Workgroup: Network Working Group
Internet-Draft: draft-hallambaker-mesh-schema
Published: 20 April 2022
Intended Status: Informational
Expires: 22 October 2022
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

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

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

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

This Internet-Draft will expire on 22 October 2022.

**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=MAMQ-ETEA-JBL3-6UKE-LRNT-DGC3-OIDF
```

The account profile created is:

```
{
  "ProfileUser":{
    "ProfileSignature":{
      "Udf":"MAMQ-ETEA-JBL3-6UKE-LRNT-DGC3-OIDF",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"ni85QjaM8wU5vRoKmwnxD0F9c4SK303Mk0Gad5WlJ8hgB
  iYWw9oNzmi32sw8XAmer6UM0SoTc24A"}}},
    "AccountAddress":"alice@example.com",
    "ServiceUdf":"MDSK-EUHS-QXGD-LKOF-AVC7-V2RH-LV6Z",
    "EscrowEncryption":{
      "Udf":"MBZP-WZAZ-B6KQ-MYYP-H7KD-VVBA-7T6U",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"tR85RCqWv8-X5Bk0NU4EVljQFJ585FNE3ZwyWzXSVtJHi
  x0FZ7jZQ7xg9uurw8KOKl5M0UW7LLOA"}}},
    "AdministratorSignature":{
      "Udf":"MBDV-XXNH-2RUB-RBMZ-5NG7-L3CD-3THV",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"HUwN4RVhGczFlOm2bDcevvVYyd6gjdq33QqV8Uq39dGas
  RzQn9_PVgCBRI_8MjiverTKdaaEI32A"}}},
    "CommonEncryption":{
      "Udf":"MDPR-FJVW-GK5Z-2LJA-LMYV-XSCH-HE2C",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"55jUkmqn3gwG0b2HzDVu3Hlf5sO6GgVlj_vaYFwAEksDc
  My3wyvUwt9ojkeUKT6304Dwfrh-Uw8A"}}},
    "CommonAuthentication":{
      "Udf":"MBVI-EWLO-EI7J-OVAK-GGZH-6YHW-ZJSU",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"fTU3TeB1-7K8SZpo4tQxZPpJAb-_d3NIdJhlkxWaiZogJ
  REK9adPf9Kns5mqr11UTToIMhzfdJaA"}}},
    "CommonSignature":{
      "Udf":"MAMP-BX4G-AKK2-YHPA-IXJV-Z2KV-UXBW",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"Y6-D2DbbKlaVXvG5ZQweLd5_kP1ECACR40bDmpg-Y4Ks9
  2FNe-uysWUrM_LmQKOIPjjr5L8NOBEA"}}}}}
```

### 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":{
    "ProfileSignature":{
      "Udf":"MA75-5N5Q-BPQF-5LMP-AN6X-NM4E-U4KS",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"pwa2YXUVQCKc31N0BL1_aSo270xT1Qo37IW6HWadhTx-b
  wqFEvdbZJ4UnjPjabKFLPs3NeXj77yA"}}},
    "Encryption":{
      "Udf":"MAAB-KTVM-DBAR-H2MO-BR65-7ADT-MLLA",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"pD72qIWSU1Z51BA0C220t-ZgE22uhnBP77VZz4gsiBj_8
  8XnpfK33J34WuKorrW32CZe_-SkqviA"}}},
    "Signature":{
      "Udf":"MBQA-M2E2-PZPR-GIM3-JCRJ-QDDC-NJXL",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"Hwu1tsJThxHMvig7PhCBnjgEYY9r7Ima0uYyKkYY5kwB9
  iD4K30jiEomSrdWFpOz6I4j_wWsFKsA"}}},
    "Authentication":{
      "Udf":"MDRS-RHS6-4XIE-34VE-2ZLM-GKWL-VMMN",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"ywFQCwMmLGNTUZh-py5Oef30Dy9j8CwWIvCZAPsuWowfM
  EUjROOPFF3q2NAg0PI3Lq87bPaUdrQA"}}}}}
```


#### 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 = MA75-5N5Q-BPQF-5LMP-AN6X-NM4E-U4KS
   Account = alice@example.com
   Account UDF = MAMQ-ETEA-JBL3-6UKE-LRNT-DGC3-OIDF
```

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


```
{
  "ActivationAccount":{
    "ActivationKey":"ZAAQ-GK4S-YPOU-UMUP-MVLX-2FO3-QN7Y-UFQK-HEPB-R
Y4L-WSUB-2NYA-VW5G-VFL5",
    "AccountUdf":"MA75-5N5Q-BPQF-5LMP-AN6X-NM4E-U4KS"}}
```


And:

```
{
  "ActivationCommon":{
    "Entries":[{
        "Resource":"MMM_Contact",
        "Key":{
          "Udf":"MDRE-XGSY-3FEQ-7JBD-5CGH-QTC7-TZEV",
          "PublicParameters":{
            "PublicKeyECDH":{
              "crv":"X448",
              "Public":"rJfOVOiZXyn_r--rH7gff3rIQFbtYMnhHsKQFYibG
  1R9W-RSXUIjHZfBx4F94e7FSe3Qi9wIb1CA"}},
          "PrivateParameters":{
            "PrivateKeyECDH":{
              "crv":"X448",
              "Private":"8yGWHtYjkzgrOEs13qsqZmS93diaEiFxP2IBMJJ7
  M3-LF_SfVv02Wzp3_557yaPh6UOYE7K2r-I"}}}},
      {
        "Resource":"MMM_Publication",
        "Key":{
          "Udf":"MDAV-VMHF-DRT3-BKDL-Y2HL-GLK2-GHEW",
          "PublicParameters":{
            "PublicKeyECDH":{
              "crv":"X448",
              "Public":"Igv5xtCKBf1hcoMivJ-sBYXF8-sn4AuTe_lzgHzKq
  _8wyiejH7-QEzrOIuxOvnFaTphUM24DXqYA"}},
          "PrivateParameters":{
            "PrivateKeyECDH":{
              "crv":"X448",
              "Private":"EhcNnOUISxV3XIdx-7Dcy8_bSPHyXv6YN1Sr-OLp
  5EPV22v5GRNQ63-4RWPe2AwGowo-JO9LQCU"}}}},
      {
        "Resource":"MMM_Inbound",
        "Key":{
          "Udf":"MCPS-VCZ3-XVV5-PBAI-QN5B-CF6E-A75G",
          "PublicParameters":{
            "PublicKeyECDH":{
              "crv":"X448",
              "Public":"P6u7tVLfiqyvmACUQJWiu_P36h38sHXcXbaVqL5nh
  wVE7g9w6IAmP22cBm-omewfEdpZN7rR1bqA"}},
          "PrivateParameters":{
            "PrivateKeyECDH":{
              "crv":"X448",
              "Private":"ItPikKDnBdWk1bzKw10zc4H1g9L96MvbVrWSlL2o
  PKg-kVtak8idY3jblfetl_WpEK8lf5mi2AI"}}}},
      {
        "Resource":"MMM_Outbound",
        "Key":{
          "Udf":"MAFH-RDQT-JWOQ-INJX-WEBR-J3HJ-M7TZ",
          "PublicParameters":{
```

```
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"LJi26cciDxYLIetfl5nwtKa1pYOeYhB9XkxUpPYxJ
wleIq06pwYX14PRxdWKvpm4vMx4V_W1Tk4A"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "crv":"X448",
            "Private":"v3Ra_hFcIHgRavFA_BKTw5jaHlFN3RR7GNwX81XX
BEymY7Jn42Zu2r9RYIxJqbf25pvbmpN03ME"}}}},
    {
      "Resource":"MMM_Network",
      "Key":{
        "Udf":"MABR-5WCQ-TNIL-GJGY-JLEH-F477-ABMS",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"s-RYfBg8hEC4BPRWnDR-DSa3Pb1qNzcozSSugzBwB
XOVUti4jDJBoF5naUr5cbtabSj0EsgLZmkA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "crv":"X448",
            "Private":"KhYa4lUPXaVe8fQL4fzbcH7-yhMhEf7LBacgxMdB
kqJ-8F1arGILPArRAF3Ib4MvphwEafcEsIA"}}}},
    {
      "Resource":"MMM_Application",
      "Key":{
        "Udf":"MBXU-FZUK-KFH2-72J2-Q4N2-A7AB-25GF",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"lwu54QyqdNjLWQHIRdZ3_bpv9JKuoJDtyCG0lWghA
xOt4toLqrdsWrC0qqZnt3edJKJKJ8m6O8CA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "crv":"X448",
            "Private":"iOUDwM6SAetyLJMC6uG_3CIWFoPWAMy9mZCl1WSL
dzkiSfnwWhz2a0NQ5bwTRLYDAYyccBx_s7c"}}}},
    {
      "Resource":"MMM_Credential",
      "Key":{
        "Udf":"MDG5-EPRO-L3LG-GGFU-WKSG-EXU3-GGAB",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"INLLEYrLIPzFvcrxknMiC6CBWpZbn8i6PkyYrTWDK
adc8DqCQ1PaW0gayF-Fjyh2nAl0sTJu8nqA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "crv":"X448",
```

```
            "Private":"csfYsFBXA0qawDzo5kA8lCku_yV-jv9tro0yNvNv
740-gzNM1jaRRNdp1xXuO1tgWDa3gEmwNRc"}}}},
    {
      "Resource":"MMM_Task",
      "Key":{
        "Udf":"MAT6-WUMY-SZ3J-ZREZ-RLV2-YJQ5-ASC7",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"_WW7zyQPuEJLZ7oVTd_EccJwZ1Ld5KAIdkW2RBBGl
btoVF1Gpna4yr4qVlgSrR0driZDaZUJIDaA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "crv":"X448",
            "Private":"JWzHJH5yBYh1tzkwuWtL2H4N2svYfem3p8oiB129
Mqz59t87R1jctfhl9kwwilllPF54xJmXmTg"}}}},
    {
      "Resource":"MMM_Bookmark",
      "Key":{
        "Udf":"MBQJ-3DZR-GNXB-W3UQ-P62O-G4RK-HX2O",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"n27OoqLidzKr9ju-p9jY-0R4vsKyUy_5e6lak4_kC
aG1Mr5jroj-w7y4VrybGm-NfGEyY-UMBKmA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "crv":"X448",
            "Private":"1IeSz2X9UOCME9mQF37f_8RziEV3LVBQgDxVZNGb
yh0YWATkwGQM17l2oXYYaWm2zdY6Bu8r7uE"}}}}
    ],
  "Encryption":{
    "Udf":"MDPR-FJVW-GK5Z-2LJA-LMYV-XSCH-HE2C",
    "PublicParameters":{
      "PublicKeyECDH":{
        "crv":"X448",
        "Public":"55jUkmqn3gwG0b2HzDVu3Hlf5sO6GgVlj_vaYFwAEksDc
My3wyvUwt9ojkeUKT6304Dwfrh-Uw8A"}},
    "PrivateParameters":{
      "PrivateKeyECDH":{
        "crv":"X448",
        "Private":"14xBD_TtMiv4VXLfv53eQqAXkGzDsI5d15IZekWy4Yi8
uTPw35kXuzIhgNvw1REvfU2JdBVh3wo"}}},
  "Authentication":{
    "Udf":"MBVI-EWLO-EI7J-OVAK-GGZH-6YHW-ZJSU",
    "PublicParameters":{
      "PublicKeyECDH":{
        "crv":"X448",
        "Public":"fTU3TeB1-7K8SZpo4tQxZPpJAb-_d3NIdJhlkxWaiZogJ
```

```
REK9adPf9Kns5mqr11UTToIMhzfdJaA"}},
    "PrivateParameters":{
      "PrivateKeyECDH":{
        "crv":"X448",
        "Private":"LqBnHkzzISgjBeoCMjlxX1p_pnrZ8Cdfn0kMTzIUf4tL
IvwRIueHQEYWP5_nvYSmYbMrJCWUA0U"}}},
  "Signature":{
    "Udf":"MAMP-BX4G-AKK2-YHPA-IXJV-Z2KV-UXBW",
    "PublicParameters":{
      "PublicKeyECDH":{
        "crv":"Ed448",
        "Public":"Y6-D2DbbKlaVXvG5ZQweLd5_kP1ECACR40bDmpg-Y4Ks9
2FNe-uysWUrM_LmQKOIPjjr5L8NOBEA"}},
    "PrivateParameters":{
      "PrivateKeyECDH":{
        "crv":"Ed448",
        "Private":"IAfy3NjVxhiNYFt16w5A99iy3TqCByxQLb9l5WoWlxN5
pjHzHeH9Ibr3n22suIvvsctPdfPAeeo"}}}}}
```


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":{
    "Authentication":{
      "Udf":"MAVT-XX2Y-B6D2-7SJ3-VVTW-MQ6C-S2MU",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"yrFrem_3XKqaQAvnlTxaZ2msYD-dBceF8NOssaE7BS5bh
  BD_ViasKtPXFncsZ-4LdAjpHE2bWKIA"}}},
    "Roles":["message",
      "web"
      ],
    "Signature":{
      "Udf":"MCJE-YAQI-I4OU-EXTX-CQ5W-IORV-HKH4",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"ayCD-NfNvsgHfxy4lyDEysG8PD36zgLq1AZmh86_R4qY2
  IpzPiymPiunbLL-pRZy8pPKDiwHPG0A"}}},
    "Encryption":{
      "Udf":"MDCW-SXW2-ROVU-4R3G-E5R3-2JGI-YBPF",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"Un-_awwiGjXgaO99A66zrVJwUi1nUaYAuftP4HTmsnZg_
  fhq3Z0rcja-z-er-BbJ9MHAqf3TxfyA"}}}}}
```

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":{
    "Authentication":{
      "Udf":"MAVT-XX2Y-B6D2-7SJ3-VVTW-MQ6C-S2MU",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"yrFrem_3XKqaQAvnlTxaZ2msYD-dBceF8NOssaE7BS5bh
  BD_ViasKtPXFncsZ-4LdAjpHE2bWKIA"}}}}}
```

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":{
    "ProfileSignature":{
      "Udf":"MDSK-EUHS-QXGD-LKOF-AVC7-V2RH-LV6Z",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"UuWD8qxdeqk6pyWkoz63qBpJPCcZOb-hySYQb_Lx5fGfY
  OoU4gB7V6VauAfG-uIBDBMqg1QmcGQA"}}},
    "ServiceAuthentication":{
      "Udf":"MDAL-ZI5N-4UKZ-H6VL-F25K-PHNF-ZUVA",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"d3bn_-qEVwBM69Z93Kabn3MqSnc9GQDlFT2_Rcx5tVRme
  b_bjy71vSRSk3ZP04Dj2cUBM4Agr-oA"}}},
    "ServiceEncryption":{
      "Udf":"MA4K-EVCK-36OZ-UHSQ-SHLK-36N3-YW7L",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"P_owWGt7wdtuvcsGCPfQo8uF5CFXG2RPwcTBlKZqx0VIf
  9hpMdeyuAjRMFeE5_3nRm0ywL6tkUQA"}}},
    "ServiceSignature":{
      "Udf":"MAC3-YJSU-42F3-BB4L-T47H-VF6M-4IXM",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"_pT0cmw66uaQbd0QhE15yUtm1UDsdoZ1zLtGrqNnDfTbh
  Q8qUqDlpPG4fszIFa9viKYE90CBA2EA"}}}}}
```


