio Audio data (e.g. VOIP, voicemail)

mmm_c_Video Video data

mmm_c_Code Content containing active code including macros, scripts and executables.

4.1.2. Null Capability

The null capability is used to affirmatively deny access to a function. This allows access requests from previously authorized devices whose credentials have been revoked to be handled separately from requests from devices that were never authorized.

4.1.3. Cryptographic Capabilities

A Mesh Service can perform cryptographic operations on a private key according to access criteria specified by the user. This capability is used to support use of threshold cryptography to mitigate compromise of a particular device or individual. The splitting of a cryptographic key into two or more parts allows the use of that key to be split into two or more roles.

Note that this approach limits rather than eliminates trust in the service. As with services presenting themselves as 'zero trust', a Mesh service becomes a trusted service after a sufficient number of breaches in other parts of the system have occurred. And the user trusts the service to provide availability of the service.

A Mesh Service **MAY** also offer to perform private key operations for other purposes. An embargo agent might offer to decrypt data under a private key but only after a specified date and time. An expiry agent might offer to decrypt data but only before a specified date and time. Such services **MAY** be reserved to the customers of a specified service or provided to the general public. Users of such services **MAY** combine key services provided by multiple service providers using threshold techniques to achieve separation of roles.

Since a service might not willingly co-operate with an account transfer request, extension of the Mesh service protocol will be required to enable threshold sharing of the keys required to effect account transfer. This would require one administration device to act as a proxy for threshold signature etc. operations being requested by another administration device. While implementation of such a scheme to support this limited function could be achieved with little difficulty, such a scheme might not support the wider range of peer-to-peer threshold capabilities that might be useful. For example, the confirmation protocol might be modified so that instead of merely providing non-repudiable evidence of the user's response to a request, the confirmation device served as a policy enforcement point through control of a necessary threshold share.

The following service cryptographic operations are specified:

4.1.3.1. Threshold Key Share

A private key share s, held by the service is split into key shares x, y such that a = x + y. One key share is encrypted under a decryption key held by the service. The other is encrypted under a public key specified by the party making the request.

This operation is not currently implemented in the Reference code. When implemented, it will allow the functions of the administration device to be threshold shared between the device and the service, thus allowing the administration capability to be revoked if the device is lost, stolen or otherwise compromised.

Implementation of this capability is expected to be based on the scheme described in . [draft-komlo-frost]

4.1.3.2. Key Agreement

A private key share s, held by the service is used to calculate the value (sl + c).P where l, c are integers specified by the requestor and P is a point on the curve.

This operation is used

4.1.3.3. Threshold Signature

A private key share s, held by the service is used to calculate a contribution to a threshold signature scheme.

The implementation of the cryptographic operations described above is described in [draft-hallambaker-threshold].

Implementation of signatures is not currently covered pending completion of [draft-irtf-cfrg-frost].

4.1.3.4. Fair Exchange

Perform a Micali Fair Exchange trusted intermediary operation.

On receipt of a signature $SIG_B(Z)$, where $Z=E_k(A, B, M)$, the service decrypts Z and returns the result to B.

4.1.4. Publication Capability

The publication capability is not currently implemented. Implementation would allow the Claim/PollClaim mechanism to be eliminated in favor of a mechanism capable of re-use for other purposes.

4.2. Application

The application catalog mmm_Application contains CatalogEntryApplication entries which describe the use of specific applications under the Mesh Service Account. Multiple application accounts for a single application **MAY** be connected to a single Mesh Service Account. Each account being specified in a separate entry.

The CatalogEntryApplication entries only contain configuration information for the application as it applies to the account as a whole. If the application requires separate configuration for individual devices, this is specified in the device activation record.

Two applications are currently defined:

Mail An SMTP email account and associated encryption and signature keys for S/MIME and OpenPGP.

SSH Secure Shell Client.

Accounts **MAY** specify multiple instances of each but each application instance is considered as describing a single application account. Thus, if Alice has email accounts alice@example.com and alice@example.net, she will have application entries for each. Accounts connected to Alice's Mesh account may be authorized to use either, both or none of the email accounts.

Note: The implementation of these features in the current specification is considered to be a 'proof of concept' rather than a proposed final form. There are many issues that need to be considered when integrating a legacy protocol with extensive deployment into a new platform.

4.2.1. Mail

Mail configuration profiles are described by one or more CatalogEntryApplicationMail entries, one for each email account connected to the Mesh profile. The corresponding activation records for the connected devices contain information used to provide the device with the necessary decryption information.

