Workgroup: Network Working Group
Internet-Draft: draft-hallambaker-mesh-schema
Published: 25 October 2021
Intended Status: Informational
Expires: 28 April 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 28 April 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> account create alice@example.com
Account=alice@example.com
UDF=MB5I-R24M-QXJT-KDBF-XFOA-DGC3-U3AA
```

The account profile created is:

```
{
  "ProfileUser":{
    "ProfileSignature":{
      "Udf":"MB5I-R24M-QXJT-KDBF-XFOA-DGC3-U3AA",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"0Q-Z5eDHtwWVYdkfyVT9R36-r0hO1fUHWpmI2mdIsi81s
  djysgsAfdKoHZpKIZtKkMXSoOkFrpOA"}}},
    "AccountAddress":"alice@example.com",
    "ServiceUdf":"MD36-Q4SC-S4YZ-KPRP-7W4P-SNR7-QMD2",
    "EscrowEncryption":{
      "Udf":"MBFO-AXQH-VEJI-J47J-W3ZG-3ZPA-7FHS",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"GChyNFuHb6_BmogqEC3_R0aXaemmDlaDGyYYdl2FSAw4E
  nKjC8AqGlpy7sQacRVj4-QbQJsz_PkA"}}},
    "AccountEncryption":{
      "Udf":"MBUH-FY45-DVNF-XMQV-SQC4-LTLI-K5AV",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"WSdlD8SLXWCFHhIHjCwQHB7b4Ym74kpM-XVZnFKWYYYpH
  gBn-JIH3aPaHzd60MH3n1evVNUsTbCA"}}},
    "AdministratorSignature":{
      "Udf":"MCBO-ZK4F-QFYM-63TK-TA2C-LHQY-7QW5",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"KZPy-O5-rDXLTTo9ckiMR5mlOjkurMLRBZW5ZkUJJ97d8
  HRtTABdLn66iOfEKCQ0si_l8O75VUQA"}}},
    "AccountAuthentication":{
      "Udf":"MAHC-QH3D-VLKC-UTFB-UEFR-M5VV-TWAH",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"EmSbhqkjgjYAGR_iNHzGi_SRB6vGlKqfIsCyQvxlVf79N
  sSEEhmyPHq7zJ1AIl1eaidaS2r263kA"}}},
    "AccountSignature":{
      "Udf":"MBUX-YI5W-NTAH-UJN2-4FFC-4PAY-NI73",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"FfvEpMucwBoxAOS_-0tZUazve5J7IBXoXpjLXTPDuoDvN
  udksR_1REfgh9Hb4bIpbZjl_8l-RiGA"}}}}}
```

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":"MB33-ROBH-6WWL-J3IZ-N76K-FUMY-HLHD",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"tksYeycyGeML8xzV85WHpRVjeQYS1VpLrUxodez9e5At7
  OjAsatGrG_dD-5xyJFzAFMG0ZU6aJGA"}}},
    "Encryption":{
      "Udf":"MAG6-3VSV-DZDM-MDLK-SGTS-O2VB-IKTS",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"5i6M3dzF1HLvyOtD5mxku58Yk81XmWLUYpE--PI1En9Rk
  9YeI0BJr6KEj-wmTSui9Bkbmf_gcbOA"}}},
    "Signature":{
      "Udf":"MC5Z-6Z5Z-ZHMC-KN2B-AEH6-VC2Y-D7PT",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"v2UJt_on-FyQqBdNPLRJxdE21a365Fa4d62RWX8o1KCc8
  uQ6hN0SAzk4oO9HcXD5hLqmI8Pug3YA"}}},
    "Authentication":{
      "Udf":"MBV4-NXKT-Y2Z5-KLXI-J3XM-NOGF-LFBJ",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"NyZDUp4n1W_RZdcIPBaaSVVHXM07YWGmbInSogiaJU4ye
  39zf00lX2C7AkIlOMHqnGkXZn8WFAKA"}}}}}
```


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> device complete
   Device UDF = MB33-ROBH-6WWL-J3IZ-N76K-FUMY-HLHD
   Account = alice@example.com
   Account UDF = MB5I-R24M-QXJT-KDBF-XFOA-DGC3-U3AA
```

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

```
{
  "ActivationDevice":{
    "ActivationKey":"ZAAQ-GWNX-ZE5I-HT5I-VGRY-4Y2N-A6CN-AU2Z-RPH5-2
QGV-6QUB-DD5Z-OHEJ-ABXP",
    "AccountUdf":"MB33-ROBH-6WWL-J3IZ-N76K-FUMY-HLHD"}}
```

And:

```
{
  "ActivationAccount":{
    "Entries":[{
        "Resource":"MMM_Contact",
        "Key":{
          "Udf":"MDBM-MZS5-4PU6-R67K-IYMD-CX7P-IBWJ",
          "PublicParameters":{
            "PublicKeyECDH":{
              "crv":"X448",
              "Public":"o-GyXZGC2B9Nwe-2FQTR7RnLyw_phzjkUPWNvblVE
  1UrPEQCLbrIJHYMq_vXU-BcXDY4Uinj55IA"}},
          "PrivateParameters":{
            "PrivateKeyECDH":{
              "crv":"X448",
              "Private":"P7oi7focyX3G2jELR25VB5rlUV4SZnr2IziiWN9g
  7_9eVN3h_XT55jB3uV9qd0DWKkmrOzQDfOw"}}}},
      {
        "Resource":"MMM_Publication",
        "Key":{
          "Udf":"MCI6-ED5E-2GOF-6GWI-7TVA-QCW4-7FX3",
          "PublicParameters":{
            "PublicKeyECDH":{
              "crv":"X448",
              "Public":"axZfYpZtKCHxAVuLXW18LQ2Hg0dyZU6ahzLnEeyJw
  zUDW3IoZV98T5Cs5Lu4NhWQNgNGp9BtC_4A"}},
          "PrivateParameters":{
            "PrivateKeyECDH":{
              "crv":"X448",
              "Private":"zIrUQg1xnIXYOz9v3fyZ8fgVUntF_y9qXqtcAuBF
  kiJk5UEcdVKF960gh_EfmE2DPR72gif2jTQ"}}}},
      {
        "Resource":"MMM_Inbound",
        "Key":{
          "Udf":"MBAG-7BKG-KJVH-6WJW-HIZW-BPRG-BAEN",
          "PublicParameters":{
            "PublicKeyECDH":{
              "crv":"X448",
              "Public":"nLLbvqa6YOSly_mr8kwtxdJFDd1FQ3cDeTZghwXDJ
  0xoZIj1uKR9wGzRz1A-jScaBKMHkI9UeLQA"}},
          "PrivateParameters":{
            "PrivateKeyECDH":{
              "crv":"X448",
              "Private":"Hk6aSE686eM7IM2gP1AEt88_cUpy0ty2kbjbA_VT
  P83AaXL8D9j8kwuUWRcfGONWkvjDrZkS43Y"}}}},
      {
        "Resource":"MMM_Outbound",
        "Key":{
          "Udf":"MBAC-FFPI-RXJR-EDH5-OQ4I-QK3A-EJKS",
          "PublicParameters":{
```

```
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"IlJ-1x6trFhAQ0pcAbnl2d2est-KRLEBJQhPVBdmr
N5pztzYqQr84uIk2yO_XYVg3ZL0qph1z7yA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "crv":"X448",
            "Private":"Ku4LVVobClGn6nn7f4gobmaZZ4D7I9Njdyjv5yqi
viTFtkFGrx_EPgLN6H2KLORZrlqmQeYKv8I"}}}},
    {
      "Resource":"MMM_Network",
      "Key":{
        "Udf":"MCGU-SSLR-FWIK-BSCL-ZLGP-Z4P7-3UYZ",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"Pd4i7IA6y1BMNEl3S6GiDQJBaIoFdXJIpI-PVtlmm
DCusS-0stTUJxNx544tD7mPRdiS-MBxx7gA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "crv":"X448",
            "Private":"pFcnzS6ivQ-Hnji8RaXlw-6d97PLDAVhD6Ou8BgP
psbzPsNBV9g9mcHi7TTVplVaq44Q-3aI3pA"}}}},
    {
      "Resource":"MMM_Application",
      "Key":{
        "Udf":"MCKR-5UNI-G6FH-V7OS-RIKX-XXHL-QQVY",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"3ma-Rp2ypt-14xHHJApZV3rknWzDyq2g99rYFEp5C
yRDlj34f2XY2wnIFpgsiAtwMbQYiwFzbL0A"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "crv":"X448",
            "Private":"DVeT_V20CObmkez8BI1t855NP4YE3SYvlQnBw88n
UoqUUNqmTtkD3J4BtyA_8cla6cXbvmHa_2s"}}}},
    {
      "Resource":"MMM_Credential",
      "Key":{
        "Udf":"MC7F-DLCK-JI67-VFL7-BOHX-T62Q-KCVG",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"n2TJxtwPNoguN414nG_aU9a8o7q1wI3RMSwwKBfOC
HZYzaTJFL9R3KqIK0izYD_NfmW2m4A-6OgA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "crv":"X448",
```

```
            "Private":"ojQyYozR_a20_HERypl5xWQsbOzmLgkNmYcpF2z_
LgMwq_qSjH_2uF63I8W6NXsps13Pm5hoqZE"}}}},
    {
      "Resource":"MMM_Task",
      "Key":{
        "Udf":"MBZM-2RTE-UXMX-O6TL-KAMK-V4WB-7AE4",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"qGkDbgJVQpHtkWTRk5WuzIAiOYcNmy7Vlecskru4g
RlCf4qzKfQq6oYSN8T_RjjrS3qEnmHimPEA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "crv":"X448",
            "Private":"TOmmssCChpEnImt_SuEQE777yTd1SwjZpHNF4b3a
Il3myFN6wSS3Q9DqW420O20J6vi6xrSxPfE"}}}},
    {
      "Resource":"MMM_Bookmark",
      "Key":{
        "Udf":"MBE4-FGWH-2XN5-4IBG-VRM2-2WTK-MXTW",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"Xk1e7PopfECElQxuFIMownIwIl1CkjBsn8NvyCy0w
UqaNkDQhNAeAc41G92t9hqOLvr4DF4983WA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "crv":"X448",
            "Private":"sIC7Hqwbs7kvpa_3WLcKixv7yrTlkOZ41MGVtQMm
tk7x9-Z0X8j5Zua8UFvt9_1_f4boXt6TqW8"}}}}
    ],
  "AccountEncryption":{
    "Udf":"MBUH-FY45-DVNF-XMQV-SQC4-LTLI-K5AV",
    "PublicParameters":{
      "PublicKeyECDH":{
        "crv":"X448",
        "Public":"WSdlD8SLXWCFHhIHjCwQHB7b4Ym74kpM-XVZnFKWYYYpH
gBn-JIH3aPaHzd60MH3n1evVNUsTbCA"}},
    "PrivateParameters":{
      "PrivateKeyECDH":{
        "crv":"X448",
        "Private":"YbrlMb1xsk7AHF2pTDQEepe0jrqMYx9XK7cZ_6Y4el9f
eYypxxnLxuwddljd1Rj7jjVl8VH_rBI"}}},
  "AccountAuthentication":{
    "Udf":"MAHC-QH3D-VLKC-UTFB-UEFR-M5VV-TWAH",
    "PublicParameters":{
      "PublicKeyECDH":{
        "crv":"X448",
        "Public":"EmSbhqkjgjYAGR_iNHzGi_SRB6vGlKqfIsCyQvxlVf79N
```

```
sSEEhmyPHq7zJ1AIl1eaidaS2r263kA"}},
    "PrivateParameters":{
      "PrivateKeyECDH":{
        "crv":"X448",
        "Private":"vfI1A6-a_ctEjhXifKAEnPpkIliRSePGU0URKMekPTkc
oGT91PHHKWhrd1tq1jv42Ljl0QCf_r4"}}},
  "AccountSignature":{
    "Udf":"MBUX-YI5W-NTAH-UJN2-4FFC-4PAY-NI73",
    "PublicParameters":{
      "PublicKeyECDH":{
        "crv":"Ed448",
        "Public":"FfvEpMucwBoxAOS_-0tZUazve5J7IBXoXpjLXTPDuoDvN
udksR_1REfgh9Hb4bIpbZjl_8l-RiGA"}},
    "PrivateParameters":{
      "PrivateKeyECDH":{
        "crv":"Ed448",
        "Private":"sjW5sxJyWdKkw0KrSt9VsMb4cpDZtRz8FtJn2XOaVPhb
2m3RyF2xGH_ibovBsOi9UOvjsv-rOVE"}}}}}
```


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":"MB7N-QSGX-3QKH-5L5E-CTPA-H5EK-LVYU",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"OIVR98unWGKRIeuasqphMZq9hTRVfJcXPHN_q-UPrn2k8
  CV85Kid06zomEoHNdAcpcCcVP3EJwwA"}}},
    "Roles":["message",
      "web"
      ],
    "Signature":{
      "Udf":"MAMU-5DAC-RMRU-J5VH-ONY6-DMML-BJVS",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"lM0UNrzn3msgwDYs6XqE_qL9eeB-mEbgg60JlsJOeL5N4
  DBcqOuQtPimcqMlFPSdxOHrn9jYBG2A"}}},
    "Encryption":{
      "Udf":"MDH6-43JJ-WGIS-VDSE-A3HA-HM6S-3TCP",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"3hhKkF4KsSDyhcxo_slEZaN__aogdH2t2pZcTg4kTyNGu
  2400-Bh7hJhT4YaB56Px3wdofX9CByA"}}}}}
```