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":{
    "Subject":"MBDH-L24Q-ZFNI-RSNS-AQ7Y-WGCQ-HRZ4",
    "Authority":"MDSK-EUHS-QXGD-LKOF-AVC7-V2RH-LV6Z",
    "Authentication":{
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"_fdKvOXPYHKFFb8oljLKA3raIGkamEuL8beeoknQpBZVc
  hhCv9QOGm47SBPow59_avyQuKO2fWSA"}}},
    "Account":"@example"}}
```

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":{
    "Key":"mailto:alice@example.net",
    "Grant":["web"
      ],
    "EnvelopedEscrow":[[{
          "enc":"A256CBC",
          "kid":"EBQL-UZXE-NDYQ-4ZWU-MD2J-ZRVB-VKMJ",
          "Salt":"YpZfSceDyfABMtX0EaezWQ",
          "recipients":[{
              "kid":"MBZP-WZAZ-B6KQ-MYYP-H7KD-VVBA-7T6U",
              "epk":{
                "PublicKeyECDH":{
                  "crv":"X448",
                  "Public":"-G2w5cKrAlMlTWkcds8EdD_Q9yXkkmVrroiG-
  S7oupxqwWa81j4D51UzOSXYw4ppzS8Wivahq1SA"}},
              "wmk":"mNAKX_Hqp6ceS_sGCcmPrEUl9f-OlS_yP9NjePwvsbtc
  BFewa1jXHQ"}
            ]},
        "c2049fWM29aBRW-li7JCScH34qT9yLKa6-lt3bpQWXGr7P6iXbokuOje
  bMaQy3hYRU72pXJGOUkUadfymayRzHKoiQwOlFyCni6JpVwBvjsHZ41nXQKzZWh5c
  nmIAMyB6Uspq8R_FKpKBf8QfzvDJcGTzOxTbfdixRHxXM1D2WGsgZ4a4vWPedSOEX
  MXHZ5C2KVYhvCOD5M0LDeinDO6RUQKajRmIxe2OUP34N3d2wr_J9KaWyByAUOLRut
  Jvh1oVa8EqIFcQ8jDoTI48le5fttEuHRPuaR8IB_ypsn3t8udgPDbIyZ_k3gNQV8_
  LHflxeGls6GfeVU4WdHPylDFpU6pz0gGU_H06wbMMwhemwYWnOKD7zHs3pwfxLp9k
  t3Ez22s6-Gt8eVcAyTUoIv6wfeqZhkBO-K7sRlPgRKnL3MQDBwtx_s5bkJns3WBuL
  1s8XExqtO2j6zC2fa5Pnevvyq0xEFQpUMhYuQHi0trPuLzHWS82x7cfCIomZIIa0K
  jgOxeA8f8lUgwf-PyFG5XZ3RPIvGBPIuhkj3JkralL4sYP9p0byq3rByDwuXVeDhJ
  CdoRiRfoKoDnEE72axBKDFHDusAum_RvKjVIOfvsJ1K8u8hPZlIPfrUGX8Ont03yF
  x3a7Vvw0Kngjw_3fAgmnJBAt4tNAlKv6K650byAeyA7ePGST9aXRpmgZqeSHuDTD4
  7LbpW9772R8IPrMPPOfB-z559sIuaKdO-Va_qKF0bDsdb-_DR_UFYc6j5kaz_CcQy
  h-Z7ENT1-5Cn8SZn6YJ_ppWGCv9BzY-zKRbMvYO40Pm7kgHis6Ypz5uWNEuwUR_Xq
  _F20ojdhPBD_V52quh8kZ5mr9Yhq5APnsNX8xe4qoBJb12D1zeYwGPq2ycU5-vzp1
  cA2LAUGJ9kAFBLimpn0uev1nJqKJmG9v0KglhByQJzyIVSn1Lb9F1CJfpMKrLZv-h
  kukW5eoZuHdMUFKWBIp0UfHOuWaoLdOEdHo06FewH7CUGF1AkpGYCwSnI5cB4N0qF
  QWDw6RMq390c46kaTeAUW6zVpMmQ5gIIt1sBR2yxpIs4m0bgfugTdEDSAKWvHQ_iX
  ZE5pPw-D7nmvx7ubtBHLp3KSBm8qHdroiQRjPQHkBtSnoVExgJQBhcHAe18Ef5kDp
  33TcRSoCW56K-CJBW8uGOp3MgeaEwWcR9psSdIWpfweIWi3uQLPfTklVGBLrcNoTw
  q2cDtrlWTTxuFmUNNyxyMoFfJ95yasEKFscz2HzPKqYN4eK9MWh3qs5fIj0Lmr-30
  HhEbaky88mvszrXg1ZrNb3TJINSHHSSVw1MH7M7o-jWdImMUowqSzdgVXyTVEPglf
  NTpIN3cWPD2WS_281j1CBZXo0K6cj-SkNxzzEvR5qg7pslatW_5ucehZXRN_MSARo
  ffWDKFjF1rXi8Z0o5gqaKPniJ2rxIKIjbiGulE2ayeKB9A5PErAEJ44LKKc34fTiH
  L5Xe_jiDDPijfT8YUcWDRULvv8PcjQexAP9AOji-hLYB2pv6z36pl2z8JxqrjskiM
  EtipMIJserHYFj3TqqNlv7PUiR5ifgSHB1B4Be2OlT95T3W5TqZ0xEMW9LJKfw6YC
  xp53HqqW7Fjc2r3mWmb1dkqKunrr1ZnXKwszWMV97gzjTgc1y4iT0TndETW8ucT1T
  DdeLae9IKFimsKgL0xkxBHT16ECSYZLOztW2E5uDF23175bZocsnLGj7JW4TMkOjq
  SODI7q2L46yhJCnBkctPABOFhi8HV_qzlJ13Lep6VGCUXdeFDWSMwZKvbwHAfF01M
  YSxhNavVycaGnIfcld4w_nLiJw2r-hLuH0zk_dy3H0sPD4h-dDyu7aCnD77Y0dFdK
  yfWPHi4lDOGrkN3XsNDihu72RNo1kEEqqbyRPxmFURJP8DnlxgJiaPdMBiDFpkJSS
```

```
olwLL1sOV4j6w8m1Dqox91rrZnpnBzANjS_BAGuLM4f7KyQ-JTIFPcybCgfbxnlaQ
cxejBDERoePNaddM_IeCQM9RRN0CTwjQZWWw17at7nxafleQw3IKva2_3U525CA9M
oFHRSJ756jcZlkLrCoJOK0iuL0-ZLOmTRPsEFfAskMuDtD6t889dA31ueToKcClBJ
xurcvo1_8LDl-oe7KyOyTEI3-w1viSiorJGtleVLnCyN5ZylH4rRE-6Tayz_23UR3
-JTNjky4mDnDCeXsZINpiZKsZd3DyoNkThwV08mv4IsTYQj_sBzEjnPkaCZMZ2CNj
u8aqHmVqLyW3JWIkTxwIPihlZ0feM6faVhh1SmhubnBneAeJWb5vJw24d1h1I2U1T
odUEMFfNjthO6rUZjZtVzIhbi4ma3G_-eTJGCOg__7zl2-V-a2wk7OHleJVGB5HNg
O6vP55GTvnkSV8NL4OkmABOY5Q33ePYxFK7TXnTlsYj3uqvHZw_-kDgy_U2MW3VTX
Fc8N5a7jFF1pwv2SWmjYCg7VXB6gqsTTBEcUMtVxc8w44RJdduHBeCLklUY36BHB3
UXt4eNCtHsY6UNynzRzuMhw5cVU4vhmCinDdx_xPTbptWQzqtQd2ARWEDWl6BEze7
tft2upt70ol8hlRg0BQRH1ywfwbmiGd35lwykRIKL2l4MAa1rbCgPLz411U5FMNBq
Jyn8irtjeNDTPm7BG6J4ehBz4pcDffIKsYtVbMT68iEll5unalo1itUfkyXV4UCHr
Dj4cWWjDa2eWgr8AKunYcTMZRFLsFxrtbmm6gQfz6gc6cxuDHHLcOItfjqmTp-Q5d
Bi17Wp2o18QQX7kgahxa6gNnTdV73nKroiWP2UhXgRctA-r1Dpal1QrMseVUZOs5W
k_9S0TW7EwnRG_ECGW8LhpJ6letVkSDP3mTV9XXCBfga71LSV2VPJKh4YvuVURD3U
f4oM0XCAy8kU1mP8sEXnmI4ejHmnWpH7LPOxTzsT6PVyDXVRccFlYjbcc6Fxm07JR
qXYajOjdTgrsXvz9fKr_Mj0oRgNCab0GdNpPHD-hNpD3P-0lxxOUEbNFyp_A9jgg7
vfmBuKWwpezONsvCv40mU5JtQbMMOytVdny4jDZHkLdqbH6N6xlP6XGJt_qrDKmz2
_CvqcuUh93Ep4Hh9wWqOsjZvnjyP13d4OsOWbIPHzDKBauBR1ovbT_qY52IHzsPf0
3Cb8QxJt6qrTJ79ay59OK0R-HvKD0k9Tqo1svFBYc8z4WA"
      ],
    [{
        "enc":"A256CBC",
        "kid":"EBQF-XDKF-XDI6-5LNG-AHLY-CH3I-KI2F",
        "Salt":"e1J8nMW9M6gT_hNYx-UMZQ",
        "recipients":[{
            "kid":"MBZP-WZAZ-B6KQ-MYYP-H7KD-VVBA-7T6U",
            "epk":{
              "PublicKeyECDH":{
                "crv":"X448",
                "Public":"8VUU5j0JTwCYCCieLOO3KrgKkvTb0L4jEpcTa
O8SiDQiGvdDfr0rYMjI8QC1A4u4OoJQ7Dy8guyA"}},
            "wmk":"bg-KGIaWlB6GqEb4V1HPhE3fOed9qOsfkkjY0O3UT1-a
5QXSBM3WDQ"}
          ]},
      "_4_lNBgGmLjlZs1fhK8ZfqkDcV-qCcQAIjCx4ExUFkmhEyz0QEPpi5Wo
Z4ovGnxXfML8xbxFUFd47_gcUvvDacloK75t4PKf1xDUukZUQZ3Qn6zFIOI5u5JQN
0mOyRgCgZ6d-KCFpY_VOdntC2LqlTtTb7yJ2Q-Ogj-eBi8yWVJcr0sG2zSXrNbzl7
Hv_2QotUcC8TTzxduUIRuXGscc2Zhz4d84XCd0Pu8f4IfEYWAyxvK9S5hK7hiuP4Y
b-o7uDwCiNmgkNKHfG3vfyCsgNdHbaDoHszXOdArhDm-qDF4f1IpFW_snXTiK0KpF
vT1URzRmwfzdKqt-J9AOx__hoR4oMH7B2YAF7IPih94Bp2ORZm2NPgya1u2Gb4zqk
6VrRTjYWzhnuuoDhpX59ik3SdyjimjOKjHy5GNoRVX9L2MZS4-1guMoJoSvjWIQtV
aEa8TXtfMyfR-EgKQRd05aAAEwIOKj500gBfIE840Js535TMbl4w6PWquZnqyrA0o
wwkAWTibtMx7gcNrxkqU_buQWZHIV-lj5HQLRGAviYlbyKCjuTZ2Ktp5w0641Pll6
db1G6SDgf1OsfJdBhG61qLL9S0tfw0thiV2XuQq8uDWi2-A6qe80EQpC0RY1UEDlC
1ebjZk2I1bpMbJHe2aBQB4MroiOwgqRyzEcaxBqd8tji6bsIGie4r_7VvVmP0lZ2a
gC78Fw_ToX7v8gwcM7f6ucSepWv165HgJYBD11wjGoI5ANP2Zcl2hPDpS96NKknpZ
PKuRqoLO112r6cfXimPBJrUCvDF5EwTrqIkb4Oh9R8U0fCfSk3VVPfm4-q41SUFPP
TldlHQrgUSxLGvg64qzzJpEUHPHlxdhxh2xpFJ3PvzPkkm_X0ux_e_MmqNFzfktzo
```

```
NXySStKv85ufWCIo-4-zX0SBkkXsVUNkSTr6W_oUAy2G_NNVNKQjLwtn4WfBEO0TV
8DesfVP0J_K1k6mtLMmY0DvBm9BOtqxnuSBHkL_7jsR-Qp6LXpM-N5jamWx4oCivr
6d5sW0A2dXbZaNRKBGUUAac6CQLqnlTOIwLhmi08iYb7k_iEqnj1UxZr_ppAkk1F8
vuSnOVv4-jfkhgADSmMmRx6XM6G9UVx0hVY43rxbUv86cf1ZmcRyX-R-QPUOSY25K
3nsIG2wpK93UvHyqlqoDX6LmH2WMI7biKJrMbKOPnTON49DDZBXp4ziFhcWif7iIO
1K-QduVF1Xc-hbKYnq8haw9_Nbl6SI5CB6ompqlwLUM5FxUJmQp8KB3BQMOC3_R3j
pKrPCUijY7Rxv3pdHN9TsdKDXyqHWr0wvAriN1A43La4Nkg9DlbVfiwR6LzdhP7gL
oGqxBDOt8g-hN9xEsE8OiMugQDsj8lF6D9J3SDadFuZAz0FCSbwWj6_zUheedmo1L
IE4wlnm55_4e93gwKZEou5DBu6z20YcrQVcRlEFPwZypXMH3AYDDGk-S58Xpk2xL_
-H20LlRSx6dvs_M_xwoSZYZW9ntX5qaeydH2zZg3sCcOIAmJT-xbfsVx202Oig08g
gI6tgOnWZuHitRDiBx0hzw2OXrx3ClLcqnft1aT9oVjMhjYwvlzwhKxJKBft9Xr1T
gnwFjWuDw28jXN-A19QwnHlz55ZPS9pQSNwzfqLmBEBOTao0f7F4oN6DhM83NafUd
RU9tAnLwHNPQZC-j3-UgOAbaY7G6Z6s8wMUX8uWYocQ2kUxVxh8vqrg2OcKM2G3gU
K-3JwL8dJrDKsWTXulacD7olJDC-GrC7oZI_gzLixPABjOqdOzN0HFys5OqThUlhg
nb4ATWO5BOrODzDY3cQHrFyscAuQsl930OK0sBBNT7zD3xsC1whLuJkDxDNWJK67w
FcipkRx_di4CtCRSNfF9VHEvG_y3okPgXFPD7HX-KRdilSXTbC387tq9MOQ_a8xZD
SCzkH6432_gwFYxctOISauSITFggcaR7iSdkyDMEwDNPcHFKHsrrtQ91MBcc1wpo-
jI5rm-Kma3pHJAhNOQ141fnKYO0NliuYBhCWPvawTeafwa15pWvRaklSfiAT2iiBT
mfH3TszxIH8McjWe4ySkQ9OWRBFwF9-CJxa3LlIWAtvTytOS5DYPs94KprYbImam4
SmmYOSgFfy77EZ7tevGWaihu-qdnS8Ue9t7Gcwt2EKvGOq1Kkw_z16b7O2fHgfiic
KsvDkzckVjDyWaU3NX3bkmxNQLLS-SGXvXN4oOCsg3j-nvHgaQ_NBelDM-66Fw82M
318soeZMXCnPokD0SfJ3LSJ8df_Wy3_lpCuzfFFVV-aC2EMbljaP933BmQ0zrBsjR
UugGdzUFqGfw1wbDQ3hVKfgHKrqQLQPRip5YbLhN_i13ipkthcWZGIlE1Rs8JCkNx
4xoyZDt84tPU1BzmKdpWsr0uxay0mSejgRrO7XZvBYL4-kWDuCnyfAZeiWl_nGnwK
Np5VuxQ-x1lTlLZGKLohcZfWneWowI4FdB6ATaFqKHbDTbvbN5jJmky4elqVLL2zS
P9nExiFoRp59AkjfiECyfox_aHWTU3dqyRtr4uSqo7qNa1SyS92Qh8f33fFcGo6OA
uq6yYc0h2qheJNqnYyGjV8TLdIVJ5pTmZvZjAJkSNGRpNS3BSKOmil83gWOjNF7rM
MCR72aPOsJKjvNyX61If3B0XwaCVAL0PnwonpRaO9r4s1ntHG4r1yf2G15iYub1HK
chpvXQFyfKhHtKvfYEV4_gqGz65teinIm1m4wzZ8L0521m71ECvIKo-xYF9t66_IV
DqRG5kiNklmKwHMx2aTz79H_ioHg-hxfulou90N_IFIBawahDFVZEPmRxtnKp7zVd
x_XKTgoY3XU60deTCxQdB0dSbd6mxpF3pEdwSguqO9khjmKKBgHy05LxaZSMGSbiR
9fBj_7lOTinFl97zDTNz-dNjRTOMo8bwtLDJCZMCtqR97k1zJVs0SKhDwjMrTvFN9
WMR5QO8bqo8MJAIDgR1wBCtHb_tm6cbm5tKxcwruArWggq7CxHTTgtp5xfu9JTGqQ
RC251rHlZXGDizWDfuTDyv42K6OSgX9zkDV2Rmjfo8y6A0eBm45I5RJQ3UGd5u6kA
xW-elPaX6n-MGgUhPu2OG9Oxx-TwxhbrwJ9YA1vGlKQC6ZCEg-TjUXUmbaDikP_4V
SZ113lwS7LH5KOxD1vo4WqE4gS7Y9CN30diBTyvewJN69Q"
      ],
    [{
        "enc":"A256CBC",
        "kid":"EBQJ-7FWU-YBJF-EK3B-DOJS-HVBX-RPLT",
        "Salt":"e5O6pSFdqyKUJUcedkyblQ",
        "recipients":[{
            "kid":"MBZP-WZAZ-B6KQ-MYYP-H7KD-VVBA-7T6U",
            "epk":{
              "PublicKeyECDH":{
                "crv":"X448",
                "Public":"NfjQmmZNM5i41TbI1iFUwGsx9uTT9ngMqxprP
JVjOevSYfgtvOe9XeMN_n04z8KObjZg-CXOPuaA"}},
            "wmk":"tPdFA95AShcouY4SKCPFltBeHn1o_nUzpuHuAeri015c
```

```
meM_1JlNoQ"}
          ]},
      "chbmpCsHHclLNKgMICThaBnUq-dr2d_sVkC4CZqM6zVlRuNGEbjeFOKb
0ToVNM2jPxhulK70x2jDpbMqSeG21ysv-UQv5PkwIDtbNp7OSPCB1hqghnyz1o1pj
6RccaM8OBLnEnWKb1T5yWwYeHWbTbHsIEeGA8t4_TVP8GFhPECtO1GduNKtso5iQl
bo83BEB507yWRzJIFBdWbvmTw6-jIWdx9lfwnU8oYwd-3URcVFO7H5MIYdQhyKwwt
eNVoG01ngdhsNDQfeEv4v6bqBckiwns54ta7pLo0zANAh6w3kyTX2ZiTs_WlPupDC
OBDKNKqSHttZXU-tCQzmlziXBtV2Z82CM0XyFibTYsF4GL-fzQhr6gQNIV4IWoxco
sYovhP2azlUmVaH86IWTzJd_6HdvXhv75obkMzbRi0WI_cXoJbh-NBdLJf7t83JGI
5DOAYYAIb9y2Zd2bb-bptKQ4efM7zZx_PqzjwKckQAWil6yTU09mzsJ3HsckgRG_S
v5GOEXeR2eQsixFxxoPlbRziTLI87Kts0iL5exFfqT9RYzI-IOJY03B8xI0Tc-IFA
mWOjF092Wtgg4FU3B43zUyRI09hVSagCpO3lEZIz7kIhNBf8-VnN6enM0-L8rtV00
-f8FDRDKPYBFgEpr6bUZk1aopJ4RspB_A64K4ukj0HXE9JVe-Bgfu2Vtzc0PjYiP3
xCWDc_6Sk5ezKYzgvFMuKkRSRzmwivCLHDCp4vUDdLMawhKXMn1kbGQAdiJZdTKg-
FQhd-KWB467kRbwmJn7ArXCMjt2cGmegOPBXPL7YhX7WP_FN2kcUUvBTuNj9VshMK
lvTieRLevlr1urpAjs0uQuppHdhVA7tJ1joPIXC8_tOr9ML0XTE5NVuYBA9PXg7Dh
i9_E9ywpTNEemzFI4ZHI1IbKhCxl-VMgs_wextW0X_otxAL94CrVl3sNNBLWJTO18
xFkeVqeEoITEq2oKMVhmB3gWKtuc5cIUyHB-a8KZGqVNfMctndRdmuX27G4eVnzsa
owXkZA5VpCoEIqUi-t5Z-DipsxZ_ts3kkh0E73cjDzXKxCTjjLhMGS3Kb86D_AKHP
MjRL1ZsmDfI8Xh2S6LFxnIdmGwUdVSHGi_RQFEw9xdWAuf1tZM-Fg_CCBucP2Wkfo
zpvY_wk7QvCE7dlfznyIx-iC_uaIpmbt0eXZuy364cLA5I1gLEJPbRCfhiBEWNtJz
-n_QMvDskaN5B5XJSpJHtwHIGFlWWJ1wI17ELfLf3yZ7QD4i7qbLkw92t32v7zIgK
kzW745Ail0XjuzI4ANJklSXk3zCvjgSTxhM124DugCHdhaEsO9U1F4U8UNPDj9US_
qy-JCm5LQWTyi3DUnJiKmIaThI9VcOEJEnw8go57IRVqNbFrL8EyRFCXQnEH52SUP
T_LQNOP44pqLx-KL1FPvSMRH4XQGSQaqPkGW70m59huT7WhnPMqV8sbUM2ldC4ElX
MrqiPME1UFybC34Btk1ogIfJ-fvTQ_0UtMuneZjtnxe5tS7bmf295voyjHhWJLSNP
eSfdXRjfgnGoDjeCsBc4CqBk3104sX-HII_J9bKzJ6eqCpC55idVrIGdq-omwYZvq
dCybvjyoQk90KNykTF10pLhx0sMqbpMw8MwhaOnlMwYiLr_Ax37rB3iUBzEGFUSgL
X3x9GqThhl5Sgk-WXEJ-_m9LPrJENwSgbcYkh3usKVBKRNjsVQfOsyU4iWGTP2Obj
UyY_duwXdLt38aGVovQHs_N8caERZDqHx5lQrbjAmPmZj4ajsyVwa6IFIy-YA1imR
x4DwAWaz5lN8JSKNZaOjU8ncL5q5xHc4JEtDcgZV2BvreE_HudyyiEf_7BUc_gmay
KC-kc3IJYyZsZAiFC3QSpLi5PacXUczT3qwmTFRvjwCPQ_tB5tfSiIh70IFRw7jPc
69zri77VMWx7QzWwkFc0a18Zvtgs1yHgCTU_Gk4atELJF1DSpQDI00WPuRZxk3-oD
Dhe64Vut8zEYiDCSuFBzzpjy_QEAS29i5VeR7WOiridKwk3OPjjhsaHEyTGxyQXlk
Oyd4dD8XBDTIvtCeboXoy9L29sjCUPUSJLcupvFLoRNy-1aLqDF3HImbZvZaJvtBB
dautoRohBx8ru2eBx-4AXooeT_BrmBdudgsymOO30-Yns3gqUhfZSUiBb2jjTvVNo
xt2jc_78SXVI7F82HAEKnTebaDzXA7UWv7akN0YpgHv_QYEJHCyEVP1tdYQqmju7f
9A2AqboZA_RHKucaotOqL9xNOzfQCjFLQxptFV2JHCnqTQBi_wroCyO9R9zdbsZ3J
E1EEVUkIUDI_LBi9vRAVUtGKDVRSisEp5D3uwZgBV_Ebe86g2CwCXuT2Hxh7KjC59
YNhM09bqIqaETbJWkUa_QMQTizhDTDdTRRkOC46lFADpibvYZT5ca4QJnRmKYEdiq
sssFO2bZpngmyqz-_g9hUAuu-S26kByYl3Mll8RaPelZnaspNdgvefFW7jSI5_cNG
IKsdVaqyF1hOqla5G7ZUcKgorulG8wfv0IGJ3YCEBaSFkce5SbPy_b3sFdV5cEICt
YUSgvfwQPHqP_N9XQFb-PgJIkuY2W_Q1ABSzb2jDvIyn4ukpm2MvCN7o82DTyQdlY
4-iE1JOvbcSM_RltmBuSTDqorytO2GsdgWobnqb5MnN7hm4uiY6zznbMXXnAdtyh_
_oKfRQvWPzwZEymgQUTXWNVu_XD-qraAwixuJZ74hWae2_QSX8jBDjzpz1TUUs2b1
q4nfo6qZHx8I5sadO2TrxfmRV3KJRD20TlBhW574uKHN9PZgWqQBqDiDfwAXifDlB
Y91sdgsobTFF5-lzuDYrPQu1uHit7c5v_rfhvn2QFkYkDRovVj0EympYC8_YRjPgl
LL0vG-2VEtTN190-vOBSFpmj3Hv5VxO0eDkbP1QOCId-w0NwHU4-uf8UDlRjGIqF6
YsmZc2ZsADZJA3FJK4YQVf6EjphUuGG5UQMhmoCnj-7mhdIiiG-xQkYxV5kQ3fksQ
```

```
HeAyHyPILVcB2hzwIvttgDPRD02mjk-2vh4euskUt4yu5uZFkUvYx1PnI3_iw2hkL
Mgyzmo0m-ex6kyOotpUcBdAfjwIoccWdX-2kak_Hb1ruY-ptHbVULSaJvPFQaUbmC
Jig7zS-fcAlSOgctaSGqLRJyyQJF_jsbPVc8RT3a1GYNUA"
      ],
    [{
        "enc":"A256CBC",
        "kid":"EBQC-5ZNE-DTJU-J43T-XAAM-GLKE-TRNW",
        "Salt":"lFwvWx2QoCeEwcP5nSF4-g",
        "recipients":[{
            "kid":"MBZP-WZAZ-B6KQ-MYYP-H7KD-VVBA-7T6U",
            "epk":{
              "PublicKeyECDH":{
                "crv":"X448",
                "Public":"uPd5bw7lmtOtKJrUeQ0YkKRzlCX2W1HEDJ4Ej
uXqvUgdWtip-Q1T3vg51aRGS7EmpXwZCgyJ3vwA"}},
            "wmk":"yikmln0p4FgWoVDC_LnLe4zYS753-ZSD3N61RVv5D7tG
9NzEC2-n5A"}
          ]},
      "2U26mTitfIVITGp4p4tV3zIvlWT63ugOLHSfF6xTCOSgf4mplIHVReJJ
6bLyWVZ4QpeqEL413psn3448AYqcUwzIDaBA5JvYvwR7Bh2-lOO7rSOnRzBZ3rpme
j_5G0FOfjRF5ZCuxLYbapP0_yaooTLIs2EkDB6y1MpLvug_jdSfEgbFC8buUxXf5V
MA2jfTqbJVgP35twvoYNpJg1ABFOYUJoCW7OVmD7m2YqfHUSWa_0zN8KEtOziXYfS
Q8IZKzowpn4M5CPSTpezAISxuTGPpl5q9zp0szpgJwNqvjPW-0qcfTJBznmXjI4H4
xYSREwJOZweTlbYKwgRCIdNrtEZwq8yPVzeHgRYrHEUjd_rGwBhMNaSugOcrgvE-1
7EkQsSr0WvxpOUkJE10JUCoDYyo5s2KRPoSbLzW1WiJGRCFSRfpaL9-Nla-SGTq80
hDqhWiNWzCpyISIQZEKjLCrcdK4vEd6DlTKZaMr5B7GIRdSlkC_tprq8iDfK_J1h2
bh-viV3DAU7dZ3_yVkyHlCATpPfIRb3elBIxmZrlMchqTlT_WXP85PoJzKcYivA9V
A6m51uo1s2vzQSeaKu83w3UPfVcL4L2hcAHW0xfMERfGEVf0AzJvhfRipWxbDOem8
3H7yzX-MdZAnqJdJlwemSDTdDS9ZeYoCE76zBYSfPooELQh_0xaIp65oQYN_geySG
D5_cAW64bRezaFtpQU0XLCcZBCt1Fzwc13yiiazg4PBr0kGPhtX9py0hK606lGjle
URh9wdvykP3_6xOTDy6TEp4YkBy7bDCpWkadZukcyT-yTyWEgUG_y3fQ8yezFCfvs
V_3vr3qcuIxrnyb9Zl1xNi3opt5qyGGZ9ebcdmWnJZa9Ckbe2uvBBUAu5kwiO0j7v
OdqHTphg5Kwdtj9wSfmZuWfmcNkv4e2lbe_SvEZz7IK1ZtbKQ8NzClKEZ5YslI8vZ
YPS2uigXXIqtF2MBXWoQr-wKK32OX_wlWEzSd0APiJdspzbfQP-56eYwdyKl0n4Bf
A61gL6ZqX-oGiNlUi8srS0OHlaxaJPqXzO2_kzu6GwowGu9X33yNACTU8kLFkuK2T
zPIcnbuWb8f6gqkDnjh_QIlB2Zcj6U4RpmiVL-xMHzzaqaKDGnxBchb2K_i-HTlT-
bxUYqCpvcw7eJx1-OkUY-ucWyOykVgpC2j-6dWzhrbrcUVyqIojYN0RUGTPQrcSJ3
NXaLM6AwMIUmbKLv1nxIZHyoYG6xIQrHUoXmHES7Q5dNp_LJ5nYIKG58CXEqauDat
ABPCu_BQ88S2UP46US-5xB0u1GZaRT1oV6JsrL9A7pGu1-mhpvpzLg8MdhuV4v9_4
rhO3NiOovR5AXRraiwuBSN7bav-OX14_D2_V2zYoNrIuClT0O-b1Zvkm_xcY7J8DX
42slBpIrSQB4labT9dwm_wmaVSJaqYOrNE7DpCaYS51BUFtwrixicxnN04aaPzZ2O
xxcQ11Rvk5zFMnqwEbAKkbMJ0i5BG6I__gmJiUP7tCOb43T7SOfefMwBjM7V5y5R7
mzTnFiBudEg0DjcG0_2d_-FOchbH2yRRzpG8Wxcd5kjBHi7DTgFvrDvvdi5XoB4v3
dkS7QDM2CvatSPsrrEaeIvWkcQXXm0eNBIJH1o4TFjwQ9GJ3FczQygun_LCof1oSe
dnXRfjgaRdt6WHHUE1Mlh694IQ3PfIA2MAsCus26RB9PyGjLNHo2KWdI4Ehs6FVDE
xnflIPE2UmRfZhU3X7CBUeiKldIaps1NWhizk8BoATi3stUTXpQx5rqUMsOrBdIwJ
6je_C2SehyarO2pI6FsBwZJ7emur_pAmkGsa0j8er4kNe7Vbp9TN3jVqu5KXd2_kL
OZcQJsXpXwi_bBPy22qO_hv3sPp5gkBPNYXw91XJiyBjd9boyQvXV43s-x2PO0zAD
SnhTUbZ-K45sc7MOrmFqOPmrNz0QYp0dSA6w0MhGeGkHGZrNdJngfvm3y04nM_21k
```

```
hD8-7OXxAbtq7jLfHmtQqjpmz1bceojWJ3wFHEuqg-ELI0pS1COVm_xGK95K2GRvz
P0chNrZxNaDT5tDL-5lvhzhN3beYKq-BTsXQ-p76xAPlRcEgBs7Y-l4MOtJgU2pa5
p9eiqVCYaNf6B72gHsNXY5sM7xyRDlBSDPLyR6sgqs5UVCHap-t4PVasnlO7vdQn-
Hci_q8KROFwawkrvHxDmkE_D0Mn1aokf3uElprEW3alrDaFd6AN2rvsFj2etl7M40
LBiJoYNqh2M_DUJ-d9Z89grHP4xi64QUNuwo4Y57v3KUSotKDECaJuZ73Ux0O70Hf
nF6sv4RbIjI8OWzxn0R3Ur9EsrfGlRyYAESeM48AN4xqfrpGT48bSCJBiwIXBK4k4
Q8cfy7draXuk6VZmIlB--bEd8U5b6QgoFgzqmXKtIr6Reh6H04cqqpby8RL_rSRkJ
18SBvv0aQPmlCdjx3m_fxDaRsYF-I3Sk4VpuiZ62YS532bsvtcVo521D7U9A1Gj-X
jrxC61bnHrQ_4FQlk5vRs1cK5yK7HFfFMERkomBmD1E3IQW0BLF774R6NJMD8Mbxw
JHiOiHqQLFnmNRvjPomjd_bc28yFNcGdRpqW3g6upZr-8GX82MM4EKKdwPamr5QsX
tONeIJxNDEVTTXnyHM3IMekbwXA-aMhRHMcK5mXGMnXL3jWOgPwEdEON48dyu53y9
osbusDhzhRINDTD6uhCjsXuGlGnhv_YQV9S-OxF9FVB2Ofjciwwg8J7SaGILqU_20
t6IP5N4ciN9zoBd2bFqIikl9Esp6HLnMLxboq-rpq6qJzABptj1LNywchMMzuFxf8
RmOvUD9F7x3EhblzoaGvAsItI-cuZFZxpeCUQFRFTxvxx6iKNZDwiYTBdSHIejXkU
L2Xqg5yNPJ23-B5q3px2wsUVyQRi336A7s6mQJuH7OWOTDLK8Ppcb_gBzwt-ndEVC
ZPE-RSpUZMDFZ6tbtKYl4P9cq9Kb0Uby2MLMLQao8vRjECTrEA5-q48rgN5kYdkCp
oI6uLdCGIqVzrIy8oHUDDzw3G4ROBJlTk1YVxn29fOvBKX2OYCuxe8bz6kktX8vSd
3vf3OA0-kWOSEwSDL3YIhbXIgVTeB7vqo-Sam-nX_IVvu0F5JD6tIK_usSkAbIrEU
flu0fV7eCQN9exXc3cVQUWRLjKh1JkQRl-f_jdKb2xMNCQ"
      ]
    ],
  "AccountAddress":"alice@example.net",
  "InboundConnect":"imap://alice@imap.example.net",
  "OutboundConnect":"submit://alice@submit.example.net",
  "SmimeSign":{
    "Udf":"MBFI-KY4H-RDBR-TZAS-ZZUP-GRQD-VGDK",
    "PublicParameters":{
      "PublicKeyRSA":{
        "kid":"MBFI-KY4H-RDBR-TZAS-ZZUP-GRQD-VGDK",
        "n":"1tp65TuDE-Bg1ALU15QM1bK-78H6oMMYZcjdCnVjynM5wYIdvb
ZG1pPexxnkjWyHx55qAS-C1dNAQ-rqCWezpk3klfwIwrFVbOnVP9fZrdFPnWLZZOC
y31mU1VGhO55TjoZrjc8g7uxc-Ea5aw9sA0Im0H5nGwtinolHsHYO5aZq_pYG0D3S
LdXkHzyyVfbrQV85iE9_szKN7OGAvlA-JxBJ1M5dLrEmUvBo40fiZvVgv1H2IJ8mL
HYJC_5fSUL5-0suIzEGrCgEoYpHLVF2YcxbHSKi2huplGyWqau80F9R6wmSCZKIjN
gTPfNeceOcN4bNkiP8FinNVcd-TnVEIQ",
        "e":"AQAB"}}},
  "SmimeEncrypt":{
    "Udf":"MA4K-FLCZ-MITB-NDNH-UUVK-IBRT-P3MC",
    "PublicParameters":{
      "PublicKeyRSA":{
        "kid":"MA4K-FLCZ-MITB-NDNH-UUVK-IBRT-P3MC",
        "n":"2bUq7peCou6gvvFFSgqGs6eLvSfcSLyl1sgZ3zKWb3vQd2K6HO
Ia1R9qht7lsypsbVfY1VXNN_Oku2t-dfmlq0G6vkvIgz5tpB4zCcQudum9MKNavbd
ieWHAFI6iVCtK6ugbPCMX7yZJwAnI0ghOTj1ICZIZ_oG9NXnlL3RAgclp-Qtw8t1v
jE_yTn1iBEUuOX0MLumQ1QbPwj_-oOMv5cU1y9RJhQDk0X66gcDOoFdInRHZX60Yh
_ojYrtVMlY66-As3sbRpJGCg69tNnHQOxoAAZYa2nuJVoQoV4Rs4zK-fWvbvXWFvZ
dcW9Ni8gqs1U13_2shC_f-wKCbMQwjEQ",
        "e":"AQAB"}}},
  "OpenpgpSign":{
```

```
    "Udf":"MBWE-RBKQ-2FVU-4YYB-E23N-ZRXC-CEOI",
    "PublicParameters":{
      "PublicKeyRSA":{
        "kid":"MBWE-RBKQ-2FVU-4YYB-E23N-ZRXC-CEOI",
        "n":"qCGk27z6pWkMB3JTTz_VNJsp2iTIon1lDThZpD66zPIweV573L
FQdziNyUt3LfZ0g3gNNRGaYu80cU8YAq4hLDggWF1Vcbh4vDhMNgnPy3Mx4l1F62x
s8nbxJSqoZwboBtp_KZoGF4yeaDuDW2Mn3DMYfJI4iFm6WjHIPxP6LFUg3hYO6Edx
uesvxS80fnc_xmH9RgMhxf4JGf1EFxBBXz6SJ4wZLYHuFx985tdEFmQdDEvZIi11g
03s5B-3S8SL15uEr945aai9-zo6IbLuuVfRlr2ycWc2fAadv4K-P76IfpigCQfdls
dVG2Q23LFw5mzHWZscQ6nZsWoeEWVL-Q",
        "e":"AQAB"}}},
  "OpenpgpEncrypt":{
    "Udf":"MDNE-BRJE-2RCO-T3BN-2KTU-NU6J-WSPU",
    "PublicParameters":{
      "PublicKeyRSA":{
        "kid":"MDNE-BRJE-2RCO-T3BN-2KTU-NU6J-WSPU",
        "n":"4qQ0ipjyNkIgg3xWU1e20tFamnda1vqluPa6KSQTCmHUNxHegV
GHHBU9yyL3I0SFca7T1a20bs5KLMvx4ITz-pxebDIhs1hs6pTdzicWSuk8zFUhM65
P1VyiHXZn630Rlc6MzMZT_WoGsSFTf0cMhbsOk0Z5-mRtWPJX88cAT3hXxeWOuOTc
_3PZUWIYhwo57txefvNqpMVjfcxCOF9gFJhT-uyl1tYYQ46cOcGOczKTdO2gkziE_
P-xhS5sQVnvJJUxqvH7XnvZ5O_3BqlLpaxalceSmC3DkaQs1vDpWaCNb9VfABAaQg
ynowqslbPRBzuFwlD1FbiWnxnF2XnAQQ",
        "e":"AQAB"}}}}}
```


