Workgroup: Network Working Group
Internet-Draft: draft-hallambaker-mesh-schema
Published: 28 June 2023
Intended Status: Informational
Expires: 30 December 2023
Authors: P. M. Hallam-Baker
         ThresholdSecrets.com

Mathematical Mesh 3.0 Part IV: Schema Reference

Abstract

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

[Note to Readers]

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

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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

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

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

2. Definitions

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

2.1. Requirements Language

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in [RFC2119].

2.2. Defined Terms

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

2.3. Related Specifications

The architecture of the Mathematical Mesh is described in the *Mesh Architecture Guide* [draft-hallambaker-mesh-architecture]. The Mesh documentation set and related specifications are described in this document.

2.4. Implementation Status

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

3. Actors

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

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

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

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

3.1. Accounts

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

ProfileUser Describes a user account.

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

Both types of profile specify the following fields:

ProfileSignature The public signature key used to authenticate the profile itself

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

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

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

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

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

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

For example, the personal account profile Alice created in

For example, Alice creates a personal account:

```
Alice> meshman account create alice@example.com
Account=alice@example.com
UDF=MB3T-WIPZ-JRCW-QZFM-SCQL-OVVO-AHO2
```

The account profile created is:

```
{
  "ProfileUser":{
    "CommonSignature":{
      "Udf":"MDE2-MKMI-773P-GJ3F-YYAI-UVCK-OMKS",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"jR9urPbosloSRu58lkKG9O4L5BNzqbEs3oq8IzwLqU2Qy
  GRk8kWDok6OOwhYOcRgXeot_eVOAGmA"}}},
    "AccountAddress":"alice@example.com",
    "ServiceUdf":"MD3E-FN6W-3G45-YQ43-QXYR-CU4X-RKG5",
    "EscrowEncryption":{
      "Udf":"MBSK-2Y6G-DK6P-T6TU-OSLD-GCFW-MPHG",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"R5W0TYUycEMhGuxGuCkBTMYB1MgKZ036y052XLVbMsjxg
  g9iDD-yVYVNe_yUCm8QGtpSt_8Eb3cA"}}},
    "AdministratorSignature":{
      "Udf":"MBL5-JSN3-V56Q-4ULY-GY7X-GM3V-KVPZ",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"Zc5n9q482E5RuHsJ4edWsr75axwzR3mWbm5T5lfnBTRIA
  icaGPogbqa54ySA7sWjh490xrvrEyaA"}}},
    "CommonEncryption":{
      "Udf":"MBUF-P7S2-WFEF-D3ML-OKCC-XYOT-6SLD",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"8ShFcmNz5BW9nGObu7_tUZD5hm5anS7Tar53RXDcg5ayi
  Ppb7L8zVC1ljjJeAu-hk9TUNuyXE7sA"}}},
    "CommonAuthentication":{
      "Udf":"MDYQ-JQB2-3EOC-N4ZD-FBJE-H3IY-WM6V",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"2zPl10qnzpGH_1idDFaVMyEymEf6ZmhMtzpmlGfZHZ2si
  FC0SHI4wamghsYS3hFL_qX6mO-SQQuA"}}},
    "ProfileSignature":{
      "Udf":"MB3T-WIPZ-JRCW-QZFM-SCQL-OVVO-AHO2",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"or2X-rP0taTX66FxY68RR4R7Gfg3r6-MIc33QeEgU1wKi
  _HVKyiBnKDXZEexYtEC2ZH7CNqUC2QA"}}}}}
```

3.2. Device

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

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

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

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

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

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

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

```
{
  "ProfileDevice":{
    "Encryption":{
      "Udf":"MCFX-IUBA-KOCW-3FUS-23Z5-NR5U-YOLN",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"BfwkRznFA0S1VirR_017VY-2B3mX8Nf_0tr5okyfLAwV0
  BgMta5eovZBEWl93_9S4VpbjOYU3cMA"}}},
    "Signature":{
      "Udf":"MCGI-AARY-CKX6-OTD6-XLON-JIZK-HGE5",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"Bz5Ze08Ub_2VY7hwZy3fa_Gw6xsDaUMKxg9h3sX_cimQf
  hUWoi5Q4wC3QYPxEomeUfxy_q8YHdkA"}}},
    "Authentication":{
      "Udf":"MCRL-42BN-TYVI-BDGP-K2NJ-OGNI-QEA3",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"Ezrv7Ku8k3odatUvfln__4zTYH9T8Ch-OP7H3MZMtRxjn
  iHBS-m8UsKGYYZUKqxXeB6lLa4_sHGA"}}},
    "ProfileSignature":{
      "Udf":"MBYI-QYCM-JXEY-OJ5D-4OW2-RPIR-SHUM",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"zuSuZB2gaW_FEdWSromPIOcqY9eYHC7Zp-MA-t4zBRWCX
  52UaLTUkDaOAEz18whmatsdZ9S7lFmA"}}}}}
```