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":"MB7N-QSGX-3QKH-5L5E-CTPA-H5EK-LVYU",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"OIVR98unWGKRIeuasqphMZq9hTRVfJcXPHN_q-UPrn2k8
  CV85Kid06zomEoHNdAcpcCcVP3EJwwA"}}}}}
```

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":"MD36-Q4SC-S4YZ-KPRP-7W4P-SNR7-QMD2",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"GuaeOHN1q9X7d1nOdBHa1EuCRRF7e9BgF8oupurCdjcOP
  kyFALXQAgxsPJSQM5egAVPDkGmhr66A"}}},
    "ServiceAuthentication":{
      "Udf":"MBGZ-7SST-4HYK-FLMM-7N5L-WVAM-P3X7",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"enmpMVpIN5Y_CStHfXSmZkVnxgXJ6pbJ1AFnf3ZQVsk_W
  dmFhDCjjlmn2lG2Xvr5DEYIiGJNmK6A"}}},
    "ServiceEncryption":{
      "Udf":"MBBR-KLL4-YRFX-K63E-2DCT-6UGQ-Z5JC",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"W2qil6gYJrfNj5zGjN0gzSEBEgu7kThfkGSaGFy-IAT63
  nKA-MvyNGHIoE1ljzThn3pzHnPNyWuA"}}},
    "ServiceSignature":{
      "Udf":"MAGX-C3MN-DHNT-YUSI-ZYPH-VQ5W-C5SW",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"yqF9WaC9GzwXRLJ8QDU4K_L6PCsVv5o5TxqyIlGtABDH-
  IpyEKsdyvAfZZwYDk1jQMooGdC1iUOA"}}}}}
```


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":{
    "Authentication":{
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"-BhevM_ydw0M9zwSyhObuw1EfR-xWYmY4BgqAMnPjOr7c
  CWrWo5bmzyRyZrKQC1SnRgMDLLiv-QA"}}},
    "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":"EBQA-DGO5-WJI3-CEPL-KVY3-FU4N-QM2T",
          "Salt":"rNmeQlDeiQkooGoNB6Fsxw",
          "recipients":[{
              "kid":"MBFO-AXQH-VEJI-J47J-W3ZG-3ZPA-7FHS",
              "epk":{
                "PublicKeyECDH":{
                  "crv":"X448",
                  "Public":"MKPHQGY_8g9taikXS8zuo0ZnjnKLAYchxJXVy
  4eIr3rBdsuLr3prs23JCgtYC1KSchYds5OIijsA"}},
              "wmk":"i2RdIj1a_c2L1lwvJb8jY5GFQXZz4kzKMEMVqvWuZ_TH
  ZjScVuhmYg"}
            ]},
        "K3d6kdxtp2h_R0qDfqW11FBLmRNmCgUJ0wSpoyPeZw-oRFeG1gxxgG05
  kSoRZIVKOstSQKQWrNGjgoYOzZ8bF2L3wdXmwZz1Tr_PW3Apv79mF70NGZOaYHygt
  3khv78MA-4dNNBZADVgvLFROVhChHB1VCdU17e7RjxBOFSxP4szEdacL7csjpU8kA
  bteNqcjgeS3pqkbmRQ_5kYcrqwKm7ABap-EDvcKfu9F8uitnyxmTvNr8AuA8loAfS
  uVBmdK1deS8IuOBUXW3OmAsiYEN9sspiSl5Txfv8LytiKwoQBYuKfoUL4oDozECek
  8Y8sjFwTqGrpqJtlsuAXoI4uDjSk0KjUQ50oILZbpA67_tB0UpIoJbFXIMcCOEmdh
  5VVMMm4eqgOTqS-uswG_NINdO9AcJ44GahrsFRuEoTyxuO9JUd2XM3lYgSxqsaW_H
  6tDXu6k4WDOgGeFlw0-OdAY8kW9HVsRiwDPJ5sJM3rSBE21eoyKbZTr596xVSbDut
  UJ3UwcvH4Fh-kw_s5UlDwSzYGYM-_f-vNOTZv0H9ervUnUuwxZTwWyvZilSz-USNr
  0XDYzSeYgSNqvrm3rMcyWBSIEO0-1PQoCT7ecy2CtcUeVIYgHr1PcFFociVFoBvLb
  hO1az3usZh4HDUltwUEcj1FYUe2gXPKdyoAup77u9zVlqjgHZhfoMnbnvo5YbJ4Rx
  p-Tj3MzSo3uOX4TfT6D4uZTKNAhaU9IoEQwc8nkVVRjZph6h_A9-5VxRRCZJ5qUyV
  bnuUfB6diuJaG6Ulxu6bOqoB9ckW59PJhMf2vB43GixArciiPvAVs0hSwQBN_qkAH
  AEtgJE3Wkml9oQwZtM_iiRdr4Ov0UhvV9pEm_Syh3jxRKljjQwaIuxVLMCo_NqiH8
  q-q8kjPhID88geWQ8edCIn45WMiDpjZqkUcAuflDRjJOnvp9AYgNgNaI8EVJ6JU5u
  4rXccYcHcQZWRvnWS4LUFJiycqd9Sedv0VPxqVO9FGMOSA5SnABkaUjlHz7I77Cj6
  2oozPboe0adY9c5DwM9G64SKReY6ItKicVqw5jSujSaW1jsl_3ix4rihSROcupdxL
  iGdUc1Yc1JXlPPWwSA0esYcynMQ-r9rKwXKNx57JhnSHzwj_qvXVGTnWPbBRhC-HM
  Z_GN275u7hzi2Rlhzccm4pqnO63yHoxf1pHAKwS50oEASDTDCn1toGj_CbV_TFZHs
  RUbUeLJuwKWUdKZsQp64fvlnDIoERRDPFvZmdW_GcbCWtVSbCuVB_K0VxRxSLSDZA
  0YjSAm7WvJ7qUs4T4qaRj8mjZIEsLDbwSQXCw8QTyrJIYwrAAQuXYz5QlfRYoboIy
  -mhTgw9mEI5TmVZUkBuFk5x3PBvWGoswCodRKhzZdsiZGofu-0lirxsvaLOQ22pdM
  pExNJZEWU24m6Yr2ORDGKAPjtyFVpi7lWboQhc12Yhy4Sd_Lvb8StKTr0zpU5f0Tm
  OFW3INJUhn3ebiFCqv3Ut9yrIS9KwpgHUrs4LjIuQzl9cP6ICmg83BfkOyPCqA9tS
  QETlkJ11d6xynS9zdJs4KlfaC-BYNpoRzfAClcydOpHqR7PMWEfeDEIB6hMrALL5L
  6URIUjdfPYygejSqVTHxvxyCNybt0xm5ATjrvCCk64j6U9P3ovc78IPbFdGCmITzi
  a9QHx8NMknRHjYKcK57lVexmxsNlp4HSgrQcDl1HD-TW9KKaYXkQRR_a5IWMfhrBB
  Eu7DaUyxc3ZfeP5qWG9HpHsFK8-trZkFGZNumIyJ_nZtUhsumiulYGD6_LD45E5xF
  j_ZBMWjs43rtMsTJWlrEZvDyPXE2gn38rsLyp5vYc2zmAAvmkrC__jW5VZZp-ws4Z
  mqr-D5UppiuoZ0xZ3S61PYPaBRoiW1NL5px_j6CTX8ii7jwD7-ApNwWvfE64x-TX3
```

```
03E8beqQ0fjgVj-RLkenkYdiYvMYV_gnzz7m05A0pRcJZxvZysuGvSt2TRYS1vokg
51n33pFOmp0PlTnlJ7GALNjkyF8zv0HkF6gFGDCQVqq5pxQijnIxX_8ac85nLq9vp
PQfwdV6fjJTnHze_PjuHVoXQtQyq0QXv8fFbPKxxcrKIsFir5W2dkTSftbW9_M5Ua
anze0NVPR3UGUA9lD8Z5yFbOS3-liKMjHKZoy6mM00YGE25nZCADP0PRRtUv9OzAb
WE0yCmZ0Zu1Vgd0U9xm4ppeJXXbWia9MC1DQPq8TCDJq9unpQ_2KI9esynPcQ2ac0
AFFIeqVVmXvVNWzEzoDxLaVnjBeVMibkArk8tHyoc4h31B_oGh3LEimByYcbtXKgX
tSREWCK_jUNiigRgFRLTWfCOlz17YmmhVg74MviYGoFokCqTo3vrgLrDwPuFiGIfs
uE9fxwB-RJ2Ifl0iq5d3t6hoqs-Oq-XkMffC2IvoGnH7WedVIHhMpzuLpu-XduK45
C055ZTgQMfV-VQNsJO8Dl7LsIh0oNh1EyCgI3f9Ty4G4MgGh1-3agtflwEnW8SEEu
QtPVSWfSaf7mANGlW36r71qCAHMowUVXlp0Hsfg0Z7O6dDdFqJBLa_6lUxXaYjx59
sNkHbASYtrUUbvgDh91u4ejZhc9vW0NwyZ6oMjjXJXz7f5-M02ppUfLLDN7J65qfR
ooPiFLhZVIX1-sxnR-0TFtSB7VT3TYHoN_wGDIlX26veEKVMA2pPhQcw1OfTm61Xm
CaT6AqH3ihyhXMR1cNLeBlQuvyk03LpCQ_1-Z0Fho0s2kYxZ2ibJ8Pl-jS8YUrc4Y
_FCp8ft2OBdqE04Q5CwgUyJW0AQjEVMXDzgelQ_eBBasyX08F7cX4grCZN6bdTicB
pvIrGdIfU3P14mz4mIKq5KK3_tromsNSojlMErfeDTebsNfiDp6Vrz0ShCg6uiGLQ
QL-c_wm9seJy2ISQpzmfpnmRg-ytTQ5wT9iXx1rK6Hd1FjXbZ7CbIfaRlLWyGowP2
r3MLrDxj00Rk7Erg_x6ET-a2DOD3lUqWGmLAu_Ksb0QpF3rw_IJ7YMw9nSU9Y23R-
3Er6xRjmzGvRpwcnXJD407cj8WWJTQYMJPKJdAm3ueGNZYu_YO7bkbRVbgbQ-gLCQ
eqE18VXBvp2NLjtxQQx-Y3I2t1GUjtH4XnPV3uXds3thYP1uvI0-KjGs0aatzpWB-
S_CDgTN-vSpNfY9mj1m4p8gKnpwfHkqh-h04Y9ta7PbEbg"
      ],
    [{
        "enc":"A256CBC",
        "kid":"EBQF-PHTE-4LEB-KWAV-26WO-3M4M-CA5Q",
        "Salt":"u9yo3dTiEs-heyXBs6BfIg",
        "recipients":[{
            "kid":"MBFO-AXQH-VEJI-J47J-W3ZG-3ZPA-7FHS",
            "epk":{
              "PublicKeyECDH":{
                "crv":"X448",
                "Public":"bjkFbLCrOl4eWttmQzaQd82IUqaFtnEdUuZKk
TD-h0slXg_T38zPU2SbT0N3YuQMtVQN8oFlr_gA"}},
            "wmk":"rLMIa4e9Mn1pD0yuDkU57RumkvHT7rPNIfjYiz_Fg-J0
DWtPs0Q6Uw"}
          ]},
      "sq2gNSTzBc46l1aedM28f83ZzoQZwN93h7nc6JQ1iGO6pS7B7F0tlYpI
eZt2Hfv_-O1y4wPI3P3Cw4-2FqeyxYZIYUiv3gvqupbkrvGwdMDvDH5gMihkzMYFR
dLkiLih9qorkhLPCX2Rpb6vbUavQgOrqhH90ZmYG8IeH2cC6_oJHJLXoFyKbqFdrx
JyBJChVLiCv1KPgnkz2Sh2zRA_jWlnn-pdZgh68ZIcxpRY5kpt3uvFcPeGBr8m_8J
t0JheyDCzKBb96ab5P5_33reC-jJxS70K94QWMTPilwBmds8k5UQXW2LoI1WeDAC3
DR9VJrQA8DIp3HayBVxkvayPyy8AqzLqsgh4HH0bFY5W4m5Vp3y_pByL8t90tco4m
oYjOvX6i-v6V1-Elt--fUMcdmagX0nJD_p-Xfrmbo_SDEaRdzOS2NGZtTK_CauNLD
DheOql4Zxwy7bPKF8RpV7teHAOvh1FGU29GXDbYHF7jQ2qklraiQUsMUb5rg3x0qm
swKfpn1yShYvvRkImQ84l9SFehy7WetuXzVspSRXjR7hqvPBi70HXTgaRIqcSbUw5
378GUjf5ZGpSJa07sWKE-cQqYezPB-H2VHjPuG8E6CbRNitpGn0Ec-BUBZD96sRL9
821p0zBLRGeA1nj2wYJLj2WHZtThuiV5i6QH9VrE-vpEQXCjgkxlBBCEQmx_JHaRp
ZUhgdqnH89fLEFZ-_B1u6RSrEzxQL9bbZM8WR0P9nHkI5lbe3mGfrhntVUR9-s_YL
_vKzB0x7Sdrct6Z-2EesK_2pTR5jv0xRMZrlje97LfgxBk5oYySa0tCA_CcfNdfFD
qtn_n8zl9MMzGzJ-qA3cUeWYizsV5LVPISViyK0JlGpifu-F5ugQ7o-N-PHEXdy_l
```