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":{
    "LocalName":"ssh",
    "Key":"MCXP-WQVY-RTKQ-ZU6P-VOM4-7U6K-FHXH",
    "Grant":["web",
      "threshold"
      ],
    "EnvelopedEscrow":[[{
          "enc":"A256CBC",
          "kid":"EBQG-TSDD-KPUM-Y3KS-TSIF-OGQ2-UIBE",
          "Salt":"KO-vj1hCiJn_L7gETkIiew",
          "recipients":[{
              "kid":"MBZP-WZAZ-B6KQ-MYYP-H7KD-VVBA-7T6U",
              "epk":{
                "PublicKeyECDH":{
                  "crv":"X448",
                  "Public":"G5PYCVsNi99zjwXBuxbzxS-yOeBeYWApIrHvM
  xPSOttBQS5wLqj6Q7x8xP-7B0c_Cbk8qwShNE-A"}},
              "wmk":"-K1Ovu8TcHok8Wo9BAHoLwaDUkBxhMDJ6FpS8vvvhQSf
  v-VEjLyw6A"}
            ]},
        "4eG28l0r231JLShpB1X4NcOjxVUp8X-LBaNaAEROX9Ngk9A-8u1ONoWr
  KCbJlRBG8orgqdSYwHEj5SruwRXO9uAkb8ru1xvg2toik5T1od1YaImeD2gY9mSTD
  iWEeUSBl_E909W1OOULjd58zjWWlhS8vIiNhlzWoQSJXMD78gyUtZhjfkoJ2mT_It
  _zCVpPmT6uAWPYlX3n33wH8Hw5b34VyF1NLIJh6Yho_6bVO8wR7kAGoOYJCs6N3V4
  JFPrmnhpZyCEF1qJ4X3quCPZchpnQsoRMtF10XsWbuaIT7sYxdh53Tf1JAnvEgrZY
  keVRORTDQlqtNJtOS1lHmBusg7NSIbv7vZOXFExOT8fQrze3Ls5QFS1HSglQN-qUR
  e1ZfMz6CBIoiZ_q-ctvkQtkMBTxWR5ABoZjGZg0aSsCt_o7JwUoDnm14hyX9Ptzw7
  0hbyTXJE1_JX2V4dIJ4YpdH8HtdUIKfB5c-_TCu-ex1B850UI7LEqqo07FTuNeWZ0
  OjHEf1t4gos2wjSThNnFcn6TY4XxXrp4KSa7Uw9060gFrYqIYDvkiGs8XabMr_Afb
  n3-9xxrHWDgzvDn3n5lkEEf-omH8goD45m-UzgVi_1fJTrNQePcJ1Js6Jb4xqPw0J
  UC2Rp-zB6nA-MzdFLnbhOVaF6l0oX-nQxVNhiVml4ABifIiDz0hDIK39aC9EzYESN
  vYJU5OaDZ_2yIfC9ADC2WabkeRgYP7-imVcBFKcARTIgcj6--DTDnFtFc4hoS_UZc
  hnuKW1PMc-AH4pej1VjnEYMG2Ch4-UDvWDu5yJLiR2asFxn1R84bcrCJf6qCZs-nX
  6xG6nzOiHo1-cDDOTqB3pvm6Hauvo4RRFtqqjy1Tg-VlY9V6kD4TfhgQKLkLfTHqe
  MRFZiVjVS_d6n4oFnPE85y54As3XEHu0P06bT47GNZJ352XFZiXK477F_5gzmRVbc
  kcHLjbmdqDKCAzKzGp0ah3VyCl2TidCEq4_qKveEMcXLehB1kPrfEzef5DtWkzRM1
  ZvahMgW3uAtNzp_7po9BrWeuBWqmrTWbvWYMDuzQktlYi6bO6uN1vPV6msQCRs_f4
  fPoBcOItMS1bjQgfgRSuLr76qK43TzoFMbCldHcZv6gZGUphiQS5BqGgWqFneJzu0
  hZJVKbPxgNZ18Xe6kOdeJNKk2TbAkQ9HfdMZ3QcAcFaGU8WjhUqWnCWmA-GEPJat5
  tO_BtovwCY_phpkbbQVyDhJvhAHYp43zcwNTNbss81FVNJPfv-bibLumK6w2oT9yk
  pLm7pHWYY__TaMl3w5zeSL7Dxbuknfiv5-SY-3o6_5s_p8_57H13TAhub0cP303DT
  uZf1OXexPGRv3zrloeXgb4tDKFXMDihE1qwdBvY00Zl8Y9-Ku3mW9M1pP6nBuHcOR
  HSbzBLjgDBMS7jz1esUvrO8wLNQ1g38o0Ja2EbtRE8ghPOE3SIl_QHl5V4fNCt17Q
  BZ_yld26leyOjYBQWApYtsEoV058Su-IpfWsC166p5O0UOeZ-GmlYoGoVodjDr002
  sBKbcFZStEM8a0p50EevhtMrPxd6sQaf7HDc422mYI2649dibVxWDgajnZc7NSE2m
  j5F2zyjkpEt3yqSSqFY9eRlwOInNtr1PG3d0bwdNqAECZnAZqIBCrr6SVrn3bgyaF
  RTpDMQRJt-vIKLfRBbJw8GV6NRNovv24VCPYCIhfKfHoAV0rvBZr-qR3MTWEbWkTA
  AUPneGZKgSVDUphKjd8vMZwnrT7xauAcxLAEl_K7bvGbu05aFgkQM1EICn42-VSfs
```