Entries specify the email account address(es), the inbound and outbound server configuration and the cryptographic keys to be used for S/MIME and OpenPGP encryption.

```
{
  "CatalogedApplicationMail":{
    "AccountAddress":"alice@example.net",
    "InboundConnect":"imap://alice@imap.example.net",
    "OutboundConnect":"submit://alice@submit.example.net",
    "SmimeSign":{
      "Udf":"MDBY-ZH7H-QX6W-NIRY-LT3L-SZHB-Z5DG",
      "PublicParameters":{
        "PublicKeyRSA":{
          "n":"rphfk_MnG0gUA8MmVABApcOMPUrv1T5LJbZWi7pz6DiTyefg1u
  Gbn8in6UUzpI-hw4KCIvnkKPsoDdZZCcsJOfs85r7uXx-qUMG7ci0gLHSw6Fpx8xt
  s6EmxeTykPlox0UtFdCSHw_o-EBcCPpoVHLSt45xXqxx91t7Xey8J2vc6cL1a4bkn
  GFKQnf8gsB49Bn7-7gj4dZweR8PtyRa4Jwpi9QAixyXanzke8LFO-Ms5qfY10DO8D
  RyEXRazEF_xFCO6rawoirzDvO-6vLDCescEcwnY0nC12YSwqbicsVHe1W288N0z8A
  ecjWcQPY9ou-1cxdVS1x6h6WhFfjzdgQ",
          "e":"AQAB",
          "kid":"MDBY-ZH7H-QX6W-NIRY-LT3L-SZHB-Z5DG"}}},
    "SmimeEncrypt":{
      "Udf":"MAIG-P2QE-E25S-CDHG-CGZI-DDD4-D3SL",
      "PublicParameters":{
        "PublicKeyRSA":{
          "n":"5Ed2RNK5cpyfpUOfRbzWN0Ad6jtGotRsK-RPDF8IX53t77C4HC
  b5oGo5WakowVvjeuL-Us3YMucN6uFOnLD4YfQWDpgMsKpzxm7NiyCJoyeRv1oZazE
  TCZcfrZ3oSO_a9GjrUh_EU_2v18g6vff_Lsyh75ubr0Zvnap9fXxFoJhOy-Kh8qRo
  Pw62wbVYmUroKhChaufTa21f5udXQC9LeD5Tfq1Yv2HR7b4TKhxeil58DISmMewwb
  30-dk3VrMSRoA3eCPHiBYCo1MN-wes4H1X_xQyqZiq7gjbgP186CpU-O9i4N5MUAE
  spawvaillUQy14z0luJ77FoXsgQs2H1Q",
          "e":"AQAB",
          "kid":"MAIG-P2QE-E25S-CDHG-CGZI-DDD4-D3SL"}}},
    "OpenpgpSign":{
      "Udf":"MDC7-X7HS-QYC4-AGAO-AAX7-XV63-UUXI",
      "PublicParameters":{
        "PublicKeyRSA":{
          "n":"ocRmpDURIJr7r0m6QTvo40ULxo2pohLMj8_pO8TPRLvRsJEByx
  nIDiVz7x5nBIdG2-TqpeqO2TvDk7Jvom2AeI80hhpaZ7HduFXmQC337gNcdv58j-M
  z6y0HfximgNXeZ8NLJYPonvKFzi_AWqRb5eLbLjKvFJssx-Erw30Cs9iJSEEQcSdS
  wZ_LNLlgqDOSv564qtfHF_Hw-1D25qJsfiTPTxT7lCAOwKEbipQ3Uby66HnyvEPJT
  9ETHzTEKzu_IlageW0jqYIYUBOBTk_NQE6GilN0UOosduX_YnUFEpfwEEx57ofsmp
  QvgbyjxBY9LUEshJRfyov1yTWxoBbLEQ",
          "e":"AQAB",
          "kid":"MDC7-X7HS-QYC4-AGAO-AAX7-XV63-UUXI"}}},
    "OpenpgpEncrypt":{
      "Udf":"MAXB-D7HP-GSGZ-OO2N-B2SW-KCWS-MG2R",
      "PublicParameters":{
        "PublicKeyRSA":{
          "n":"t8YNU3OPO2KD2RJ-OsZGR47lA6dLp3KUoJD9i8dfIiIPD5-6rC
  0DK3h9GiGO5NwFcuOQYHsZbgdoSBP-ROwiBBg2ETLA6g20MtuZQzKC-O_hcpB7GsK
  ujErH0H2Zg90HtvUJyrdrblcpQ5VGHoKu-36i2LgBv1I9zKcNP76QTN6Vx4LXglcJ
  VYE-SRbSB8P1Iob_wvUDt7fUSG1DaJQdEI7ns1b8GD_gCsykE9kOyafQmacNJ760H
sOVQ2S0SE0xJQWsqSb-4KJBZSXBqIxjr3q3b0I0YUyowVGzDrhZCo-at131rn90Or
SUD353BDn3mV6lEg5ey7k4tMDYU1pUkQ",
        "e":"AQAB",
        "kid":"MAXB-D7HP-GSGZ-OO2N-B2SW-KCWS-MG2R"}}},
  "Key":"mailto:alice@example.net",
  "Grant":["web"
    ],
  "EnvelopedEscrow":[[{
        "enc":"A256CBC",
        "kid":"EBQF-JZ4P-AOEC-ERXM-GUCI-32AJ-DMUX",
        "Salt":"1vVvewZfe2ZN1cSqmxHxZA",
        "recipients":[{
            "kid":"MBMT-KJJW-FU7U-HRMR-K4OI-OKMY-XCYO",
            "epk":{
              "PublicKeyECDH":{
                "crv":"X448",
                "Public":"4D3hrOay-5bwNbomCMB9ZiF2t-yBTcyCK8HsI
nyCX0mVybwjg4yrCTj9BBCMATWtNaG5QELQRrWA"}},
            "wmk":"Nu1dyuzlcPq8EBg-cKNPuIzSXGsOSCwbai5SlWVZGH06
LJ4p0IoPKQ"}
          ]},
      "ZRIQAAejkf5TRQhZHCXFdRq7Zj-HVQvtNEUzFlvOGzrMqEy29phfGU2i
tPITSGgjdr2i12cOE_ObtJYMtd-J4NVIQB6DlRQgXINIyTs2M1cLAR0VP9WHxkdNV
AG3S4eg1OJnYmKKQJfsn63z2VqJT5JFZ0Zknj6tYV_pkatlf9qeVnHzP7HYMSpwkx
UYocTFCUmCyES4WJfrVw2ngvjGYR8hDOqH0CFKNVtC3lZHYjKWxRkQUFnU9SALV4L
Ljy8dahSjLXnIHGet7TUo_YzO5ctnndw0EvMQBeZsEoIzcfBUayeF7NxjFZL5Vwok
JeeBVeXP9kTxiZPYYbHelK3E6x7QC4LEmqzocfRZtFL-rzXx07R0cPEIk4P3uSvRc
WTIq5LRIwpWGYxNiqJFnsnknVTKZgIR7JM6cWpmcKy_qW1X3iINCbsUgIiY5qIjy_
JW57b0TnCEtHOoappq-KXqUVuBWclKrhnqGfoehk3Yn7ERFlSUO9VfoJW0GJBZLyL
LUDH7cOzVFfWCNuP9qCcgDLZ7ZB_v8xlCGdHvFs5ndWrQMlrqQD2N76hFenBYyP65
jQVB5MZsBSJt88-LFqiUUv98kHiD32GY4aJG-p0YFb_HUdP1wqd_wY2URwNuJElfn
r8N1K-TfB1qD2PiAin2UM--6NddAhIKFoIXNlkKShvb0FkbADkn-7ls6p7QSZ-Uk-
CrU4NV1_6nps--DDFC1XgCtfwOw7hApRHsTW268zy5iLe_3VGq1eyQQs-bMkwYUkP
tWfvygjuXkYFcgAtLv3wFWsIRiuZLag2Kh5Dda_yY-Aymy_EjB4ADAo73TkbyDa80
P-72VHboi9HFcVPriWzzSqvbD21rreklCt8Z0-wVMMz-d-HsmT_fYQVamcaJ3pFqv
EuOtu9_7hv5hvknh1vwmFWm10ZLKsHa0mp3E58P9qaahKLzVc8lYh9dUHiwHN4Hpc
NZdQh4EAlh3hb0MSdQN2zYI55ugzzRC16i4T8WSor6JeEqGald8BsABRkCwViCwJf
1mjChxcKnXiWDV0tsDzU2uSlkb-GeZy4_imvJ2wW0dZ61DUIriNIvl2QoUY87_1aI
MdnPbpvCi2OSLPL6cunIRgtK8TWLiAvFF7ehX9GXiEQ5Qh-jBTcciRrDAtv3sDD0w
_UyMBZnSh-0XNM-SdJrPm7So7ShC-AgB48kqGdkJC-U431lZJ_3GtykDCxdcP2M7n
5vdE8iSqZDshNDU-9q6wfN-tKiT2AOn3ezKz1eZvjWrRnBDXKkfkj3SH_s4aamiCc
VvZAGGZGZedJ1CoOuvlB26BNjW8VUeTap3g2oiKn6X08ePxZfFryq0B-LafQxnzIZ
dIlF8Cb9Ix8Bd3rEKVsbIH2iolsDN575Y9DI34J2VjcUlWkkOoaLoUL1ROnjbhPPL
Wz6FQoh8mavDFc0p9TQt9NvoZmdcasP00YD--5yzwK91tcSuIYgJ7bGj1FHTbWNki
35TTlMaEh5Pclr5aFY75W-grS7TWVAP0ZlHNaSoBGSrAFT4nPjPqRkw32K6DHRLu0
fN1ctOpgSkadwjSjymWdfnBe3lfehpvbP3I1js7b9m_qanYE6V80xXurnbTjDJmxl
0ENOPgUYqVRZylvNVVDT188wwga-2tX3ykgv0KrrT_ki3Vc3LHy4F3-6JioYvLZKR
soZFEbWnsHTA_j23xXb3FV8_YTCcjWgWKRFzZIdTkVGX5oaGMB3vu4u1Si9WOkdfq
PZK_ui1Q80r_AXfZ-FNz-52cbrW1w5ePFJcxvJimeZI2teAkC3zmbqUEsRQ4iDKjv
```

```
bJl-eMmOK5VnybVf4pmqrJ6gSoz7z6OFviy2cvSlyl_gHGpSyKzMEdcKnLFEuq5rI
nqxGw4AZJlXHD5kwGTX7d-PH7inZbJw4YOXZmaPmSm5wNfvnF6Vc2RbcqiJIUdEnf
W6-yVL-nbRmeBRoKJxMII-9XdJ03CVfBJyb1V3CJOQoQrsfEwH_1c58TQQbnKYUOC
QqDpeRtFpldHe2HjBIknuAgT5Vwx726lcYCrAIkJCe6-97bi8viEEyxfD0cMt4C_Q
XlSYi4ltwIJWAuaOgcE6aTfbEYHOeSTs_WwX2eplsdrYMiUwcdrhUGUM4ZzbwOXGK
P21Tzq024ylNOwpVmTZXRYdmv-o4VMTjaNDq4jqIVwTc3p7jjd2Cq3Vexgi0Tr4ub
F5pFfD4pZ_aqXl3KZzc7NBW-Zi3q71VtG-E6-dS8TXraKed4q5cgWWN-bPtni4yXL
z9DaeOXT_oSOdPUeGLEIKibpr3djV1VkuVqULw3NyCMq7mQxFq6GBpHVi66kEBDvG
pRxau91N7BWrm576RBDGbHDQv6LBQzHsH7GFXrrYGzluTJ1qa7XO9S0q_1fBTTyU_
HCZtVEoxM-sUwQVdcVDNL2a5PuTfoAjum0WCOBGl3gVpgpcmK0Crhask1wGaLLdny
wMA_l_Kzx5QIkSe3gArxlfpPHfj7AfTXhcz-dm0PdE3fBWdYUsUmH7Vfa3Vtmi8vk
3XgUKVLH8FBFXu1MrXgiEKUWsHNN1ijZRIF-dQ3ywKlbhIBGffjt_mtzH5CEVW4Le
Ao_I0oKy-a6nnHkBlowwHY-j_80uVlXHIX_DvGP1id2QfxWB6ruzLahfzanjAo3JY
FvCEe2MqbckCcX3vyWwZResCjGeVRnJgwRHDgnnNiWmdrYHZmeZQRj0XzQuuAdENR
61kNel3qk-3bNlcL9Le67Eaarbg68d45rxjLJKYN1UbLEKiY_v55Hh60_7L14w5NU
JRrvfnxlhfuKkDptay3QzkkcnuXmIs8XpI0jPVcLTeGPe2lajGy8Xg9kegHpD5Nm7
CmQyGEQ9h2lpZkbv6AUhQMO9yNm4B-MWtFa-hGIzn5zDTtKV6GAIZLi6rdu1MaHny
6ed3PjVITxJ1yCELVM1hEsBcSO2Amx6kgxbLCoDtYw4S_3KkS6QqApBGpEKOmjxDP
dS5wHjOTB_2-6elkJLEJXHlqpP_FpXPa3KQ_nEsa7EG-Q9ItxUvbuWUhsJAJhb6Hc
n7AXooWcA-SwVZKhVN14Gbjq4YhwmkDyj6DGxuE1VubfeA7PHHQFt4uWc6vlYYqzI
7Cagft9sN32oAPUmVVlfTL8djZN6a9dbhxmxy9asbT7PbrHnbuXDeeSKgj1pO6xRM
jxU84LIFoGDjGiIxtmyNLRcEKr75KXro3z4NmNhixjqKwQ"
      ],
    [{
        "enc":"A256CBC",
        "kid":"EBQF-IQ6V-5H7H-MZ6E-MANZ-ZBWS-ZQW3",
        "Salt":"_RfmDRMAn5Ea0xSvHPTLhA",
        "recipients":[{
            "kid":"MBMT-KJJW-FU7U-HRMR-K4OI-OKMY-XCYO",
            "epk":{
              "PublicKeyECDH":{
                "crv":"X448",
                "Public":"azwxbxdW99wv6UaiCqT3tUYKVXiAVWYsFy6Bi
b5Aa_RKgPIeaanH0-RZyWtIWR1Vysa6h8eZWeQA"}},
            "wmk":"wzgI3XItF23Ba_vFuTdfbpEjc8z6-hvX3MZ8SmWuu257
PUc00bCtIg"}
          ]},
      "XK8015qFIGPcGCIGT-fNbL5cPCKSKkqqw5cXTpwVASoS46riXA7TI9dt
dIMQk3Qjok3OnXUJuzu61uuZAn6Ri8vKsvAzD0bzPAKoYlynVz53zjmvOXKCsLSYE
ZPXI_AAlQBS-5PfV2h8_GH3iBlEu5Y_0_cV9--mRMOkcHeEbLfCcCaI7voTiKfuz2
bdMEeK21HNeQ9cjYRpjFbtgaEVB7HMdl8CPVLQJK2vpXGpKWPUz-NK_VhNVSL5EAh
sQKOPK8FJPzSyqsdMlX_lhtLQXWPwhukfQgF_AQP7lyCin3APTfE1t76yhsmb_hKm
XpJytnSHIaYyxuD5UvR99J5gJEScGjZI2XPGxBtATTT04trRjaAxbWxLAjExhQW6x
OT1pQjNVYmlLckZQXfc1zXF2edf5RyiaDe6K45w8a9TC9DILpcM1PLkO0JDkvHfuE
1IZDMA12hDVbyVliugepKiiSSCYYsixIeif7aNRwNfS8uPXWiUWsNzjPjo9rSSiue
gz6Vbq8QqXsJN0TA0MhiFPrFCHr-HaelMXHnBl3C15NTd6zvrgZOvJ7FUE8D37tdl
A0_6px3xbD_Q87Fuq42GkGDgshzDMOf3Oi4TvLrGUA2MF9TNEvGG6OlcUYrFyEmub
XLsDe0oWNKCqjoMN8rRgDPwp7kLh3OQLZtK8qIyiF0Gm40yy-b29R5LDZLyMYsbyI
pF0J9WuC7R7qCyCbJf6l9HxSXt1-J1YrnT98QOcLBW4Eo1tebdsk02DEOopDrq_hi
```

```
mi-s32KSOYLUGb8gYsAHw00eusfzzP-0nrWKhpkUSO5i9haqpe_vSjm1wlNMrtDEu
tfEuQrF46RSfNv42D5Vn7X4kstYqMhWl43TOwUiD7WlT6-wrK0iTE2PAwFKUxSkKU
-JYBCPCBUUgfwKdzHGcSnny4cAf-YZ5F4UbC6PTmMygsznBoFgwipUP9EYJ7HnQUZ
7LjQ4axlDJMRiR0jlk9RaNxiJS6FVQgCZ9unCqg5dhnL2GSy5w9UTnA9vtdeTugvx
kZlc-oGtBE1KpUQKBvrkaT-_DEz_EnxAFGkXye6BhToTYmpT4e3tND6XkjiZY1ene
d0266iMQlWkl2ty4GF-03-vZe4uUyxBMkfw3zznYxGIKYZPFvKgyDZq2RfJASPMfy
z5_nAb4Uxfm5krYctDaBBlbZrsqIUFei755tFQA7WCS3SSDFg2ecqY7zt4SCDTETZ
PiltqA1nVYyLOW-Z4iNb2DOiVjg1YKvaysDBaiPV3FRAwhLe9gXfkFRConBN0HR6B
rWZWXmu3DH8nUft4arqyW_b65Zesqdsqpc3GzyU1tPQ5l6SKKaOIfu3dGUJ-AdDDO
Y0dsWSkRytt9lKCTuK7Tb8El5pAmcDBkdO4P5tPW2wuZTYaGldbw4F3UsRsm3vcaZ
M52CIVNheN9XJozr0Rw6SZx4X1EEMRnc_t7hC2Eep1b1DReMK-fGcT7TGmlySDAdc
xChYLNJ7jkGfMqNptQ5yapGFkJThv30LRkXB3a4oNxelxHv5_tmBryHHt5xwxUI2i
8_lOm74pj0hZOQmwkN7aJ-ADa26OeqTsn9GYETKCCKI4jdqRhSWMQ9-ZosCBkAKpX
6F6YAi9myfjwMDNl-KGx31kObwiJ_ZnwYYVoyJ9lDRJW2bcFYZiiinbnQjmXQ-wt2
_qp_btYP8qNdmlRm7BipkDhQYmSNjuLKWLSXh1Je-1ZwAOSJTK7BGny0OQb4q7HSn
r9rP1SFjEbTfGs--Uz83G6cm86rLxMf9asGbHOsoB_jd95obb_xjIDSGX1SzZ9OEp
RGSQoiMor2MMStwfJet5b1kz0KgmgWQOqc5Uujc--KL6oH5ICZV0KaAafvDTVNQKV
sW3gMz6JGNMhReuN-C-cdVR4WgUX3-_Povt3dJauMBFc6mIJSdoc6VgX-_qciOp92
MgJ2mnotzwj53B912IAENulNPuoGW1mB3RTXemx13csgLTa990xrYRpadamSJHxm6
DdbceNUCeJ8ztSE06IVxKtW6i_y0QmEcLNMmbpGhBV9S77QHp7_C5oGj6YZThTTIu
MddfifSFJvgQJ7ntkR0zc6YIY-7NqPpUBtlwYfcdT6xFBwOS-CxuJSDm5ahDFWMhu
PnK21Y32juSc1B7fOr8Jk3mZeIYwzXDqfGeIZyh_UvUjvtAwNzN-TmAoQ4Hz2vAbX
HqHFhKR9Iby2Ntn7zFlmCoQzWYsNywUwojovx2XxEfIjkwVFM56bfV4yv70ElqY_E
BvU0mAUhzv7FWQ9zP7iibm5Aou_dmqUxQRXIw9gBrGgsprKLqsjDOORrQNEmxER3H
z9clWzp80juiH5F6bTmTHm7JNODDM2dd6QRT1yLa2gcUeprcJg8Q-uwlkkEr_0ZJI
FJak3QKx3in-GUFaUCYkvh4NGEkUo4ycje3QYm_IYcuWZaimACNFkSESV35j0hD7T
mT84ow_Ws65H3OiIJcVUQQXIkZipxw5faDmA1NYWpf8bOOX0aSu0U_g_UBWyd3S_C
rVOJCpSFeSMN-xUziVUrHW3I3i9kk2DKDHE3v7frPnZBEziv2BsUX6wEJRyy6Rm9K
i1VhPZaqT1MafXFjOhdBZWMBT_xmdSKnjsBwq0piWVeIsL5KOxAJh9Qc46Ecg5Sjv
Y4m1w_rStGbdhiXdTYaeYLv4CpVQXTeCHzWX1Wp9CXs6YE2KvaCxyJl3wBlLDuy_v
0pR5P2Xt8qszRPW9FagJbo2Cpuwrv-hNtN86qjaIsiLplU4nZmlgtwcuGUZaaD9Cu
Xd6pdjhRbd54P932aWRku318CzU4Q1c1hBjxrVTmwdXMmhYswGqGYGg8qO1bDSp8r
jAiXpdV0efAIBIMNBRN6WT7xoEnXkTsUIgTsIm7ftUJ66Vb_fCQBK_u45M-S5BW3l
48bjHomHH-9LAQAT2xftIPRZkhOw7K8SC-ajplVJl4MrXTfeub0Jip7OOKT2yofz-
n-pTIUhCEjP6BzKnQbSq0len8W4B486dK3IBxyXFHipIGRYASrAeA6wnH6whzSObv
3m2peXd_nfW5ADGhaspxafeFsfHmBgSGkXI9Df_LrVCmfK0WDJ9XiW6UjJBoR1IGc
6q9te4pq6CSANvilR_4_fwP76BrGGK3gQfZXZ28-4KUkhwnq5u0VglxoZ6kZkolw2
RrWX8LiELe6Hrr6-qgxJbIjADJctU-tP8U9SW3bvGskmPg"
      ],
    [{
        "enc":"A256CBC",
        "kid":"EBQO-3KIW-XTLG-YHMI-HIQC-73PD-IBXH",
        "Salt":"X7anKIzYdCfBTk9IxZyRJQ",
        "recipients":[{
            "kid":"MBMT-KJJW-FU7U-HRMR-K4OI-OKMY-XCYO",
            "epk":{
              "PublicKeyECDH":{
                "crv":"X448",
                "Public":"nTWeYltIhsQ1IZmaSc4Uvl466A_JqGqbkhBeW
xLnJ2mEVfDWgv8m0Tp6UKOUHo5JHDeqr7bstwuA"}},
         "wmk":"zF3Ed_HTX_1m9Tu_743Z5sgJ5aHTjyM8zm2ajqRZKmzS
tiLKri1y3w"}
        ]},
     "zSusUD77qIY6miDDIHw_EUE91USxVOqKrbgB7ZHNqVBIjYATPAEE-usd
OZ52SOz_t8-EvscbA3GWPEdRFj1Rx-Z8dYX9Bydcvj4eSFS-xxCaGDLgWynwnpkwP
I0Q-cryO5T6v_p9p_37fWpb074MG7724e--YhG8JF-1vWxETz9geBme7RaRx8SQmL
8d_Bj9cUBUiFQ3BMrFH50546onEhabeK8AMtO-pbhd9eC3rksMegGnY2f4lW8nOfP
onyvKmGEE7fGvbtxzx73knqQHkUkC9O5IkXyWSo4A4bkFDDiOkLunoF5BZPGPUjDy
JJmcYkFyL4U5VT9f9oJG1UXzyHLrpsEQa3mYzuITzAIDvneLMmn38PoV9JsKd0s1s
Uhqmt-EmlQdJEfp-s8xBPrisOh1NL0ys-pPe71JUbhhXBkVe-M6tPStoSkW0SBpm8
F7KDCfmU-765wwMYjFIuPoDtC7iZLVcuz8uOsOMFysktE2UD8_yQn7O1JY8tJIz1C
Q6qeFWcN7wsOSypxEPAjI0YH46ZPJL0wpN26qktP4qkp4iIybA66m7Pbl_RZt6-dE
-bcRPohbTFR7G4VEjoCbNM9TKdYYZJBwwRnb6bkJ2NpoTSA8NGD0qiaLBGfrLxnyx
mMJJ6_1JCwMHjSunIDBtTO3GN_JMNtsnwuGRcXbrHh256S1bD6nU8_hO2jnJJdEzS
DVwINpwl1g20vltBUW447Cd-S4whZ9j2JZ3XKCJEo-Z6YZs5djyHgDVn6OqwHtV-J
CWbVAp4cPaIdRdW4vsk8Gi04JWf_DPS1TvyVf32Su4wopyE_wCFaiE1tH3iTO3DjR
E0pN-Ju-ud-MLlJP4-OvTGva0xvyB1UzQmV0-RQ2aRtu1HCqzLjN0ZcwqGfv_DR5K
7_4zz-7Ig8k0Sx5AWowND68DMoxQn2N0sgwuNyiLQYiN6kg5KWLorLEwaQhqurOa_
4M5d4b0hg5wUxaQ3XAfGG0ILTmc8HtilrhMZ4Ch3GI4yz2Km-gmRMAftX1GiNrCu7
MHa3x3zmE-5s3-K2iXrApgZ9YmcsDv4dHpxpkyHEwhEfsMN_rCaat6LGl3B7dul8b
lW7CmJy2Wwe1H9ToCMjr7IrPgPeWUJWGbFLjUjQeBF-3Cr19KE2HF99OTx0yoJkV9
zu8o6tYVqTQQg56CVW0iezqR97YAw-vDjHTPOC2e9wccqrbozwuBs9IhGGL5K038d
KvrJlcBfCQocMWmwnKXMvENtHWu2VMSLDGiHs_r9MnXz1D5cnrv8ZZZyXphP9stOc
v_ipK5PONtvsePNqX_2VbcXFUWREbGbgIDcoU5WSv-Th7_2Mlo0QXlCU83R_JQKU3
g2NgZtW7gPd46tYxkRrgpgXqagvW8zCRARBTvk4UvawhhpFsnh4r70u-jB13NDVCt
6mM1CrgeBp5ZnJVFpz5M9yqhFB6bLBfQJtPPjLTUk3fxeSvcMstsXDu7MKZWhX5ma
k_gexDKSMW4a58pF4QCVqhAh75lx767b57JUThZJaDGNffPUaPYjOakyHkkIcmi1w
zMlyvkbh0U6Gbgp-mNL9MJslSNfsnjSfoHOpFzzxuWGMaYo_m47f1RrC68qDCXpTo
f6x75-1O6L5OPwF2MOjjMpMOtZ3qg5-91IGjNySVb_pmwewZlhYr7pm9pEbYfkpPI
mAWooIHIaqcu2wTnJioQvjSiXUSq-6vCY1cw1trIruotkzTtLuWUAvAf7wF9nqo1H
Z2oC1Py8Mbl21cea1R-CfBbJj2eyRKfyidAcr3EaN2JkBrkp1GIQQIL-7bHDlXJv9
_zuu8OCIJQXiqlZtb5IAUgNUDb_N8cSccEFPd5hqwa1xOCSktU_5SuLsK5pdt-Tsk
snErMPtnRvfCakUstwMeRS1txH6k0AEpC9Vx-bygnS8DlptoEl2xCgzAM2VrNUgsL
S0EpeEmgeLOM_0W-ay2gMUYWJRa2PNs10YJOWzIWj5wc9mbWd7gT5yYXrHuqvpD4G
duVWneDk66Q4Ej24qAuNic9LyS5Y9QnOSSvRwAGg8d-H1Ye5Z3SV7Bce3B5VhjG6J
IEW4iHu6eI_ZlXUdlYkxbpOXcr-mZTAkkNF7WlJGScH91iZd7CwhWrtUR-Ol_sVhc
eRqrP_NHV0rq9HwuxFno3sqG6BCFlFb46n_RXsJKBurubNiTZ9l93x3VbENqT5p8m
6nc7Xpr_cco-EZ06rYUSoQAU0IRDVJItwbGtJdGJK55XFuQZBfBIzblxC-qrFQ0Fo
HMI3tx1fI41Eg9gh9fCKQwAxbTKRET4icZ9vOTGBPTMSyCUTpiyaxLatZHRWtrzYN
4Y1NKR_jgR8jX7BZDgzImCkaT35b8vUIR3nZ_TEE2ya-G4lCt41fMbseogcL_EMH2
sMgyxZlQFdqvLn_jHocSIhVANEdvzKr2MCv43Im8aPnKHRpZxjt7QhcXo-b8zNAlB
C68HYhKG6dAoib9I_b3k19CqcZXzil4AToocSMc6sjXpbD0d5QVUk0osCrPe9bnGO
0JPuIprts0qIeeWcOJAxHMpwSKkPFnIwPGnzC_XGHl6I8RV_RMVikk6rqn3UeaqZW
pZ7-TexKNmIuo_m9dO9mA9mwrxuApqGi4GMJp49mwRzAvFjT4HpmyHh_2J3JuyVeW
kktHhkqCBGvhbxao3eI7uQws-gFNkiN_zevRAgvRvP9bXR31_PKwXe8SPW93gWt8J
FQGGxSiiGltJCQcVlOxmiEYJFzS7YvPePMnJ1ovI2DuSpyySzgj99NCOEg9GzMwJG
UfCg9vMfX0Hds4-l-gbYG0xosUZ_L2VjrqW76bRJnOkbLY7FfnX4x6CFOJvWMdaNw
QoiwXM7JG-cPKg7Vsil3mCNdwEWARRLgKzZZH6Z8yqSoziBgHW8uKOK6bS2aAgIwf
```

```
hGaT75jW5TjASRcnOYdiNfs2Zc0WXXHEohWJajt1FaTsVyFDRkVJEmyafpcIo_w0Y
pPU9r9KE510vhE2IGw37f-Zcs48xJmH6lcBVyMdWAknnz8_tdKSqh9NV1jQnzrJu1
CRP6oAjnGiqOofXOY3FBkqqbbtx3Oc0vq_MgTKcFt5vAkMRsbR5r-NWJPyioyNBBF
VAcrkK7WWMh9ZqZ3TK5HYb6HRoBrbxGgsaWJnqG5DsZFH-FAZhWXIRIr5kkXcB6Fb
oML2ceOQRusdTfTsMWGKkJfl0c_TtI0LwfoCUwmnvq-pjg"
      ],
    [{
        "enc":"A256CBC",
        "kid":"EBQI-M3C2-C55S-XL7O-NU2G-3MFY-CFVT",
        "Salt":"jZ5J5fRRrX5IOPZfjmw_oA",
        "recipients":[{
            "kid":"MBMT-KJJW-FU7U-HRMR-K4OI-OKMY-XCYO",
            "epk":{
              "PublicKeyECDH":{
                "crv":"X448",
                "Public":"KYthWtMfrALo0e4hekTqP-jg5dECp0F8YVr9M
_9sew1qkxBC2Ez8zXN1vak2itTqfs0CcHH2IgwA"}},
            "wmk":"ttLWQbGoMkuE4Zgxy9r6djeKFKy0wMdnqO2C4OtlRNpS
1mYXoUAR-g"}
          ]},
      "KJAOghNFeDgbfPjcTWuC3XSZYGzDB5ZfpRL5TE5Nw8hMDAXud6BmQy4S
sWrBBX2D_Ls4rUevP9xuyjDRBw8JmTcSfGV7lKvomjDleFyMxJZaV92TcWGFhU2Hy
0qwn5q7IjyaJONUybT_9MfKuFm9zP2ofcLpFsMNv2Zh4KdKjt3xgUVtYofKOFP0AD
bixAxfyhBCivoPv_Q7u4ZgoHdFEZefzOJSHK1uAcXRdDFUE0bW91yfQQGhh7yjyjj
49HKwZZCle5lbG2JyxZ0NvCka3dxiVTcB8UVrILIBBpILCUP9DuMQXaSyOAfzNI7x
OzoGRUZtlC3-2Z1VTAuir3hsgaPdRN1baY3y9xqFbA9VXMVdE1aTr4KU82QM_h8N-
7o789gjcbYDaL85fX13Q-r8v9BpBIMARmAGr-QgHA0emX7PijPhRCCUVqE4j9dx7f
Vff_QvjaIj9BisNNy_MAvBA9xYEu-zKRYjCJy0duwwXuznBFi0FozyL1wMCKj8xTp
sNTeae6yYY3WCSv-N_llhDBRq00iiSPZ313h2F_JVAca_ehtbu9jJsVhyuhaBBaYf
1GEIfPK7weSMxgFq98T3WRIQslMsM0KKr5oZ2Jqj9d1aUuVqLNla1p5Ew5GwtGU1X
unZe4JT6rqLdySlAsgtH96DlTAf79_doBhp-a8829VbuJ-e0djQKhpED3E-_Sk_00
JPIWZ2e7gVc2e5xai-6YEUib7fT8AaABS116XWL4UU3Nur3eWhGW_MZklNsjBJcMC
9P1hzoFeiZR-THiZvH1pAj3Mr1DGZb3KaC0hC4Lr-BsSAFi7hxIv8X36-4pd2KwTD
_dknlIbxw6POj4Csw6RYRdLqOZ6ezzPwxyD1MMblm-tM9XM4U10RQsu-VmKZN4U-U
IasTyVuZdmb-DMPyns9ovPToe4BBmBn9g-QMC7zxLKmSYPD03K1SKJAle15Zm1a67
LGhM6UIc06rI00fnYYAyycJS_H97PZGJCicURkAc8dwlUmcYACa4FNb04ipYfYumC
pAVfWZAI_oacg8Z9U6Y1rVs4IUYGprB3QU1xtIvQ-jidYuH2bv2RNEIbaz54QPoBb
Qhw-mEPxjy3acWZfXIANo_AcsqhboeIYboEy05TniVNay7hBA7InuxDXy22DK-QZO
wTGCJpqxLQJvqt-ptjGaeu23lMc2q0hIJS1f1QZSBdY5vUy5ZhabsTBk-xBE3hIg9
rZMI2TIeyyuk3NkA8nRYZOP1_tHbLniYYGaurDAfuC5dXLJ8bL2r7Wdinun_dCDBz
XEjLdn_fzvw2HVNZ0P6D2rllHodEjP4-kj9RhkfVTqRLZyPo0mzNHjd20S7gsCtdD
W0RhpiJ-yVILRi3jz3g4Q3c5nLhr8K7GIOmiGYhgCmUIw7Wxz6X9zBIv-fesxOl2g
fRezJXnW5LVIUTiKGycGSpPv5csK1zWaHRaOtM4_oDorGeOcKbeS9yn-iN34dSIXM
vbUKbzFotQ2lpr-Mdpo94WISdi5wb0PgPVO7Ze2wIiKmGSS7PNIwdzScLOIfT3v65
LmqogSxQ8Tmjju2Ufy9RQmNLD5hV8DhwjGdqU22YVA73Upts3QZ8QQSbagAmL3806
9CEPhWYy4D5y7DQwLvLpPCiIqxLGNMqWiqsZ6vyhLgQBkINqn503y-lb6RMwFBxKE
NFZHcghSinegFTjDCr8NTmEM56L0Zrc-VYV6BkMM7J7gcza-8cQjP8Wj52dUUgiNE
05wIvM0-YBg8tRgk8s1OO81vMYiPHYVH1YNKu3KD-apNve8YJaOlX29uZIRDRQRPo
AEzBXM0O08Wy7YkvObkukj7hiU9J1ZSf_ShkM6XtEpEZxkd2DyEcui-EL33pCVSfR
```

```
P9zYGqEEA5Yl-3M1pRjaL23tvkcGJhcVZz4UYMBxyEN94zIt64Wo6iRmjhTmC82kJ
9_btQxMx2Ei9BCcQiE7l1viTEyKCTECQM6eUl0l40sSRIH5BiRFllbctJTa2zb3fa
CnkJ4KS14aOnQwRWiD3xaQFykanLe8qfnksvwMEtTpNwtPIQ6edmlvYjNCTjkmxSS
aOZFqsaM6ptYI8fsx-hj9yxgfOQvfrDeQLXSkIy59kjHtO7wGANHh-GX5oBHAmWjr
EXd5sHfqDf9bPlUM8FDiWNTRwaRI6LLbZ4XXxk93FzRIQK_4sX0gNSdkXwJTmhrpA
dycwFk5w44DYYH9SKKT7QL5blrXRsuL0Nm2Ih5an3gcqoEA-fcLnaCZCRr8GZWGIf
gJzOnYrEF_jUnrMtlgwehBLZdOvblQPOpneM9EIViZNGOT00viRTJzG_6QcbFO7p9
qUFsy0YbGLbft7W_nPLUsD49tvpjfEN_Gq0Gy58UZVi3ps0daXrYEnMlny33cUAEd
LsHNzB8wXEuppghIMWR5UzdgTlKIA9-n-mfYUC5idK19uUWA9QD8gZH9gzRp1KuLa
6TLs6sMpWmRQuY3lQu_bolHz2Ccc2wXov97ae2gEh4BABRSlI6qkKYkQPNHzITikC
pNAu4yTG6xJM9qyu39CjbHq2kK6JIu3xt8a4ZzKHT1qnkuB3VWQchPe14ygBif5EX
mE-VLgEeQbMeHqqDPd2B8PaW6TwAuJ0TGriYJXPcJv2SBzQmakH6f4Nrx8p_6yhLL
aeEn3vachROl2FTPfkW5ITJkPoWHa6CAMcxwtax5oY8GemAt85JU023djPbZ-yWpH
29fHqV2Ci9v8Y_ygrOeg5ZTTG_XRmcC9BaiCiWJ2D2YmLfC37HR20KelcX6NH3p57
Sp0Me1tKJxigy155HKjrizDEkQTyDxQtjGuSZzA4S4RCa4-1U8RIBgmtq4U1SC1qg
46BSrHibNwvYzRpAPSUmNRejEq14k_iWzxeDSYGt3920DeA5Zu9ZmTSfGhBf5oGPu
iOU6IfEUE9EgOe111NyyarTblo-1_WdCKgZUcAgWzMiOKU04lHOL346tylYRuizO8
QBHB7z3qSfLQUUUx5pRJuMhpRGRevVv21vF33sZh6v9HrBf7bbkcCPElFY6RwXXIr
kyhLE9-9uY64I6g41ZbswdG7pRrJ2icB2OGi8iCddpj0eV2JPH073N5zlajSHtNSq
EFYU2X5w2NdM-h7-Ph4uq_n_Kb9gSgunzMsnce0r0EFTefOglxuWRfeb6LHL-m5Sh
VBSLONdc5vTfARi0oUc227x7CsFQuPjNkVtXuESMOt_n9g"
      ]
    ]}}
```