```
TUITW48EsLBWI7nOhVBdCdi1wAWzQE3RE5G8qllz9wzMk6uyRWlvWnPAt3blk77P9
Y6KMjQ-srSLGUEjO4lH3V2QiylH5LOjp1XSz5XBCajMOJb1nHp0V2VMk6GCYcMqko
3_rHBIj5hD_Ds77nYvyNtjx1zi55sfphWTqv0dAq7Bnhe7d6RkWc2y5iLFm11LGLJ
CDRtcCvat1zeIMvBoRbnjeVeOwhTFJpiPXXW9AWgY_ClayRwnHuP785fA0ZyfwTpU
JWFOnoemWBQw_d9HYPGFLAAWHBa2WhN2zhn8y_e7-UOUy-cHHxCZLuzm8GEN4QQNB
D5N8z7oeU8DCeabWyLSZz9jriGQpXl4qR3oXZ5BPMwva9j5_Y2amZIShIirjdCmAJ
XzXCHIGgSMOx9zmgwpcqdkBieDJ60PBXnQcN25Ke3EuDRd_nklN2QNIe-YKhVIWbM
XDCgbUoWAkegW7QJkwr2-N8Y7_agZ67zgxvD90E4h7JbxuK4V3fwMPeY-RA77S3Gw
tVFZ3DPmjaJaqxMX7NdIIXKdjCdALE8wdpKwC9pfR8zmv5bL2iWzKeKfMiCx8nKqj
sNuT7518gMSYdnmYjmdVd0u3U6B4W2TgBxkatquB3vTz8vSQdMrLinku0NcxZSA9v
3l8rolcsSjdFmKZUBwuwVEdOgjnGCKINAnla7fpZudEodyLrktHuA1aTYgZWeCyNM
y15UdHxc3Q1MMiC5sm6ejySTMXZMtq5jm1ERKbTrIV5xUU5TTnDwJ6B6DujTbrFf8
kpTjcd5PZjvO5Xa_aPpUb7VwYzVANdy8hmhYo6-rRx-qVxFy_dLUEc48p9JWReA1C
eVaOgOFFihJ_P3sxTrwHGJhLm_lu58MXfVDszCCtupW_IMzZLyiQQOVrTh-dQzcGR
1lHeibgSbpsB9gJCw_vqMAGYfFUZkNIS1ov5sH-KoxaSv8krxarCLy3KOa6LFqnwv
wm_l_a3AlI1C_IKqrtl15xhUwGNmzgl2UFbViVAlxMMIGlEhQQ1xPkh3caF2ZAEDL
kxV6CqB41fM_0F0eVv5-ULQLwvqpo86sOaasOFkDz9dxlEBd6M1sap5aInKEoP6PZ
Nz92ihvp4ikDNQ45wfw2uZlsjrHInKqHvZGF6OphWuNroAVDS2_PY_4MQgvA_TEng
nvGanNX32IPYFpd9lMS5T-4JmgbrFUOYIhliELo6BNcDQ8p6kQRzMraDF4YBgIIA1
mZxq-hi3HGzGFoHnhzdcVxEx1OmYOk-fuSCR7DsMH7l44dbFJ1p9OY9U-YwI015Oi
QVgCu_l3foDNf3QuOWJzcFT0FYzYR41NohlIFyE1bwzTAGowqZZ2-oLrTvjD9dV7k
9uhJsBw6AUy9P8EtFNhPQ9MSQnrBScaUVxNQcZJtAUOkjW5nTm-aCcF88Zv26GeOA
9yh_frFQAWfalmCj8q70IvU4CELNO5QfcuA3rBeVyGymASyXI92vqnmbDGFxshnHW
RixLnZB-8w-2uryrWDkutK19RBOWLXzyjN3sM5D47N8CAi6FapZc1XrmG0dB-VCoY
8TtpjDKXGRNCi7-FgfmvLFBE7hTpWAKD_Zyxhns8r7xwYDKpZVD0Nm_CcJBdr6IFs
DGml2kfuMrafDc-wMn3OQ7PDYvNFrnq2R-rIu9dWpguAcuw-XEFzrZklNKNKYjLr-
tx3I3XIjfFEHYHqmRECZH3oK36x2KXNHCyg8e-re7ZZ9IMOVwzPuKrnq9P1L9haPz
m-1vabtqzSHL9nkBIDUU4rH8wdJpD72FklapNDXeUByZMCzK_giYiEylUCpfyVAAW
_GeFigDjP0G64Hejp8w7V5QtCOLTrIq13CTxT8xgjQztIyC1kjgLueuPQ5qy14c8g
DRZIxi-MqBqRqTbL3HrqIY-bBswCha1X1wcOdEwEd2Mj-X_q3JsUZf2AhFPYQUEDr
3BpXLk5o9pS8H1sWuAZUftIh50yZJ40fmM4sPcIFI0j2ImbNrezITLKNWstjO0n6l
B4qXYnPYi0c78kV20nA5LjKrfwxR46bZ5ogDxZ5-2opf54dtMxMwykhNU8EOOtrfn
Ey_if-PC_XfBEY-tn159Om7_Hs-pbZxAtBcHzUFn3cGVp7NmB_N6SB2yo1Scz1ufe
W3E1gP2gVKpF7knXflwpZOxRXdPxXkDy_kePjKL73rb-tvN9HHdi_DD0AZD5suy_
k7ESFyxfRNYVSzuQvn55BO6JrE60aE6wOwR1nf9OiWETtscWa_rc69toLf2hnhYBl
rkmDzO1M_3IazALIas3vPONUngdc56kSNO1P2fQ4X3GavQ"
      ],
    [{
        "enc":"A256CBC",
        "kid":"EBQA-U6UQ-QKTY-6YKT-OJM6-ZMUF-JUOE",
        "Salt":"wzSCqTRJjlzQMPT_ZxwrVw",
        "recipients":[{
            "kid":"MBFO-AXQH-VEJI-J47J-W3ZG-3ZPA-7FHS",
            "epk":{
              "PublicKeyECDH":{
                "crv":"X448",
                "Public":"z9qmGsdjZBWPS7AcklLZ5kCHls50wT4RC5clV
EHnsTjMiVFcfjPRSV8S1w3wBeH--0OrTvbJLAiA"}},
            "wmk":"cV04tFR_rjwNsGInwCa_FCy03lLC938FTBdBNdPKwo-4
```

```
VuWkzXEuJA"}
          ]},
      "uwmHS2h-e3IYxOwjP3eS8zrwg-cwXDofC45fGABY3jYZayXxhkms_jSQ
qHicSJsi8s55A1JLXBvpNJ7Ncg5YWBKgGsucMYy6NhZB07qcAxSPXB2cb6_7ube-o
ICT8BLtemP-afn2mY2VJjpbDoxK0Ez-i35mwD3Y-7u3lAS3UMiQhJxGwITp1dyy7a
SB2ygPlUdEqQ15d0YQkXy_E3YxLLUYa8m-Zss-o-rGhTqwD8za_pps0d1JVThljOZ
Gbc42Ru24XZNTE3erQ1BfXAZb29_VjDoMerWDtcqZXtEFSu2Oy92rhgtTp48wTJxo
6NG_EMvGSx0v1oMKaL39Q2XJsKgZyz6yTVadNZ19qZxjq6Kq8xPiVT9e_HnDOCb0B
hAgLSyf42bdXQltLNPBWVPcNEuAesJXzr6R7g99P8gK75FrJc7SY1KaWwFgBFtNZX
QjEVFLKuj9e5vp1l6RaZBEJhx_7zSUPMhYUkc5v_cgUsxQDVHGlY69JtEpFXutkep
AsycCzDW073rw2B_igK-XPtFLJoHv0F0Qxi6cqCxBHoIH8j1WuiksU3w8YB6Wui2d
qcQQfIsTGTy_0O_Q7ov855lEiorHjtvGr3_lJ11ZgAGBa8krcW27lKHQNIp14GdQZ
y-7n7Li2ttWli-goqSU64XM8H5JF1bUkJ-ZnCRAmjWooO2eDk7av_4ijTNKaVUGS6
y8vnMPKRMSBjDFghNryKIdpwOA7ilTR3_ZbtWimhIHi-RPVypISbKb-y4nF5YEsF1
4eeRB6TTI2e-L8NiEukPYeBUu-17M6JhEaw-sgPwL-xwUeDIj1vUlYjZom1JVgpcI
FOTsV12FCkx8VcR9Rv0pV6nfDyPsZ83TQvjyCVypCHmqOqpnEEnOvQyTXyxmEC1Bf
VAs8xwoR-OAXTroNhqh3JL2GeQkIXUN_h5PlmgrUK0NxE1BqH8yf88eLUL575Btre
SdCAWMoHH1drH19uNmk01ESbDGXzZCWsvcviIXWhAQU9luX9FCg9fQ2PAhIvcrjVR
p53ZAuu1uHPSAhm3M-feuhCyV16pA_EUhnTYUZ2maL1qm5fRDHjLZGephJJS3TbLE
FpxlN6NKV8ziSf6Gr8eBYbc6z8Y5FUAeS8SR6A1H9WhsGvS0jRFZ0jhGEvGf-IU4-
d5j0YMDfXZFrC2odObEr11C7owzwEf0SJFIsSh4f7febKLpuBwgXxIl5KqxWy6IrX
jdTqt32QC1kQRtPoKCgjjdpF-09mM-2RQ8iidX8ofgWz_P_xpRDQtsTq9bz4nRBj8
Gr9s4xPpblV0XWDFXH-K1NNBdAwMhsMdsHP2AJo7tL_f4vWw7hTaSPQlL_bHI9eTB
18z5JspPe8YfenapBLSsP_tJ2LDsfGdIljqBSD-tkYq199jQjx5Se2EvnIQUk_A9u
6TGIX9C7kKFKchueDt4kZYrMkw7DsUaRZP5hN8DDQ1KP6641LeUjkGaRstzBw1nI6
5pd-O2cGZquxUC8AKsGMIxlqwlPbOqZbThCUno_jmDsrbVteMRcIHF7UDAc_xkZZq
IKyivt4Q7Zlpk9lT8crCIoXuMLc-spptjzKK2q5GTvCXvaSYypKiSkkA9UBfNUgDj
HGjFFLVy3YIpSmhCW3DbrokDE-ZGQGPiW2zK1KnDQmJvSF563Q-h2J8IUgq8nteGj
JB3LPj-j0jrIbAxnEghWc-hKEyVD6qB42MR1UGluisZP8NN0klhalA60dvHnFd6OZ
GuXy_3YhiTj4xdJXCGLfiDkMJ3Ol2IMfdm4JrtC_PRnhFh9rSluxop3lKCNn2NALQ
PuhiBEYEP3hlFxz26eTnPWCmOFgSkufBb--CUNJHswBiEnU3VEkvRQuhoc-WWaRCF
pUfR-ujQqib1NXv_OdXGrAkt7r2gh97LD8TVr5Vf5HjWQP30qbvKVJqXPw02iSSvC
cGKti8MRJtUWnWwYcZE1yksQ9fCgVJHgyHWjc1w2233vGCy3Vhoq7rYSMwEr6xKk3
aCWER704RO3dCOnI_t2T6Domyu4ArgdOCy7V2xAeEEqcbVlB0dPGXHXMO6Dw2FrcL
qZ76EA4dqbU-isayqTxmOK7OeMUbJZf9BiWdEEFOxbQ09DrFRyZaWAmt3eN1nQBR0
wOicO6h2dtUTGS1aWhWoLhxCv4wbK6NIb0XY4DUqOVnKq1ruTAw1NM2vXBncXd8vp
pA4upVKrJ5p4BhygQnJfrBLxq7XLdJz_qkV8BxFeb0PI_jp-psBBeM8yvFwbOfaKj
tv0a2D1RF5_iqdiZAGm7fjpDNe3kj2xoZ1kLnRVnuw8Gt1BKCEBN4fRSPyK4Gl9sq
Nbt1gsaux8bQNme201Yik0Y5YSS5pk_U2KGOpgIRo_pJI1b98RC_47ml7t-SCpk9G
WR6U6qVrbHF7VXpS_GTeZBBVMO8zKPJbNIcHcPJCoUIJCOmvGm_DRN0lAEwIt0Sf_
tthkrWv91pUoXXnz2zgho-zUtCLF5CPpYm4VbG356DHsSdCqIlMQ5nE4ERIANh_mB
Pi_uFSklPH7BbzKBqgqgxwThedsok14lkUe6MP0JmQaqbUrJZWd5h3I9Yd6jrjiaH
VFD8Jnl_GJuxmA5i26h0rEzJSGG5p8eQW4QvfcN43mJCU7I67M9jdPaCuKCcdZc6g
A1pVT80PmGJXr9FKBDDP8Vn9cfMUSuP4rK36vasoupFOJkvGAfAHcVBE3wtBT0foi
7TMRpkNT163w-dqrxe78Sb6gjgu5YxKoEIloLSZQlB6zlr3gXD21pCm6hgO6ztkd3
D9ArKPgrvTpRPcgxU23EUyAG5yiSYy6C7fShQ3w3B-Sjt-nYC-tre1fXYe12EoCRk
2uihdQ2rSnPo5hmtzMcKJUuUfOOwXySvpGZLuZoO9UMRcbtIN-f3jq7ThE9mX-lLF
ORjnuPsFAQRnQ22S9Uh6iRg9FgVF2OIR5dcya1ZgOyRg5xOvvzmIewIpd4QturG_y
W2oJlIy_uDpPKzNQINIKAc4Byj9xJ_M534DAqeS73-Wq8xtjaaGP69y_cS6u9i8Sd
```