```
UDhYv4ZsOdOPEymKrgzr3Pb6N6pKz8YuznB6RmmBhkNgz8_DHGbFVgaovMLpk9ZL7
0wMVh7hiX5XcgAvk8b3ZarwbniOdERXE4-zRw_j7Rnt7twmfFSDVInPhFPIciFixg
R4hnG1Ecn8s82Q-3QJ9BrMBE5xWScoJh_BxeAk1LE2Epb95UkQBV6b9Xp5gz-6xOJ
dBYsJCWVBnWXHl9OQNeta9TQOeM-7k7Xc5n3HVkn0q040SsL2FNTxaFUgwZW4dSCp
XZPzaGFRdDG4nr71OEkBBQD_Lpv7UzSdAQcluSxwfT7DdxnhktoSx8yMFR2KIDtVI
UmDN4EfulAKiD_fZJQllNRk4GndN-ePAZGgZr3mHZqTguCmmfC8y4qPQIR3OHofJx
rWHU-uzZDKN2Kq3TydR0GQxLkL-fCW3ejebtbU_q2xDDlKUpVzCYEeP8PuwYawCak
adYsTFJrVzPSybOQrboldGk-PyLTQ1vY45Xp3I4tiGnKWrEBM2CJAmT_vLEa77ru9
rM3fY-Z2WrBOILIL943PxPHFx2aqQ2s0W6AAf1grIIi8sTLL7GEyhsqTO_Xzui-q9
5ZBbrz-mpy1MMphpNAgvv9hz49vyEvmQZY8Gd1M7IPO0DGaL14tNPn3gcmCiQ8CCZ
9NxYLCAzfaqIUMZU9BukkAG9pO4LY3amI_cOlFInKbSAmqcTFLcKJFPOFhskxqubG
dr8VNH5MdZUM5bmoiqZWvMDt-az39M_MZYAfWvy9opM0-oae1nI4Bw2Kh4aoteIOi
mEi7kbucpth03r7VN46n5SXf1GrbKR4LsDAWyROBURvRLDthbKP9a2pt3MWuGvgFa
W9ntaSx51LKf471vyvtFkmX_eJslRZGyhDt21Pf5iJ3RO2bqYGpkxoFDfP0iJYTvs
Soz8MiF5KrGFB213k7aXkzhQOqnSuIpzeIzKW6SJjqf7O_mGugetW5CCZq73H39zZ
BqzkeQtti0ZVmIsnvcsOSRs0Nc7nRxocka6W9HexdE_HOkPajl_fND8BloM5H13zQ
raZuxfV_K3-yNgltDBMEFPtAgVWgE28Pvame14HDFfmDMoLVjmqyjhVv5JBcPTeCD
Tph99ZFh4285NzyOo4PUPUIOBO-XzRn6MmsaDh7ySmtdNEDYdJTIJEQDpHWTfAXoi
we0Ijd0anDgDg55LuGhyhafRlE57pZYuIEc1gioFY_uA_xm6g4HSTVkcN99r-M7x0
tl2l4SIBF24lpUiW-wMMLpRBWHQPFaG-HeK85oBGrnE4kMVlPb7ax7nQnpd1hQd7m
_dW9x4Je5-nZGInlS7WC4iL4_hu0RPpUcsHaBUAM4wjLsGpPftg8YW-RrmL0VHToi
MY6HhB6lbObwQvSQgXjA3DMEYBCfZ52wtc5OKQd8R8aVrw"
      ]
    ],
  "ClientKey":{
    "Udf":"MCXP-WQVY-RTKQ-ZU6P-VOM4-7U6K-FHXH",
    "PublicParameters":{
      "PublicKeyRSA":{
        "kid":"MCXP-WQVY-RTKQ-ZU6P-VOM4-7U6K-FHXH",
        "n":"v0EWseYtsQP3dC_eBaDEK76z7Sg_fMmYaMiq_WrR_tJJvcxxrV
3rHFLAuqg4NAH4evuCjq99W07T4PLNNR3Dee6HrFpf9ktKplHina37_ZqvOUbpLSY
DGCnV_4ghAun1qYcyREcZ-x88NuXbHSni09k2KAc5HxSfKQPuhUOnTBcK8xR83psR
u4jpYTM31Djga8iFVJQRaC9t0Q1aD3BXHKtak3mMMV0GGYBX55xLcYTsIggXLEmOx
ZhJqLgY3pNE77jIqmyWL8aryPBVrdYIYne8uNSCaDa-mE-ao_9jsjGYseOeTrkJ6g
1Ne1CpL4iiNzpJmP4kAI_3Si4jJk8xyQ",
        "e":"AQAB"}}}}}
```


### 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":{
    "LocalName":"Sites-1",
    "Uid":"NDU5-XXSS-6KLM-MO6Q-S3F5-SJ7P-FO73",
    "Uri":"http://www.example.com",
    "Title":"site1"}}
```