3.2.1. Activation

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

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

```
Alice2> meshman device complete
   Device UDF = MBYI-QYCM-JXEY-OJ5D-4OW2-RPIR-SHUM
   Account = alice@example.com
   Account UDF = MB3T-WIPZ-JRCW-QZFM-SCQL-OVVO-AHO2
```

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

```
{
  "ActivationAccount":{
    "AccountUdf":"MBYI-QYCM-JXEY-OJ5D-4OW2-RPIR-SHUM",
    "ActivationKey":"ZAAQ-GK4M-VGOY-6ZKY-O4GL-PR4C-VLDX-2REE-JWKT-U
Z4P-WURR-AU5I-7643-GZKJ"}}
```

And:

```
{
  "ActivationCommon":{
    "Encryption":{
      "Udf":"MBUF-P7S2-WFEF-D3ML-OKCC-XYOT-6SLD",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"8ShFcmNz5BW9nGObu7_tUZD5hm5anS7Tar53RXDcg5ayi
  Ppb7L8zVC1ljjJeAu-hk9TUNuyXE7sA"}},
      "PrivateParameters":{
        "PrivateKeyECDH":{
          "Private":"sVRiHsLzUyNdLATaeG3d75Lb7vzWwse3uruhJC0Ndmmr
  KlM5wMcrq1EWUpb5rwZm2jc65olD628",
          "crv":"X448"}}},
    "Authentication":{
      "Udf":"MDYQ-JQB2-3EOC-N4ZD-FBJE-H3IY-WM6V",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"2zPl10qnzpGH_1idDFaVMyEymEf6ZmhMtzpmlGfZHZ2si
  FC0SHI4wamghsYS3hFL_qX6mO-SQQuA"}},
      "PrivateParameters":{
        "PrivateKeyECDH":{
          "Private":"iiKAixqSrf-v04PvRp3dTfr7rFU11ndEQdwP_FoXtrJ3
  CxLrBF9t_28p7smWM97Glrom4KuhaWA",
          "crv":"X448"}}},
    "Signature":{
      "Udf":"MDE2-MKMI-773P-GJ3F-YYAI-UVCK-OMKS",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"jR9urPbosloSRu58lkKG9O4L5BNzqbEs3oq8IzwLqU2Qy
  GRk8kWDok6OOwhYOcRgXeot_eVOAGmA"}},
      "PrivateParameters":{
        "PrivateKeyECDH":{
          "Private":"uZPrQk8XtRKRHxDaOFDimB7kWcpsDjgV2Zx2SVlwpmZc
  7wmVPWnScBmVkMyGJFNelgRCTAYANtI",
          "crv":"Ed448"}}},
    "Entries":[{
        "Resource":"Contact",
        "Key":{
          "Udf":"MCE3-MLLT-AICU-MQBL-ZQJ5-TBT4-A7AE",
          "PublicParameters":{
            "PublicKeyECDH":{
              "crv":"X448",
              "Public":"pm2dwPeH2FR2wtt7mK5PGPquXtRh4qUJKNPbrkPZf
  uJ709z-EQ9iBF2GOViCoOtySu6ILRuzvx6A"}},
          "PrivateParameters":{
            "PrivateKeyECDH":{
```

```
            "Private":"4A8VYsoBNKkzLp_clmuXSKo37DjSVu7C_jO-vTj-
k9-NU9eYBz3zt9S9DZcsXVGms16qtwF1ZG0",
            "crv":"X448"}}}},
    {
      "Resource":"Publication",
      "Key":{
        "Udf":"MA7P-HMO7-OBDU-K5KP-ZTEJ-F4VI-MAP7",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"Vb8cnHtbqcb_67XkyNuxObW-j0LrxnKjVqa2DAtOk
r9YjCaI9BdhjaOjYPQctqdXFSQ5p9_LHGSA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "Private":"7HgWOueHftKdmOtn3T0As9hw8_zAES5U82KJ5KEa
PyV8y0bNKWADYbuklEHMcVht_9goAOMl_hE",
            "crv":"X448"}}}},
    {
      "Resource":"Inbound",
      "Key":{