```
L2-x3jEnfjyA5ggyZRvzAIAkbgdI1Q-G_XmZ2SCoC68bRBzW9dRxtNZQ_iRr1yVgF
WmC8GN4mKzCJsAyzi_5SWLCh0NxGPrHUfZezFHmO3HDlkxJqWYKiWwaoxJWU504pz
IROz-6o06P4o6GBaB1jxhwmL0AxUB-W_OqsEyfLBOGJ3UQ"
      ],
    [{
        "enc":"A256CBC",
        "kid":"EBQJ-ALJ5-QUTU-MXMI-SZHS-YZ2A-AROR",
        "Salt":"qYOy7-oQsdShDcqNjxJZ-w",
        "recipients":[{
            "kid":"MBFO-AXQH-VEJI-J47J-W3ZG-3ZPA-7FHS",
            "epk":{
              "PublicKeyECDH":{
                "crv":"X448",
                "Public":"vNAdusBDn2D7jSHQomfNbvkV1wf-Xu0LsEI-B
Os5hnE-Z1BBStFgqIJW-TX7atRfYswqaDtFnxIA"}},
            "wmk":"qowqtU0WeoH6IUwq2ebdmXkAgAHf3I3dQuvQCBgOlApG
D6NHKoo34Q"}
          ]},
      "663KgrWaOL4lmdC4iEvOWLRZR-dvYreMDrXcHEh1hV99dV2Gpa3WxUcg
Xgp9jx5gTfgm979qgFiJpx9kriWGrMWk-yUryV7R6YDiRgVtPtNHmVAn82seykPeQ
ctIKribpahUt8c7NIHNnhzyBJWZpNS6hzlqtoo82iPbVRkiY-WAYvJNkHCjgeNbzZ
94z_6aAcfbctVGz5aHAJtOZLtHn9wfye8J1bQ1veHydCinXfH3d0ifaOL49OD5fdS
RY63Rrz4qxcaSOvJGGjW08ZhvK7dpSTIlS3oFW5PK5bXVfpBIU0-ZQS7ZnTZqLd-K
GqLMR2CBejHICQ_FzI8CoEbne4a_hNiJl-U08Yn0EwNuvwt33w9MpjpKdSN7C0YZb
oGM6hrCK5NL-faduhb8T13V20nItpkvqV3L1vbl-MOmx8LmWp7xyHkiYecCP8k1VM
DtdkMLtkG1nox7V0PCgU8DmDFt9bhblLNuW_jYP1Tm5b1PjF0VxjW0no4n4dkDfyz
X2S-DKU3jTMQDpNRB_v73Ozr45oqoPogKtQfPtUx5vaApRbPVWceIWLLXNpgGx8P0
swnArpwMNdivmqHBLccgWqhbj3UUUY3NEvBKxXuHn3kpydHfcJnSyXrHcG34sQ0NA
JbXMeBVuTsN7zWiGOUj4ecBeXX24ZkKJ8lCcpIULSaDCuccWBA2BotN213M4AVnEy
vbmR2-sHtlRBmn7sRoeuLRwSiEe71784RSrkoGRJ3j3hTfi8JviBqqtAU3mDwbpJV
DdMRXpFpiDyeJIgRLFUpaWjjWIPlaDNJNqARycVfu86MVwiQtLJS4CRVaWSDadPfq
e0VunhybKfP9XTKd--g18mfuaUeaxtSL5oDdwdZgAxAY182eC957Aowiclwkndru
x5wKQR0Le3mOuzjphM_fbV6jJNmJu7SqNiBTuppToF1A45_SqShP3qcc409pLdsDU
ccx3FFvu5w5rwoKSV3Ooe7vNKvyN6RPrlbET-o-CqO0YrLvU8P3mMj-n7fK1fUzll
VZ-DicZfJ4IUdnGeNAvcrLhUeKmkB_ovfJSPDjog4w3vCuVSAaQD1tokuH6scj8qx
n1QDjSk9StgC1QCPaX3ouMqeDmjfMRt9TfMJnkGuzuBnn4iixj948cgjPgKlS6mpy
FWuvwv3uPLN1JNlUuqzddGdlYJdVXsuyGJaHIkY00eVW2Y0fELoxuToGKO8yV--yo
FYy5JE3E8Lx0ZOjtKNIe3tBfBAOxxtEfMMR8RYujhVq3CkkoE_swjKFYaqiFIXByU
_SXIoapl2yxx6Yx2rCLpdq0T_KHek5KkQxvhvw-crM9RbRBFS199Kt8DmP9tkc1E2
lO9OBjktiyGQ3x7HUYyj1ppHaaoXZA4GOJxgnPOlJWZLnb6uihkJ0UXhmFyVpjzb8
uG6atIrlM3WhneT0HAaN2oCnhm7GFqB-YF_NeFlu_95YtV0KoIdS-fWBPWjxGZ3c0
YSpAxwv_mY9gMMelERqC3hxAUxKzwD5036VFQ0xHT27eyftzZjGICmLreC_9gxdyP
qO1i8lK6q8gjcbSb8zJLqRxks1sX3MKLGuopX89kZQ74m7YiKadn32fpOC-QSeHiB
OczzzcxSqAQLnJIDzCusgg4_lW_MOcxOAc60E5EX4j0BecOjkJoaIdh1NoDr3dqwg
nu25cpXwYl7AogfFNdo_HlTfBFMgOU67k_9IXThFTqMcVKZtl72E04aVvDvRKoCzv
yBZ8D8tbKmHHWJ2mWV-y_1KaiptHtDmi1jV8HL4bCAggRbrwVcyi75gyOyFrsfors
O4vH4aEA4Pgq2o5y_mdDnQ6UQsoBP2U-92cor4RoCeTX2nnDfCp313WyXwfoqyYEP
leu5fpu1mACFDpc0Xvq7_h52yQvip0Uy9KdM5ePiuS8av4sLFJ-BZZgu70jkPmNdS
OF0wwlzgUvKuT_V6Tw9LWFead_K3gJ42iN1Vhku7VNuiwy5iQmXUlSh3LsLjTu-oG
```

```
H86l9bJ7x9GOrCdqtIVA4kbDUbbS569z-24CuUYCnGuyxnsU5bgHyRVP26uEah2u7
1VvfQ4JJWn-TKyFpFEzT3VMo4zvqB98b370OdtQ_qKgYWOxkuOsTlq--rxqYIwse9
goqm61cUSk1y58aSLDu2TCAcSOzqixFMK0vpl3hKoWYH8qpWBKi-wDSOjb9J8jCwU
kEHW20QXLcy74PJNtixQ0TM6ESwVs-ld8hiZ5QlRwqGytbRZx5O_ZvzKv5U78zb2k
k-y1T18oRXWW3xqYd0O-ADm0e_7FqMlvBuvdIJGn9k2LmErB7PSNaiWfouNDxTkDS
O3oAeRePbHnD7xrUBsyovaEfQ7sqUTzgCk4-Pz0aeJY7OS23SXPMv3xpqoKvQplY3
0FxuVx7H8CTGjW4e11xyWSuq4e6fIL519yFFgOjqHKBXxW-imhGHl6A2HBen2R6Xn
EjIJ4XX5kQ9GS3x9XN6fvBHplszHRACseFtAthxTD08XjvXqKoRe5_0iwqyZ2gpdT
fhh35HWnMy3ww86jezlq_tZ7MVhSfTQC1eyT1IHOwnYSEbyu6lr6B_-r3ayigJwqB
ka9vgOLDENV5S3Vx4RayqGVr8EU0vuoX3RkhX2p8y_qY-mXtId5C9LKg9s-AjQvZI
PmNxmSvpI7lML4UdGY5t6HALz8is9m-d541QKtFKRwzPoF2AG0Eg-7B6GJ9tZIz-Q
qa3-K2U3cTs7UyJVw5TrD439vMAcith05-K9FQmDeUfgPsuvJHTEMGvvIrOl1inXf
8mPyLzMY4kRbTa5o8ers5WlGUTMZDom05Gb90qu_hvwXdD_tnyNYW9p4KgkPDv3pP
-DtIjmRETrKk3PC1Wf18j0aXtxhuthAbExCjxRWvEtRekAa6KhQn6hXEtsk7Cnt0b
mUPoUBK8uDI-vN40oWerWT9NnDSf88MKjymgy0sjvODdUfAVfSBADNoA2MkULIHI1
4IAlf6l5LJ9tPLFbQZbr0TkvnHE9uT2Y7p_NmN_tQRArjae8j_hIadzOY61ZQw7we
f_lSo1JWXwCWdTuB9vYYbR4l0U5MeF1UZN0vslQ7b2m5hlFrHDd9wltV5zJiEohsl
qDcY4lzsnNWVv6YVXgFihO1c_XiMr_-oi-3CVOqNI9j2My3r_KTm41a2CYnfCKL-K
iVEH4KUKVAsS9w0fBR2o-tRwb6HFQXEVrW4J0P5a8o_J9A"
      ]
    ],
  "AccountAddress":"alice@example.net",
  "InboundConnect":"pop://alice@pop3.example.net",
  "OutboundConnect":"submit://alice@submit.example.net",
  "SmimeSign":{
    "Udf":"MDVA-7WX4-LF2D-GDK3-CZTB-M2V5-NNCR",
    "PublicParameters":{
      "PublicKeyRSA":{
        "kid":"MDVA-7WX4-LF2D-GDK3-CZTB-M2V5-NNCR",
        "n":"1OYN0_ik4_xumv55F65fc7RHoBKUcoPmYL2uoeEtjry2XW0hMh
4Bp6tURQIaQRjoW7sMAukC2vnvF5LhdLiAJbxYdN-9R0FCnkg9ntiruXBhNujBg6l
yXdu94-3glFOBqmjwMqKcKu49wBuZlJODY0T7ZNZp1p4H3OPnMDZ7Pguh0a-GRagQ
CD7_m3yyNRk5RpV_NtYcTSaKtrUPhVmCxKAQvkgIcyojQ5rSecIEdn8R5I8AMiEoI
ChyYY63k8FDHo_rmYu7230byoJsthTFk2AReURAaRfkr4Ogub4ZxCuG-kD6gW6itd
KdnlL0yds5tXUZ1WrrnTJoXMWExm07BQ",
        "e":"AQAB"}}},
  "SmimeEncrypt":{
    "Udf":"MAHU-MIR5-BZVE-IOUS-DYDQ-MHCG-IF6V",
    "PublicParameters":{
      "PublicKeyRSA":{
        "kid":"MAHU-MIR5-BZVE-IOUS-DYDQ-MHCG-IF6V",
        "n":"uS2nl0O9hC5LHCSNZHG80cc0hvKrJnhZz0dosIJdWggxt9eUY3
RPPaRKexe55hsUzkxqaLEF1c6x5kscaRSsKUef2ZYlQoG16HQAXUdII53mG7TJuwP
6Bduxjs2lkQOzNX7S16qBpI9iQaQBi5L8OGVEGn5Pzx7lDOcQeM8Hd4g3aML3K8hx
sx_x78SUm4Q03Orqc8DHmfcdyayMcMX29s1TfBSX3PUjhcMs_u5vBptyZkQ4QdbK8
sny32GPWUaS6i8c5uNFaPjcVi5gUSqnQIigYTnohWP0gV3S-fvkztvJidFofdfr01
kW_tvIyKV6LmF7Srqpqf966Nf1XmGpMQ",
        "e":"AQAB"}}},
  "OpenpgpSign":{
```

```
    "Udf":"MA3J-Y2O2-TYHP-2IAZ-3IW5-ANG2-AGIR",
    "PublicParameters":{
      "PublicKeyRSA":{
        "kid":"MA3J-Y2O2-TYHP-2IAZ-3IW5-ANG2-AGIR",
        "n":"xYf5DpQSgleszlwTxKztxQ-G41TU4lhFrFQ85sbND5bAcHy3qp
  chFgVJhh7hiMqRE-P0swWXrqHWV-7WubhVOmQbqeAhwlbr811Z4gKiJeyljQXb1C6
  Lz5GKPlNd8MqOGg_6ZHgraPSyySqVxPlO6Afq3LtUvrqFLewNX1hqTghkZ8JfHyHb
  ZLuOCjRS7ex8CR7uIiI6JfAJt9rUhnogiXrzTaM2z9M9esVjAsomtSOSMDUxtRPpL
  j_-8dLCJN7hh93hdim8WoCUcOQ_QqtpQ0IHkguDeKveJn2yITR7tWgX0E3-TBmhFx
  QSN78awk_yucvT-4YWHiAQWGDzCea9FQ",
        "e":"AQAB"}}},
  "OpenpgpEncrypt":{
    "Udf":"MD5K-DR4V-5DK3-J4ZR-V4DX-DBBB-Q4HZ",
    "PublicParameters":{
      "PublicKeyRSA":{
        "kid":"MD5K-DR4V-5DK3-J4ZR-V4DX-DBBB-Q4HZ",
        "n":"um2rozPmSCnZQD9nm4G8C_YvZd8L3BSkiylc1M0Ia96-C4PBrK
  C4O9VkOxp5gTr4Vi07mv2W7ONc9HLvOamk86TaMxn4AFNs854etwbMliVqHClfeqj
  0Sc0T8ugIM4s9kZ_Uj2pHdEQ4Ob11Rmm3XRnnHCHdzM2-flP55O018Su82r5wS36-
  BF4ar7IERCVNd3CBLTtUZvHRSzlOg4LykgfeHJwc2df3ZVPkEUhIRM6ws7P1I5OtQ
  ysm3bgLg8Q8St0hlwNOpouebmtUZG9fMfvuL7ovjisw4ui0RB660rKkROiocLBd_c
  lBCbvIuorbvt1BdUEF6_feKAmGgkm7NQ",
        "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 CatalogEntryApplicationSSH entries. The corresponding activation records for the connected devices contain the contributions used to derive the private keys.