### 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":"MAMQ-ETEA-JBL3-6UKE-LRNT-DGC3-OIDF",
    "Self":true,
    "Contact":{
      "ContactPerson":{
        "Id":"MAMQ-ETEA-JBL3-6UKE-LRNT-DGC3-OIDF",
        "Anchors":[{
            "Udf":"MAMQ-ETEA-JBL3-6UKE-LRNT-DGC3-OIDF",
            "Validation":"Self"}
          ],
        "NetworkAddresses":[{
            "Address":"alice@example.com",
            "EnvelopedProfileAccount":[{
                "EnvelopeId":"MAMQ-ETEA-JBL3-6UKE-LRNT-DGC3-OIDF",
                "dig":"S512",
                "ContentMetaData":"ewogICJVbmlxdWVJZCI6ICJNQU1RLU
  VURUEtSkJMMy02VUtFLUxSTlQtREdDMy1PSURGIiwKICAiTWVzc2FnZVR5cGUiOiA
  iUHJvZmlsZVVzZXIiLAogICJjdHkiOiAiYXBwbGljYXRpb24vbW1tL29iamVjdCIs
  CiAgIkNyZWF0ZWQiOiAiMjAyMi0wNC0yMFQxNjoxNzoxN1oifQ"},
              "ewogICJQcm9maWxlVXNlciI6IHsKICAgICJQcm9maWxlU2lnbm
  F0dXJlIjogewogICAgICAiVWRmIjogIk1BTVEtRVRFQS1KQkwzLTZVS0UtTFJOVC1
  ER0MzLU9JREYiLAogICAgICAiUHVibGljUGFyYW1ldGVycyI6IHsKICAgICAgICAi
  UHVibGljS2V5RUNESCI6IHsKICAgICAgICAgICJjcnYiOiAiRWQ0NDgiLAogICAgI
  CAgICAgIlB1YmxpYyI6ICJuaTg1UWphTTh3VTV2Um9LbXdueEQwRjljNFNLMzAzTW
  swR2FkNVdsSjhoZ0JpWVd3OW9OCiAgem1pMzJzdzhYQW1lcjZVTTBTb1RjMjRBIn1
  9fSwKICAgICJBY2NvdW50QWRkcmVzcyI6ICJhbGljZUBleGFtcGxlLmNvbSIsCiAg
  ICAiU2VydmljZVVkZiI6ICJNRFNLLUVVSFMtUVhHRC1MS09GLUFWQzctVjJSSC1MV
  jZaIiwKICAgICJFc2Nyb3dFbmNyeXB0aW9uIjogewogICAgICAiVWRmIjogIk1CWl
  AtV1pBWi1CNktRLU1ZWVAtSDdLRC1WVkJBLTdUNlUiLAogICAgICAiUHVibGljUGF
  yYW1ldGVycyI6IHsKICAgICAgICAiUHVibGljS2V5RUNESCI6IHsKICAgICAgICAg
  ICJjcnYiOiAiWDQ0OCIsCiAgICAgICAgICAiUHVibGljIjogInRSODVSQ3FXdjgtW
  DVCazBOVTRFVmxqUUZKNTg1Rk5FM1p3eVd6WFNWdEpIaXgwRlo3aloKICBRN3hnOX
  V1cnc4S09LbDVNMFVXN0xMT0EifX19LAogICAgIkFkbWluaXN0cmF0b3JTaWduYXR
  1cmUiOiB7CiAgICAgICJVZGYiOiAiTUJEVi1YWE5ILTJSVUItUkJNWi01Tkc3LUwz
  Q0QtM1RIViIsCiAgICAgICJQdWJsaWNQYXJhbWV0ZXJzIjogewogICAgICAgICJQd
  WJsaWNLZXlFQ0RIIjogewogICAgICAgICAgImNydiI6ICJFZDQ0OCIsCiAgICAgIC
  AgICAiUHVibGljIjogIkhVd040UlZoR2N6RmxPbTJiRGNldnZWWXlkNmdqZHEzM1F
  xVjhVcTM5ZEdhc1J6UW45X1AKICBWZ0NCUklfOE1qaXZlclRLZGFhRUkzMkEifX19
  LAogICAgIkNvbW1vbkVuY3J5cHRpb24iOiB7CiAgICAgICJVZGYiOiAiTURQUi1GS
  lZXLUdLNVotMkxKQS1MTVlWLVhTQ0gtSEUyQyIsCiAgICAgICJQdWJsaWNQYXJhbW
  V0ZXJzIjogewogICAgICAgICJQdWJsaWNLZXlFQ0RIIjogewogICAgICAgICAgImN
  ydiI6ICJYNDQ4IiwKICAgICAgICAgICJQdWJsaWMiOiAiNTVqVWttcW4zZ3dHMGIy
  SHpEVnUzSGxmNXNPNkdnVmxqX3ZhWUZ3QUVrc0RjTXkzd3l2VQogIHd0OW9qa2VVS
  1Q2MzA0RHdmcmgtVXc4QSJ9fX0sCiAgICAiQ29tbW9uQXV0aGVudGljYXRpb24iOi
  B7CiAgICAgICJVZGYiOiAiTUJWSS1FV0xPLUVJN0otT1ZBSy1HR1pILTZZSFctWkp
  TVSIsCiAgICAgICJQdWJsaWNQYXJhbWV0ZXJzIjogewogICAgICAgICJQdWJsaWNL
  ZXlFQ0RIIjogewogICAgICAgICAgImNydiI6ICJYNDQ4IiwKICAgICAgICAgICJQd
  WJsaWMiOiAiZlRVM1RlQjEtN0s4U1pwbzR0UXhaUHBKQWItX2QzTklkSmhsa3hXYW
```