        "Udf":"MBF6-A6BS-M3M3-BT6K-JS3R-UEUI-IRBT",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"wqSCFcY3p6-4jO32u3DWI-8YGLpoBj3nboEM_gm4v
j_at2HY_aD4qCV1nkWRhVMrMD-WTox7G0sA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "Private":"hSbexrjzeJzgJFVtOy4XOP2n84N4VLttgP6tbUDA
NdUoULQ0W7QhRHEILN6iuuq5XuxQqL2LTB4",
            "crv":"X448"}}}},
    {
      "Resource":"Outbound",
      "Key":{
        "Udf":"MAY4-ZV3E-CCLL-GJ54-2TJU-K323-BJAE",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"Yo4h5mJsW0pocasjvnd839ICa-dqJK0Q6mNRByzmH
0Wrmff62DuPUgH5Si1mQzQ3eYDj21vACQ0A"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "Private":"iWQoqRUkCeN90KMHPpNMH17fOiOfXNbexlCz-Y9j
KeRllsfoZ-QpOHbru8HM_si8Ma0_Zan0vTw",
            "crv":"X448"}}}},
    {
      "Resource":"Network",
      "Key":{
        "Udf":"MDMP-ZV3Q-TL33-JKBK-ONM5-CZOY-LPJD",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"7yrGvA-wVorqDfuS4KoxD4rosyCu1ae9qXLlhyS8s
ubnwjM9xA_hvkwc7LQeaCOgD7SiV39Kj16A"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "Private":"EYOG1Vzx9yu_GJXj4b1g0M1VGM_OKS_NuHF18c-m
IqHTpioTf2bOkesUJKpzqwMvciISqz4STzE",
            "crv":"X448"}}}},
    {
      "Resource":"Application",
      "Key":{
        "Udf":"MCYA-ZVF4-3QCB-MEQN-7K3L-L6MW-FPMV",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"IoJUXGyFCjQZP0ZoLF5oCiJeGA9UyZbATCi0FJxfm
qdrkKWy138f8VZRKzBSefP3mgPcUARiXIOA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "Private":"GR5BbtfJGwAtLsD_VWS9wofM6PadfcARfHI0TRGv
aki2GwoKT6rwc3UnbWKYnXwdgakQcpnAphA",
            "crv":"X448"}}}},
    {
      "Resource":"Credential",
      "Key":{
        "Udf":"MD5W-K3CL-LJ5Q-YVXA-CMFG-5SPX-TUXV",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"TdGC2V98FOC6E5WC0SJ0T8sTHvN8I4h0_AcJp1Qob
Vo0ZiAKkkCE3zUsjVmR4dHwFTlARj6swnuA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "Private":"eIhnA06b9mENMyqvUxznx61t4DiEXW1hL7yKOiDK
b0tT7BQkZkDQ3p_Dwby_IErcKqExCQqS3pU",
            "crv":"X448"}}}},
    {
      "Resource":"Task",
      "Key":{
        "Udf":"MBI3-CXJZ-ZGOU-U72A-QL5W-ZHNF-RLG3",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"FSl9fvdqdhRYdzzSjSwBuuc51nyDGzXWHjWj28u_L
r7Ppq3qpk7Tr447CwBwTQHdtTNkkrddqW2A"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "Private":"m4TFMtSo5LiaCLhMXQtTHuTksfocg29jDYT9umr9
jiRMCJWGDfMz24sptRRY15E8BjLRYCOOw8w",
            "crv":"X448"}}}},
    {
      "Resource":"Bookmark",
      "Key":{
        "Udf":"MDEC-T7A4-FMKQ-ARG4-A7XL-C5YG-6Z7P",
        "PublicParameters":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"ax1r7_wqg5r1ddHKMWCesT0NhjN5kn8lN0rI9_RTN
CDhda76PxB5piX7Z3_mIoUp9kC8SmR-KCeA"}},
        "PrivateParameters":{
          "PrivateKeyECDH":{
            "Private":"mPdKM_St3NhCqz4pvxOJ3suxYGP11yXpKc3ESn8z
IPSko9hffME6LGaorxBZX78AC6imabN_ur4",
            "crv":"X448"}}}}
    ]}}
```