Note that the inbound and outbound server configuration does not specify the access credentials to be used to access the service. These are specified in the Credential catalog.

Future: The mail application should support automated means of credentialling the public key including obtaining an X.509v3 certificate or uploading the key to a key service.

4.2.2. SSH

SSH configuration profiles are described by entries in multiple catalogs

CatalogedApplicationSsh entries in the Applications catalog. Specify an SSH client credential or certificate signing credential

CatalogedCredential entries in the Credential catalog. Specify SSH host keys (i.e. contents of the known hosts file)

CatalogedContact entries in the Contacts catalog. Specify SSH client keys (i.e. material from which an authorized_key file entry might be constructed).

Future: Client and Host certificates are not currently supported. This is clearly desirable but requires additional implementation considerations.

Future: Provisioning of SSH host private keys is currently out of scope. This is best considered as part of the device provisioning and authorization flow and will lead to entries being created/ updated in the device catalog.

A user may have separate SSH configurations for separate purposes within a single Mesh Account. This allows a system administrator servicing multiple clients to maintain separate SSH profiles for each of her customers allowing credentials to be easily (and verifiably) revoked at contract termination.

```
{
  "CatalogedApplicationSsh":{
    "ClientKey":{
      "Udf":"MDCT-IRNQ-JDWH-IRIP-FZK2-YP4G-5MGS",
      "PublicParameters":{
        "PublicKeyRSA":{
          "n":"r3BGZS5ksJmWK-SLRdfXr13-mMS3jtac2pMtkqqX5EfZ22CNhL
  1JBK4emB3L5CLHA74zz-ALk95t_V02VJRUEZCRga242COMmIP_D6Og1y053VHsb5r
  8Ov8w3ujCqiKrQ2s4PTwyYHPyEDbgkX8PcdC4kWGobTUQ6_ll_McVyCzwb-Ha6Slh
  PdOcYNQFhb3YfPp7dkapUmygN9qySGefOs2HSeylRwsnm75vdkuCzFyLsWpb4ajve
  rEGJ8QQO1WYckfQOPrS7EB7NPfJgc0UcrX93uPyKN-Pee4eETsiEwcCIHrMMyGQoJ
  xRVgaQDrFqx484PKsvP4O5cesy9VVjOQ",
          "e":"AQAB",
          "kid":"MDCT-IRNQ-JDWH-IRIP-FZK2-YP4G-5MGS"}}},
    "Key":"MDCT-IRNQ-JDWH-IRIP-FZK2-YP4G-5MGS",
    "Grant":["web",
      "threshold"
      ],
    "EnvelopedEscrow":[[{
          "enc":"A256CBC",
          "kid":"EBQH-G57X-AWCC-FCHD-6X5J-Z7P3-I2NQ",
          "Salt":"R5yBxnxmNK4Ha9VKwYu2iQ",
          "recipients":[{
              "kid":"MBMT-KJJW-FU7U-HRMR-K4OI-OKMY-XCYO",
              "epk":{
                "PublicKeyECDH":{
                  "crv":"X448",
                  "Public":"rEPPH0rrnI_G_4pc7_TWSyYnm86aL7mGUOfTD
  HtVykD9uQb1LtHXyvatXzXNXg39i-A2mioXeUaA"}},
              "wmk":"-zCF4nTooNfFAdOFPEwSSL9uxxOoLfGLXnPbadSjz8Uu
  d-OqNC7Hqw"}
            ]},
        "P2FL0S-xOs2wRXxaoH52RD78RhSmFUGSXKfG2JzOKTYdquqvJegMcjUP
  9037R7iEs-dQOlHovY5C8I5S6J-JSKxot8RvuWzMExNkSCgqPZJ9TnZVLRozMssxK
  9YYGGdvmFcQazA-Q3htZWtWWTlyYDi0EK6hFmcvkIAJmvA9kkqTGA5GzRUZsYJGgf
  9VAN78ksRHKFNm9NwLUpNMruhWLNAwrhICeIyTGFRr-am0g4XE3hfiSmElSTX_HJG
  EFmhftOkfA8Enp7OPGKXzWAM0bdZ7j-50t0RxFlp-mzVu2xu1pG_TuTg3OuyQSQdL
  qauHqFftz_6yQLcYZCqiS7rgUmOFXDtE1TfovxnhZdtVZG1mqJgdQvxNkXdgeFSEE
  yPG_o7BPO4k6AGnRhBopX-iiOilnjVbegXOaDFqOU2PKFrLmQ1M8vxMJKcUO3S-Hf
  mXMywQaTsPI1QAlwRJFHUlUsAw0YBZGhuxDIQsOsHl2gz8G5K3AgIEq3fSD6Jt9Sa
  gdA_jTFRy88L6RhNgBTiJ08HM1cXVzzN1Kh3IH1ZlFFlmLvtMqVhHWN9Ub8izOU-f
  _tnhdGiGlmgamqS---jxqB9MNqnbUMD14XgZzRq0so4TcgEEkmhLav6BiXmb2GRBD
  9Dr6cc-lrlhRJfjrepbbKvpO_c_8qHp6nfbroj6sy4e9_XeZGMspX38YLupH0nzOl
  EFsgG7cwdHEDnAgxOU7qrVbR6AmAHwOsdseVMlXpwrGILS0b0ylRz7PBC2zREeE-Z
  nOlFhGCOqgT4N9UrV4RD4Zz9TzBaASesmvUtsCAm1LkVi3xChu0hLS_NislI-Cg3q
  a5rQK-OpB1Y06fAxLRSyV8PjHJCmJPy-7Svp0rWNDXEi_AOJntvcKmA5zuL0Q5o59
  tKGd-AP9kA1dVqTN2IzJ-jHf-rwluuSUIHeNx3F6x9X--BdpXzwvtuYzm5BwYP1Pw
  DVEw5GM6veiluS7tjzrB5TFcZwVNJx-evoy-GHLaBJHmbKy1djfDzHHIbSzYbAMhC
  wC5alnZfges3PcSgcam8yYyjLXTiKxY7Ty6T5-8A6zJR4Ts3c7kItKXnxt8IMNzvd
DT0CyGmPCSAdJe4_07hx0N9tjmOPPFkYMUSv5brB3MnMQbSzItD2anw23IvOhntRQ
pglC4zizE4X5IHl7FYzqth6m8_fJaUI5jlW823-TlcmOgVgnzifz0XsH7Z3AnBgfW
tGgAlWhW_XkvDY4Jpxr73jv_7tkuuf4hCIOyTuejXGD0qYHMfWe3CiXwv86kccimx
6j9xNViYG_67vLz1NSPAQ9nJUGBogXgkjwQLEb5aiTfMwAIUoUKvmB8EQguVTSxv2
x5R2QmDHgRAWLGSg06hkK9LXouGil3uXJLbkvANWSc1NswnpDBLnG0BsMEX2RJcMr
5UMK91-oRUOx-WGZZMtDkx8VZxKKHknstJzogb2WOuTlnJ5LkjYy372YFRdP0Olzh
c3iMgPOCnfoSG5nYz6v0j8sUoNIEluQey77R38RGch7JxpDNRJG8y4DsiJqLrImaJ
_jtJpKFHE3gC3Pbfz1PHjRVizhRwmeJINNkIQ9GyKmAQp0BkloTuo3XIQsD1FQlCz
R5RkogEBiBCw7R-R-TjeVkFtNP45e06vyrHbPvWdDdkfTAbd6kKH6PODVOD7rTTwq
MGtMuNPH55PYfEYhKQzDddFtV84Uj_QJoWe3_YepdHg0On4XHc7xjJL9ioMIV5KJp
1mQzhcQKNRtXrhrL5UjtAmIpgDcrP-KBPWOcsM0HXlWRFjnFSGYYOeeJBMyjOmypA
cIfmGDI8wjnqCEr8sCdkud8-YJPgpY1p4PGt2F5U7AeRKMrSmANAeAbw_HJ0LNJzS
TT_0SS2SgMterOKFPkM0Cs7EKNJZsQRpv5fVUDuCpp1D84RWQ9x1eXusjH7WTZsl7
nFTzDE5MmUAeoMqydSLP6w-XGCZkVQacmi3-fsyWv1cPXVZ1CCaQoN_rPC9LLlNSe
oiOy6NfZos2nuu_LIOiSTbs7Ewr32pAORWlAY0maAVVLd_N6bOpUa6LV3xR64F-5G
XvEBXR5DSUcwvlPSiIh-Ft4rEMAPqX2yd2P3Sccm2YbCxb3zzEElRAPRNDTo5UEpp
aKPprkZ1SV2uAnV0EoKje83RHSmXgJ-cH8sTZEsaY5SXnxPI3045LoQrqYI2FMgHM
adPOe8b2AglkLsl64kXBEew78_IQEDxWneVjDvdObbU89PSQ7CNEWR13XwXbqccRn
AyaIX16FAu-SAMeW6bg2oxVer5H3yBoC_9tNTe73wcAthNMgs0EsNPAHvI-WnbgHl
ASjHdWsx_Azrn110U9LZ_2DwOod-uySyicrN22DopuBz_UK0tbZyQke__BQ9WcbU5
uuaJDQd4hKGn5K33_4S_i5LJzTRO3GPn6MblgmU4RhJncgQsdyCv83HlOtdBbfpB9
5C6vujPt2QYldXwAqhDDMmiuJTpXKOw_yBM9cbIISYUbGt9_V1f2lCGKqAKFvY4HP
rfCXbE0cy9oWgIHVRHMh48Wb92XUxbq6cq3DY7vgitIAVlpVKEv9sDflVghRDgvuw
jravIN0rjYNkHa5g5TAeEF3Hh7m8GZKmcLuawZ2gq_J_pmGIbV666wpcckH3gJIm2
hhcF7gjflt5WzQazi7R01Mi-B9ZdPnSi5VzyVdNAcR66SR69I_63GrTMtMa52NgTR
2ANgXD-hv346LKH0mbzn7uHohlZw3RfU_-5Fwl6C5v1vWZ9Bj183afX-9d5mJf5u_
QiwE4EpKJc1aQJ298PxM4oR4TUigYvqw1nzRgzO3PW6pf1kboRosjkMe01h2xsNQ4
0slpSspHVPYGKZawkVsDR7vmdEdElmHRLqwmjm3EM8pUoIKMYafkD8VJWDFt1SwSh
aMT1mxedgl3y2PwPIwieyHF6P2I1T68DTX3PiMTm92f86sF57-3dWeeLkxULCiHQl
2jWLFXaL2-VonNInTg6sHKh6L6hJTRaq-QLEJHxxydxMwjChlvgUVxWKuo-snBGaU
gnNT-evaEO2Li-8iNnE5BuIvWKVwqzeOlhc1vR20PXxOEi9SAZTqEGWdZ9gvPadeM
ZsjjS8sl07ZWzMvCX7NXF_ZBS660YNcalGQVoI0-u1FFT0B8iG2VpWs14DZl98fyK
NleGKeppLVX-Z3pS9WJQ6cfAs8kjFS-Sj9kGuL44uZEJWA"
      ]
    ],
  "LocalName":"ssh"}}
```

4.3. Bookmark

The bookmark catalog mmm_bookmark contains CatalogEntryBookmark entries which describe Web bookmarks and other citations allowing them to be shared between devices connected to the profile.

The fields currently supported by the Bookmarks catalog are currently limited to the fields required for tracking Web bookmarks. Specification of additional fields to track full academic citations is a work in progress.

```
{
  "CatalogedBookmark":{
    "Uri":"http://www.example.com",
    "Title":"site1",
    "LocalName":"Sites-1",
    "Uid":"NCQL-JSFV-RDXN-GTGT-LYZQ-H7YK-CE74"}}
```

4.4. Contact

The contact catalog mmm_contact contains CatalogEntryContact entries which describe the person, organization or location described.