```
lab2dKUkVLOWFkUAogIGY5S25zNW1xcjExVVRUb0lNaHpmZEphQSJ9fX0sCiAgICA
iQ29tbW9uU2lnbmF0dXJlIjogewogICAgICAiVWRmIjogIk1BTVAtQlg0Ry1BS0sy
LVlIUEEtSVhKVi1aMktWLVVYQlciLAogICAgICAiUHVibGljUGFyYW1ldGVycyI6I
HsKICAgICAgICAiUHVibGljS2V5RUNESCI6IHsKICAgICAgICAgICJjcnYiOiAiRW
Q0NDgiLAogICAgICAgICAgIlB1YmxpYyI6ICJZNi1EMkRiYktsYVZYdkc1WlF3ZUx
kNV9rUDFFQ0FDUjQwYkRtcGctWTRLczkyRk5lLXV5CiAgc1dVck1fTG1RS09JUGpq
cjVMOE5PQkVBIn19fX19",
            {
              "signatures":[{
                  "alg":"S512",
                  "kid":"MAMQ-ETEA-JBL3-6UKE-LRNT-DGC3-OIDF",
                  "signature":"FOqGS7sd-l-iXeW0NnWOIUbmJxw0SLBH
k_F4VYya8AIu23JVKebgbH-MtSAK_-0FVuXyWcRUdT8AsHeGljsGe7Y9tN4q_NT8t
IASs9ZsZa4HXUyAB3vOzMuSO6wi5bHehc-zWhkEPZhvdiBMcizkODYA"}
                ],
              "PayloadDigest":"pbnx3FGeWuZWOrANRD5vo3UYnkZRpHGm
pLwSWVJnsNZ4SFe4qVn-hfNrZ557hnJhp4aD7EN2p6B7IVNMmuK_9w"}
            ],
          "Protocols":[{
              "Protocol":"mmm"}
            ]}
        ],
      "Sources":[{
          "Validation":"Self",
          "EnvelopedSource":[{
              "dig":"S512",
              "ContentMetaData":"ewogICJNZXNzYWdlVHlwZSI6ICJDb2
50YWN0UGVyc29uIiwKICAiY3R5IjogImFwcGxpY2F0aW9uL21tbS9vYmplY3QiLAo
gICJDcmVhdGVkIjogIjIwMjItMDQtMjBUMTY6MTc6MTdaIn0"},
            "ewogICJDb250YWN0UGVyc29uIjogewogICAgIkFuY2hvcnMiOi
BbewogICAgICAgICJVZGYiOiAiTUFNUS1FVEVBLUpCTDMtNlVLRS1MUk5ULURHQzM
tT0lERiIsCiAgICAgICAgIlZhbGlkYXRpb24iOiAiU2VsZiJ9XSwKICAgICJOZXR3
b3JrQWRkcmVzc2VzIjogW3sKICAgICAgICAiQWRkcmVzcyI6ICJhbGljZUBleGFtc
GxlLmNvbSIsCiAgICAgICAgIkVudmVsb3BlZFByb2ZpbGVBY2NvdW50IjogW3sKIC
AgICAgICAgICAgIkVudmVsb3BlSWQiOiAiTUFNUS1FVEVBLUpCTDMtNlVLRS1MUk5
ULURHQzMtT0lERiIsCiAgICAgICAgICAgICJkaWciOiAiUzUxMiIsCiAgICAgICAg
ICAgICJDb250ZW50TWV0YURhdGEiOiAiZXdvZ0lDSlZibWx4ZFdWSlpDSTZJQ0pOU
VUxUkxVVlVSVUV0U2tKTU15MAogIDJWVXRGTFV4U1RsUXRSRWRETXkxUFNVUkdJaX
dLSUNBaVRXVnpjMkZuWlZSNWNHVWlPaUFpVUhKdlptbHNaCiAgVlZ6WlhJaUxBb2d
JQ0pqZEhraU9pQWlZWEJ3YkdsallYUnBiMjR2YlcxdEwyOWlhbVZqZENJc0NpQWdJ
a04KICB5WldGMFpXUWlPaUFpTWpBeU1pMHdOQzB5TUZReE5qb3hOem94TjFvaWZRI
n0sCiAgICAgICAgICAiZXdvZ0lDSlFjbTltYVd4bFZYTmxjaUk2SUhzS0lDQWdJQ0
pRY205bWFXeAogIGxVMmxuYm1GMGRYSmxJam9nZXdvZ0lDQWdJQ0FpVldSbUlqb2d
JazFCVFZFdFJWUkZRUzFLUWt3ekxUWlZTCiAgMFV0VEZKT1ZDMUVSME16TFU5SlJF
WWlMQW9nSUNBZ0lDQWlVSFZpYkdsalVHRnlZVzFsZEdWeWN5STZJSHMKICBLSUNBZ
0lDQWdJQ0FpVUhWaWJHbGpTMlY1UlVORVNDSTZJSHNLSUNBZ0lDQWdJQ0FnSUNKam
NuWWlPaUFpUgogIFdRME5EZ2lMQW9nSUNBZ0lDQWdJQ0FnSWxCMVlteHBZeUk2SUN
KdWFUZzFVV3BoVFRoM1ZUVjJVbTlMYlhkCiAgdWVFUXdSamxqTkZOTE16QXpUV3N3
UjJGa05WZHNTamhvWjBKcFdWZDNPVzlPQ2lBZ2VtMXBNekp6ZHpoWVEKICBXMWxja
```

lpWVFRCVGIxUmpNalJCSW4xOWZTd0tJQ0FnSUNKQlkyTnZkVzUwUVdSa2NtVnpjeU
k2SUNKaGJHbAogIGpaVUJsZUdGdGNHeGxMbU52YlNJc0NpQWdJQ0FpVTJWeWRtbGp
aVlZrWmlJNklDSk5SRk5MTFVWVlNGTXRVCiAgVmhIUkMxTVMwOUdMVUZXUXpjdFZq
SlNTQzFNVmpaYUlpd0tJQ0FnSUNKRmMyTnliM2RGYm1OeWVYQjBhVzkKICB1SWpvZ
2V3b2dJQ0FnSUNBaVZXUm1Jam9nSWsxQ1dsQXRWMXBCV2kxQ05rdFJMVTFaV1ZBdF
NEZExSQzFXVgogIGtKQkxUZFVObFVpTEFvZ0lDQWdJQ0FpVUhWaWJHbGpVR0Z5WVc
xbGRHVnljeUk2SUhzS0lDQWdJQ0FnSUNBCiAgaVVIVmliR2xqUzJWNVJVTkVTQ0k2
SUhzS0lDQWdJQ0FnSUNBZ0lDSmpjbllpT2lBaVdEUTBPQ0lzQ2lBZ0kKICBDQWdJQ
0FnSUNBaVVIVmliR2xqSWpvZ0luUlNPRFZTUTNGWGRqZ3RXRFZDYXpCT1ZUUkZWbX
hxVVVaS05UZwogIDFSazVGTTFwM2VWZDZXRk5XZEVwSWFYZ3dSbG8zYWxvS0lDQlJ
OM2huT1hWMWNuYzRTMDlMYkRWTk1GVlhOCiAgMHhNVDBFaWZYMTlMQW9nSUNBZ0lr
RmtiV2x1YVhOMGNtRjBiM0pUYVdkdVlYUjFjbVVpT2lCN0NpQWdJQ0EKICBnSUNKV
lpHWWlPaUFpVFVKRVZpMVlXRTVJTFRKU1ZVSXRVa0pOV2kwMVRrYzNMVXd6UTBRdE
0xUklWaUlzQwogIGlBZ0lDQWdJQ0pRZFdKc2FXTlFZWEpoYldWMFpYSnpJam9nZXd
vZ0lDQWdJQ0FnSUNKUWRXSnNhV05MWlhsCiAgRlEwUklJam9nZXdvZ0lDQWdJQ0Fn
SUNBZ0ltTnlkaUk2SUNKRlpEUTBPQ0lzQ2lBZ0lDQWdJQ0FnSUNBaVUKICBIVmliR
2xqSWpvZ0lraFZkMDQwVWxab1IyTjZSbXhQYlRKaVJHTmxkblpXV1hsa05tZHFaSE
V6TTFGeFZqaAogIFZjVE01WkVkaGMxSjZVVzQ1WDFBS0lDQldaME5DVWtsZk9FMXF
hWFpsY2xSTFpHRmhSVWt6TWtFaWZYMTlMCiAgQW9nSUNBZ0lrTnZiVzF2YmtWdVkz
SjVjSFJwYjI0aU9pQjdDaUFnSUNBZ0lDSlZaR1lpT2lBaVRVUlFVaTEKICBHU2xaW
ExVZExOVm90TWt4S1FTMU1UVmxXTFZoVFEwZ3RTRVV5UXlJc0NpQWdJQ0FnSUNKUW
RXSnNhV05RWQogIFhKaGJXVjBaWEp6SWpvZ2V3b2dJQ0FnSUNBZ0lDSlFkV0pzYVd
OTFpYbEZRMFJJSWpvZ2V3b2dJQ0FnSUNBCiAgZ0lDQWdJbU55ZGlJNklDSllORE0
SWl3S0lDQWdJQ0FnSUNBZ0lDSlFkV0pzYVdNaU9pQWlOVFZxVld0dGMKICBXNHpaM
2RITUdJeVNIcEVWblV6U0d4bU5YTlBOa2RuVm14cVgzWmhXVVozUVVWcmMwUmpUWG
t6ZDNsMlZRbwogIGdJSGQwT1c5cWEyVlZTMVEyTXpBMFJIZG1jbWd0VlhjNFFTSjl
mWDBzQ2lBZ0lDQWlRMjl0Ylc5dVFYVjBhCiAgR1Z1ZEdsallYUnBiMjRpT2lCN0Np
QWdJQ0FnSUNKVlpHWWlPaUFpVFVKV1NTMUZWMHhQTFVWSk4wb3RUMVoKICBCU3kxS
FIxcElMVFpaU0ZjdFdrcFRWU0lzQ2lBZ0lDQWdJQ0pRZFdKc2FXTlFZWEpoYldWMF
pYSnpJam9nZQogIHdvZ0lDQWdJQ0FnSUNKUWRXSnNhV05MWlhsRlEwUklJam9nZXd
vZ0lDQWdJQ0FnSUNBZ0ltTnlkaUk2SUNKCiAgWU5EUTRJaXdLSUNBZ0lDQWdJQ0Fn
SUNKUWRXSnNhV01pT2lBaVpsUlZNMVJsUWpFdE4wczRVMXB3YnpSMFUKICBYaGFVS
EJLUVdJdFgyUXpUa2xrU21oc2EzaFhZV2xhYjJkS1VrVkxPV0ZrVUFvZ0lHWTVTMj
V6TlcxeGNqRQogIHhWVlJVYjBsTmFIcG1aRXBoUVNKOWZYMHNDaUFnSUNBaVEyOXR
iVzl1VTJsbmJtRjBkWEpsSWpvZ2V3b2dJCiAgQ0FnSUNBaVZXUm1Jam9nSWsxQlRW
QXRRbGcwUnkxQlMwc3lMVmxJVUVFdFNWaEtWaTFhTWt0V0xWVllRbGMKICBpTEFvZ
0lDQWdJQ0FpVUhWaWJHbGpVR0Z5WVcxbGRHVnljeUk2SUhzS0lDQWdJQ0FnSUNBaV
VIVmliR2xqUwogIDJWNVJVTkVTQ0k2SUhzS0lDQWdJQ0FnSUNBZ0lDSmpjbllpT2l
BaVJXUTBORGdpTEFvZ0lDQWdJQ0FnSUNBCiAgZ0lsQjFZbXhwWXlJNklDSlpOaTFF
TWtSaVlrdHNZVlpZZGtjMVdsRjNaVXhrTlY5clVERkZRMEZEVWpRd1kKICBrUnRjR
2N0V1RSTGN6a3lSazVsTFhWNUNpQWdjMWRWY2sxZlRHMVJTMDlKVUdwcWNqVk1PRT
VQUWtWQkluMQogIDlmWDE5IiwKICAgICAgICAgIHsKICAgICAgICAgICAgInNpZ25
hdHVyZXMiOiBbewogICAgICAgICAgICAgImFsZyI6ICJTNTEyIiwKICAgICAg
ICAgICAgICJraWQiOiAiTUFNUS1FVEVBLUpCTDMtNlVLRS1MUk5ULURHQzMtT
0lERiIsCiAgICAgICAgICAgICAgICAic2lnbmF0dXJlIjogIkZPcUdTN3NkLWwtaV
hlVzBObldPSVVibUp4dzBTTEJIa19GNFZZeWE4QUl1MjNKVksKICBlYmdiSC1NdFN
BS18tMEZWdVh5V2NSVWRUOEFzSGVHbGpzR2U3WTl0TjRxX05UOHRJQVNzOVpzWmE0
SFhVeQogIEFCM3ZPek11U082d2k1YkhlaGMteldoa0VQWmh2ZGlCTWNpemtPRFlBI
n1dLAogICAgICAgICAgICAiUGF5bG9hZERpZ2VzdCI6ICJwYm54M0ZHZVd1WldPck

```
FOUkQ1dm8zVVlua1pScEhHbXBMd1NXVkpuc05aNFMKICBGZTRxVm4taGZOclo1NTd
obkpocDRhRDdFTjJwNkI3SVZOTW11S185dyJ9XSwKICAgICAgICAiUHJvdG9jb2xz
IjogW3sKICAgICAgICAgIlByb3RvY29sIjogIm1tbSJ9XX1dfX0",
            {
              "signatures":[{
                  "alg":"S512",
                  "kid":"MAMP-BX4G-AKK2-YHPA-IXJV-Z2KV-UXBW",
                  "signature":"P5Zhrm_5gMxQ2QlEQKXSDr03F6xjL1TR
CjS568xsRv_o13mr84x80mEVOUWwBVLltpaD5ezjLEGAYyjupBS1qtRVxWLLyY8w-
Vje3zocM-kn_wQgxbBjWE6GwrLoSjlKICFDO8Brg1SkZMtgpw97FzEA"}
                ],
              "PayloadDigest":"aRSD7Lw6GWggbqxAhn77PNOe2ekZNQR1
bCVj-ESSgdDH836wVdwzFXwkMe63uvysVSdtoR4mAYojoG2LU5j_nA"}
            ]}
          ]}}}}
```

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":"NC4X-EQN6-S6RF-NJKY-PTPW-2SI7-QELL"}}
```

## 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
    = NCAA-7UYA-TG2C-6XUC-UG3B-4XGT-OBIE

EnvelopeID
    = MBHZ-QYVP-T5DQ-FQAP-AWD4-FLMO-ZZJT

ResponseID
    = MB2Z-JQXS-7IEO-K5OJ-YI3P-FZC2-OGFU
```

### 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 ADFR-TEQU-3HJD-IRND-P4TS-CRBD-NI used to authenticate a an action of type Device:

```
pin = ADFR-TEQU-3HJD-IRND-P4TS-CRBD-NI
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)
    = AAV6-EBKF-JIUO-B2UV-UQX7-OKHB-OAAX
```

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