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

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

3.2.2. Connection Assertion

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

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

```
{
  "ConnectionDevice":{
    "Roles":["message",
      "web"
      ],
    "Signature":{
      "Udf":"MA72-GAIH-AETH-DMWS-6WMP-TX4P-4DSR",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"JT9yarZYOkJlcW43IopCv6oS4Ode9JCgzOTys9xRvtDRE
  Ajywqk70JbSzBucJL9u3egYKCOdUo4A"}}},
    "Encryption":{
      "Udf":"MARU-OXNG-MA6F-7LZQ-R75I-2DSO-7RHN",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"sa2gDOJ6fqmAYsT2FM96o01XjIfMF_DsCzSrGAtQp0YA8
  gsF8_GGYTl3xr1wJcYXkdT0pdUkNEoA"}}},
    "ProfileUdf":"MB3T-WIPZ-JRCW-QZFM-SCQL-OVVO-AHO2",
    "Authentication":{
      "Udf":"MBGO-55A4-MBVM-L2G7-4T5B-NILX-AODX",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"UnCaMwkkjEbs7jHvvKvzlLYCdnOjIIRRSn_xEx-0_OEge
  LLH2bsK2PqYr7tIsjmJGi84ry7FDB-A"}}}}}
```

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

The corresponding ConnectionService assertion is:

```
{
  "ConnectionService":{
    "ProfileUdf":"MB3T-WIPZ-JRCW-QZFM-SCQL-OVVO-AHO2",
    "Authentication":{
      "Udf":"MBGO-55A4-MBVM-L2G7-4T5B-NILX-AODX",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"UnCaMwkkjEbs7jHvvKvzlLYCdnOjIIRRSn_xEx-0_OEge
  LLH2bsK2PqYr7tIsjmJGi84ry7FDB-A"}}}}}
```

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

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

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

3.3. Service

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

```
{
  "ProfileService":{
    "ServiceAuthentication":{
      "Udf":"MBC7-GBU6-CLL7-USJM-UCQ2-72R2-2H7O",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"TqIu0kEwjnjeXeZeID53a9893eRH-BRPc5iQ1_d-J_A_E
  zGclqGTlaHJSOyf1bfu0Q2VNkWtiwUA"}}},
    "ServiceEncryption":{
      "Udf":"MBIH-BWP2-Z43Z-NIR6-SWLY-WIMD-S7AK",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"2eapnJ1gDZN4bHJGoHYLciCcLCMmnTZtaDF-e26BxJwE_
  0vO5PAWCkc5xZeGcGr3ITpdzrWhAFiA"}}},
    "ServiceSignature":{
      "Udf":"MAYH-WLYU-U6TX-S64V-E5WS-5LLO-C4ZW",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"H_yTB3rN_OP86NjklUOHr_WP1vgXQHoWXnOXRsQkv9kgc
  XrMD3FEBJAhxS4bFrQJl7p-GTVe20YA"}}},
    "ProfileSignature":{
      "Udf":"MD3E-FN6W-3G45-YQ43-QXYR-CU4X-RKG5",
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"Ed448",
          "Public":"X-VsXh60gID4HH8Jh-H4Y1sgrCIyYSO3rbErd7be7MV0E
  6xG7XxsSeO4OCFQIBszhUaO41KX2AMA"}}}}}
```

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

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

```
{
  "ConnectionService":{
    "ProfileUdf":"MB7I-UXTA-P3GC-ARHP-QZFB-5254-LESZ",
    "Subject":"MBCS-LJG2-NL6V-RZQB-HRVJ-JFII-C5VI",
    "Authority":"MD3E-FN6W-3G45-YQ43-QXYR-CU4X-RKG5",
    "Authentication":{
      "PublicParameters":{
        "PublicKeyECDH":{
          "crv":"X448",
          "Public":"8HS0cmDkhaPVc-ZCUg91bc6CzPkN5hU7884DodLYakJSu
  JBE1Zc1movkB7hc4WRQMI794NAPdzoA"}}}}}
```

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

4. Catalogs

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

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

<prefix>_<name>

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

The following catalogs are currently specified within the Mathematical Mesh.