The fields of the contact catalog provide a superset of the capabilities of vCard [RFC2426].

```
{
  "CatalogedContact":{
    "Key":"MDRR-5W72-3RJO-VZB3-VUVQ-IOEC-6UNA",
    "Self":true,
    "Contact":{
      "ContactPerson":{
        "Id":"MDRR-5W72-3RJO-VZB3-VUVQ-IOEC-6UNA",
        "Anchors":[{
            "Udf":"MDRR-5W72-3RJO-VZB3-VUVQ-IOEC-6UNA",
            "Validation":"Self"}
          ],
        "NetworkAddresses":[{
            "Address":"alice@example.com",
            "EnvelopedProfileAccount":[{
                "EnvelopeId":"MDRR-5W72-3RJO-VZB3-VUVQ-IOEC-6UNA",
                "dig":"S512",
                "ContentMetaData":"ewogICJVbmlxdWVJZCI6ICJNRFJSLT
  VXNzItM1JKTy1WWkIzLVZVVlEtSU9FQy02VU5BIiwKICAiTWVzc2FnZVR5cGUiOiA
  iUHJvZmlsZVVzZXIiLAogICJjdHkiOiAiYXBwbGljYXRpb24vbW1tL29iamVjdCIs
  CiAgIkNyZWF0ZWQiOiAiMjAyMi0xMC0xOFQxMjo0MzoyOFoifQ"},
              "ewogICJQcm9maWxlVXNlciI6IHsKICAgICJDb21tb25TaWduYX
  R1cmUiOiB7CiAgICAgICJVZGYiOiAiTUNERy1UUzdULVVQREQtVjY2Ny1PWFNYLVF
  KNUctRlFSWiIsCiAgICAgICJQdWJsaWNQYXJhbWV0ZXJzIjogewogICAgICAgICJQ
  dWJsaWNLZXlFQ0RIIjogewogICAgICAgICAgImNydiI6ICJFZDQ0OCIsCiAgICAgI
  CAgICAiUHVibGljIjogImhBZTdpaUNZbm51MGpyVFNhdTVXdWNPNzRNajBaQTlEY1
  N6VFd5ck5RVXg3dDVuSnNsZkIKICB6VjBqYnpaWWprb29HalFsYnZJclVUR0EifX1
  9LAogICAgIkFjY291bnRBZGRyZXNzIjogImFsaWNlQGV4YW1wbGUuY29tIiwKICAg
  ICJTZXJ2aWNlVWRmIjogIk1CWUgtQkozSS1FVVdMLTdRQUktTkdJRS1UUEM2LVg0S
  1UiLAogICAgIkVzY3Jvd0VuY3J5cHRpb24iOiB7CiAgICAgICJVZGYiOiAiTUJNVC
  1LSkpXLUZVN1UtSFJNUi1LNE9JLU9LTVktWENZTyIsCiAgICAgICJQdWJsaWNQYXJ
  hbWV0ZXJzIjogewogICAgICAgICJQdWJsaWNLZXlFQ0RIIjogewogICAgICAgICAg
  ImNydiI6ICJYNDQ4IiwKICAgICAgICAgICJQdWJsaWMiOiAiak1XbTJvRGpvQWdJZ
  053SkV3eGk2MkZvRnhrN002R0VMX1FUcGZySmhvd2k2eUFJOTFHVAogIDh4X3pFVG
  9NYnVheDA5VkpDRU9QWnphQSJ9fX0sCiAgICAiQWRtaW5pc3RyYXRvclNpZ25hdHV
  yZSI6IHsKICAgICAgIlVkZiI6ICJNQkZNLVhXMkgtQ0JMVC1BTU5RLVpXVlotVVNH
  SS1LT0dJIiwKICAgICAgIlB1YmxpY1BhcmFtZXRlcnMiOiB7CiAgICAgICAgIlB1Y
  mxpY0tleUVDREgiOiB7CiAgICAgICAgICAiY3J2IjogIkVkNDQ4IiwKICAgICAgIC
  AgICJQdWJsaWMiOiAid0loNFhfcnpEMzQ2OFRFWnhLdGZWd0xSdHRlRFBZUEpqeWF
  UUUMwckl5bzFOazZQTnNkUQogIHZNa0FPNzZBejlCR19aTGxVNE50T2tnQSJ9fX0s
  CiAgICAiQ29tbW9uRW5jcnlwdGlvbiI6IHsKICAgICAgIlVkZiI6ICJNQzdWLVhWT
  UotNzNPTC1ZV0dMLTVNSUstUk9YUS1HTDNZIiwKICAgICAgIlB1YmxpY1BhcmFtZX
  RlcnMiOiB7CiAgICAgICAgIlB1YmxpY0tleUVDREgiOiB7CiAgICAgICAgICAiY3J
  2IjogIlg0NDgiLAogICAgICAgICAgIlB1YmxpYyI6ICJjbERrUVQ0bDBxV3E4eFJ4
  SlNsNmp0eV9NdXFsWTM5ZE1jOUhheFEwSWk5Nk00aThFVWVRCiAgeW9VT1pRM2IxY
  jQwVFc3eUtBb3U5SHlBIn19fSwKICAgICJDb21tb25BdXRoZW50aWNhdGlvbiI6IH
  sKICAgICAgIlVkZiI6ICJNQVgzLUU2V1AtQk1JUy1JWFBJLU1ZUFItTTU2Qy1PSVU
  zIiwKICAgICAgIlB1YmxpY1BhcmFtZXRlcnMiOiB7CiAgICAgICAgIlB1YmxpY0tl
  eUVDREgiOiB7CiAgICAgICAgICAiY3J2IjogIlg0NDgiLAogICAgICAgICAgIlB1Y
  mxpYyI6ICJwamdjdmlIRU9yYW4yWmFMa2E5ZmVnbmFqN3V0OU5Sd2NTNUZHWmlGOD
```

```
BvSmUzRnpVeHZzCiAgeE1xdXRJNFpxNW5zbVAwbDhEa1FPUUlBIn19fSwKICAgICJ
Qcm9maWxlU2lnbmF0dXJlIjogewogICAgICAiVWRmIjogIk1EUlItNVc3Mi0zUkpP
LVZaQjMtVlVWUS1JT0VDLTZVTkEiLAogICAgICAiUHVibGljUGFyYW1ldGVycyI6I
HsKICAgICAgICAiUHVibGljS2V5RUNESCI6IHsKICAgICAgICAgICJjcnYiOiAiRW
Q0NDgiLAogICAgICAgICAgIlB1YmxpYyI6ICI4MXN3cG0wNVQ5b2x5cWJNSE8wZGF
EVFdSMmktUEtGaEhtQnRHdjVwTkowNmg2a0tFNk5VCiAgMGJDTHY2U3k3cGJuc3dX
bUZzekt0U3FBIn19fX19",
            {
              "signatures":[{
                  "alg":"S512",
                  "kid":"MDRR-5W72-3RJO-VZB3-VUVQ-IOEC-6UNA",
                  "signature":"UNtyhJFuwLPmj8uuSw6Ts61ACoOkEoLF
63rSbHT35bDRuS8VFhnkyNX2mQ4SIGHuBPPSURZB84kAGRhq0MRAR32jbTJr4We3L
Sy_PdeGh5hVaGbRMUhX2V40SVzy7SxLcGYW8iXqXq9PVYL3S315fBIA"}
                ],
              "PayloadDigest":"6P0GfqW3b_kYhYrWG0e0oXy0uENOr_Yx
xcU3CgLaNO3tLeTmWkUCGtlZUMvEptTtN-Ysu4KqmXr7OmphX03qow"}
            ],
          "Protocols":[{
              "Protocol":"mmm"}
            ]}
        ],
      "Sources":[{
          "Validation":"Self",
          "EnvelopedSource":[{
              "dig":"S512",
              "ContentMetaData":"ewogICJNZXNzYWdlVHlwZSI6ICJDb2
50YWN0UGVyc29uIiwKICAiY3R5IjogImFwcGxpY2F0aW9uL21tbS9vYmplY3QiLAo
gICJDcmVhdGVkIjogIjIwMjItMTAtMThUMTI6NDM6MjhaIn0"},
            "ewogICJDb250YWN0UGVyc29uIjogewogICAgIkFuY2hvcnMiOi
BbewogICAgICAgICJVZGYiOiAiTURSUi01VzcyLTNSSk8tVlpCMy1WVVZRLUlPRUM
tNlVOQSIsCiAgICAgICAgIlZhbGlkYXRpb24iOiAiU2VsZiJ9XSwKICAgICJOZXR3
b3JrQWRkcmVzc2VzIjogW3sKICAgICAgICAiQWRkcmVzcyI6ICJhbGljZUBleGFtc
GxlLmNvbSIsCiAgICAgICAgIkVudmVsb3BlZFByb2ZpbGVBY2NvdW50IjogW3sKIC
AgICAgICAgICAgIkVudmVsb3BlSWQiOiAiTURSUi01VzcyLTNSSk8tVlpCMy1WVVZ
RLUlPRUMtNlVOQSIsCiAgICAgICAgICAgICJkaWciOiAiUzUxMiIsCiAgICAgICAg
ICAgICJDb250ZW50TWV0YURhdGEiOiAiZXdvZ0lDSlZibWx4ZFdWSlpDSTZJQ0pOU
kZKU0xUVlhOekl0TTFKS1R5MQogIFdXa0l6TFZaVlZsRXRTVTlGUXkwMlZVNUJJaX
dLSUNBaVRXVnpjMkZuWlZSNWNHVWlPaUFpVUhKdlptbHNaCiAgVlZ6WlhJaUxBb2d
JQ0pqZEhraU9pQWlZWEJ3YkdsallUbiMjR2YlcxdEwyOWlhbVZqZENJc0NpQWdJ
a04KICB5WldGMFpXUWlPaUFpTWpBeU1pMHhNQzB4T0ZReE1qbzBNem95T0ZvaWZRI
n0sCiAgICAgICAgICAiZXdvZ0lDSlFjbTltYVd4bFZYTmxjaUk2SUhzS0lDQWdJQ0
pEYjIxdGIyNQogIFRhV2R1WVhSMWNtVWlPaUI3Q2lBZ0lDQWdJQ0pWWkdZaU9pQWl
UVU5FUnkxVVV6ZFVMVlZRUkVRdFZqWTJOCiAgeTFQV0ZOWUxWRktOVWN0UmxGU1dp
SXNDaUFnSUNBZ0lDSlFkV0pzYVdOUVlYSmhiV1YwWlhKeklqb2dld28KICBnSUNBZ
0lDQWdJQ0pRZFdKc2FXTkxaWGxGUTBSSUlqb2dld29nSUNBZ0lDQWdJQ0FnSW1OeW
RpSTZJQ0pGWgogIERRME9DSXNDaUFnSUNBZ0lDQWdJQ0FpVUhWaWJHbGpJam9nSW1
oQlpUZHBhVU5aYm01MU1HcHlWRk5oZFRWCiAgWGRXTlBOelJOYWpCYVFUbEVZMU42
VkZkNWNrNVJWWGczZERWdVNuTnNaa0lLSUNCNlZqQnFZbnBhV1dwcmIKICAyOUhhb
```

```
EZzWW5aSmNsVlVSMEVpZlgxOUxBb2dJQ0FnSWtGalkyOTFiblJCWkdSeVpYTnpJam
9nSW1Gc2FXTgogIGxRR1Y0WVcxd2JHVXVZMjl0SWl3S0lDQWdJQ0pUWlhKMmFXTmx
WV1JtSWpvZ0lrMUNXVWd0UWtvelNTMUZWCiAgVmRNTFRkUlFVa3RUa2RKUlMxVVVF
TTJMVmcwUzFVaUxBb2dJQ0FnSWtWelkzSnZkMFZ1WTNKNWNIUnBiMjQKICBpT2lCN
0NpQWdJQ0FnSUNKVlpHWWlPaUFpVFVKTlZDMUxTa3BYTFVaVk4xVXRTRkpOVWkxTE
5FOUpMVTlMVAogIFZrdFdFTlpUeUlzQ2lBZ0lDQWdJQ0pRZFdKc2FXTlFZWEpoYld
WMFpYSnpJam9nZXdvZ0lDQWdJQ0FnSUNKCiAgUWRXSnNhV05MWlhsRlEwUklJam9n
ZXdvZ0lDQWdJQ0FnSUNBZ0ltTnlkaUk2SUNKWU5EUTRJaXdLSUNBZ0kKICBDQWdJQ
0FnSUNKUWRXSnNhV01pT2lBaWFrMVhiVEp2UkdwdlFXZEpaMDUzU2tWM2VHazJNa1
p2Um5ock4wMAogIDJSMFZNWDFGVWNHWnlTbWh2ZDJrMmVVRkpPVEZIVkFvZ0lEaDR
YM3BGVkc5TlluVmhlREE1VmtwRFJVOVFXCiAgbnBoUVNKOWZYMHNDaUFnSUNBaVFX
UnRhVzVwYzNSeVlYUnZjbE5wWjI1aGRIVnlaU0k2SUhzS0lDQWdJQ0EKICBnSWxWa
1ppSTZJQ0pOUWtaTkxWaFhNa2d0UTBKTVZDMUJUVTVTTFZwWFZsb3RWVk5IU1MxTF
QwZEpJaXdLSQogIENBZ0lDQWdJbEIxWW14cFkxQmhjbUZ0WlhSbGNuTWlPaUI3Q2l
BZ0lDQWdJQ0FnSWxCMVlteHBZMHRsZVVWCiAgRFJFZ2lPaUI3Q2lBZ0lDQWdJQ0Fn
SUNBaVkzSjJJam9nSWtWa05EUTRJaXdLSUNBZ0lDQWdJQ0FnSUNKUWQKICBXSnNhV
01pT2lBaWQwbG9ORmhmY25wRU16UTJPRlJGV25oTGRHWldkMHhhTZEhSbFJGQlpVRX
BxZVdGVVVVTQogIHdja2w1YnpGT2F6WlFUbk5rVVFvZ0lIWk5hMEZQTnpaQmVqbEN
SMTlhVEd4Vk5FNTBUMnRuUVNKOWZYMHNDCiAgaUFnSUNBaVEyOXRiVzl1Ulc1amNu
bHdkR2x2YmlJNklIc0tJQ0FnSUNBZ0lsVmtaaUk2SUNKTlF6ZFdMVmgKICBXVFVvd
E56TlBUQzFaVjBkTUxUVk5TVXN0VWs5WVVVTMUhURE5aSWl3S0lDQWdJQ0FnSWxCMV
lteHBZMUJoYwogIG1GdFpYUmxjbk1pT2lCN0NpQWdJQ0FnSUNBZ0lsQjFZbXhwWTB
0bGVVVkRSRWdpT2lCN0NpQWdZ0lDQWlZM0oySWpvZ0lsZzBORGdp
TEFvZ0lDQWdJQ0FnSUNBZ0lsQjFZbXhwWXlJNklDSmpiRVJyVVZZRMGIKICBEQnhWM
0U0ZUZKNFNsTnNOYXAwZVY5TmRYRnNXVE01WkUxak9VaGhlRkV3U1drNU5rMDBhVG
hGVldUUkNpQQogIGdlVzlWVDFwUk0ySXhZalF3VVdEJiM1U1U0hsQkluMTl
mU3dLSUNBZ0lDSkRiRmJF0YjI1QmRYUm9aCiAgVzUwYVdOaGRHbHZiaUk2SUhzS0lD
QWdJQ0FnSWxWa1ppSTZJQ0pOUVZZnekxVVTJWMUF0UWsxSlV5MUpXRkIKICBKTFUxW
lVGSXRUVFUyUXkxUFNWVXpJaXdLSUNBZ0lDQWdJbEIxWW14cFkxQmhjbUZ0WlhSbGNuTWlPaUI3Q2l
NuTWlPaUI3QwogIGlBZ0lDQWdJQ0FnSWxCMVlteHBZMHRsZVVWRFJFZ2lPaUI3Q2l
BZ0lDQWdJQ0FnSUNBaVkzSjJJam9nSWxnCiAgME5EZ2lMQW9nSUNBZ0lDQWdJQ0Fn
SWxCMVlteHBZeUk2SUNKd2FtZGpkbWxJUlU5eVlXNHlXbUZNYTJFNVoKICBtVm5ib
UZxTjNWME9VNVNNk5UTlVaSFdtbEdPREJ2U21VelJucFZlSFp6Q2lBZ2VFMXhkWF
JKTkZweE5XNQogIHpiVkF3YkRoRWExRlBVVWxCSW4xOWZTd0tJQ0FnSUNKUWNtOW1
hV3hsVTJsbmJtRjBkWEpsSWpvZ2V3b2dJCiAgQ0FnSUNBZ0lsVmtaaUk2SUNKTlFV
SXROVmMzTWkwelFsWldVzFKVDBWRExUWlZa0UKICBCcFRFFvZ
0lDQWdJQ0FpVUhWaWJHbGpVR0Z5WVcxbGRHVnljeUk2SUhzS0lDQWdJQ0FnSUNBaV
VIVmliR2xqUwogIDJWNVJVTkVTQ0k2SUhzS0lDQWdJQ0FnSUNBZ0lDSmpjbllpT2l
BaVJXUTBORGdpTEFvZ0lDQWdJQ0FnSUNBZ0lsQjFZbXhwWXlJNklDSTRNWE4z
Y0cwd05WUTViMng1Y1dKTlNFOHdaR0ZFVkZkU01ta3RVRXRXRHYUVodFEKICBuUkhka
lZ3VGtvd05tZZzJhMHRGTms1VkNpQWdNR0pEVEhZMlUzazNjR0p1YzNkWGJJVVdnpla3
QwVTNGQkluMQogIDlmWDE5IiwKICAgICAgICAgIHsKICAgICAgICAgICAgInNpZ25
hdHVyZXMiOiBbewogICAgICAgICAgICAgICAgImFsZyI6ICJTNTEyIiwKICAgICAg
ICAgICAgICAgICJraWQiOiAiTURSSUi01VzcyLTNSSk8tVlpCMy1WVVZZRLUlPRUMtN
lVOQSIsCiAgICAgICAgICAgICAgICAic2lnbmF0dXJlIjogIlVOdHloSkZ1d0xQbW
o4dXVTdzZUczYxQUNvT2tFb0xGNjNyU2JIVDM1YkRSdVM4VkYKICBobmt5TlgybVE
0U0lHSHVCUFBTVVJaQjg0a0FHUmhxME1SQVIzMmpiVEpyNFdlM0xTeV9QZGVHaDVo
VmFHYgogIFJNVWhYMlY0MFNWenk3U3hMY0dZVzhpWHFYcTlQVllMM1MzMTVmQklBI
n1dLAogICAgICAgICAgICAiUGF5bG9hZERpZ2VzdCI6ICI2UDBHZnFXM2Jfa1loWX
```

```
  JXRzBlMG9YeTB1RU5Pcl9ZeHhjVTNDZ0xhTk8zdEwKICBlVG1Xa1VDR3RsWlVNdkV
  wdFR0Ti1Zc3U0S3FtWHI3T21waFgwM3FvdyJ9XSwKICAgICAgICAiUHJvdG9jb2xz
  IjogW3sKICAgICAgICAgICAgIlByb3RvY29sIjogIm1tbSJ9XX1dfX0",
              {
                "signatures":[{
                    "alg":"S512",
                    "kid":"MCDG-TS7T-UPDD-V667-OXSX-QJ5G-FQRZ",
                    "signature":"vNRYwmXv2J3oZ3FBsDkkGw7acTiVw-tV
  Kptb9jB3zrNYMBSuDXVwNi_OpdVZnTSViU0fnESrDFUAL7YuKMzwQth9aiTFqfFWx
  l9bq8c-6L0-T4fUxP03Z7F8Xh3dLHfPJgQMw6oMnIRmva1lsPetLzkA"}
                  ],
                "PayloadDigest":"qvRHyBm7El55dSLGleU8R-FWGZa1sEnb
  MoHtkFp4On8Z7dSNwnvmHiySY92jsmbKjeMd31gYdmeTHr915O0vLw"}
              ]}
          ]}}}}
```

The Contact catalog is typically used by the MeshService as a source of authorization information to perform access control on inbound and outbound message requests. For this reason, Mesh Service **SHOULD** be granted read access to the contacts catalog by providing a decryption entry for the service.

4.5. Credential

The credential catalog mmm_credential contains CatalogEntryCredential entries which describe credentials used to access network resources.

```
{
  "CatalogedCredential":{
    "Service":"ftp.example.com",
    "Username":"alice1",
    "Password":"password"}}
```

Only username/password credentials are stored in the credential catalog. If public key credentials are to be used, these **SHOULD** be managed as an application profile allowing separate credentials to be created for each device.