```
Account =  alice@example.com

PinId = UDFPresent (MAC (Account, saltedPINData))
    = ADDU-7BE6-DN7R-U2BB-VST6-DYZL-YEZR
```

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.

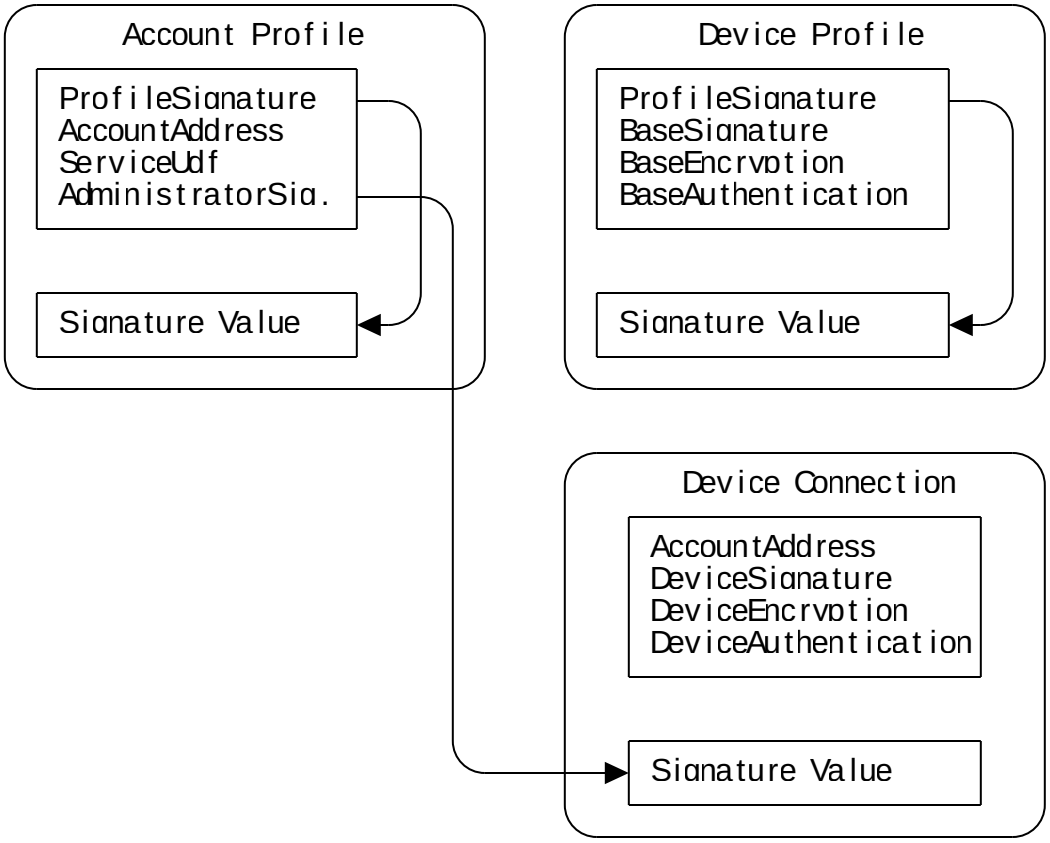


Figure 1: Profiles And Connections

### 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":{
    "EnvelopedProfileDevice":[{
        "EnvelopeId":"MBOB-5GVY-Q43B-KODG-UJ3E-LY7V-36UV",
        "dig":"S512",
        "ContentMetaData":"ewogICJVbmlxdWVJZCI6ICJNQk9CLTVHVlktUT
  QzQi1LT0RHLVVKM0UtTFk3Vi0zNlVWIiwKICAiTWVzc2FnZVR5cGUiOiAiUHJvZml
  sZURldmljZSIsCiAgImN0eSI6ICJhcHBsaWNhdGlvbi9tbW0vb2JqZWN0IiwKICAi
  Q3JlYXRlZCI6ICIyMDIyLTA0LTIwVDE2OjE3OjU3WiJ9"},
      "ewogICJQcm9maWxlRGV2aWNlIjogewogICAgIlByb2ZpbGVTaWduYXR1cm
  UiOiB7CiAgICAgICJVZGYiOiAiTUJPQi01R1ZZLVE0M0ItS09ERy1VSjNFLUxZN1Y
  tMzZVViIsCiAgICAgICJQdWJsaWNQYXJhbWV0ZXJzIjogewogICAgICAgICJQdWJs
  aWNLZXlFQ0RIIjogewogICAgICAgICAgImNydiI6ICJFZDQ0OCIsCiAgICAgICAgI
  CAiUHVibGljIjogIkZXaWlfWUV0VERYNUt6ZUQtLW44QW5LcWlFUFQzODN6YWZPOW
  VFREt0QjNjc2pMa2VaV2UKICBXMjNhQlEtd01pZFVNLVZGX1VsYTFtSUEifX19LAo
  gICAgIkVuY3J5cHRpb24iOiB7CiAgICAgICJVZGYiOiAiTUNLMi1PRlNZLUNBUEot
  RVpVNS1LTzM3LUlJTkMtNkhYTCIsCiAgICAgICJQdWJsaWNQYXJhbWV0ZXJzIjoge
  wogICAgICAgICJQdWJsaWNLZXlFQ0RIIjogewogICAgICAgICAgImNydiI6ICJYND
  Q4IiwKICAgICAgICAgICJQdWJsaWMiOiAiNkNwVFVfWlp1QWE3bENOYkE4ZUs4c2h
  EeUdsQy05YldXckwteFQybTFZNjcwZVpFVzI1NwogIHR2SnREVDFLSTN3aXotaXB0
  bjFBVHBhQSJ9fX0sCiAgICAiU2lnbmF0dXJlIjogewogICAgICAiVWRmIjogIk1CS
  DYtUEQyNy02Tjc2LVIyNTctQlUzTS1CUUpYLVFEQlMiLAogICAgICAiUHVibGljUG
  FyYW1ldGVycyI6IHsKICAgICAgICAiUHVibGljS2V5RUNESCI6IHsKICAgICAgICA
  gICJjcnYiOiAiRWQ0NDgiLAogICAgICAgICAgIlB1YmxpYyI6ICJXV0xIN0hjb0Vl
  SzdhRzMtYWdMdHI2UlltWTJnYWtiekNyWm00aWppWERGbXhWVFJIamJlCiAgaUItV
  1dLOS1JVDQydW5OaHRXRmxPdXdBIn19fSwKICAgICJBdXRoZW50aWNhdGlvbiI6IH
  sKICAgICAgIlVkZiI6ICJNQlRKLU9CNEYtQVlIRC1YQzRJLUpaTkctTUJaVS1ISTN
  HIiwKICAgICAgIlB1YmxpY1BhcmFtZXRlcnMiOiB7CiAgICAgICAgIlB1YmxpY0tl
  eUVDREgiOiB7CiAgICAgICAgICAiY3J2IjogIlg0NDgiLAogICAgICAgICAgIlB1Y
  mxpYyI6ICJWd0hYcHQxdmZKV21zNUNjazluc2dlam92WkxOa1ctcEFxalpHdkdWNW
  5lb0UtcnVyZWJDCiAgaTdYLTR3bnhxbXV4RkxIVHF5cFdJRjhBIn19fX19",
      {
        "signatures":[{
            "alg":"S512",
            "kid":"MBOB-5GVY-Q43B-KODG-UJ3E-LY7V-36UV",
            "signature":"m10FQkPJzhAR2Cg2VfPzvSUt3XyQh0yjgqggXSep
  nwz3NpDWrH6TZLNeO0Gq-moqahTzGn_ZW8aA6vuiuiqtDMy_avBf0g31nDpFyRDk6
  9D5qXBh8Br-4utT_Zxyzz3S2i63FGczDekAZTwZTQoQwTUA"}
          ],
        "PayloadDigest":"-irGyEMwNtkfLTM8Ygprqww7Lr41K_2Recre2O2H
  DP5CyC4VklJfYiDMR8822Sp5oALA-2aqQjDzJKKEt50nhA"}
      ],
    "EnvelopedConnectionDevice":[{
        "dig":"S512",
        "ContentMetaData":"ewogICJNZXNzYWdlVHlwZSI6ICJDb25uZWN0aW
  9uRGV2aWNlIiwKICAiY3R5IjogImFwcGxpY2F0aW9uL21tbS9vYmplY3QiLAogICJ
  DcmVhdGVkIjogIjIwMjItMDQtMjBUMTY6MTc6NTdaIn0"},
      "ewogICJDb25uZWN0aW9uRGV2aWNlIjogewogICAgIkF1dGhlbnRpY2F0aW
  9uIjogewogICAgICAiVWRmIjogIk1DSzItT0ZTWS1DQVBKLUVaVTUtS08zNy1JSU5
```

```
DLTZIWEwiLAogICAgICAiUHVibGljUGFyYW1ldGVycyI6IHsKICAgICAgICAiUHVi
bGljS2V5RUNESCI6IHsKICAgICAgICAgICJjcnYiOiAiWDQ0OCIsCiAgICAgICAgI
CAiUHVibGljIjogIjZDcFRVX1padUFhN2xDTmJBOGVLOHNoRHlHbEMtOWJXV3JMLX
hUMm0xWTY3MGVaRVcyNTcKICB0dkp0RFQxS0kzd2l6LWlwdG4xQVRwYUEifX19LAo
gICAgIlNpZ25hdHVyZSI6IHsKICAgICAgICAgIlVkZiI6ICJNQkg2LVBEMjctNk43Ni1S
MjU3LUJVM00tQlFKWC1RREJTIiwKICAgICAgICAgIlB1YmxpY1BhcmFtZXRlcnMiOiB7C
iAgICAgICAgICAgIlB1YmxpY0tleUVDREgiOiB7CiAgICAgICAgICAgICAiY3J2IjogIkVkND
Q4IiwKICAgICAgICAgICAgICJQdWJsaWMiOiAiV1dMSDdIY29FZUs3YUczLWFnTHRyNlJ
ZbVkyZ2FrYnpDclptNGlqaVhERm14VlRSSGpiZQogIGlCLVdXSzktSVQ0MnVuTmh0
V0ZsT3V3QSJ9fX0sCiAgICAiRW5jcnlwdGlvbiI6IHsKICAgICAgICAgIlVkZiI6ICJNQ
0syLU9GU1ktQ0FQSi1FWlU1LUtPMzctSUlOQy02SFhMIiwKICAgICAgICAgIlB1YmxpY1
BhcmFtZXRlcnMiOiB7CiAgICAgICAgICAgIlB1YmxpY0tleUVDREgiOiB7CiAgICAgICAgICA
gICAiY3J2IjogIlg0NDgiLAogICAgICAgICAgICAgIlB1YmxpYyI6ICI2Q3BUVV9aWnVB
YTdsQ05iQThlSzhzaER5R2xDLTliV1dyTC14VDJtMVk2NzBlWkVXMjU3CiAgdHZKd
ERUMUtJM3dpei1pcHRuMUFUcGFBIn19fX19",
    {
      "signatures":[{
          "alg":"S512",
          "kid":"MBGZ-R2AS-DPME-4KOZ-KKF5-WLDO-IBZO",
          "signature":"pe4KEfz7NgyGS4nz7VxBPZNcX04Fnf5EVQXCg4AO
Z_XDKD3egMEeg5cStZALTB-yOkk44XLobyWAbxbhyeVFif7qZAdZ0hdk-h_o-di3h
aX-SVPdFpGHXeCeOMaEAfsCOXTb9oSvHqDNLUaRIfq0wiIA"}
        ],
      "PayloadDigest":"oa0Yms70Z_buemEpSstfNdKSVlxUy7NoHKkZv_bA
9OX9ZJGkB3E4nNBfLG85arEixWQhkxFCwkHLvmInqkjYIQ"}
    ],
  "EnvelopedConnectionService":[{
      "dig":"S512",
      "ContentMetaData":"ewogICJNZXNzYWdlVHlwZSI6ICJDb25uZWN0aW
9uU2VydmljZSIsCiAgImN0eSI6ICJhcHBsaWNhdGlvbi9tbW0vb2JqZWN0IiwKICA
iQ3JlYXRlZCI6ICIyMDIyLTA0LTIwVDE2OjE3OjU3WiJ9"},
    "ewogICJDb25uZWN0aW9uU2VydmljZSI6IHsKICAgICJBdXRoZW50aWNhdG
lvbiI6IHsKICAgICAgIlVkZiI6ICJNQ0syLU9GU1ktQ0FQSi1FWlU1LUtPMzctSUl
OQy02SFhMIiwKICAgICAgIlB1YmxpY1BhcmFtZXRlcnMiOiB7CiAgICAgICAgIlB1
YmxpY0tleUVDREgiOiB7CiAgICAgICAgICAgICAiY3J2IjogIlg0NDgiLAogICAgICAgI
CAgIlB1YmxpY1YyI6ICI2Q3BUVV9aWnVBYTdsQ05iQThlSzhzaER5R2xDLTliV1dyTC
14VDJtMVk2NzBlWkVXMjU3CiAgdHZKdERUMUtJM3dpei1pcHRuMUFUcGFBIn19fX1
9",
    {
      "signatures":[{
          "alg":"S512",
          "kid":"MBGZ-R2AS-DPME-4KOZ-KKF5-WLDO-IBZO",
          "signature":"mGzTozZ5fDt4p9-VSDGwx6b9AUo_YDR9pLwXAj1m
oN5de75NXuZRdz_ENeTLu1AtEzyYENDaQskAho664biW8I7DuRbNbLJ_AJLXQD99b
5kiiz1Ljavg1RAdrdfH05TDGHw7eMP5aCEir_o4oS7zjTEA"}
        ],
      "PayloadDigest":"97C6-ryQFiyRF-8NAP9pX7YvJEtcz-hexhvkHgsJ
2GUEl7yW_-uhclWSu0F7eRrdENFRq8g-qJDXPJTmo8TyEA"}
    ],
```

```
    "PrivateKey":{
      "PrivateKeyUDF":{
        "PrivateValue":"ZAAQ-A5KD-OPXN-5E7X-ZXRU-CRYP-B2N2-G6FY-MCO
H-GAIH-72GR-EZXO-LQIM-Z5GA",
        "KeyType":"MeshProfileDevice"}},
    "ConnectUri":"mcu://maker@example.com/EBKG-ED3O-HBHK-ZQGS-EX4H-
X22S-X4"}}
```


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.

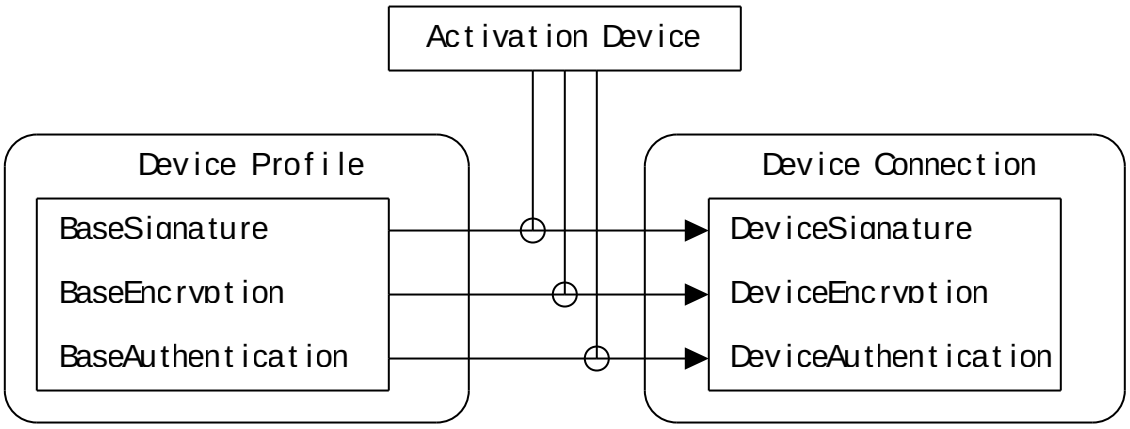


Figure 2: Mapping of Device Profile and Device Private to Device Connection Keys.