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

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

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

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

Credential: mmm_Credential Describes credentials used to access network resources.

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

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

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

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

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

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

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

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

4.1. Access

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4.1.1. Access Capability

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

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

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

4.1.1.1. Operation Rights

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

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

mmm_o_UnbindAccount UnbindAccount

mmm_o_Connect Connect

mmm_o_Complete Complete

mmm_o_Status Status (of specified catalogs or all catalogs)

mmm_o_Download Download (of specified catalogs or all catalogs)

mmm_o_Transact Transact (of specified catalogs or all catalogs)

mmm_o_Post
Post outbound message

4.1.1.2. Messaging

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

mmm_m_Contact Contact messages from the specified subject.

mmm_m_Confirmation Confirmation messages from the specified subject.

mmm_m_Async Asynchronous delivery messages (e.g. mail)

mmm_m_Sync Synchronous delivery messages (e.g. chat)

mmm_m_Presence Forward presence request.

The following media are defined

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

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

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

mmm_c_Video Video data

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

4.1.2. Null Capability

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

4.1.3. Cryptographic Capabilities

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

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

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

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

The following service cryptographic operations are specified:

4.1.3.1. Threshold Key Share

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

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

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

4.1.3.2. Key Agreement

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

This operation is used

4.1.3.3. Threshold Signature

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

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

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

4.1.3.4. Fair Exchange

Perform a Micali Fair Exchange trusted intermediary operation.

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

4.1.4. Publication Capability

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

4.2. Application

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

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

Two applications are currently defined:

Mail

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

SSH Secure Shell Client.