4.6. Device

The device catalog mmm_Device contains CatalogEntryDevice entries which describe the devices connected to the account and the permissions assigned to them.

Each device connected to a Mesh Account has an associated CatalogEntryDevice entry that includes the activation and connection records for the account. These records are described in further detail in section ???.

4.7. Network

The network catalog contains CatalogEntryNetwork entries which describe network settings, IPSEC and TLS VPN configurations, etc.

```
{
  "CatalogedNetwork":{
    "Service":"myWiFi",
    "Password":"securePassword"}}
```

4.8. Publication

[Note, this catalog is obsolete, the functions provided by this catalog are being merged with the Access catalog]

The publication catalog mmm_Publication contains CatalogEntryPublication entries which describe content published through the account.

If the data being published is small, it **MAY** be specified in the CatalogEntryPublication entry itself as enveloped data. Otherwise a link to the external content is required.

The Publication catalog is currently used to publish two types of data:

Contact Used in the Static QR Code Contact Exchange interaction.

Profile Device Used in the Preconfigured Device Connection interaction.

The interactions using this published data are described in [draft-hallambaker-mesh-protocol].

>>>> Unfinished SchemaEntryPublication

Missing example 11

4.9. Task

The Task catalog mmm_Task contains CatalogEntryTask entries which describe tasks assigned to the user including calendar entries and to do lists.

The fields of the task catalog currently reflect those offered by the iCalendar specification [RFC5545]. Specification of additional fields to allow task triggering on geographic location and/or completion of other tasks is a work in progress.

```
{
  "CatalogedTask":{
    "Title":"SomeItem",
    "Key":"NCA3-YB4P-SDYT-4YLO-NIBC-O5WN-JH32"}}
```

5. Spools

Spools are DARE Sequences containing an append only list of messages sent or received by an account. Three spools are currently defined:

Inbound Messages sent to the account. These are encrypted under the account encryption keys of the sender and receiver that were current at the time the message was sent.

Outbound Messages sent from the account. These are encrypted under the account encryption keys of the sender and receiver that were current at the time the message was sent.

Local Messages sent from the account for internal use. These are encrypted under the encryption key of the intended recipient alone. This is either the account administration encryption key or a device encryption key.

Every Mesh Message has a unique message identifier. Messages created at the beginning of a new messaging protocol interaction are assigned a random message identifier. Responses to previous messages are assigned message identifiers formed from the message identifier to which they respond by means of a message digest function.

Every Mesh Message stored in a spool is encapsulated in an envelope which bears a unique identifier that is formed by applying a message digest function to the message identifier. Each stored message has an associated state which is initially set to the state Initial and **MAY** be subsequently altered by one or more MessageComplete messages subsequently appended to the spool. The allowable message states depending upon the spool in question.

5.1. Outbound

The outbound spool stores messages that are to be or have been sent and MessageComplete messages reporting changes to the status of the messages stored on the spool.

Messages posted to the outbound spool have the state Initial, Sent, Received or Refused:

Initial The initial state of a message posted to the spool.

Sent The Mesh Service of the sender has delivered the message to the Mesh Service of the recipient which accepted it.

Received The Mesh Service of the sender has delivered the message to the Mesh Service of the recipient and the recipient has acknowledged receipt.

Refused The Mesh Service of the sender has delivered the message to the Mesh Service of the recipient which refused to accept it.

MessageComplete messages are only valid when posted to the spool by the service.

5.2. Inbound

The inbound spool stores messages that have been received by the Mesh service servicing the account and MessageComplete messages reporting changes to the status of the messages stored on the spool.

Messages posted to the outbound spool have the state Initial, Read:

Initial The initial state of a message posted to the spool.

Read The message has been read.

A message previously marked as read **MAY** be returned to the unread state by marking it as being in the Initial state.

5.3. Local

The local spool stores messages that are used for administrative functions. In normal circumstances, only administrator devices and the Mesh Service require access to the local spool.

The local spool is used to store MessagePin messages used to notify administration devices that a PIN code has been registered for some purpose and RespondConnection messages used to inform a device of the result of a connection request.

The local spool is used in a device connection operation to provide a device with the activation and connection records required to access the service as an authorized client. Servicing these requests requires that the service be able to access messages stored in the spool by envelope id.

Messages posted to the outbound spool have the states Initial, Closed:

Initial The initial state of a message posted to the spool.

Closed The action associated with the message has been completed.

Future: Redefining the role of the Local spool would allow the Claim/PollClaim operations used in device connection to be eliminated and greater consistency achieved between the device connection interactions.

5.4. Log

The log spo

6. Logs

The logging functions are not currently implemented.

Logs are records of events. Mesh logs **SHOULD** be encrypted and notarized.

The following logs are specified:

Service A log written by the Mesh Service containing a list of all actions performed on the account

Exception A log written by the Mesh Service containing a list of all exception events such as requests for access that were refused.

Notary A log written by administration devices connected to the account containing a sequence of status entries and cross notarization receipts.

The notary log will perform a particularly important role in future Mesh versions as it provides the ultimate root of trust for the account itself through cross notarization with the account holder's MSP which in turn achieves mutual cross notarization with every other MSP by cross notarizing with the Callsign registry. Thus every Mesh user is cross notarized with every other Mesh user making use of the Callsign registry through a graph with a diameter of 4.

7. Cryptographic Operations

The Mesh makes use of various cryptographic operations including threshold operations. For convenience, these are gathered here and specified as functions that are referenced by other parts of the specification.

7.1. Key Derivation from Seed

Mesh Keys that derived from a seed value use the mechanism described in [draft-hallambaker-mesh-udf]. Use of the keyname parameter allows multiple keys for different uses to be derived from a single key. Thus escrow of a single seed value permits recovery of all the private keys associated with the profile.

The keyname parameter is a string formed by concatenating identifiers specifying the key type, the actor that will use the key and the key operation:

7.2. Message Envelope and Response Identifiers.

Every Mesh message has a unique Message Identifier MessageId. The MakeID() function is used to calculate the value of Envelope Identifier and Response identifier from the message identifier as follows:

```
static string MakeID(string udf, string content) {
    var (code, bds) = UDF.Parse(udf);
    return code switch
        {
            UdfTypeIdentifier.Digest_SHA_3_512 =>
                UDF.ContentDigestOfDataString(
                bds, content, cryptoAlgorithmId:
                    CryptoAlgorithmId.SHA_3_512),
            _ => UDF.ContentDigestOfDataString(
            bds, content, cryptoAlgorithmId:
                    CryptoAlgorithmId.SHA_2_512),
            };
```

Where the values of content are given as follows:

application/mmm/envelopeid The proposed IANA content identifier for the Mesh message type.

application/mmm/responseid The proposed IANA content identifier for the Mesh message type.

For example:

```
MessageID
    = NBKV-TDNI-KV6R-O6U6-B4UI-3INK-AAFG

EnvelopeID
    = MCRT-4U7E-2EFA-6GT6-ATYZ-NKLT-DNU6

ResponseID
    = MBHI-EYTN-YPH2-U3AN-UO52-SGWT-CO57
```

7.3. Proof of Knowledge of PIN

Mesh Message classes that are subclasses of MessagePinValidated **MAY** be authenticated by means of a PIN. Currently two such messages are defined: MessageContact used in contact exchange and RequestConnection message used in device connection.

The PIN codes used to authenticate MessagePinValidated messages are UDF Authenticator strings. The type code of the identifier specifies the algorithm to be used to authenticate the PIN code and the Binary Data Sequence value specifies the key.

The inputs to the PIN proof of knowledge functions are:

PIN: string A UDF Authenticator. The type code of the identifier specifies the algorithm to be used to authenticate the PIN code and the Binary Data Sequence value specifies the key.

Action: string A code determining the specific action that the PIN code **MAY** be used to authenticate. By convention this is the name of the Mesh message type used to perform the action.

Account: string The account for which the PIN code is issued.

ClientNonce: binary Nonce value generated by the client using the PIN code to authenticate its message.

PayloadDigest: binary The PayloadDigest of a DARE Envelope that contains the message to be authenticated. Note that if the envelope is encrypted, this value is calculated over the ciphertext and does not provide proof of knowledge of the plaintext.

The following values of Action are currently defined:

Device Action info for device PIN

Contact Action info for contact PIN

These inputs are used to derive values as follows:

```
alg =           UdfAlg (PIN)
pinData =       UdfBDS (PIN)
saltedPINData = MAC (Action, pinData)
saltedPIN =     UDFPresent (HMAC_SHA_2_512 + saltedPINData)
PinId =         UDFPresent (MAC (Account, saltedPINData))
```

The issuer of the PIN code stores the value saltedPIN for retrieval using the key PinId.

The witness value for a Dare Envelope with payload digest PayloadDigest authenticated by a PIN code whose salted value is saltedPINData, issued by account Account is given by PinWitness() as follows:

```
witnessData =   Account.ToUTF8() + ClientNonce + PayloadDigest
witnessValue =  MAC (witnessData , saltedPINData)
```

For example, to generate saltedPIN for the pin AAIT-WXRD-BVB7-3BBT-D6JS-44GE-B4 used to authenticate a an action of type Device:

```
pin = AAIT-WXRD-BVB7-3BBT-D6JS-44GE-B4
action = message.

alg = UdfAlg (PIN)
    = Authenticator_HMAC_SHA_2_512

hashalg = default (alg, HMAC_SHA_2_512)

pinData = UdfBDS (PIN)
    = System.Byte[]

saltedPINData
    = hashalg(pinData, hashalg);
    = System.Byte[]

saltedPIN = UDFPresent (hashalg + saltedPINData)
    = ADGS-TMEV-G2MR-2NPD-ZJO3-NH2F-363W
```

The PinId binding the pin to the account alice@example.com is

```
Account =  alice@example.com

PinId = UDFPresent (MAC (Account, saltedPINData))
    = AD3I-LNZ6-JCHV-UYO6-JDRO-GPQG-R2VC
```

Where MAC(data, key) is the message authentication code algorithm specified by the value of alg.

When an administrative device issues a PIN code, a Message PIN is appended to the local spool. This has the MessageId PinId and specifies the value saltedPIN in the field of that name.

When PIN code authentication is used, a message of type MessagePinValidated specifies the values ClientNonce, PinWitness and PinId in the fields of those names. These values are used to authenticate the inner message data specified by the AuthenticatedData field.

7.4. EARL

The UDF Encrypted Authenticated Resource Locator mechanism is used to publish data and provide means of authentication and access through a static identifier such as a QR code.

This mechanism is used to allow contact exchange by means of a QR code printed on a business card and to connect a device to an account using a static identifier printed on the device in the form of a QR code.

In both cases, the information is passed using the EARL format described in [draft-hallambaker-mesh-udf].

8. Mesh Assertions

Mesh Assertions are signed DARE Envelopes that contain one of more claims. Mesh Assertions provide the basis for trust in the Mathematical Mesh.

Mesh Assertions are divided into two classes. Mesh Profiles are self-signed assertions. Assertions that are not self-signed are called declarations. The only type of declaration currently defined is a Connection Declaration describing the connection of a device to an account.

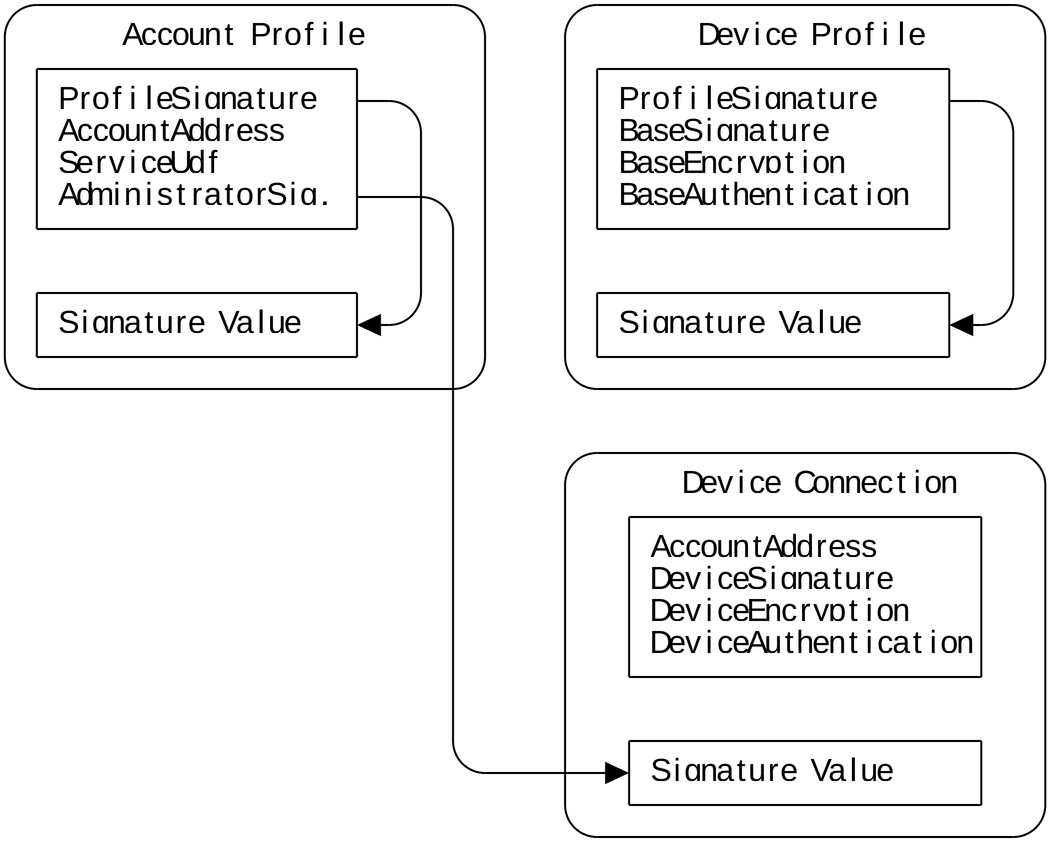

Figure 1: Profiles And Connections

8.1. Encoding

The payload of a Mesh Assertion is a JSON encoded object that is a subclass of the Assertion class which defines the following fields:

Identifier An identifier for the assertion.

Updated The date and time at which the assertion was issued or last updated

NotaryToken An assertion may optionally contain one or more notary tokens issued by a Mesh Notary service. These establish a proof that the assertion was signed after the date the notary token was created.

Conditions A list of conditions that **MAY** be used to verify the status of the assertion if the relying party requires.

The implementation of the NotaryToken and Conditions mechanisms is to be specified in [draft-hallambaker-mesh-callsign] at a future date.

Note that the implementation of Conditions differs significantly from that of SAML. Relying parties are required to process condition clauses in a SAML assertion to determine validity. Mesh Relying parties **MAY** verify the conditions clauses or rely on the trustworthiness of the provider.

The reason for weakening the processing of conditions clauses in the Mesh is that it is only ever possible to validate a conditions clause of any type relative to a ground truth. In SAML applications, the relying party almost invariably has access to an independent source of ground truth. A Mesh device connected to a Mesh Service does not. Thus the types of verification that can be achieved in practice are limited to verifying the consistency of current and previous statements from the Mesh Service.

8.2. Mesh Profiles

Mesh Profiles perform a similar role to X.509v3 certificates but with important differences:

- Profiles describe credentials, they do not make identity statements

- Profiles do not expire, there is therefore no need to support renewal processing.

- Profiles may be modified over time, the current and past status of a profile being recorded in an append only log.

Profiles provide the axioms of trust for the Mesh PKI. Unlike in the PKIX model in which all trust flows from axioms of trust held by a small number of Certificate Authorities, every part in the Mesh contributes their own axiom of trust.

It should be noted however that the role of Certificate Authorities is redefined rather than eliminated. Rather than making assertions whose subject is represented by identities which are inherently mutable and subjective, Certificate Authorities can now make assertions about immutable cryptographic keys.

Every Profile **MUST** contain a SignatureKey field and **MUST** be signed by the key specified in that field.

A Profile is valid if and only if:

- There is a SignatureKey field.

- The profile is signed under the key specified in the SignatureKey field.

A profile has the status current if and only if:

- The Profile is valid
- Every Conditions clause in the profile is understood by the relying party and evaluates to true.

8.3. Mesh Connections

A Mesh connection is an assertion describing the connection of a device or a member to an account.

Mesh connections provide similar functionality to 'end-entity' certificates in PKIX but with the important proviso that they are only used to provide trust between a device connected to an account and the service to which that account is bound and between the devices connected to an account.

A connection is valid with respect to an account with profile *P* if and only if:

- The profile *P* is valid
- The AuthorityUdf field of the connection is consistent with the UDF of *P*
- The profile is signed under the key specified in the AdministrationKey field of *P*.
- Any conditions specified in the profile are met

A connection has the status current with respect to an account with profile if and only if:

- The connection is valid with respect to the account with profile *P*.
- The profile P is current.

A device is authenticated with respect to an account with profile P if and only if:

- The connection is valid with respect to the account with profile *P*.
- The device has presented an appropriate proof of knowledge of the DeviceAuthentication key specified in the connection.

8.4. Device Pre-configuration

The DevicePreconfiguration record provides a means of bundling all the information used to preconfigure a device for use in the Mesh. This comprises:

- The Enveloped ProfileDevice.

- A ConnectionDevice assertion credentialing the device to the configuration provider Mesh Service.

- A ConnectionService assertion credentialing the device to the configuration provider Mesh Service.

- The secret seed used to create the ProfileDevice data.

The DevicePreconfiguration record **MAY** be used as the means of preconfiguring devices to allow connection to a user's account profile using the Preconfigured/Static QR Code device connection interaction.