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":{
    "Key":"ssh",
    "Grant":["web"
      ],
    "EnvelopedEscrow":[[{
          "enc":"A256CBC",
          "kid":"EBQA-XLRF-3R62-CM7L-MPZG-6DQB-7CG7",
          "Salt":"LLLbmrexmMC0waYowoa8zg",
          "recipients":[{
              "kid":"MBFO-AXQH-VEJI-J47J-W3ZG-3ZPA-7FHS",
              "epk":{
                "PublicKeyECDH":{
                  "crv":"X448",
                  "Public":"oBoI3gVHFS1wmdljqrWnd1WsHcl51i_2PvK7V
  4nbcpr6XX-TCNG2DW-zBt1Ma_Uf-sTja_OkIdUA"}},
              "wmk":"LGmUWipX6xbyLOX-eLNonwE1XwDIG6jAAefr6cAKQoVR
  vzEJwVwltQ"}
            ]},
        "WA3lo9E0y3SNKZGjj9ZX1wic_VNZ06AUcaeRpb-lopdLiQ6TFsWtHA03
  Z-Tpq8lCqL2Da-3ZBT3DAGoR3xr8hyMtU9Ogv3C5Emd3Map9eHA0kDvxC6iNEq0jt
  yFPrQGsPq_CJ6GLHvsv-ba5UzkFaky8YL6c0H4vkjxXQwLooIyKada5_g0xRvzOMn
  4l556Jpo1G3qoryO1p_S0g8HBN_uQrYDpyxTuHkG68DRunbapcBVsGk8PnNI5LWTX
  GwcfwoqqAk7qlRHk6soiDlLdnkbuFxzoRPrLMCW0-cbAtm8wzgiZ_6dGJ4EjvFW6v
  uUSHZ-zHjN1zAbrko-SnQwoXvTkLn_AKVBMLZ3BPc9x2Aq-Hpc7pqxnZcePVFHDp5
  CXtNHC4Wf4t866xQCCLv7UIPcW7RJVRo0e-VdT-WJJjCcrwFXh0jVUcAiMxcc-cTO
  exytfjuDLVCJfjxx4zLRlpRUOnebjyyHAssmyVvbWN7bkAQ01XSMZxrrv3dBu2DVD
  EZlybk7b6UNdkSatJTDwPRXhlEHbJpkZmi5do7dG_nVBmIFw7AWVmiv_-VUroC6JT
  VYdTKUOvJuTzUgE98bkrHtAHMlye6SsNPcJeG_72ImOlWsxT3fazJiwvWLdjBGI_9
  YOYhs-PUiYAlu9rEWlwa292fNs4mSBcVc6isNhxOSFVprjBBtO0hv8hqh8AADNErB
  cwuu0-iKMPnv1XeeXSXDe7nxtUkwnLmBXP9WOn198e8sqmK0G4XFeYmczoohDJaeX
  L23FT1bNT8JOnQzOyp_bC2sZZkD4TEqkc0kfBnnPdw7xYQ72uMO74cCTTvKSTbZy0
  zGaQSyXpJJ264z9pwrp7zQc3j_tvwGKxFrvEJ_rDP8v7uPyCFqNK0LvCFZf_TMDTK
  f-GJbciVDWc2v07qLoYJ4F8DUU9x2TTDAA_AX8G2wz66SRbfOlpJ3smTIyuartcay
  x9JN4Wn4K4awdbjfXvkGeoH8XDJ2bWOjfUYxux0IPVOsxnAd8S0wzoCnPDsA9kvzf
  Bx_vWcWrp_LuJmy3iOUZLzEMgTyHMOjsnqecnKcEz_G5T2NSQG0XDnYmRkXYrzHXY
  TrW7xaNmPbPrf7HGVsm4nsye92lSwVhZAg-2bHCondmKjpoTMuItCCtIrHG2Gp7Vx
  tIAZ1sdCs9Z-bdo6azjg2B84m8lbquCAS_XkgK3y760WGmuMlNiPZASBGLZNogvyn
  IViSReG7_LUyo3Vs9hoY9Tv_Q-i64MyNwwGn1PEZ5WZQf33wfJR693Il-rHvoYAUT
  oxcuHbQ8honRMEdmMz4bYEJ5pcGzYXrQhleVuo3c1roD9C1NviFZXWdnd8EZtnk9D
  8n22fWeQyfrZ_i6PypZwIQT48Sv8cifU-G0zWoH8Vh3Up_lB2_S9-D17RwcmeH20T
  eG-iUifMzKBDuhbDILpNA3eIVFJ7ffJwYfdin_zK-XY7FgJBeJVDQ8Qzdz2_Gqtc0
  inzurlBk6mWdeJYEHX2r9XchjUB71RRy6EZFmFs8f5HoUzMiQGHI8XG3DpEaRj1Xf
  gYSlA8-clNIGFswjS5rtFK91krxX_2CaxyulskDA-zaGwHGc3EyhnaWeqIEFpLPhq
  B9olt90OxOELDAI7wdnn7csAKg2ODJZZnriiIIwNupDd6WF--Szuy5NEKReiPaBeT
  K_n0EQiTg7RW5ySnS3Z_DG6HJ1_huEk6QHuCa2D46qSkYClx97shzbNaW2ZzqKKPp
  t4T4afua6q6PvcszBtNdPgdI_-sDmxgxfXWaUVbYHoGoI-imZMUSw8h2dmTmxKRJE
  lGrD_-8vdSyi-amcSTTmesMDH0gsw6YjV6r8QG8Pr0AyLQ98797yZjqxmNQaqUkPj
  evxCaPXI4h_EttKBCyMPIhBGuHwiXTvry597yAgHY0yIW57RaNsMeSoN7SkzrU8pR
```

```
r015wWSavurUUqO4LgYGzxYs8GzSe_prq-Yy7yQQy6dcbRpEUshidYgf8I-dgDYan
6noxJ0YsZGhxS3mRxsWhpHlyTicCe806dkX2UfPfZPFCJTQwCobXm1UCrM_DrnV9h
_xckgKfzStUgFjQrEzJzehLRTIkxP8sdgOxTImYxcXzJFIEiB6-IRHGi7llb1AfRz
EQZbcBtHwlep4Fa7SCFKlMXeyTUabCpu6noM-GsZPqHu8Ts8hTSRZVcR0Sd95i0Sp
oZwEYQlJBWV4csl3KqXEh8Jx3gTEjoTUG6doA9EludNCCl4Y4svJ1ysn-3nic2ZhP
s2jSmS_y83Zcxa3RlsNV0j79Hk3TZb4IXHYIlHJw2MHgn7shMRAieG2qZ2SWFR-Sq
Gqw6nMrpr3O2_ctNs0SAXQjJOpQtH5EpkQqq8gGnETtGYaB_LNj745UubENtU2GWZ
sbeZThwELnAgQrreej0Pr49G33QAZjvIksUDxu-YNY9Mg8yy7RgdGYMgv3k7Y5hbx
uaLyf2G_DnGOBdZKB6HVkn5ut87-fhn2pgdQioIB8CaJ6DyvS3NiaTIIQM1X4cV0e
Pzu9AztZaE4twsdhln34CKFV-9SZSjATd6UjAAe0dQ5BHSVhd6PrFyIiRMA9VcnCt
5xBrjkGA-VFDrU6nhMrQ7ZhaxQzV64leZvdGaQ2sF3srE7LeblE0bLP_bm8Igl_W0
9nr3ROsgmpYfhJVNkoaens7CqKaJCMiCsWCJbxmDGkHT5VfMO-niINqPGsaG6wDcb
cAG4EQT2B4RnyKdVSUB4M1SQtn4l6oHodjjiscSoHUZOh7lZgTbCtRVRCKCg8yhkC
vdlz8481gSRYkHBqK2L-4hKRUJRVjJKrsONQlXVaLcedCTyoZXXj4SaYw1zeu3ocJ
EwWsM_-y8ehP4TR18y4gPSSHC_Upgmhqe_4-6qzgY0Bi4TxnBbbDiTZhg_uAxedCq
OTpX2cMbEKmtbtBzzawAnDgob9K3mbdI_voTeBF98rgsgoIFZqrte5kR5qUqQ8Hhn
Y6_8CCUElZP06Jikf3UKK3_MTbioGJmYrNKno6f7n1lSmN7pt5Kdd7Tr6MjKzV9IS
T7emY74YwqbUgFEa47ELxNOsON6N-QEWiTq5NzeLWm2AXACOPcM3hBHLL_5zPBacB
uP18WsLMA7jibp0_E7J0fdvzll8Eb-5Yzrtugp3UVarAuRzGD-wCA3JYLouyDKQv5
0-ipn3tmajyZqA9ENw29tnVGyhkuH_3a80fiFi1nymrUEw"
      ]
    ],
  "ClientKey":{
    "Udf":"MA7Z-CDPN-MRFL-XNJA-J2FQ-MQK6-ESHI",
    "PublicParameters":{
      "PublicKeyRSA":{
        "kid":"MA7Z-CDPN-MRFL-XNJA-J2FQ-MQK6-ESHI",
        "n":"0TPeFQGpnkzgDlETAWjUwnlX8T3OiB0O3c0UoZuMFvpiDM2CS0
1wRCZ3WGsKTlqlgOtQ41UEvKKQesD-4YRCzfop6pyk3D7mJwO-hw3dERtsDwgNbPZ
Lst9c8yNtk2VcCpeQlLbWOTVXxbfZaq4OXbCptydtAAERbjsqOfSK8PqfNmVbM1Jk
NKc04qr6Ug_jU2-msaR3Gzw8SWe1G9-NXpVcX_PzWCYCB1wc11dDqZa01DNgG8nfg
XmluB8QKMp4MVv-Jzda8Q_3Fuwx32DJ2OE6rZ8P-heMXoRlpgSVp4DzegaYUUXyZn
HeMOCVZitEAOyHeNuo1OjNxwc9vdyGuQ",
        "e":"AQAB"}}}}}
```


Future: The SSH application is only used to track the SSH client private key which is the same for all of a user's devices. This approach is obviously not ideal from the point of view of cryptographic hygiene: we would much prefer to specify a different client key for every device.

Future: The present specification only manages client keys. In the context of managing IoT devices, the ability to manage service keys is highly desirable.

4.3. Bookmark

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

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