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

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

4.2.1. Mail

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

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

```
{
  "CatalogedApplicationMail":{
    "AccountAddress":"alice@example.net",
    "InboundConnect":"imap://alice@imap.example.net",
    "OutboundConnect":"submit://alice@submit.example.net",
    "SmimeSign":{
      "Udf":"MAE2-33FZ-R2AP-BN5O-YAO4-VGQD-YTIZ",
      "PublicParameters":{
        "PublicKeyRSA":{
          "n":"0iK8bfwAC98EJt6mcDSNxBk7ybZFsHAvJJRBWWymr9fz95u9yr
  qp1HBNrH1j1z4eCff1lzHiLFVK_QjPbf1liy1aq1sFcrZw8ZppkcXmaUVwk0uDsm6
  vEbHLdPn3f4W1TkGYhfC0TfaXu_XOJJQmg3RwWmqBaWz5BZK9zUq_KzoHf5WUfUHq
  giY_ix9Kd6XGpb46O1aCgQl1FGHt039QtnoNeg5mHn4egI7AX042xPwGvyvIWvstj
  eM5TX77Iow56l4-8MJlu5B2KuwdmxaFJetss7paAf0o3GsZpq9pKQ5b5sAvdl3PVg
  wFFZDi6lLXG3AnmFan1V0pS2jyLtEnhQ",
          "e":"AQAB",
          "kid":"MAE2-33FZ-R2AP-BN5O-YAO4-VGQD-YTIZ"}}},
    "SmimeEncrypt":{
      "Udf":"MANV-7HIN-RGQG-VOJF-IYQI-ORA4-ZDVA",
      "PublicParameters":{
        "PublicKeyRSA":{
          "n":"qxhcmT2vmt9aXflFd1LCYWBwAN4Y23S_FDneSi4JIrQNer5ztl
  Ctuc4AtJdoLMuMyfBFhbH6NW_5QzbuPgnK13uxeQDai5Olbrg1izblPnJFTM6ZYb1
  xNX9hoMaveARs_A7EK6k0ij6BWz6aRhEUNB9R5h9O06beKoOliBS4aTmtym586EL_
  MOk9guFf1vCAu0XCfRy6lyNYZQRIx11DTLFAszYMbDEjGeQbkcXRujWZcQgthiF0r
  77KhEdnfcDh9f8Co2DqD5pa6vzHmiYcsJl-By-TsMb7l4gP7DvRTE8iyUx2dYd-Gs
  HFfUzA8jjKLLtYhMrKWoooyOVX4RtvgQ",
          "e":"AQAB",
          "kid":"MANV-7HIN-RGQG-VOJF-IYQI-ORA4-ZDVA"}}},
    "OpenpgpSign":{
      "Udf":"MDGI-YJTT-N3G5-VQRV-V6HH-XNKA-5DG4",
      "PublicParameters":{
        "PublicKeyRSA":{
          "n":"3_XUiJgDWbZn-hlNZDO7t805kp5K1hQv_MbrT1fTlLbRW1gxj7
  xflcduhzBafh_Do7NWDtps21U53ZauC9dws4JHa7WkTRVfVAPleoj5EHux3xJQfPS
  C_0WU88c7ff0xetJEfIbTpDB7Hr7S6CsFpAGEk-7sIYK6_U8jUG_R3WD4z-GNgqhb
  qxjj9_v37lDzIFdwFl7srxjStdVnkGsytuVvsgyfJmzeXP9uKamxj5yWBYbrRKMq0
  rnKGN57HNHT7x55DoxH7_5Kw5Hq9J0XPD0mBrgGOVvuI5kESkLCtJAMp4hpoAsPdQ
  582XlxUJ0euS3h6KEwaDWSIKfoibJ5VQ",
          "e":"AQAB",
          "kid":"MDGI-YJTT-N3G5-VQRV-V6HH-XNKA-5DG4"}}},
    "OpenpgpEncrypt":{
      "Udf":"MC2E-4BAB-AVZC-B6QY-UJZR-P73B-V26M",
      "PublicParameters":{
        "PublicKeyRSA":{
          "n":"oFFbPPtTlpbNHOe2b1WGaeQCT-l8AWUzkeOfLOvVxZDjVzmQW2
  IAnezIiH_xnh-RAR8i63skPV9pMGLaQBJw2Ld9ozjPilPIAq1gHWA_qsece0lfHmU
  dGlHNc_gmkjD3t7IIm2Dn17G9hgTRTdLb2Ktp3KdCJhSLBwHgcR3FGgqLRqV3ueZS
  fBLW6ZVCOQiN_aIH8yuvF2KwE9bRfEYul85k_L91onplWW7o9R4DbbSf53GvO0yol
nm2gjPl91sijTeavSmGxZ8B6m6gJUrACa38bVewyGWf2lWP_Dbvg5h50kkn83r0i-
37syrqqa-KZzXmb8XkdssYBFfWGhXvIQ",