For example, Alice's coffee pot was preconfigured for connection to a Mesh account at the factory and the following DevicePreconfiguration record created:

```
{
 "DevicePreconfigurationPrivate":{
  "EnvelopedConnectionDevice":[{
    "dig":"S512",
    "ContentMetaData":"ewogICJNZXNzYWdlVHlwZSI6ICJDb25uZWN0aW
 9uRGV2aWNlIiwKICAiY3R5IjogImFwcGxpY2F0aW9uL21tbS9vYmplY3QiLAogICJ
 DcmVhdGVkIjogIjIwMjItMTAtMThUMTI6NDg6MTdaIn0"},
   "ewogICJDb25uZWN0aW9uRGV2aWNlIjogewogICAgIlNpZ25hdHVyZSI6IH
 sKICAgICAgIlVkZiI6ICJNQkZPLVdNN0stSTdDNy1ZUVNVLUNJVVotSlFFUC1USDR
 RIiwKICAgICAgIlB1YmxpY1BhcmFtZXRlcnMiOiB7CiAgICAgICAgIlB1YmxpY0tl
 eUVDREgiOiB7CiAgICAgICAgICAiY3J2IjogIkVkNDQ4IiwKICAgICAgICAgICJQd
 WJsaWMiOiAiWXcwcnhJMWZZTlpnRFA1ZGwtTkRPZ05GVF9TLVVaUGpLYTBvWWJvRE
 xUakpRTXFWVWFJOQogIGYzMFRDS29FX2NEcUd4ck11VUlYUTRvQSJ9fX0sCiAgICA
 iRW5jcnlwdGlvbiI6IHsKICAgICAgIlVkZiI6ICJNQjYzLU1BR04tRVNOVy1OR0tI
 LUZQWFotSVNHTC1PTE9XIiwKICAgICAgIlB1YmxpY1BhcmFtZXRlcnMiOiB7CiAgI
 CAgICAgIlB1YmxpY0tleUVDREgiOiB7CiAgICAgICAgICAiY3J2IjogIlg0NDgiLA
 ogICAgICAgICAgIlB1YmxpYyI6ICJfNjZDY1VSdThhYTlFXT3hXZm5fTVdkRVVXMmx
 paVRXVUtNeHVla3R1ZWoxOEltbnlXb1NJCiAgNkp6UGMwVWpYZWRKZUhnNm5IazhO
 bUtBIn19fSwKICAgICJBdXRoZW50aWNhdGlvbiI6IHsKICAgICAgIlVkZiI6ICJNQ
 jYzLU1BR04tRVNOVy1OR0tILUZQWFotSVNHTC1PTE9XIiwKICAgICAgIlB1YmxpY1
 BhcmFtZXRlcnMiOiB7CiAgICAgICAgIlB1YmxpY0tleUVDREgiOiB7CiAgICAgICA
 gICAiY3J2IjogIlg0NDgiLAogICAgICAgICAgIlB1YmxpYyI6ICJfNjZDY1VSdThhY
 TlFXT3hXZm5fTVdkRVVXMmxpaVRXVUtNeHVla3R1ZWoxOEltbnlXb1NJCiAgNkp6U
 GMwVWpYZWRKZUhnNm5IazhObUtBIn19fX19",
   {
    "signatures":[{
      "alg":"S512",
      "kid":"MCBJ-UITH-2BQD-PX3A-SR3Z-S4UV-BNWK",
      "signature":"TpL0FOcO64HC2B13c-uQrBqlZtXFzPxvsznY9sb_
 sKosFnrjmlhBQNR55A58DgxRiinXtHTnOqqAZAHcnDVcdgnAQV9qY9znPNzsDVmjN
 3EmXr9R1fNtJU_vhLzJKk6jQc1Wp5GCygtwSQNRsaTjFjQA"}
     ],
    "PayloadDigest":"eajU4hdXOEvO8gdTYhwG33txVBGqZFp2PyD4WtE5
 mCRi2ZZ5w0K5r6HciY6zlqas4-6-dxb5XMAQ3S3gcYJtNg"}
   ],
  "EnvelopedConnectionService":[{
    "dig":"S512",
    "ContentMetaData":"ewogICJNZXNzYWdlVHlwZSI6ICJDb25uZWN0aW
 9uU2VydmljZSIsCiAgImN0eSI6ICJhcHBsaWNhdGlvbi9tbW0vb2JqZWN0IiwKICA
 iQ3JlYXRlZCI6ICIyMDIyLTEwLTE4VDEyOjQ4OjE3WiJ9"},
   "ewogICJDb25uZWN0aW9uU2VydmljZSI6IHsKICAgICJBdXRoZW50aWNhdG
 lvbiI6IHsKICAgICAgIlVkZiI6ICJNQjYzLU1BR04tRVNOVy1OR0tILUZQWFotSVN
 HTC1PTE9XIiwKICAgICAgIlB1YmxpY1BhcmFtZXRlcnMiOiB7CiAgICAgICAgIlB1
 YmxpY0tleUVDREgiOiB7CiAgICAgICAgICAiY3J2IjogIlg0NDgiLAogICAgICAgI
 CAgIlB1YmxpYyI6ICJfNjZDY1VSdThhYTlFXT3hXZm5fTVdkRVVXMmxpaVRXVUtNeH
 Vla3R1ZWoxOEltbnlXb1NJCiAgNkp6UGMwVWpYZWRKZUhnNm5IazhObUtBIn19fX1
 9",
   {
    "signatures":[{
            "alg":"S512",
            "kid":"MCBJ-UITH-2BQD-PX3A-SR3Z-S4UV-BNWK",
            "signature":"nsQ8vwj0eO4OgnmHKe1IDjmB_yW9vJFl7eXWVVcI
  Q5aHBGEUiVtqHbcnED3VNWZDwUYb3KavpuSAcdy8rgGRQVXtrDbT59EQupuwx2sKA
  Nx4ifkwM4z1_FmJdv4QJxGM0Zoh0Qcx5omEGnLxJCyjPAEA"}
          ],
        "PayloadDigest":"v7-o_VKzsUxg2rb3_mg9MTRA8-_9C-0ZJLv2SzZn
  0j2FIGl28RV4TXpDPieXTXBnHAtjrJePIxWM_tQKEHmz9g"}
      ],
    "PrivateKey":{
      "PrivateKeyUDF":{
        "PrivateValue":"ZAAQ-BVPA-BOCZ-6SIX-ZZP3-GP3R-ETLG-BBKB-2YD
M-WPNI-5RXJ-CVG2-4G5Z-GUCM",
        "KeyType":"MeshProfileDevice"}},
    "ConnectUri":"mcu://maker@example.com/ED6B-KIW3-TSCC-P4LM-4D3I-
IAPD-LE",
    "EnvelopedProfileDevice":[{
        "EnvelopeId":"MALQ-6D3Y-ERRF-TIFW-36LR-6GJK-4OZI",
        "dig":"S512",
        "ContentMetaData":"ewogICJVbmlxdWVJZCI6ICJNQUxRLTZEM1ktRV
  JSRi1USUZXLTM2TFItNkdKSy00T1pJIiwKICAiTWVzc2FnZVR5cGUiOiAiUHJvZml
  sZURldmljZSIsCiAgImN0eSI6ICJhcHBsaWNhdGlvbi9tbW0vb2JqZWN0IiwKICAi
  Q3JlYXRlZCI6ICIyMDIyLTEwLTE4VDEyOjQ4OjE3WiJ9"},
      "ewogICJQcm9maWxlRGV2aWNlIjogewogICAgIkVuY3J5cHRpb24iOiB7Ci
  AgICAgICJVZGYiOiAiTUI2My1NQUdOLUVTTlctTkdLSC1GUFhaLUlTR0wtT0xPVyI
  sCiAgICAgICJQdWJsaWNQYXJhbWV0ZXJzIjogewogICAgICAgICJQdWJsaWNLZXlF
  Q0RIIjogewogICAgICAgICAgImNydiI6ICJYNDQ4IiwKICAgICAgICAgICJQdWJsa
  WMiOiAiXzY2Q2NVUnU4WE5RV094V2ZuX01XZEVVVzJsaWlUV1VLTXh1ZWt0dWVqMT
  hJbW55V29TSQogIDZKelBjMFVqWGVkSmVIZzZuSGs4Tm1LQSJ9fX0sCiAgICAiU2l
  nbmF0dXJlIjogewogICAgICAiVWRmIjogIk1CRk8tV003Sy1JN0M3LVlRU1UtQ0lV
  Wi1KUUVQLVRINFEiLAogICAgICAiUHVibGljUGFyYW1ldGVycyI6IHsKICAgICAgI
  CAiUHVibGljS2V5RUNESCI6IHsKICAgICAgICAgICJjcnYiOiAiRWQ0NDgiLAogIC
  AgICAgICAgIlB1YmxpYyI6ICJZdzByeEkxZllOWmdEUDVkbC1ORE9nTkZUX1MtVVp
  QakthMG9ZYm9ETFRqSlFNcVZVYUk5CiAgZjMwVENLb0VfY0RxR3hyTXVVSVhRNG9B
  In19fSwKICAgICJBdXRoZW50aWNhdGlvbiI6IHsKICAgICAgIlVkZiI6ICJNQUNEL
  TNSSUItRjJMTy1HTFhKLVJJT1MtNzJaVC1FVVA0IiwKICAgICAgIlB1YmxpY1Bhcm
  FtZXRlcnMiOiB7CiAgICAgICAgIlB1YmxpY0tleUVDREgiOiB7CiAgICAgICAgICA
  iY3J2IjogIlg0NDgiLAogICAgICAgICAgIlB1YmxpYyI6ICJHNXQ0OEVHYnJTbWU5
  YWMxSGhseHFzaUYyemVRN2pmcV8tZkI1a0wxam1ac0NxN1ZmS2VKCiAgNjM3eHVwb
  ENjOFlFMEp2V2R6RFlCR0tBIn19fSwKICAgICJQcm9maWxlU2lnbmF0dXJlIjogew
  ogICAgICAiVWRmIjogIk1BTFEtNkQzWS1FUlJGLVRJRlctMzZMUi02R0pLLTRPWkk
  iLAogICAgICAiUHVibGljUGFyYW1ldGVycyI6IHsKICAgICAgICAiUHVibGljS2V5
  RUNESCI6IHsKICAgICAgICAgICJjcnYiOiAiRWQ0NDgiLAogICAgICAgICAgIlB1Y
  mxpYyI6ICIxVUp6VzBBc1hMbkd4UjhqVGozUzM3VUtIQVRSdmlLWnpvUXJwWEZ6eS
  0tdUctaGwyUUlvCiAgaWRZck1kYm1zZ1MzWlNCSkRpSXRsQTRBIn19fX19",
      {
        "signatures":[{
            "alg":"S512",
            "kid":"MALQ-6D3Y-ERRF-TIFW-36LR-6GJK-4OZI",
          "signature":"nM9Y8MDljAp7Bms8jCNdgpZqpC-Q7uVBH6EfiNf7
dH4zAJ8g3ee24DDpWGGkaIUYTjixCqyH_8uAxNLMwWhRGzmipnwEUy20UmrjMBjqI
hu2TshN1yrC5VtftF-AK5JEg0dnJsZuIuT4bro50ON7OAMA"}
        ],
      "PayloadDigest":"jLTOUGaU-Y26uQ6Xczvc-ycCrD-4vfT3Ud0RLH35
b2hm1dvcF2Iy-F4A9Jx8u3OPSjkQ1WePfCDfw4hUzQUcHQ"}
    ]}}
```

The use of the publication mechanism in device connection is discussed further in [draft-hallambaker-mesh-protocol].

9. Architecture

The Mesh architecture has four principal components:

Mesh Account A collection of information (contacts, calendar entries, inbound and outbound messages, etc.) belonging to a user who uses the Mesh to management.

Mesh Device Management The various functions that manage binding of devices to a Mesh to grant access to information and services bound to that account.

Mesh Service Provides network services through which devices and other Mesh users may interact with a Mesh Account.

Mesh Messaging An end-to-end secure messaging service that allows short messages (less than 32KB) to be exchanged between Mesh Accounts and between the Mesh devices connected to a particular account.

The separation of accounts and services as separate components is a key distinction between the Mesh and earlier Internet applications. A Mesh account belongs to the owner of the Mesh and not the Mesh Service Provider which the user may change at any time of their choosing.

A Mesh Account May be active or inactive. By definition, an active Mesh account is serviced by exactly one Mesh Service, an inactive Mesh account is not serviced by a Mesh Service. A Mesh Service Provider **MAY** offer a backup service for accounts hosted by other providers. In this case the backup provider is connected to the account as a Mesh device, thus allowing the backup provider to maintain a copy of the stores contained in the account and facilitating a rapid transfer of responsibility for servicing the account should that be desired. The use of backup providers is described further in [draft-hallambaker-mesh-discovery].

9.1. Mesh Account

Mesh Accounts contains all the stateful information (contacts, calendar entries, inbound and outbound messages, etc.) related to a particular persona used by the owner.

By definition a Mesh Account is active if it is serviced by a Mesh Service and inactive otherwise. A Mesh user **MAY** change their service provider at any time. An active Mesh Account is serviced by exactly one Mesh Service at once but a user **MAY** register a 'backup' service provider to their account in the same manner as adding an advice. This ensures that the backup service is pre-populated with all the information required to allow the user to switch to the new provider without interruption of service.

Each Mesh account is described by an Account Profile. Currently separate profile Account Profile are defined for user accounts and group accounts. It is not clear if this distinction is a useful one.

9.1.1. Account Profile

A Mesh account profile provides the axiom of trust for a mesh user. It contains a Master Signature Key and one or more Administration Signature Keys. The unique identifier of the master profile is the UDF of the Master Signature Key.

An Account Profile **MUST** specify an EscrowEncryption key. This key **MAY** be used to escrow private keys used for encryption of stored data. They **SHOULD NOT** be used to escrow authentication keys and **MUST NOT** be used to escrow signature keys.

A user should not need to replace their account profile unless they intend to establish a separate identity. To minimize the risk of disclosure, the Profile Signature Key is only ever used to sign updates to the account profile itself. This allows the user to secure their Profile Signature Key by either keeping it on hardware token or device dedicated to that purpose or by using the escrow mechanism and paper recovery keys as described in this document.

9.1.1.1. Creating a ProfileMaster

Creating a ProfileMaster comprises the steps of:

0. Creating a Master Signature key.

1. Creating an Online Signing Key

2. Signing the ProfileMaster using the Master Signature Key

3. Persisting the ProfileMaster on the administration device to the CatalogHost.

4. (Optional) Connecting at least one Administration Device and granting it the ActivationAdministration activation.

9.1.1.2. Updating a ProfileMaster

Updating a ProfileMaster comprises the steps of:

0. Making the necessary changes.

1. Signing the ProfileMaster using the Master Signature Key

2. Persisting the ProfileMaster on the administration device to the CatalogHost.

9.2. Device Management

Device management allows a collection of devices belonging to a user to function as a single personal Mesh. Two catalogs are used to manage this process:

- The Access catalog is used to instruct the Mesh Service how to respond to requests from the device.

- The Device catalog records information for use by administration devices managing the device.

9.2.1. The Device Catalog

Each Mesh Account has a Device Catalog CatalogDevice associated with it. The Device Catalog is used to manage the connection of devices to the Personal Mesh and has a CatalogEntryDevice for each device currently connected to the catalog.

Each Administration Device **MUST** have access to an up-to-date copy of the Device Catalog in order to manage the devices connected to the Mesh. The Mesh Service protocol **MAY** be used to synchronize the Device Catalog between administration devices in the case that there is more than one administration device.

The CatalogEntryDevice contains fields for the device profile, device private and device connection.

9.2.2. Mesh Devices

The principle of radical distrust requires us to consider the possibility that a device might be compromised during manufacture. Once consequence of this possibility is that when an administration device connects a new device to a user's personal Mesh, we cannot put our full trust in either the device being connected or the administration device connecting it.

This concern is resolved by (at minimum) combining keying material generated from both sources to create the keys to be used in the context of the user's personal Mesh with the process being fully verified by both parties.

Additional keying material sources could be added if protection against the possibility of compromise at both devices was required but this is not supported by the current specifications.

A device profile provides the axiom of trust and the key contributions of the device. When bound to an account, the base keys specified in the Device Profile are combined with the key data provided in the Activation device to construct the keys the device will use in the context of the account.

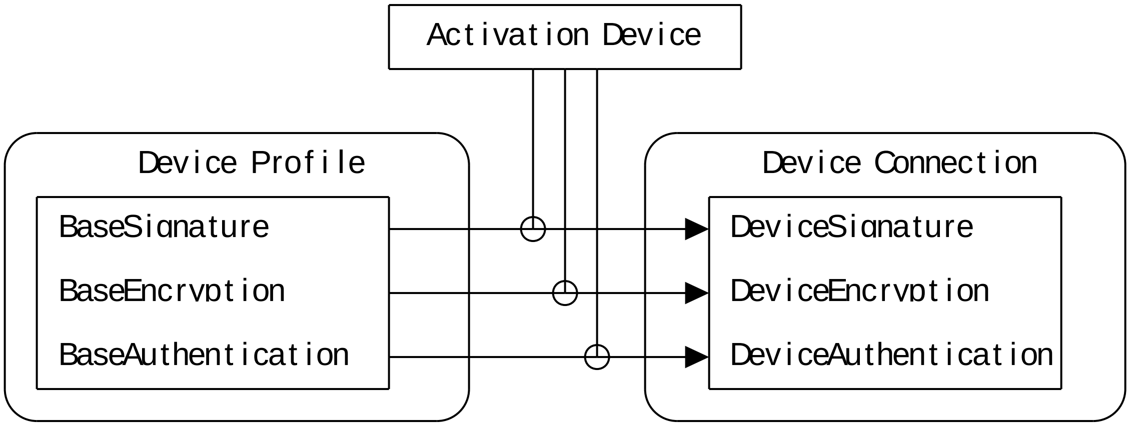

Figure 2: Mapping of Device Profile and Device Private to Device Connection Keys.

Unless exceptional circumstances require, a device should not require more than one Device profile even if the device supports use by multiple users under different accounts. But a device **MAY** have multiple profiles if this approach is more convenient for implementation.

9.2.2.1. Creating a ProfileDevice

Creating a ProfileDevice comprises the steps of:

0. Creating the necessary key

1. Signing the ProfileDevice using the Master Signature Key

2. Once created, a ProfileDevice is never changed. In the unlikely event that any modification is required, a completely new ProfileDevice **MUST** be created.

9.2.2.2. Connection to a Meh Account

Devices are only connected to a personal Mesh by an administration device. This comprises the steps of:

0. Generating the PrivateDevice keys.

1. Creating the ConnectionDevice data from the public components of the ProfileDevice and PrivateDevice keys and signing it using the administration key.

2. Creating the Activations for the device and signing them using the administration key.

3. Creating the CatalogEntryDevice for the device and adding it to the CatalogDevice of the account.

4. Creating an AccessCapability granting the necessary access rights for the device and adding that to the CatalogAccess of the account.

These steps are usually performed through use of the Mesh Protocol Connection mechanism. However, Mesh clients **MAY** support additional mechanisms as circumstances require provided that the appropriate authentication and private key protection controls are provided.

9.3. Mesh Services

A Mesh Service provides one or more Mesh Hosts that support Mesh Accounts through the Mesh Web Service Protocol.

Mesh Services and Hosts are described by Service Profiles and Host Profiles. The means by which services manage the hosts through which they provide service is outside the scope of this document.