```
{
  "CatalogedBookmark":{
    "Uri":"http://www.example.com",
    "Title":"site1",
    "Path":"Sites.1"}}
```

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":"MB5I-R24M-QXJT-KDBF-XFOA-DGC3-U3AA",
    "Self":true,
    "Contact":{
      "ContactPerson":{
        "Id":"MB5I-R24M-QXJT-KDBF-XFOA-DGC3-U3AA",
        "Anchors":[{
            "Udf":"MB5I-R24M-QXJT-KDBF-XFOA-DGC3-U3AA",
            "Validation":"Self"}
          ],
        "NetworkAddresses":[{
            "Address":"alice@example.com",
            "EnvelopedProfileAccount":[{
                "EnvelopeId":"MB5I-R24M-QXJT-KDBF-XFOA-DGC3-U3AA",
                "dig":"S512",
                "ContentMetaData":"ewogICJVbmlxdWVJZCI6ICJNQjVJLV
  IyNE0tUVhKVC1LREJGLVhGT0EtREdDMy1VM0FBIiwKICAiTWVzc2FnZVR5cGUiOiA
  iUHJvZmlsZVVzZXIiLAogICJjdHkiOiAiYXBwbGljYXRpb24vbW1tL29iamVjdCIs
  CiAgIkNyZWF0ZWQiOiAiMjAyMS0xMC0yNVQxNTo0ODo0NFoifQ"},
              "ewogICJQcm9maWxlVXNlciI6IHsKICAgICJQcm9maWxlU2lnbm
  F0dXJlIjogewogICAgICAiVWRmIjogIk1CNUktUjI0TS1RWEpULUtEQkYtWEZPQS1
  ER0MzLVUzQUEiLAogICAgICAiUHVibGljUGFyYW1ldGVycyI6IHsKICAgICAgICAi
  UHVibGljS2V5RUNESCI6IHsKICAgICAgICAgICJjcnYiOiAiRWQ0NDgiLAogICAgI
  CAgICAgIlB1YmxpYyI6ICIwUS1aNWVESHR3V1ZZZGtmeVZUOVIzNi1yMGhPMWZVSF
  dwbUkybWRJc2k4MXNkanlzZ3NBCiAgZmRLb0hacEtJWnRLa01YU29Pa0ZycE9BIn1
  9fSwKICAgICJBY2NvdW50QWRkcmVzcyI6ICJhbGljZUBleGFtcGxlLmNvbSIsCiAg
  ICAiU2VydmljZVVkZiI6ICJNRDM2LVE0U0MtUzRZWi1LUFJQLTdXNFAtU05SNy1RT
  UQyIiwKICAgICJFc2Nyb3dFbmNyeXB0aW9uIjogewogICAgICAiVWRmIjogIk1CRk
  8tQVhRSC1WRUpJLUo0N0otVzNaRy0zWlBBLTdGSFMiLAogICAgICAiUHVibGljUGF
  yYW1ldGVycyI6IHsKICAgICAgICAiUHVibGljS2V5RUNESCI6IHsKICAgICAgICAg
  ICJjcnYiOiAiWDQ0OCIsCiAgICAgICAgICAiUHVibGljIjogIkdDaHlORnVIYjZfQ
  m1vZ3FFQzNfUjBhWGFlbW1EbGFER3lZWWRsMkZTQXc0RW5LakM4QXEKICBHbHB5N3
  NRYWNSVmo0LVFiUUpzel9Qa0EifX19LAogICAgIkFjY291bnRFbmNyeXB0aW9uIjo
  gewogICAgICAiVWRmIjogIk1CVUgtRlk0NS1EVk5GLVhNUVYtU1FDNC1MVExJLUs1
  QVYiLAogICAgICAiUHVibGljUGFyYW1ldGVycyI6IHsKICAgICAgICAiUHVibGljS
  2V5RUNESCI6IHsKICAgICAgICAgICJjcnYiOiAiWDQ0OCIsCiAgICAgICAgICAiUH
  VibGljIjogIldTZGxEOFNMWFdDRkhoSUhqQ3dRSEI3YjRZbTc0a3BNLVhWWm5GS1d
  ZWVlwSGdCbi1KSUgKICAzYVBhSHpkNjBNSDNuMWV2Vk5Vc1RiQ0EifX19LAogICAg
  IkFkbWluaXN0cmF0b3JTaWduYXR1cmUiOiB7CiAgICAgICJVZGYiOiAiTUNCTy1aS
  zRGLVFGWU0tNjNUSy1UQTJDLUxIUVktN1FXNSIsCiAgICAgICJQdWJsaWNQYXJhbW
  V0ZXJzIjogewogICAgICAgICJQdWJsaWNLZXlFQ0RIIjogewogICAgICAgICAgImN
  ydiI6ICJFZDQ0OCIsCiAgICAgICAgICAiUHVibGljIjogIktaUHktTzUtckRYTFRU
  bzlja2lNUjVtbE9qa3VyTUxSQlpXNVprVUpKOTdkOEhSdFRBQmQKICBMbjY2aU9mR
  UtDUTBzaV9sOE83NVZVUUEifX19LAogICAgIkFjY291bnRBdXRoZW50aWNhdGlvbi
  I6IHsKICAgICAgIlVkZiI6ICJNQUhDLVFIM0QtVkxLQy1VVEZCLVVFRlItTTVWVi1
  UV0FIIiwKICAgICAgIlB1YmxpY1BhcmFtZXRlcnMiOiB7CiAgICAgICAgIlB1Ymxp
  Y0tleUVDREgiOiB7CiAgICAgICAgICAiY3J2IjogIlg0NDgiLAogICAgICAgICAgI
  lB1YmxpYyI6ICJFbVNiaHFramdqWUFHUl9pTkh6R2lfU1JCNnZHbEtxZklzQ3lRdn
```

```
hsVmY3OU5zU0VFaG15CiAgUEhxN3pKMUFJbDFlYWlkYVMycjI2M2tBIn19fSwKICA
gICJBY2NvdW50U2lnbmF0dXJlIjogewogICAgICAiVWRmIjogIk1CVVgtWUk1Vy1O
VEFILVVKTjItNEZGQy00UEFZLU5JNzMiLAogICAgICAiUHVibGljUGFyYW1ldGVyc
yI6IHsKICAgICAgICAiUHVibGljS2V5RUNESCI6IHsKICAgICAgICAgICJjcnYiOi
AiRWQ0NDgiLAogICAgICAgICAgIlB1YmxpYyI6ICJGZnZFcE11Y3dCb3hBT1NfLTB
0WlVhenZlNUo3SUJYb1hwakxYVFBEdW9Edk51ZGtzUl8xCiAgUkVmZ2g5SGI0Yklw
YlpqbF84bC1SaUdBIn19fX19",
            {
              "signatures":[{
                  "alg":"S512",
                  "kid":"MB5I-R24M-QXJT-KDBF-XFOA-DGC3-U3AA",
                  "signature":"Z935mSJZSJRi1kXTEsD-Q9AAkAu3IuD_
-QJXHa8WVr2xMXcA-23dcvYx9duavojUCUVkKvl1W8iAsxPtl2n0HoAKUATgpSQmW
1X28In4RZ9e60BCW7kFIqbADT4jF0fBOVI7bf15uh3coVtpXAtHehAA"}
                ],
              "PayloadDigest":"0_av1I9T_vQ-6biLixf0vQ-_JLiUttOy
Ynb5fPbqu5l3agCn0lgRFl8uGdSgmzVqzUSIxQl36g-SDrhwApbyEw"}
            ],
          "Protocols":[{
              "Protocol":"mmm"}
            ]}
        ],
      "Sources":[{
          "Validation":"Self",
          "EnvelopedSource":[{
              "dig":"S512",
              "ContentMetaData":"ewogICJNZXNzYWdlVHlwZSI6ICJDb2
50YWN0UGVyc29uIiwKICAiY3R5IjogImFwcGxpY2F0aW9uL21tbS9vYmplY3QiLAo
gICJDcmVhdGVkIjogIjIwMjEtMTAtMjVUMTU6NDg6NDRaIn0"},
            "ewogICJDb250YWN0UGVyc29uIjogewogICAgIkFuY2hvcnMiOi
BbewogICAgICAgICJVZGYiOiAiTUI1SS1SMjRNLVFYSlQtS0RCRi1YRk9BLURHQzM
tVTNBQSIsCiAgICAgICAgIlZhbGlkYXRpb24iOiAiU2VsZiJ9XSwKICAgICJOZXR3
b3JrQWRkcmVzc2VzIjogW3sKICAgICAgICAiQWRkcmVzcyI6ICJhbGljZUBleGFtc
GxlLmNvbSIsCiAgICAgICAgIkVudmVsb3BlZFByb2ZpbGVBY2NvdW50IjogW3sKIC
AgICAgICAgICAgIkVudmVsb3BlSWQiOiAiTUI1SS1SMjRNLVFYSlQtS0RCRi1YRk9
BLURHQzMtVTNBQSIsCiAgICAgICAgICAgICJkaWciOiAiUzUxMiIsCiAgICAgICAg
ICAgICJDb250ZW50TWV0YURhdGEiOiAiZXdvZ0lDSlZibWx4ZFdWSlpDSTZJQ0pOU
WpWSkxWSXlORTB0VVZoS1ZDMQogIExSRUpHTFZoR1QwRXRSRWRETXkxVk0wRkJJaX
dLSUNBaVRXVnpjMkZuWlZSNWNHVWlPaUFpVUhKdlptbHNaCiAgVlZ6WlhJaUxBb2d
JQ0pqZEhraU9pQWlZWEJ3YkdsallYUnBiMjR2YlcxdEwyOWlhbVZqZENJc0NpQWdJ
a04KICB5WldGMFpXUWlPaUFpTWpBeU1TMHhNQzB5TlZReE5UbzBPRG8wTkZvaWZRI
n0sCiAgICAgICAgICAiZXdvZ0lDSlFjbTltYVd4bFZYTmxjaUk2SUhzS0lDQWdJQ0
pRY205bWFXeAogIGxVMmxuYm1GMGRYSmxJam9nZXdvZ0lDQWdJQ0FpVldSbUlqb2d
JazFDTlVrdFVqSTBUUzFSV0VwVUxVdEVRCiAga1l0V0VaUFFTMUVSME16TFZVelFV
RWlMQW9nSUNBZ0lDQWlVSFZpYkdsalVHRnlZVzFsZEdWeWN5STZJSHMKICBLSUNBZ
0lDQWdJQ0FpVUhWaWJHbGpTMlY1UlVORVNDSTZJSHNLSUNBZ0lDQWdJQ0FnSUNKam
NuWWlPaUFpUgogIFdRME5EZ2lMQW9nSUNBZ0lDQWdJQ0FnSWxCMVlteHBZeUk2SUN
Jd1VTMWFOV1ZFU0hSM1YxWlpaR3RtZVZaCiAgVU9WSXpOaTF5TUdoUE1XWlZTRmR3
YlVreWJXUkpjMms0TVhOa2FubHpaM05CQ2lBZ1ptUkxiMGhhY0V0SlcKICBuUkxhM
```

DFZVTI5UGEwWnljRTlCSW4xOWZTd0tJQ0FnSUNKQlkyTnZkVzUwUVdSa2NtVnpjeU
k2SUNKaGJHbAogIGpaVUJsZUdGdGNHeGxMbU52YlNJc0NpQWdJQ0FpVTJWeWRtbGp
aVlZrWmlJNklDSk5SRE0yTFZFMFUwTXRVCiAgelJaV2kxTFVGSlFMVGRYTkZBdFUw
NVNOeTFSVFVReUlpd0tJQ0FnSUNKRmMyTnliM2RGYm1OeWVYQjBhVzkKICB1SWpvZ
2V3b2dJQ0FnSUNBaVZXUm1Jam9nSWsxQ1JrOHRRVmhSU0MxV1JVcEpMVW8wTjBvdF
Z6TmFSeTB6VwogIGxCQkxUZEdTRk1pTEFvZ0lDQWdJQ0FpVUhWaWJHbGpVR0Z5WVc
xbGRHVnljeUk2SUhzS0lDQWdJQ0FnSUNBCiAgaVVIVmliR2xqUzJWNVJVTkVTQ0k2
SUhzS0lDQWdJQ0FnSUNBZ0lDSmpjbllpT2lBaVdEUTBPQ0lzQ2lBZ0kKICBDQWdJQ
0FnSUNBaVVIVmliR2xqSWpvZ0lrZERhSGxPUm5WSVlqWmZRbTF2WjNGRlF6TmZVak
JoV0dGbGJXMQogIEViR0ZFUjNsWldXUnNNa1pUUVhjMFJXNUxha000UVhFS0lDQkh
iSEI1TjNOUllXTlNWbW8wTFZGaVVVcHplCiAgbDlRYTBFaWZYMTlMQW9nSUNBZ0lr
RmpZMjkxYm5SRmJtTnllWEIwYVc5dUlqb2dld29nSUNBZ0lDQWlWV1IKICBtSWpvZ
0lrMUNWVWd0UmxrME5TMUVWazVHTFZoTlVWWXRVMUZETkMxTVZFeEpMVXMxUVZZaU
xBb2dJQ0FnSQogIENBaVVIVmliR2xqVUdGeVlXMWxkR1Z5Y3lJNklIc0tJQ0FnSUN
BZ0lDQWlVSFZpYkdsalMyVjVSVU5FU0NJCiAgNklIc0tJQ0FnSUNBZ0lDQWdJQ0pq
Y25ZaU9pQWlXRFEwT0NJc0NpQWdJQ0FnSUNBZ0lDQWlVSFZpYkdsakkKICBqb2dJb
GRUWkd4RU9GTk1XRmREUmtob1NVaHFRM2RSU0VJM1lqUlpiVGMwYTNCTkxWaFdXbT
VHUzFkWldWbAogIHdTR2RDYmkxS1NVZ0tJQ0F6WVZCaFNIcGtOakJOU0ROdU1XVjJ
WazVWYzFSaVEwRWlmWDE5TEFvZ0lDQWdJCiAga0ZrYldsdWFYTjBjbUYwYjNKVGFX
ZHVZWFIxY21VaU9pQjdDaUFnSUNBZ0lDSlZaR1lpT2lBaVRVTkNUeTEKICBhU3pSR
0xWRkdXVTB0TmpOVVN5MVVRVEpETFV4SVVWa3ROMUZYTlNJc0NpQWdJQ0FnSUNKUW
RXSnNhV05RWQogIFhKaGJXVjBaWEp6SWpvZ2V3b2dJQ0FnSUNBZ0lDSlFkV0pzYVd
OTFpYbEZRMFJJSWpvZ2V3b2dJQ0FnSUNBCiAgZ0lDQWdJbU55ZGlJNklDSkZaRFEw
T0NJc0NpQWdJQ0FnSUNBZ0lDQWlVSFZpYkdsaklqb2dJa3RhVUhrdFQKICB6VXRja
1JZVEZSVWJ6bGphMmxOVWpWdGJFOXFhM1Z5VFV4U1FscFhOVnByVlVwS09UZGtPRW
hTZEZSQlFtUQogIEtJQ0JNYmpZMmFVOW1SVXREVVRCemFWOXNPRTgzTlZaVlVVRWl
mWDE5TEFvZ0lDQWdJa0ZqWTI5MWJuUkJkCiAgWFJvWlc1MGFXTmhkR2x2YmlJNklI
c0tJQ0FnSUNBZ0lsVmtaaUk2SUNKTlFVaERMVkZJTTBRdFZreExReTEKICBWVkVaQ
0xWVkZSbEl0VFRWV1ZpMVVWMEZJSWl3S0lDQWdJQ0FnSWxCMVlteHBZMUJoY21GdF
pYUmxjbk1pTwogIGlCN0NpQWdJQ0FnSUNBZ0lsQjFZbXhwWTB0bGVVVkRSRWdpT2l
CN0NpQWdJQ0FnSUNBZ0lDQWlZM0oySWpvCiAgZ0lsZzBORGdpTEFvZ0lDQWdJQ0Fn
SUNBZ0lsQjFZbXhwWXlJNklDSkZiVk5pYUhGcmFtZHFXVUZIVWw5cFQKICBraDZSM
mxmVTFKQ05uWkhiRXR4WmtselEzbFJkbmhzVm1ZM09VNXpVMFZGYUcxNUNpQWdVRW
h4TjNwS01VRgogIEpiREZsWVdsa1lWTXljakkyTTJ0QkluMTlmU3dLSUNBZ0lDSkJ
ZMk52ZFc1MFUybG5ibUYwZFhKbElqb2dlCiAgd29nSUNBZ0lDQWlWV1JtSWpvZ0lr
MUNWVmd0V1VrMVZ5MU9WRUZJTFZWS1RqSXRORVpHUXkwMFVFRlpMVTUKICBKTnpNa
UxBb2dJQ0FnSUNBaVVIVmliR2xqVUdGeVlXMWxkR1Z5Y3lJNklIc0tJQ0FnSUNBZ0
lDQWlVSFZpYgogIEdsalMyVjVSVU5FU0NJNklIc0tJQ0FnSUNBZ0lDQWdJQ0pqY25
ZaU9pQWlSV1EwTkRnaUxBb2dJQ0FnSUNBCiAgZ0lDQWdJbEIxWW14cFl5STZJQ0pH
Wm5aRmNFMTFZM2RDYjNoQlQxTmZMVEIwV2xWaGVuWmxOVW8zU1VKWWIKICAxaHdha
3hZVkZCRWRXOUVkazUxWkd0elVsOHhDaUFnVWtWbVoyZzVTR0kwWWtsd1lscHFiRj
g0YkMxU2FVZAogIEJJbjE5ZlgxOSIsCiAgICAgICAgICB7CiAgICAgICAgICAgICAgICJ
zaWduYXR1cmVzIjogW3sKICAgICAgICAgICAgICAgICJhbGciOiAiUzUxMiIsCiAg
ICAgICAgICAgICAgICAia2lkIjogIk1CNUktUjI0TS1RWEpULUtEQkYtWEZPQS1ER
0MzLVUzQUEiLAogICAgICAgICAgICAgICAgInNpZ25hdHVyZSI6ICJaOTM1bVNKWl
NKUmkxa1hURXNELVE5QUFrQXUzSXVEXy1RSlhIYThXVnIyeE1YY0EtCiAgMjNkY3Z
ZeDlkdWF2b2pVQ1VWa0t2bDFXOGlBc3hQdGwybjBIb0FLVUFUZ3BTUW1XMVgyOElu
NFJaOWU2MEIKICBDVzdrRklxYkFEVDRqRjBmQk9WSTdiZjE1dWgzY29WdHBYQXRIZ
WhBQSJ9XSwKICAgICAgICAgICAgICAgIlBheWxvYWREaWdlc3QiOiAiMF9hdjFJOVRfdl

```
EtNmJpTGl4ZjB2US1fSkxpVXR0T3lZbmI1ZlBicXU1bDNhCiAgZ0NuMGxnUkZsOHV
HZFNnbXpWcXpVU0l4UWwzNmctU0RyaHdBcGJ5RXcifV0sCiAgICAgICAgIlByb3Rv
Y29scyI6IFt7CiAgICAgICAgICAgICJQcm90b2NvbCI6ICJtbW0ifV19XX19",
            {
              "signatures":[{
                  "alg":"S512",
                  "kid":"MBUX-YI5W-NTAH-UJN2-4FFC-4PAY-NI73",
                  "signature":"4gyw3dH34sz4KhwN7Xazd_2HZhjQBkeE
6h2KoYJ2k4M0NmdqTnwlFgoEkqA6TXYXjDfyxayGC-aAuVkNOHqlktmqSmMmyMfHO
Uk3xfWlk0DVOxaN-muMyZ5IugL5gYHdUgeQJ_wwQr5YexhoNyS2dwgA"}
                ],
              "PayloadDigest":"wKyLuwSyXzEw7soSKcGk_CIsHE_y7nZm
lmrix1BzQpLA12o_gTB9vO4WO6XO_mvD54crjS07iICsq4EXOh4XVA"}
            ]}
          ]}}}}
```