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:

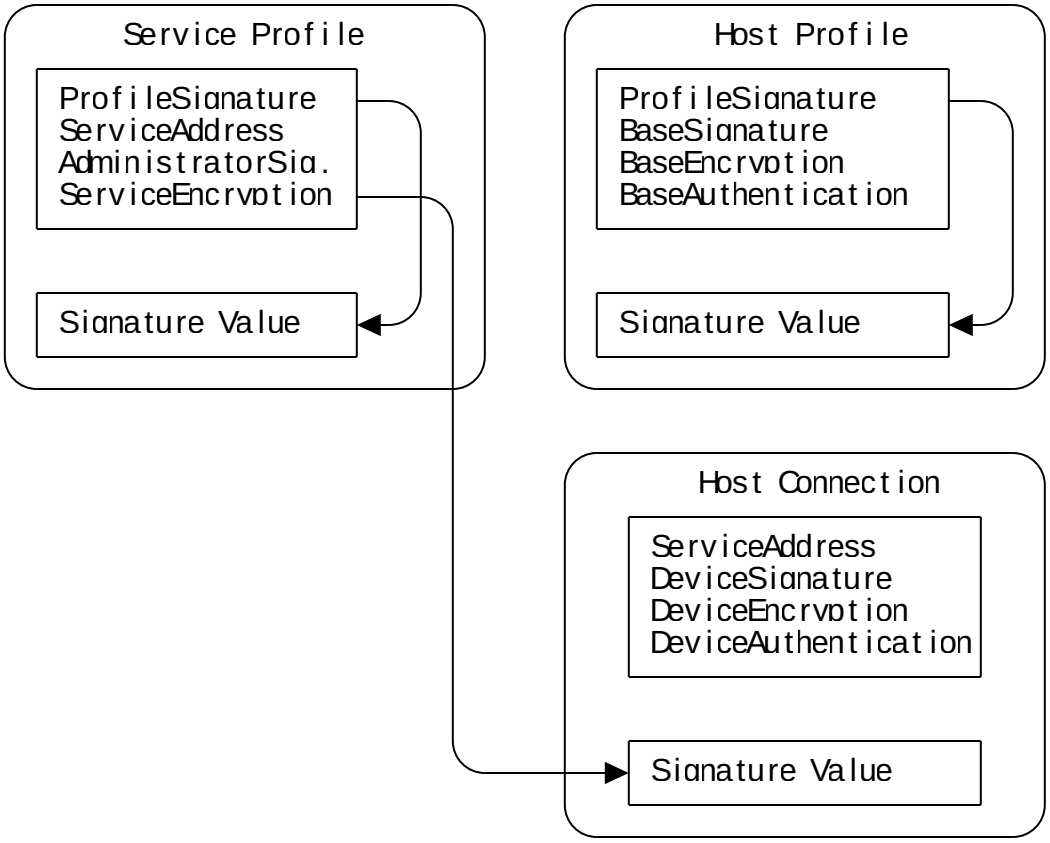


Figure 3: Service Profile and Delegated Host Assertion.

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

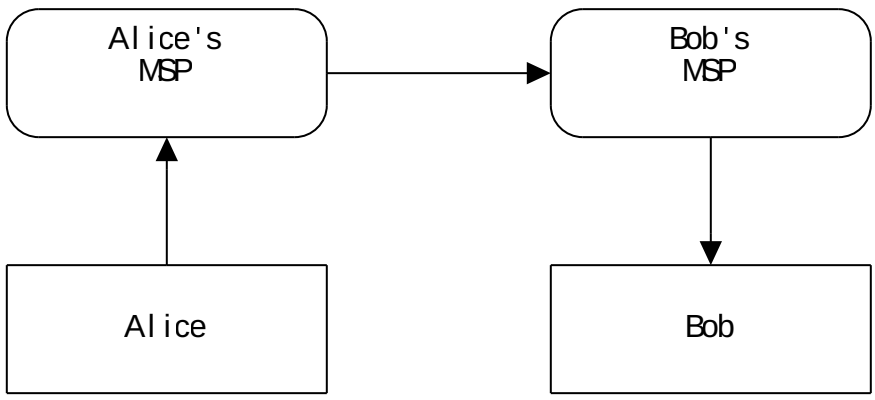

Figure 4: Four Corner Messaging Model

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.

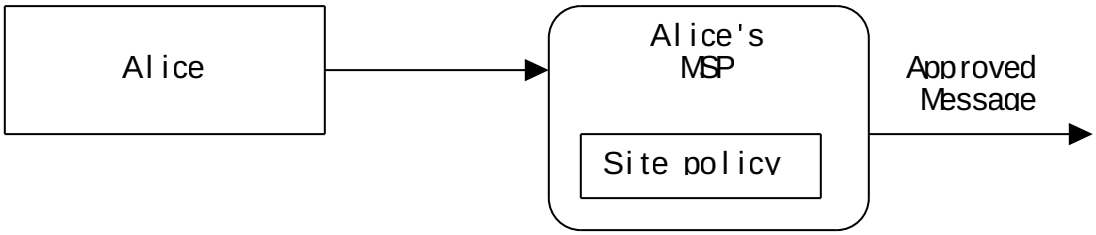


Figure 5: Performing Access Control on Outbound Messages

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.

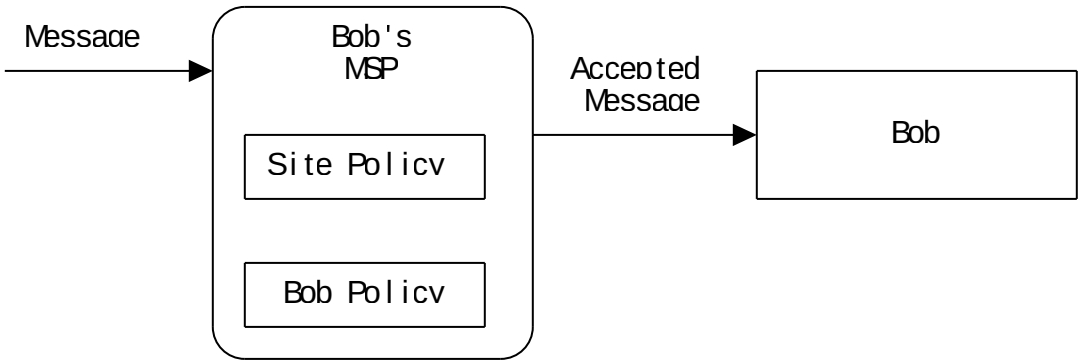


Figure 6: Performing Access Control on Inbound Messages

### 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: 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: Connection

**Inherits: Assertion** UDF of the connection target.
**SubjectUdf: String (Optional)**
**AuthorityUdf: String (Optional)** UDF of the connection source.

#### 11.2.5. 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 resources.

#### 11.2.6. Structure: ActivationEntry

**Resource: String (Optional)** Name of the activated resource

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

#### 11.2.7. Mesh Profile Classes

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

#### 11.2.8. Structure: Profile

**Inherits: Assertion**
Parent class from which all profile classes are derived

**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.

#### 11.2.9. Structure: ProfileDevice

**Inherits: Profile**
Describes a mesh device.

**Description: String (Optional)** Description of the device

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

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

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

### 11.2.10. 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.

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

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

### 11.2.11. Structure: ProfileUser

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

**AccountAuthentication: KeyData (Optional)** Key used to authenticate requests made under this user account.

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

### 11.2.12. 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.

[No fields]

### 11.2.13. 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.

#### 11.2.14. Structure: ProfileHost

**Inherits: Profile** Key used to authenticate service connections.
**KeyAuthentication: KeyData (Optional)**
**KeyEncryption: KeyData (Optional)** Key used to pass encrypted data to the device such as a

#### 11.2.15. 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.

#### 11.2.16. Structure: ConnectionDevice

**Inherits: Connection**
Connection assertion used to authenticate service requests made by a device.

**AccountAddress: String (Optional)** The account address

**DeviceSignature: KeyData (Optional)** The signature key for use of the device under the profile

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

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

#### 11.2.17. Structure: ConnectionApplication

**Inherits: Connection**
Connection assertion stating that a particular device is

[No fields]

#### 11.2.18. Structure: ConnectionGroup

Describes the connection of a member to a group.

**Inherits: Connection**

[No fields]

#### 11.2.19. Structure: ConnectionService

**Inherits: Connection**
[No fields]

#### 11.2.20. Structure: ConnectionHost

**Inherits: Connection**
[No fields]

#### 11.2.21. Activation Assertions

#### 11.2.22. Structure: ActivationDevice

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)**

#### 11.2.23. Structure: ActivationAccount

**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.

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

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

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

#### 11.2.24. Structure: ActivationApplication

**Inherits: Activation**
[No fields]

### 11.3. Data Structures

Classes describing data used in cataloged data.

#### 11.3.1. Structure: Contact

**Inherits: Assertion**
Base class for contact entries.

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

**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.

#### 11.3.2. Structure: Anchor

Trust anchor

**Udf: String (Optional)** The trust anchor.

**Validation: String (Optional)** The means of validation.

#### 11.3.3. 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.

#### 11.3.4. Structure: ContactGroup

**Inherits: Contact**
Contact for a group, including encryption groups.

[No fields]

#### 11.3.5. Structure: ContactPerson

**Inherits: Contact** List of person names in order of preference
**CommonNames: PersonName [0..Many]**

#### 11.3.6. Structure: ContactOrganization

**Inherits: Contact** List of person names in order of preference
**CommonNames: OrganizationName [0..Many]**

#### 11.3.7. 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.

#### 11.3.8. 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).

#### 11.3.9. 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

#### 11.3.10. Structure: NetworkProtocol

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

#### 11.3.11. 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.

#### 11.3.12. Structure: Location

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

#### 11.3.13. Structure: Bookmark

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

#### 11.3.14. 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.

#### 11.3.15. Structure: Task

**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)**

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

### 11.4. Catalog Entries

#### 11.4.1. Structure: CatalogedEntry

Base class for cataloged Mesh data.

#### 11.4.2. Structure: CatalogedDevice

**Inherits: CatalogedEntry**
Public device entry, indexed under the device ID Hello

**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

**EnvelopedProfileDevice: Enveloped (Optional)** The device profile

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

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

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

**EnvelopedActivationApplication: Enveloped [0..Many]** Application activations granted to the device.

#### 11.4.3. 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)

#### 11.4.4. Structure: CatalogedCredential

**Inherits: CatalogedEntry**
**Protocol: String (Optional)**
**Service: String (Optional)**
**Username: String (Optional)**
**Password: String (Optional)**

#### 11.4.5. Structure: CatalogedNetwork

**Inherits: CatalogedEntry**
**Protocol: String (Optional)**
**Service: String (Optional)**
**Username: String (Optional)**
**Password: String (Optional)**

#### 11.4.6. 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.

#### 11.4.7. Structure: CatalogedAccess

**Inherits: CatalogedEntry**

[No fields]

#### 11.4.8. Structure: CryptographicCapability

**Id: String (Optional)** The identifier of the capability. If this is a user capability, MUST match the KeyData identifier. If this is a serviced capability, MUST match the value of ServiceId on the corresponding service capability.

**KeyData: KeyData (Optional)**
The key that enables the capability

**EnvelopedKeyShares: Enveloped [0..Many]** One or more enveloped key shares.

**SubjectId: String (Optional)** The identifier of the resource that is controlled using the key.

**SubjectAddress: String (Optional)** The address of the resource that is controlled using the key.

#### 11.4.9. Structure: CapabilityDecrypt

**Inherits: CryptographicCapability**
The corresponding key is a decryption key

[No fields]

#### 11.4.10. Structure: CapabilityDecryptPartial

**Inherits: CapabilityDecrypt**
The corresponding key is an encryption key

**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.]

#### 11.4.11. 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]

#### 11.4.12. Structure: CapabilitySign

**Inherits: CryptographicCapability**
The corresponding key is an administration key

[No fields]

### 11.4.13. Structure: CapabilityKeyGenerate

**Inherits: CryptographicCapability**

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

[No fields]

### 11.4.14. 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]

### 11.4.15. Structure: CatalogedBookmark

**Inherits: CatalogedEntry**
**Uri: String (Optional)**
**Title: String (Optional)**
**Path: String (Optional)**

### 11.4.16. Structure: CatalogedTask

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

### 11.4.17. Structure: CatalogedApplication

**Inherits: CatalogedEntry** Enveloped keys for use with Application
**Key: String (Optional)**
**EnvelopedCapabilities: DareEnvelope [0..Many]**

### 11.4.18. Structure: CatalogedMember

**ContactAddress: String (Optional)**
**MemberCapabilityId: String (Optional)**
**ServiceCapabilityId: String (Optional)**
**Inherits: CatalogedEntry**

### 11.4.19. Structure: CatalogedGroup

**Inherits: CatalogedApplication** The Mesh profile
**EnvelopedProfileGroup: Enveloped (Optional)**
**EnvelopedActivationAccount: Enveloped (Optional)** The activation of the device within the Mesh account

### 11.4.20. Structure: CatalogedApplicationSSH

**Inherits: CatalogedApplication**

[No fields]

### 11.4.21. Structure: CatalogedApplicationMail

**Inherits: CatalogedApplication**

[No fields]

#### 11.4.22. Structure: CatalogedApplicationNetwork

**Inherits: CatalogedApplication**

[No fields]

### 11.5. Publications

#### 11.5.1. Structure: DevicePreconfiguration

A data structure that is passed

**EnvelopedProfileDevice: Enveloped (Optional)** The device profile

**EnvelopedConnectionDevice: Enveloped (Optional)** The device connection

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

### 11.6. Messages

#### 11.6.1. 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**

#### 11.6.2. Structure: MessageError

**Recipient: String (Optional)**

**Inherits: Message**

**ErrorCode: String (Optional)**

#### 11.6.3. Structure: MessageComplete

**Inherits: Message**

**References: Reference [0..Many]**

#### 11.6.4. Structure: MessagePinValidated

**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)

### 11.6.5. 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.

### 11.6.6. Structure: RequestConnection

Connection request message. This message contains the information

**Inherits: MessagePinValidated**
**AccountAddress: String (Optional)**

### 11.6.7. 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)**

### 11.6.8. 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.

### 11.6.9. Structure: MessageContact

**Inherits: MessagePinValidated** If true, requests that the recipient
**Reply: Boolean (Optional)** 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.

### 11.6.10. Structure: GroupInvitation

**Inherits: Message**
**Text: String (Optional)**

### 11.6.11. Structure: RequestConfirmation

**Inherits: Message**
**Text: String (Optional)**

### 11.6.12. Structure: ResponseConfirmation

**Inherits: Message**
**Request: Enveloped (Optional)**
**Accept: Boolean (Optional)**

### 11.6.13. Structure: RequestTask

**Inherits: Message**
[No fields]

### 11.6.14. Structure: MessageClaim

**Inherits: Message**
**PublicationId: String (Optional)**
**ServiceAuthenticate: String (Optional)**
**DeviceAuthenticate: String (Optional)**
**Expires: DateTime (Optional)**

### 11.6.15. 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-19, 25 October 2021, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-architecture-19>.

**[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-14, 25 October 2021, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-dare-14>.

**[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-12, 25 October 2021, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-protocol-12>.

**[draft-hallambaker-mesh-security]**
Hallam-Baker, P., "Mathematical Mesh 3.0 Part IX Security Considerations", Work in Progress, Internet-Draft, draft-hallambaker-mesh-security-08, 20 September 2021, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-security-08>.

**[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-15, 25 October 2021, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-udf-15>.

**[draft-hallambaker-threshold]**
Hallam-Baker, P., "Threshold Modes in Elliptic Curves", Work in Progress, Internet-Draft, draft-hallambaker-

threshold-06, 5 August 2021, <https://datatracker.ietf.org/doc/html/draft-hallambaker-threshold-06>.

**[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-04, 29 March 2022, <https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-frost-04>.

**[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>.