As with a Device connected to a Mesh Account, a the binding of a Host to the service it supports is described by a connection record:

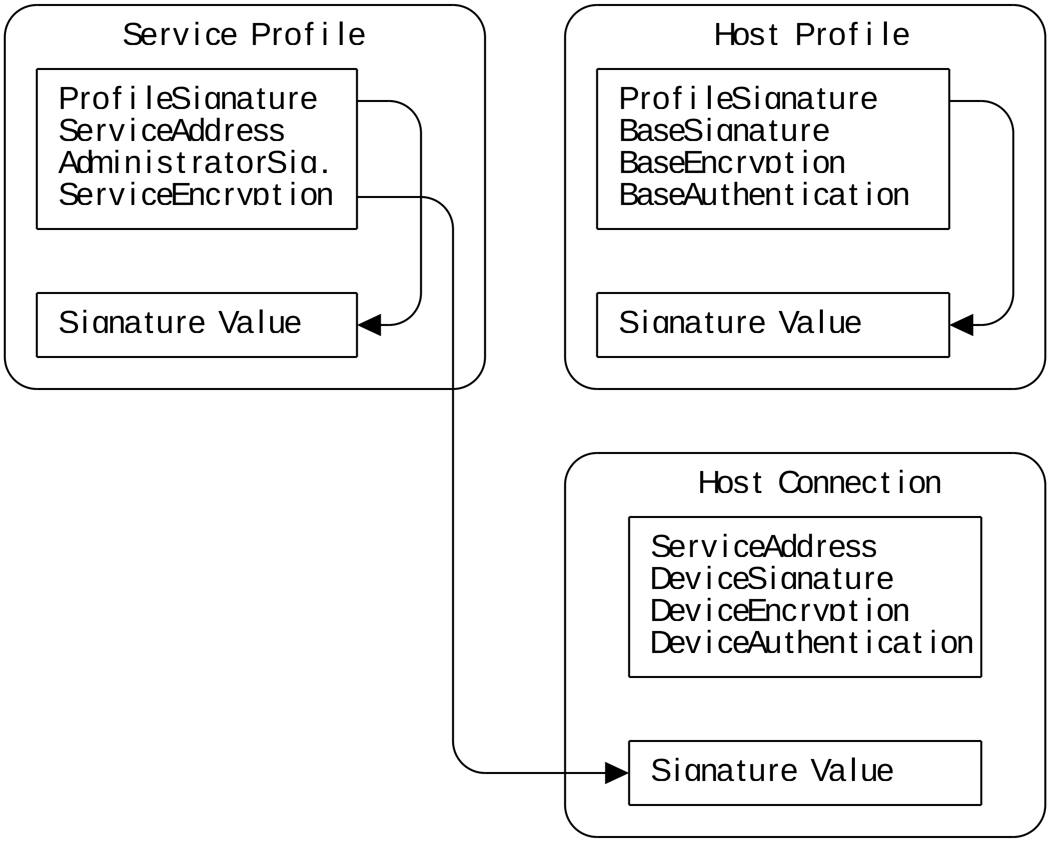

Figure 3: Service Profile and Delegated Host Assertion.

The credentials provided by the ProfileService and ProfileHost are distinct from those provided by the WebPKI that typically services TLS requests. WebPKI credentials provide service introduction and authentication while a Mesh ProfileHost only provides authentication.

Unless exceptional circumstances require, a service should not need to revise its Service Profile unless it is intended to change its identity. Service Profiles **MAY** be countersigned by Trusted Third Parties to establish accountability.

9.4. Mesh Messaging

Mesh Messaging is an end-to-end secure messaging system used to exchange short (32KB) messages between Mesh devices and services. In cases where exchange of longer messages is required, Mesh Messaging **MAY** be used to provide a control plane to advise the intended message recipient(s) of the type of data being offered and the means of retrieval (e.g an EARL).

All communications between Mesh accounts takes the form of a Mesh Message carried in a Dare Envelope. Mesh Messages are stored in two spools associated with the account, the SpoolOutbound and the SpoolInbound containing the messages sent and received respectively.

This document only describes the representation of the messages within the message spool. The Mesh Service protocol by which the messages are exchanged between devices and services and between services is described in [draft-hallambaker-mesh-protocol].

9.4.1. Message Status

As previously described in section ###, every message stored in a spool has a specified state. The range of allowable states is defined by the message type. New message states **MAY** be defined for new message types as they are defined.

By default, messages are appended to a spool in the Initial state, but a spool entry **MAY** specify any state that is valid for that message type.

The state of a message is changed by appending a completion message to the spool as described in [draft-hallambaker-mesh-protocol].

Services **MAY** erase or redact messages in accordance with local site policy. Since messages are not removed from the spool on being marked deleted, they may be undeleted by marking them as read or unread. Marking a message deleted **MAY** make it more likely that the message will be removed if the sequence is subsequently purged.

9.4.2. Four Corner Model

A four-corner messaging model is enforced. Mesh Services only accept outbound messages from devices connected to accounts that it services. Inbound messages are only accepted from other Mesh Services. This model enables access control at both the outbound and inbound services

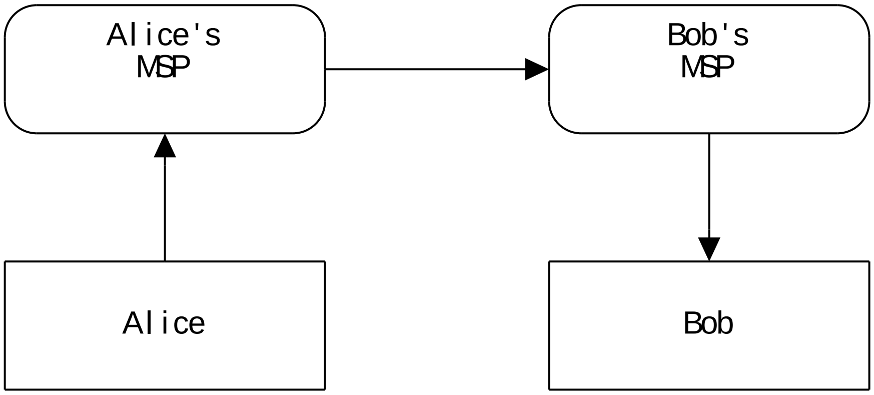

Figure 4: Four Corner Messaging Model

The outbound Mesh Service checks to see that the request to send a message does not violate its acceptable use policy. Accounts that make a large number of message requests that result in complaints **SHOULD** be subject to consequences ranging from restriction of the number and type of messages sent to suspending or terminating messaging privileges. Services that fail to implement appropriate controls are likely to be subject to sanctions from either their users or from other services.

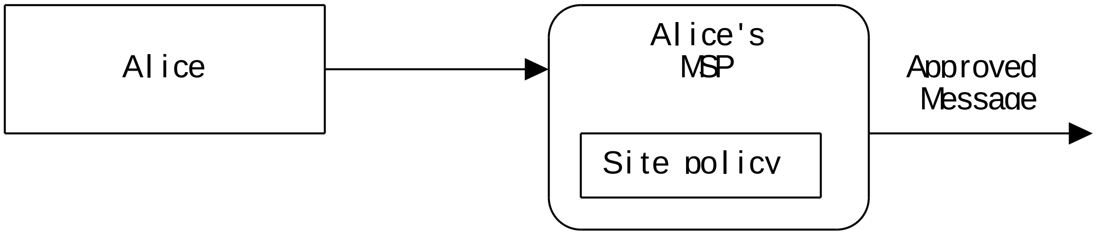

Figure 5: Performing Access Control on Outbound Messages

The inbound Mesh Service also checks to see that messages received are consistent with the service Acceptable Use Policy and the user's personal access control settings.

Mesh Services that fail to police abuse by their account holders **SHOULD** be subject to consequences in the same fashion as account holders.

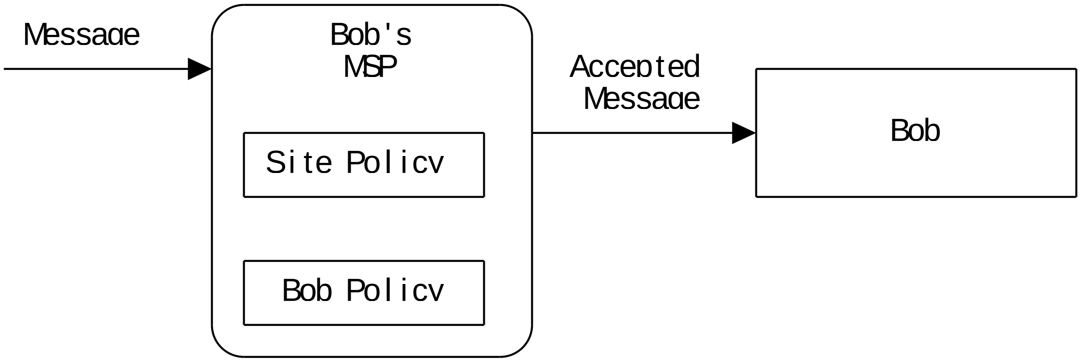

Figure 6: Performing Access Control on Inbound Messages

9.4.3. Traffic Analysis

The Mesh Messaging protocol as currently specified provides only limited protection against traffic analysis attacks. The use of TLS to encrypt communication between Mesh Services limits the effectiveness of na?ve traffic analysis mechanisms but does not prevent timing attacks unless dummy traffic is introduced to obfuscate traffic flows.

The limitation of the message size is in part intended to facilitate use of mechanisms capable of providing high levels of traffic analysis such as mixmaster and onion routing but the current Mesh Service Protocol does not provide support for such approaches and there are no immediate plans to do so.

10. Publications

Static QR codes **MAY** be used to allow contact exchange or device connection. In either case, the QR code contains an EARL providing the means of locating, decrypting and authenticating the published data.

The use of EARLs as a means of publishing encrypted data and the use of EARLs for location, decryption and authentication is discussed in [draft-hallambaker-mesh-dare] .

10.1. Profile Device

10.2. Contact Exchange

When used for contact exchange, the envelope payload is a CatalogedContact record.

Besides allowing for exchange of contact information on a business card, a user might have their contact information printed on personal property to facilitate return of lost property.

11. Schema

11.1. Shared Classes

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

11.1.1. Classes describing keys

11.1.2. Structure: KeyData

The KeyData 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.

NotBefore: DateTime (Optional) If present specifies a time instant that use of the private key is not valid before.

NotOnOrAfter: DateTime (Optional) If present specifies a time instant that use of the private key is not valid on or after.

11.1.3. Structure: KeyShare

Inherits: Key The identifier used to claim the capability from the
ServiceId: String (Optional) service.[Only present for a partial key.]

ServiceAddress: String (Optional) The service account that supports a serviced capability. [Only present for a partial key.]

11.1.4. Structure: CompositePrivate

Inherits: Key UDF fingerprint of the bound device key (if used).
DeviceKeyUdf: String (Optional)

11.2. Assertion classes

Classes that are derived from an assertion.

11.2.1. 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.

11.2.2. Structure: Condition

Parent class from which all condition classes are derived.

[No fields]

11.2.3. Base Classes

Abstract classes from which the Profile, Activation and Connection classes are derrived.

11.2.4. Structure: Activation

Inherits: Assertion

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

ActivationKey: String (Optional) Secret seed used to derive keys that are not explicitly specified.

Entries: ActivationEntry [0..Many] Activation of named account resource activations. These are separate from Application activations which are

11.2.5. Structure: ActivationEntry

Resource: String (Optional) Name of the activated resource

Key: KeyData (Optional) The activation key or key share

ServiceId: String (Optional) The identifier used to claim the capability from the service.[Only present for a partial capability.]

ServiceAddress: String (Optional) The service account that supports a serviced capability. [Only present for a partial capability.]

11.2.6. Mesh Profile Classes

Classes describing Mesh Profiles. All Profiles are Assertions derrived from Assertion.

11.2.7. Structure: Profile

Inherits: Assertion

Parent class from which all profile classes are derived

Description: String (Optional) Description of the profile

ProfileSignature: KeyData (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.

11.2.8. Structure: ProfileDevice

Inherits: Profile
Describes a mesh device.

Encryption: KeyData (Optional) Base key contribution for encryption keys. Also used to decrypt activation data sent to the device during connection to an account.

Signature: KeyData (Optional) Base key contribution for signature keys.

Authentication: KeyData (Optional) Base key contribution for authentication keys. Also used to authenticate the device during connection to an account.

11.2.9. Structure: ProfileAccount

Base class for the account profiles ProfileUser and ProfileGroup. These subclasses may be merged at some future date.

Inherits: Profile The account address. This is either a DNS service
AccountAddress: String (Optional) address (e.g. alice@example.com) or a Mesh Name (@alice).

ServiceUdf: String (Optional) The fingerprint of the service profile to which the account is currently bound.

EscrowEncryption: KeyData (Optional) Escrow key associated with the account.

AdministratorSignature: KeyData (Optional) Key used to sign connection assertions to the account.

CommonEncryption: KeyData (Optional) Key currently used to encrypt data under this profile

CommonAuthentication: KeyData (Optional) Key used to authenticate requests made under this user account. This key SHOULD NOT be provisioned to any device except for the purpose of enabling account recovery.

11.2.10. Structure: ProfileUser

Inherits: ProfileAccount
Account assertion. This is signed by the service hosting the account.

CommonSignature: KeyData (Optional) Key used to sign data under the account.

11.2.11. Structure: ProfileGroup

Inherits: ProfileAccount
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.

Cover: Binary (Optional) HTML document containing cover text to be presented if a document encrypted under the group key cannot be decrypted.

11.2.12. Structure: ProfileService

Inherits: Profile
Profile of a Mesh Service

ServiceAuthentication: KeyData (Optional) Key used to authenticate service connections.

ServiceEncryption: KeyData (Optional) Key used to encrypt data under this profile

ServiceSignature: KeyData (Optional) Key used to sign data under the account.

11.2.13. Structure: ProfileHost

Inherits: ProfileDevice
Profile of a Mesh Host providing one or more Mesh Services.

[No fields]

11.2.14. Connection Assertions

Connection assertions are used to authenticate and authorize interactions between devices and the service currently servicing the account. They SHOULD NOT be visible to external parties.

11.2.15. Structure: Connection

Inherits: Assertion UDF of the connection target.
Subject: String (Optional)
Authority: String (Optional) UDF of the connection source.

Authentication: KeyData (Optional) The authentication key for use of the device under the profile

11.2.16. Structure: CallsignBinding

Inherits: Assertion

Canonical: String (Optional) The canonical form of the callsign.

Display: String (Optional) The display form of the callsign. This MAY include characters such as whitespace, trademark signifiers, etc. that are omitted of trranslated in the canonical form.

ProfileUdf: String (Optional) The profile to which the name is bound.

Services: NamedService [0..Many] List of named services. If multiple service providers are specified for a given service, these are listed in order of priority, most preferred first.

11.2.17. Structure: Accreditation

Registration of a trusted third party accreditation of a callsign/profile binding.

Callsign: String (Optional) The callsign to which the accreditation applies

ProfileUdf: String (Optional) The profile to which the accreditation applies.

SubjectNames: String [0..Many] The validated names of the subject

SubjectLogos: String [0..Many] Mesh strong URIs from which a validated logo belonging to the subject MAY be retreived and validated.

Issued: DateTime (Optional) The time the assertion was issued.

Expires: DateTime (Optional) The time the assertion is due to expire

Policy: String (Optional) The issuing policy under which the validation was performed.

Practice: String (Optional) The issuing practices under which the validation was performed.

11.2.18. Structure: ConnectionStripped

Asserts that a profile is connected to an account address.

Inherits: Connection Stripped down connection assertion

Account: String (Optional) To be removed

11.2.19. Structure: ConnectionService

Inherits: Connection
Asserts that a device is connected to an account profile

ProfileUdf: String (Optional) The account address

Callsign: CatalogedCallsign (Optional) The account callsign

11.2.20. Structure: ConnectionDevice

Inherits: ConnectionService
Asserts that a device is connected to an account profile

Roles: String [0..Many] The signature key for use of the device
Signature: KeyData (Optional) under the profile

Encryption: KeyData (Optional) The encryption key for use of the device under the profile

11.2.21. Structure: ConnectionApplication

Inherits: Connection
Connection assertion stating that a particular device is

[No fields]

11.2.22. Structure: ConnectionGroup

Describes the connection of a member to a group.

Inherits: Connection
[No fields]

11.2.23. Structure: AccountHostAssignment

Inherits: Assertion The account being bound
AccountAddess: String (Optional)
HostAddresses: String [0..Many] Host address in Callsign, DNS or IP format in order of preference.

AccessEncrypt: KeyData (Optional) Encryption key to be used to encrypt data for the service to use.

11.2.24. Structure: ConnectionHost

Inherits: Connection
[No fields]

11.2.25. Activation Assertions

11.2.26. Structure: ActivationAccount

Contains activation data for device specific keys used in the context of a Mesh account.

Inherits: Activation The UDF of the account
AccountUdf: String (Optional)

11.2.27. Structure: ActivationHost

Contains activation data for device specific keys used in the context of a Mesh host

Inherits: ActivationAccount

[No fields]

11.2.28. Structure: ActivationCommon

Inherits: Activation Grant access to profile online signing key
ProfileSignature: KeyData (Optional) used to sign updates to the profile.

AdministratorSignature: KeyData (Optional) Grant access to Profile administration key used to make changes to administrator catalogs.

Encryption: KeyData (Optional) Grant access to ProfileUser account encryption key

Authentication: KeyData (Optional) Grant access to ProfileUser account authentication key

Signature: KeyData (Optional) Grant access to ProfileUser account signature key

11.2.29. Structure: ActivationApplication

Inherits: Activation

[No fields]

11.2.30. Structure: ActivationApplicationSsh

Inherits: ActivationApplication The SSH client key.
ClientKey: KeyData (Optional)

11.2.31. Structure: ActivationApplicationMail

Inherits: ActivationApplication The S/Mime signature key
SmimeSign: KeyData (Optional)
SmimeEncrypt: KeyData (Optional) The S/Mime encryption key

OpenpgpSign: KeyData (Optional)
The OpenPGP signature key

OpenpgpEncrypt: KeyData (Optional) The OpenPGP encryption key

11.2.32. Structure: ActivationApplicationGroup

Inherits: ActivationApplication Key or capability allowing account
AccountEncryption: KeyData (Optional) encryption keys to be created for new members.

AdministratorSignature: KeyData (Optional) Key or capability allowing account updates, connection assertions etc to be signed.

AccountAuthentication: KeyData (Optional) Key or capability allowing administration of the group.

EnvelopedConnectionService: Enveloped (Optional) Signed connection service delegation allowing the device to access the account.

11.2.33. Structure: ActivationApplicationCallsign

Inherits: ActivationApplication
[No fields]

11.3. Application Data

11.3.1. Structure: ApplicationEntry

Identifier: String (Optional)

11.3.2. Structure: ApplicationEntrySsh

Inherits: ApplicationEntry
EnvelopedActivation: Enveloped (Optional)

11.3.3. Structure: ApplicationEntryGroup

Inherits: ApplicationEntry
EnvelopedActivation: Enveloped (Optional)

11.3.4. Structure: ApplicationEntryMail

Inherits: ApplicationEntry
EnvelopedActivation: Enveloped (Optional)

11.3.5. Structure: ApplicationEntryCallsign

Inherits: ApplicationEntry
EnvelopedActivation: Enveloped (Optional)

11.4. Data Structures

Classes describing data used in cataloged data.

11.4.1. Structure: Contact

Inherits: Assertion

Base class for contact entries.

Id: String (Optional) The globally unique contact identifier.

Local: String (Optional) The local name.

Anchors: Anchor [0..Many] Mesh fingerprints associated with the contact.