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

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":"NDX6-MX46-AFIL-BMLW-BTMN-SUTR-7DT5"}}
```

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
    = NA46-HSVG-N5NU-EXKZ-4X7G-GSF7-DUWS

EnvelopeID
    = MDKW-3KOD-ZTW6-MRIB-AARK-UACM-PDOZ

ResponseID
    = MAWR-ARTQ-H4GO-VXNI-LT5L-6ZB7-3HXR
```

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 ABQR-GO5I-FPIE-TK5O-M4VU-DALE-WM used to authenticate a an action of type Device:

```
pin = ABQR-GO5I-FPIE-TK5O-M4VU-DALE-WM
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)
    = ADL6-MGFR-DK2V-XMCH-Y4VK-FG4R-AIDL
```

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

```
Account =  alice@example.com

PinId = UDFPresent (MAC (Account, saltedPINData))
    = ACEE-R3XJ-23LL-A562-JOYB-UXNX-W6VO
```

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.

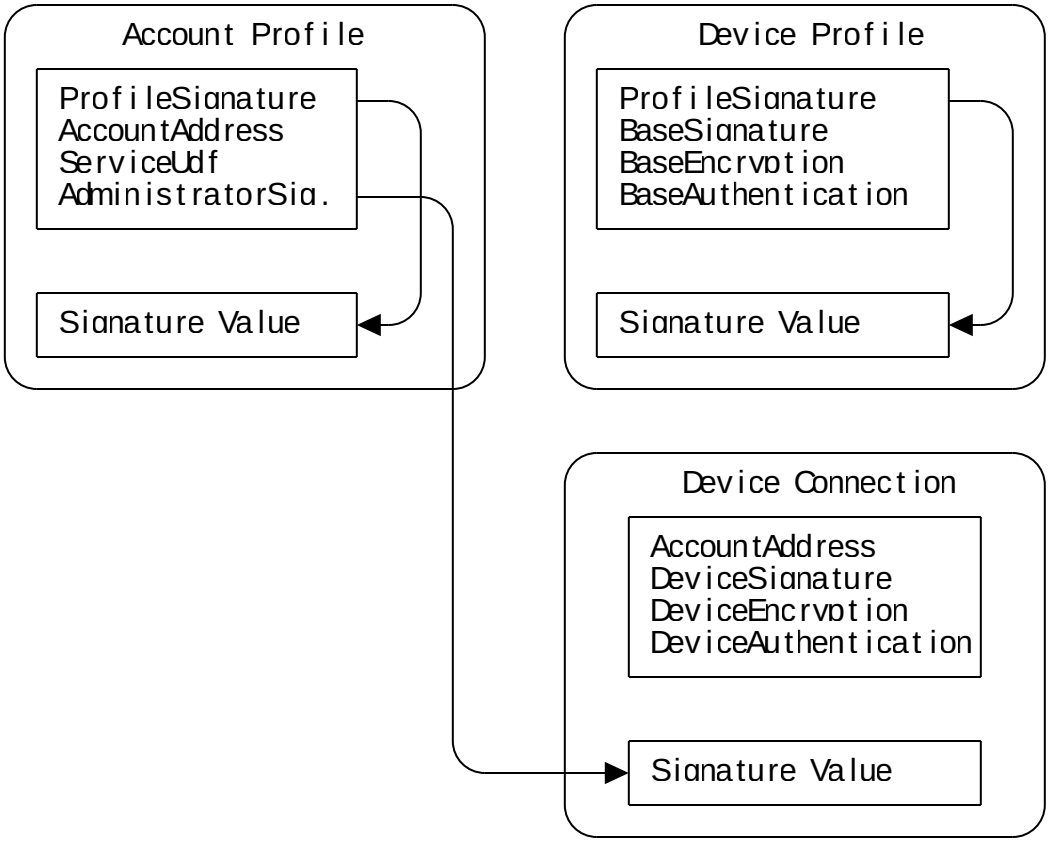

Figure 1: Profiles And Connections

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":"MDDT-KTDT-AZ62-55HV-FFVY-JYNU-Y3YE",
        "dig":"S512",
        "ContentMetaData":"ewogICJVbmlxdWVJZCI6ICJNRERULUtURFQtQV
  o2Mi01NUhWLUZGVlktSllOVS1ZM1lFIiwKICAiTWVzc2FnZVR5cGUiOiAiUHJvZml
  sZURldmljZSIsCiAgImN0eSI6ICJhcHBsaWNhdGlvbi9tbW0vb2JqZWN0IiwKICAi
  Q3JlYXRlZCI6ICIyMDIxLTEwLTI1VDE1OjQ5OjA3WiJ9"},
      "ewogICJQcm9maWxlRGV2aWNlIjogewogICAgIlByb2ZpbGVTaWduYXR1cm
  UiOiB7CiAgICAgICJVZGYiOiAiTUREVC1LVERULUFaNjItNTVIVi1GRlZZLUpZTlU
  tWTNZRSIsCiAgICAgICJQdWJsaWNQYXJhbWV0ZXJzIjogewogICAgICAgICJQdWJs
  aWNLZXlFQ0RIIjogewogICAgICAgICAgImNydiI6ICJFZDQ0OCIsCiAgICAgICAgI
  CAiUHVibGljIjogInF0TDhCYVN3UUptNk12bE1BUXY0MkpsSk9MWFZMY0gxTWNweU
  p1SWxJazhXbVpvYTlHd2MKICB4WjFIMmI5VE5MZGFZUGp1VlVaWHRkb0EifX19LAo
  gICAgIkVuY3J5cHRpb24iOiB7CiAgICAgICJVZGYiOiAiTUFCQy1MR1k1LUJVMk8t
  U0FaTi1ESjJFLVMzQ0ItQkc2NSIsCiAgICAgICJQdWJsaWNQYXJhbWV0ZXJzIjoge
  wogICAgICAgICJQdWJsaWNLZXlFQ0RIIjogewogICAgICAgICAgImNydiI6ICJYND
  Q4IiwKICAgICAgICAgICJQdWJsaWMiOiAiWDZUaF9IOEJZOC1zRHpydWNVV3F4S0c
  1YVloenhTVC12dDE5STlKOU83TmlnRGYxZmhEcQogIGZCT1pWWk9uUDhYNVdTMkJJ
  WGQ3SjlTQSJ9fX0sCiAgICAiU2lnbmF0dXJlIjogewogICAgICAiVWRmIjogIk1EW
  lQtREFFNC02TkJQLUJSQ08tUzVUTC01Q1E2LVNDWTMiLAogICAgICAiUHVibGljUG
  FyYW1ldGVycyI6IHsKICAgICAgICAiUHVibGljS2V5RUNESCI6IHsKICAgICAgICA
  gICJjcnYiOiAiRWQ0NDgiLAogICAgICAgICAgIlB1YmxpYyI6ICJpM1hia3lpT201
  WnlXaWxBeU9DZnFUalBMaUtVLUgyNTJZVUdqRVd3MWgtZ2haR3Nkb09aCiAgcXRkQ
  0k4Q0hRYWtzS3JHTWZDdDMxbjRBIn19fSwKICAgICJBdXRoZW50aWNhdGlvbiI6IH
  sKICAgICAgIlVkZiI6ICJNQk1DLVE3SFctNUlOSy1RU1pPLVBLRFEtS01aNS1BT01
  GIiwKICAgICAgIlB1YmxpY1BhcmFtZXRlcnMiOiB7CiAgICAgICAgIlB1YmxpY0tl
  eUVDREgiOiB7CiAgICAgICAgICAiY3J2IjogIlg0NDgiLAogICAgICAgICAgIlB1Y
  mxpYyI6ICJSX08tZnpLUnp4aExsdHh1Nko5VG05MVNHSWFCY2g0LXFfNnFwNTZ4WU
  YtVTZqa0hSall2CiAgT2hjNm12OUdLOVhNUjZtVFNOUEstV0tBIn19fX19",
      {
        "signatures":[{
            "alg":"S512",
            "kid":"MDDT-KTDT-AZ62-55HV-FFVY-JYNU-Y3YE",
            "signature":"VFD-9f8AXHdm38HR7y7JKsPStGNRu7wW5SXsJgc1
  lbRyzQ0XVyDyNtqR5el9TCEuJKC0vU4lq4QAQfzJlUaa-viM7xhTcvJhVZ_YGiYEW
  wq3Nb1-sortDNUdi7FGmG9C5Nh-ErWxy2oKkH8Nht19LDQA"}
          ],
        "PayloadDigest":"PRkvfQ8djpN_Z3tY_p8qPRR4rTy_ZFEFW_WAqBcQ
  2WpffnNZf_dPVKtW1XW9IpGjxYg2h0zB-hSVnCWViSUiEQ"}
      ],
    "EnvelopedConnectionDevice":[{
        "dig":"S512",
        "ContentMetaData":"ewogICJNZXNzYWdlVHlwZSI6ICJDb25uZWN0aW
  9uRGV2aWNlIiwKICAiY3R5IjogImFwcGxpY2F0aW9uL21tbS9vYmplY3QiLAogICJ
  DcmVhdGVkIjogIjIwMjEtMTAtMjVUMTU6NDk6MDdaIn0"},
      "ewogICJDb25uZWN0aW9uRGV2aWNlIjogewogICAgIkF1dGhlbnRpY2F0aW
  9uIjogewogICAgICAiVWRmIjogIk1BQkMtTEdZNS1CVTJPLVNBWk4tREoyRS1TM0N
```

```
CLUJHNjUiLAogICAgICAiUHVibGljUGFyYW1ldGVycyI6IHsKICAgICAgICAiUHVi
bGljS2V5RUNESCI6IHsKICAgICAgICAgICJjcnYiOiAiWDQ0OCIsCiAgICAgICAgI
CAiUHVibGljIjogIlg2VGhfSDhCWTgtc0R6cnVjVVdxeEtHNWFZaHp4U1QtdnQxOU
k5SjlPN05pZ0RmMWZoRHEKICBmQk9aVlpPblA4WDVXUzJCSVhkN0o5U0EifX19LAo
gICAgIlNpZ25hdHVyZSI6IHsKICAgICAgIlVkZiI6ICJNRFpULURBRTQtNk5CUC1C
UkNPLVM1VEwtNUNRNi1TQ1kzIiwKICAgICAgIlB1YmxpY1BhcmFtZXRlcnMiOiB7C
iAgICAgICAgIlB1YmxpY0tleUVDREgiOiB7CiAgICAgICAgICAiY3J2IjogIkVkND
Q4IiwKICAgICAgICAgICJQdWJsaWMiOiAiaTNYYmt5aU9tNVp5V2lsQXlPQ2ZxVGp
QTGlLVS1IMjUyWVVHakVXdzFoLWdoWkdzZG9PWgogIHF0ZENJOENIUWFrc0tyR01m
Q3QzMW40QSJ9fX0sCiAgICAiRW5jcnlwdGlvbiI6IHsKICAgICAgIlVkZiI6ICJNQ
UJDLUxHWTUtQlUyTy1TQVpOLURKMkUtUzNDQi1CRzY1IiwKICAgICAgIlB1YmxpY1
BhcmFtZXRlcnMiOiB7CiAgICAgICAgIlB1YmxpY0tleUVDREgiOiB7CiAgICAgICAgICA
gICAiY3J2IjogIlg0NDgiLAogICAgICAgICAgIlB1YmxpYyI6ICJYNlRoX0g4Qlk4
LXNEenJ1Y1VXcXhLRzVhWWh6eFNULXZ0MTlJOUo5TzdOaWdEZjFmaERxCiAgZkJPW
lZaT25QOFg1V1MyQklYZDdKOVNBIn19fX19",
    {
      "signatures":[{
          "alg":"S512",
          "kid":"MDQJ-G5K2-BJ66-MPLM-FWSA-665O-MILP",
          "signature":"r-JxVZxihprjMs3buV4yqmgXO7NdXlAEI-Cn2nYF
HB3rlbcNPwmi5z_0f5HpAXkQfFlVJefnxsMAffF8GNbOocmVEdaIXR8rHDkBMa1xd
6iCaWZdv8SAGdTHK0wLHkeAUDGj2wXsINFTMfDqhh_TjRUA"}
        ],
      "PayloadDigest":"aT7dqhsuhW15GSExnBrO1nHQqAcT-uLaCUkJPhqg
AevgNUtTUuWkHC63T2ensFiSjCAAXd1YOvp7L8V7twmvZg"}
    ],
  "EnvelopedConnectionService":[{
      "dig":"S512",
      "ContentMetaData":"ewogICJNZXNzYWdlVHlwZSI6ICJDb25uZWN0aW
9uU2VydmljZSIsCiAgImN0eSI6ICJhcHBsaWNhdGlvbi9tbW0vb2JqZWN0IiwKICA
iQ3JlYXRlZCI6ICIyMDIxLTEwLTI1VDE1OjQ5OjA3WiJ9"},
    "ewogICJDb25uZWN0aW9uU2VydmljZSI6IHsKICAgICJBdXRoZW50aWNhdG
lvbiI6IHsKICAgICAgIlVkZiI6ICJNQUJDLUxHWTUtQlUyTy1TQVpOLURKMkUtUzN
DQi1CRzY1IiwKICAgICAgIlB1YmxpY1BhcmFtZXRlcnMiOiB7CiAgICAgICAgIlB1
YmxpY0tleUVDREgiOiB7CiAgICAgICAgICAiY3J2IjogIlg0NDgiLAogICAgICAgI
CAgIlB1YmxpY yI6ICJYNlRoX0g4Qlk4LXNEenJ1Y1VXcXhLRzVhWWh6eFNULXZ0MT
lJOUo5TzdOaWdEZjFmaERxCiAgZkJPWlZaT25QOFg1V1MyQklYZDdKOVNBIn19fX1
9",
    {
      "signatures":[{
          "alg":"S512",
          "kid":"MDQJ-G5K2-BJ66-MPLM-FWSA-665O-MILP",
          "signature":"BwF9R7byEqkzaUblEujRrko0zPuHn7NwH__14VRv
YH0jTblJSrmG40hujXOKqs9ElXe8F0jM26EAXm6l0Okhi_stdxotXwa8CHLZgzTGO
T9qEKdJElqkZIWLYJ9Tv_vM-VowlOz7jlzP4ThsVkI4fhcA"}
        ],
      "PayloadDigest":"KUSigElHIQenRINVDSSgH5M9Dt5GJLzKUk5yylWM
TNdJ_4bW-JKREQiwutelFZvKv0-rX4XFnfBPwzmUflNY2A"}
    ],
```

```
    "PrivateKey":{
      "PrivateKeyUDF":{
        "PrivateValue":"ZAAQ-APQL-QS4L-SY3L-RER2-TYEA-V4EF-Q3OB-6N2
F-DKDP-UJQ6-KXUN-LI2H-7RXH",
        "KeyType":"MeshProfileDevice"}},
    "ConnectUri":"mcu://maker@example.com/EC6P-KOIX-T3B4-YIKE-OLX3-
BUUD-64"}}
```

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.

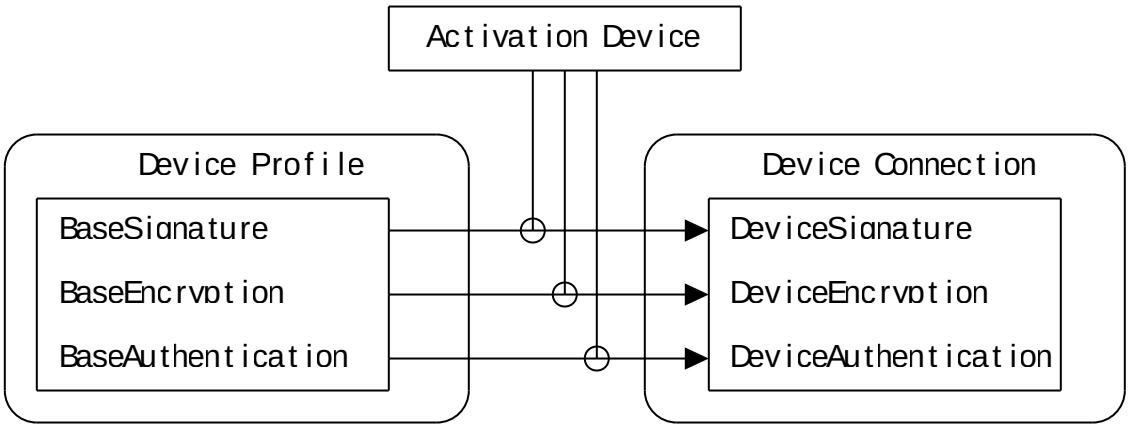

Figure 2: Mapping of Device Profile and Device Private to Device Connection Keys.

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:

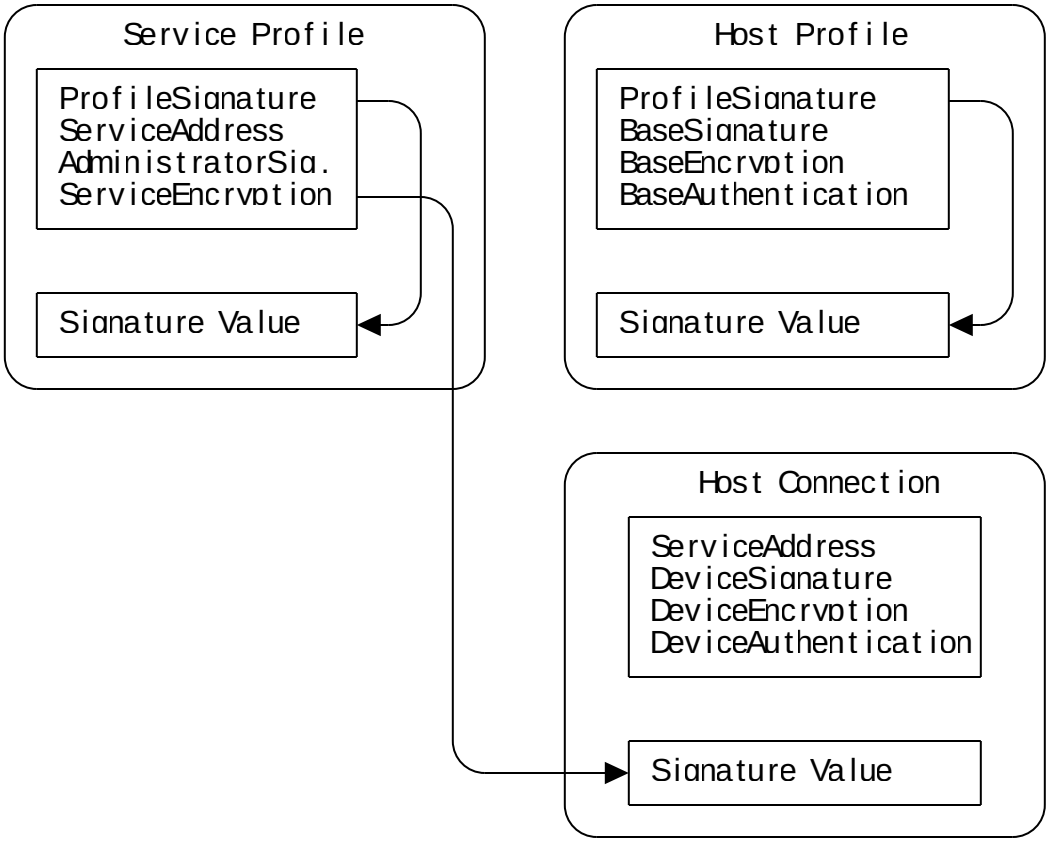

Figure 3: Service Profile and Delegated Host Assertion.

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

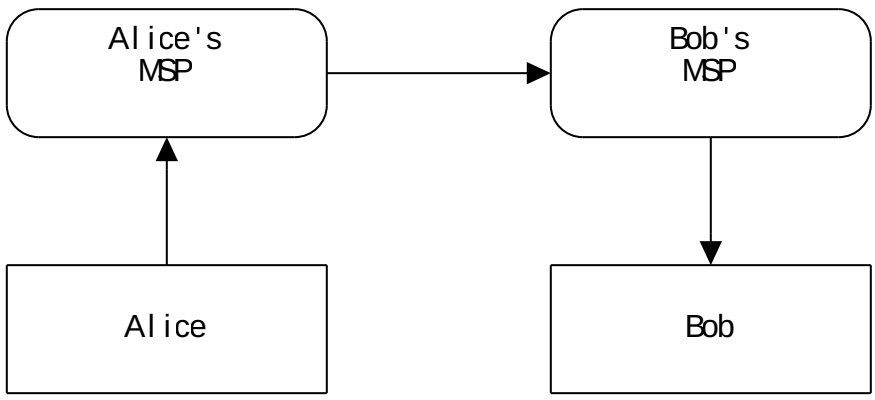

Figure 4: Four Corner Messaging Model

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.

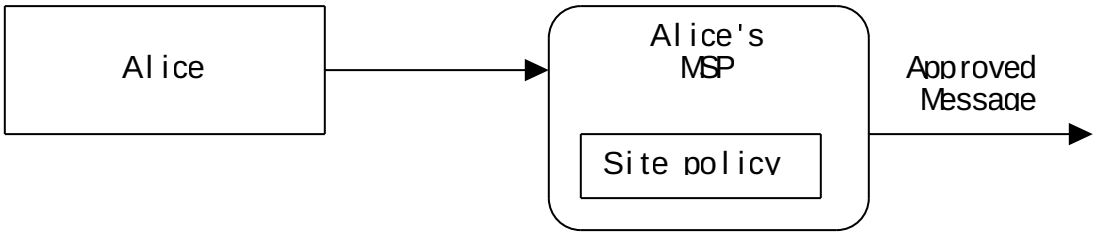

Figure 5: Performing Access Control on Outbound Messages

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.

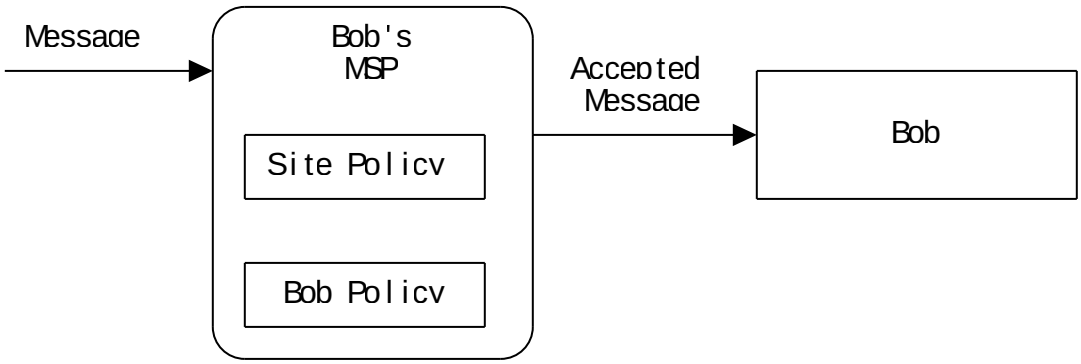

Figure 6: Performing Access Control on Inbound Messages

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-18, 20 September 2021, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-architecture-18>.

[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-13, 20 September 2021, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-dare-13>.

[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-10, 20 September 2021, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-protocol-10>.

[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-14, 20 September 2021, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-udf-14>.

[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] Komlo, C., Goldberg, I., and T. Wilson-Brown, "Two-Round Threshold Signatures with FROST", Work in Progress, Internet-Draft, draft-irtf-cfrg-frost-01, 11 August 2021, <https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-frost-01>.

[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>.