NetworkAddresses: NetworkAddress [0..Many] Network address entries

Locations: Location [0..Many] The physical locations the contact is associated with.

Roles: Role [0..Many] The roles of the contact

Bookmark: Bookmark [0..Many] The Web sites and other online presences of the contact

Sources: TaggedSource [0..Many] Source(s) from which this contact was constructed.

11.4.2. Structure: Anchor

Trust anchor

Udf: String (Optional) The trust anchor.

Validation: String (Optional) The means of validation.

11.4.3. Structure: TaggedSource

Source from which contact information was obtained.

LocalName: String (Optional) Short name for the contact information.

Validation: String (Optional) The means of validation.

BinarySource: Binary (Optional) The contact data in binary form.

EnvelopedSource: Enveloped (Optional) The contact data in enveloped form. If present, the BinarySource property is ignored.

11.4.4. Structure: ContactGroup

Inherits: Contact

Contact for a group, including encryption groups.

[No fields]

11.4.5. Structure: ContactPerson

Inherits: Contact List of person names in order of preference

CommonNames: PersonName [0..Many]

11.4.6. Structure: ContactOrganization

Inherits: Contact List of person names in order of preference

CommonNames: OrganizationName [0..Many]

11.4.7. Structure: OrganizationName

The name of an organization

Inactive: Boolean (Optional) If true, the name is not in current use.

RegisteredName: String (Optional) The registered name.

DBA: String (Optional) Names that the organization uses including trading names and doing business as names.

11.4.8. Structure: PersonName

The name of a natural person

Inactive: Boolean (Optional) If true, the name is not in current use.

FullName: String (Optional) The preferred presentation of the full name.

Prefix: String (Optional) Honorific or title, E.g. Sir, Lord, Dr., Mr.

First: String (Optional) First name.

Middle: String [0..Many] Middle names or initials.

Last: String (Optional) Last name.

Suffix: String (Optional) Nominal suffix, e.g. Jr., III, etc.

PostNominal: String (Optional) Post nominal letters (if used).

11.4.9. Structure: NetworkAddress

Provides all means of contacting the individual according to a particular network address

Inactive: Boolean (Optional) If true, the name is not in current use.

Address: String (Optional) The network address, e.g. alice@example.com

NetworkCapability: String [0..Many] The capabilities bound to this address.

EnvelopedProfileAccount: Enveloped (Optional) The account profile

Protocols: NetworkProtocol [0..Many] Public keys associated with the network address

11.4.10. Structure: NetworkProtocol

Protocol: String (Optional) The IANA protocol|identifier of the network protocols by which the contact may be reached using the specified Address.

11.4.11. Structure: Role

OrganizationName: String (Optional) The organization at which the role is held

Titles: String [0..Many] The titles held with respect to that organization.

Locations: Location [0..Many] Postal or physical addresses associated with the role.

11.4.12. Structure: Location

Appartment: String (Optional)
Street: String (Optional)
District: String (Optional)
Locality: String (Optional)
County: String (Optional)
Postcode: String (Optional)
Country: String (Optional)

11.4.13. Structure: Bookmark

Uri: String (Optional)
Title: String (Optional)
Role: String [0..Many]

11.4.14. 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.

11.4.15. Structure: Engagement

Key: String (Optional) Unique key.

Start: DateTime (Optional)
Finish: DateTime (Optional)
StartTravel: String (Optional)
FinishTravel: String (Optional)
TimeZone: String (Optional)
Title: String (Optional)
Description: String (Optional)
Location: String (Optional)
Trigger: String [0..Many]
Conference: String [0..Many]
Repeat: String (Optional)
Busy: Boolean (Optional)

11.5. Catalog Entries

11.5.1. Structure: CatalogedEntry

Base class for cataloged Mesh data.

Labels: String [0..Many] The set of labels describing the entry

LocalName: String (Optional) User specified identifier.

Uid: String (Optional) Globaly unique identifier

11.5.2. Structure: CatalogedDevice

Inherits: CatalogedEntry

Public device entry, indexed under the device ID Hello

Updated: DateTime (Optional) Timestamp, allows

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

DeviceUdf: String (Optional) UDF of the offline signature key of the device

SignatureUdf: String (Optional) UDF of the account online signature key

EnvelopedProfileUser: Enveloped (Optional) The Mesh profile. Why is this still here? This is not specific to the device.

EnvelopedProfileDevice: Enveloped (Optional) The device profile

EnvelopedConnectionService: Enveloped (Optional) Slim version of ConnectionDevice used by the presentation layer

EnvelopedConnectionDevice: Enveloped (Optional) The public assertion demonstrating connection of the Device to the Mesh

EnvelopedActivationAccount: Enveloped (Optional) The activation of the device within the Mesh account

EnvelopedActivationCommon: Enveloped (Optional) The activation of the device within the Mesh account

11.5.3. Structure: CatalogedPublication

Inherits: CatalogedEntry

A publication.

Id: String (Optional) Unique identifier code

Authenticator: String (Optional) The witness key value to use to request access to the record.

EnvelopedData: DareEnvelope (Optional) Dare Envelope containing the entry data. The data type is specified by the envelope metadata.

NotOnOrAfter: DateTime (Optional) Epiration time (inclusive)

11.5.4. Structure: CatalogedCredential

Inherits: CatalogedEntry Specifies the client identification key
Protocol: String (Optional)
Service: String (Optional) Means of authenticating the host key
Username: String (Optional)
Password: String (Optional)
ClientAuthentication: KeyData [0..Many]
HostAuthentication: KeyData [0..Many]

11.5.5. Structure: CatalogedApplicationSsh

Inherits: CatalogedApplication The S/Mime encryption key
ClientKey: KeyData (Optional)

11.5.6. Structure: CatalogedNetwork

Inherits: CatalogedEntry
Protocol: String (Optional)
Service: String (Optional)

Username: String (Optional)
Password: String (Optional)

11.5.7. Structure: CatalogedContact

Inherits: CatalogedEntry Unique key.
Key: String (Optional)
Self: Boolean (Optional) If true, this catalog entry is for the user who created the catalog.

11.5.8. Structure: CatalogedAccess

Inherits: CatalogedEntry
[No fields]

11.5.9. Structure: Capability

Id: String (Optional) The identifier of the capability. If this is a cryptographic capability, this is the KeyIdentifier of the primary key that was shared. If this is an access capability, this is the KeyIdentifier of the authentication key being authorized for access.

Active: Boolean (Optional) The authentication mode: Device,
Issued: Integer (Optional) Account, PIN
Mode: String (Optional)
Udf: String (Optional) Identifies the authentication credential. For a device, this is the authentication key identifier, for an account, the profile identifier, for a PIN, the locator value of the PIN.

Witness: String (Optional) The verification value used to perform proof of knowledge of the secret.

11.5.10. Structure: NullCapability

Inherits: Capability
[No fields]

11.5.11. Structure: AccessCapability

Inherits: Capability Access rights associated with the key
Rights: String [0..Many]
EnvelopedCatalogedDevice: Enveloped (Optional) Digest value used to
CatalogedDeviceDigest: String (Optional) signal updates to envelope

11.5.12. Structure: PublicationCapability

Inherits: Capability Selector allowing a specific document to be
Identifier: String (Optional) requested.

Digest: String (Optional)
Document digest, this allows a status/ claim request to request an update to be returned only if the document has changed.

Data: Binary (Optional) The published document.

11.5.13. Structure: CryptographicCapability

Inherits: Capability The key that enables the capability
KeyData: KeyData (Optional)
GranteeAccount: String (Optional) One or more enveloped key shares.
GranteeUdf: String (Optional)
EnvelopedKeyShare: Enveloped (Optional)

11.5.14. Structure: CapabilityDecrypt

Inherits: CryptographicCapability
The corresponding key is a decryption key

[No fields]

11.5.15. Structure: CapabilityDecryptPartial

Inherits: CapabilityDecrypt
The corresponding key is an encryption key

[No fields]

11.5.16. Structure: CapabilityDecryptServiced

Inherits: CapabilityDecrypt
The corresponding key is an encryption key

AuthenticationId: String (Optional) UDF of trust root under which request to use a serviced capability must be authorized. [Only present for a serviced capability]

11.5.17. Structure: CapabilitySign

Inherits: CryptographicCapability
The corresponding key is an administration key

[No fields]

11.5.18. Structure: CapabilityKeyGenerate

Inherits: CryptographicCapability

The corresponding key is a key that may be used to generate key shares.

[No fields]

11.5.19. Structure: CapabilityFairExchange

Inherits: CryptographicCapability
The corresponding key is a decryption key to be used in accordance with the Micali Fair Electronic Exchange with Invisible Trusted Parties protocol.

[No fields]

11.5.20. Structure: CatalogedCallsign

Inherits: CatalogedApplication Fast lookup for the canonical form
Canonical: String (Optional) of the callsign.

ProfileUdf: String (Optional) Fast lookup for the profile to which the name is bound.

EnvelopedCallsignBinding: Enveloped (Optional) The enveloped binnding of the callsign to the profile.

11.5.21. Structure: NamedService

Prefix: String (Optional) The IANA service name (e.g. dns)

Mapping: String (Optional) Optional name mapping, (e.g. alice@example.com -> alice.mesh)

Endpoint: String [0..Many] The service endpoint. This MAY be specified as a callsign (@alice), a DNS address (example.com), an IP address (10.0.0.1) or a fully qualified URI.

11.5.22. Structure: CatalogedBookmark

Inherits: CatalogedEntry User comments on bookmark entry
Uri: String (Optional)
Title: String (Optional) **11.5.23. Structure: CatalogedTask**
Comments: String [0..Many]

Inherits: CatalogedEntry Unique key.
EnvelopedTask: Enveloped (Optional)
Title: String (Optional) **11.5.24. Structure:**
Key: String (Optional) **CatalogedApplication**

Inherits: CatalogedEntry
Default: Integer (Optional)

Key: String (Optional)
Grant: String [0..Many] Enveloped keys for use with Application
Deny: String [0..Many]
EnvelopedCapabilities: DareEnvelope [0..Many] Escrow entries for
EnvelopedEscrow: Enveloped [0..Many] the application.

11.5.25. Structure: CatalogedMember

ContactAddress: String (Optional)
MemberCapabilityId: String (Optional) **11.5.26. Structure:**
ServiceCapabilityId: String (Optional) **CatalogedGroup**
Inherits: CatalogedEntry

Inherits: CatalogedApplication The connection allowing control of
EnvelopedConnectionAddress: Enveloped (Optional) the group.

EnvelopedProfileGroup: Enveloped (Optional) The Mesh profile

EnvelopedActivationCommon: Enveloped (Optional) The activation of
the device within the Mesh account

11.5.27. Structure: CatalogedApplicationMail

Inherits: CatalogedApplication The S/Mime signature key
AccountAddress: String (Optional)
InboundConnect: String (Optional) The S/Mime encryption key
OutboundConnect: String (Optional)
SmimeSign: KeyData (Optional) The OpenPGP signature key
SmimeEncrypt: KeyData (Optional)
OpenpgpSign: KeyData (Optional) The OpenPGP encryption key
OpenpgpEncrypt: KeyData (Optional)
11.5.28. Structure:
CatalogedApplicationNetwork

Inherits: CatalogedApplication
[No fields]

11.5.29. Structure: MessageInvoice

Inherits: Message
[No fields]

11.5.30. Structure: CatalogedReceipt

Inherits: CatalogedEntry
[No fields]

11.5.31. Structure: CatalogedTicket

Inherits: CatalogedEntry
[No fields]

11.6. Publications

11.6.1. Structure: DevicePreconfigurationPublic

EnvelopedProfileDevice: Enveloped (Optional) The device profile

Hailing: String [0..Many] A list of URIs specifying hailing transports that may be used to initiate a connection to the device. This allows a device to specify that it can be reached by WiFi transport to a particular private SSID, or by Bluetooth, IR etc. etc.

11.6.2. Structure: DevicePreconfigurationPrivate

Inherits: DevicePreconfigurationPublic
A data structure that is passed

EnvelopedConnectionDevice: Enveloped (Optional) The device connection

EnvelopedConnectionService: Enveloped (Optional) The device connection

ConnectUri: String (Optional) The connection URI. This would normally be printed on the device as a QR code.

11.7. Messages

11.7.1. Structure: Message

MessageId: String (Optional) Unique per-message ID. When encapsulating a Mesh Message in a DARE envelope, the envelope EnvelopeID field MUST be a UDF fingerprint of the MessageId value.

Sender: String (Optional) **11.7.2. Structure: MessageError**
Recipient: String (Optional)

Inherits: Message
ErrorCode: String (Optional) **11.7.3. Structure: MessageComplete**

Inherits: Message
References: Reference [0..Many]

11.7.4. Structure: MessageValidated

Inherits: Message Enveloped data that is authenticated by means of
AuthenticatedData: DareEnvelope (Optional) the PIN

ClientNonce: Binary (Optional) Nonce provided by the client to validate the PIN

PinId: String (Optional) Pin identifier value calculated from the PIN code, action and account address.

PinWitness: Binary (Optional) Witness value calculated as KDF (Device.Udf + AccountAddress, ClientNonce)

11.7.5. Structure: MessagePin

Account: String (Optional) If true, authentication against the PIN
Inherits: Message code is sufficient to complete the associated
Expires: DateTime (Optional) action without further authorization.
Automatic: Boolean (Optional)
SaltedPin: String (Optional) PIN code bound to the specified action.

Action: String (Optional) The action to which this PIN code is bound.

Roles: String [0..Many] The set of rights bound to the PIN grant.

11.7.6. Structure: RequestConnection

Connection request message. This message contains the information

Inherits: MessageValidated
AccountAddress: String (Optional)

11.7.7. Structure: AcknowledgeConnection

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

Inherits: Message The client connection request.
EnvelopedRequestConnection: Enveloped (Optional)
ServerNonce: Binary (Optional)
Witness: String (Optional)

11.7.8. Structure: RespondConnection

Respond to RequestConnection message to grant or refuse the connection request.

Inherits: Message The response to the request. One of "Accept",
Result: String (Optional) "Reject" or "Pending".

CatalogedDevice: CatalogedDevice (Optional)

The device information. MUST be present if the value of Result is "Accept". MUST be absent or null otherwise.

11.7.9. Structure: MessageContact

Inherits: MessageValidated

Reply: Boolean (Optional) If true, requests that the recipient return their own contact information in reply.

Subject: String (Optional) Optional explanation of the reason for the request.

PIN: String (Optional) One time authentication code supplied to a recipient to allow authentication of the response.

11.7.10. Structure: GroupInvitation

Inherits: Message

Text: String (Optional)

11.7.11. Structure: RequestConfirmation

Inherits: Message

Text: String (Optional)

11.7.12. Structure: ResponseConfirmation

Inherits: Message

Request: Enveloped (Optional)

Accept: Boolean (Optional)

11.7.13. Structure: RequestTask

Inherits: Message

[No fields]

11.7.14. Structure: MessageClaim

Inherits: Message

PublicationId: String (Optional)

ServiceAuthenticate: String (Optional)

DeviceAuthenticate: String (Optional)

Expires: DateTime (Optional)

11.7.15. Structure: ProcessResult

For future use, allows logging of operations and results

Inherits: Message

Success: Boolean (Optional)

ErrorReport: String (Optional) The error report code.

12. Security Considerations

The security considerations for use and implementation of Mesh services and applications are described in the Mesh Security Considerations guide [draft-hallambaker-mesh-security].

13. IANA Considerations

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

14. Acknowledgements

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

15. Normative References

[draft-hallambaker-mesh-architecture]
Hallam-Baker, P., "Mathematical Mesh 3.0 Part I: Architecture Guide", Work in Progress, Internet-Draft, draft-hallambaker-mesh-architecture-20, 20 April 2022, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-architecture-20>.

[draft-hallambaker-mesh-callsign]
Hallam-Baker, P., "Mathematical Mesh 3.0 Part VII: Mesh Callsign Service", Work in Progress, Internet-Draft, draft-hallambaker-mesh-callsign-01, 23 October 2021, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-callsign-01>.

[draft-hallambaker-mesh-dare]
Hallam-Baker, P., "Mathematical Mesh 3.0 Part III : Data At Rest Encryption (DARE)", Work in Progress, Internet-Draft, draft-hallambaker-mesh-dare-15, 20 April 2022, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-dare-15>.

[draft-hallambaker-mesh-discovery]
Hallam-Baker, P., "Mathematical Mesh 3.0 Part VI: Mesh Discovery Service", Work in Progress, Internet-Draft, draft-hallambaker-mesh-discovery-01, 13 January 2021, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-discovery-01>.

[draft-hallambaker-mesh-protocol]
Hallam-Baker, P., "Mathematical Mesh 3.0 Part V: Protocol Reference", Work in Progress, Internet-Draft, draft-hallambaker-mesh-protocol-13, 20 April 2022, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-protocol-13>.

[draft-hallambaker-mesh-security]
Hallam-Baker, P., "Mathematical Mesh 3.0 Part IX Security Considerations", Work in Progress, Internet-Draft, draft-

hallambaker-mesh-security-09, 20 April 2022, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-security-09>.

[draft-hallambaker-mesh-udf] Hallam-Baker, P., "Mathematical Mesh 3.0 Part II: Uniform Data Fingerprint.", Work in Progress, Internet-Draft, draft-hallambaker-mesh-udf-16, 20 April 2022, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-udf-16>.

[draft-hallambaker-threshold] Hallam-Baker, P., "Threshold Modes in Elliptic Curves", Work in Progress, Internet-Draft, draft-hallambaker-threshold-07, 20 April 2022, <https://datatracker.ietf.org/doc/html/draft-hallambaker-threshold-07>.

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

16. Informative References

[draft-hallambaker-mesh-developer] Hallam-Baker, P., "Mathematical Mesh: Reference Implementation", Work in Progress, Internet-Draft, draft-hallambaker-mesh-developer-10, 27 July 2020, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-developer-10>.

[draft-irtf-cfrg-frost] Connolly, D., Komlo, C., Goldberg, I., and C. A. Wood, "Two-Round Threshold Schnorr Signatures with FROST", Work in Progress, Internet-Draft, draft-irtf-cfrg-frost-11, 7 October 2022, <https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-frost-11>.

[draft-komlo-frost] Komlo, C. and I. Goldberg, "FROST: Flexible Round-Optimized Schnorr Threshold Signatures", Work in Progress, Internet-Draft, draft-komlo-frost-00, 7 August 2020, <https://datatracker.ietf.org/doc/html/draft-komlo-frost-00>.

[RFC2426] Dawson, F. and T. Howes, "vCard MIME Directory Profile", RFC 2426, DOI 10.17487/RFC2426, September 1998, <https://www.rfc-editor.org/rfc/rfc2426>.

[RFC5545] Desruisseaux, B., "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", RFC 5545, DOI

10.17487/RFC5545, September 2009, <https://www.rfc-editor.org/rfc/rfc5545>.