        "e":"AQAB",
        "kid":"MC2E-4BAB-AVZC-B6QY-UJZR-P73B-V26M"}}},
  "Key":"mailto:alice@example.net",
  "Grant":["web"
    ],
  "EnvelopedEscrow":[[{
        "enc":"A256CBC",
        "kid":"EBQC-5T3E-MZPA-A5HZ-AJNY-GJIM-ID5L",
        "Salt":"6sHLmXq_WNMYj_UUBJWkOg",
        "recipients":[{
            "kid":"MBSK-2Y6G-DK6P-T6TU-OSLD-GCFW-MPHG",
            "epk":{
              "PublicKeyECDH":{
                "crv":"X448",
                "Public":"HKVJsj9T6h000KweUdJhBJtF2sdF9752AHW2K
Lvyx-RbCV6yApzCSBDIfUgJ3rpbVB7pLEi-sb6A"}},
            "wmk":"QsJDbr0Nhq8EvpVMO5k1aakXjeCwIjU2nemfiFUVYgiA
jfGFSFak8w"}
          ]},
      "KMx91W6yFjbHZxrNDYmyfksxYAz0rAsBYDhna8XG9FrYy-9mtB-xir2e
H78qbXlnIiEo-vBjqzKJqJVW_ugKd7RXBzR5YhoI7KlKFL0EILIE8BQRmrAMhxvdd
sropv5NnvXeoa_xwDVeMiCNrak4CpfqB_x0JuVx7aMcEgYL8Xg4mCehrCxn_L7oNa
nedsxHIdk6Lmt8yJYqB6E5Pyc2B1PwtExKtNMs-9vXKoJsCCaW6_7GAPKImx-Sfor
c1AzpAtJK0j1UaeWNdoLGesUBkAiXcx2GJbP8VifjmWGaUO_lV2Km04NNj8FwwMQw
4tFlqlOl9T2_ttJAXvKZHtlaT_X4wKcXhs3hwC9GI_YsB7_O4HxdZS4FaFfSQtJMw
XETMan4EUMfoKf3e0ITUwUg39VB0EjwvlYot3Bg2ZxeGiXcRjVmbf7L28vACUBwCj
5d2YxbuOyqRr_6-4O5WW8vg0bb8JZHeWpbQWc2Ku64iJtuW7KdD8OHKmrmks1gPk9
Lzc-_3ei9LqNrAaqUG_qBgpskfhFJK_ed1h-YwJt-GEp4iFmNb4YoFpYnXAlBbIeL
1_f6mHmrQqw5FxfVrAohjBRHmcv2CnZakjCAxZ3yCwrXGzkxDoBAwKegToj1rb8kg
nlEAWTcPvoT3SVkitxsErE2eQTTZf8NrjB_h6sqasHjyvN3auB4W_9vGPMwf5TKMq
iIGYdn_lC_rkpG9iARkr_2u-BDnvndHsGEt2CC0FgOT9d7eGSU9zn70QHc4uro6pz
O-lgMAFpgViMSWgHDZrQk63u7Stj61t-5IjmQFDgicH0NQjWuR1BlggHkY73B2DpH
HQcB4YJAbvemOfPiI5uInezFwwUU5OGhsh-3vJdGFg-Lt3N0vEvaSvAFukR8WwKZo
K2nJ2CXbh4UPdW6M3tRfbgIgdYHe6HmiIv28fsW9o8X-eQaSCLWzJuOQIZjkdNp0n
X4O1buLNSV7oL-yaOOI_L_evK4FV2XIDg5Vof4q6EmHsMkKG5pml7CwnKcS47ey5E
isB4_Y00NdiqFSLgLZxGJB8xsberxBt5ym95GOk3Tu_R2N8zOu-hX7Q3LR8TTLygp
XzfAMOsjymK0WcwgK_jYFWrpekhf7HZukik8s0OycF275qoMVxex7TjlrOESispos
Psg0LUCRULKLmL1Fx6A8-0UwCGL9lXCQQvzkGHX7UPxv-hA480jYU6x3c0snWM0V4
HYHbGtjbBdTcEYYYGoWaNQL0IkweLPVeL0EfurcIifu7GR5Anscu3C4Tx0a__Ff3K
6HmcuXrBaTNCudng8X69g3aBlXn08bD48KYL6Zrel_E13on7jUqUdJH0ha4c-XMeu
LV0uSD2aVL90md0U1tNVCw9i6-BCJh0wt6LGPpcCVDNI3Z6X7-4p4FSl2K-nG6WcM
RfWhDIGjZKLuAwDC6tREt2zyfehe77eLXIoC46vSZCD4XGtHdBQoxyy-BEbjZBdUj
4PAv0i6oEGviISMUhIb8q2rfgs8-NxjzSpt23EWAKnShQJUVD-04Q5YqgUeUQ2GIp
LYFjr9IXMTCHgrOKafgZm5bySEyGCh7Vq9-VTPFsgpsPhS8SnmOxU6cUG-_mF_R_L
SrvQ1b0HuC92kqdyWFCv2DtkOchIQjusGaEktQj_FqL9m0l9QQ2x1ZMhllx4kVY5n
zxozNsu_MXitpFkm6jBAMn6YvLXW-thClHC-SLlfkvcJDgC7ZCdgdb30GTcewVYPo
oAV2-2xSE9wFbI6NaTWEYgB5qfbK176mjetsv7WdmXGXEe7tHn_B36PtCLJQqt-o3
```

```
kIhrmQhaA_mJNtdTB0UKuaCcTQPLsKPfx6wXvRNiG2vdic6fiBpQS-Ej2-fTNtLR-
D-i65vsGvFirSpoakXM-97lcCHGIMyvIErnifJPLnoioTYVIHoyGrsgrroOJ8wuyP
JRI7OCLpo0wZpA3sz6I_trBKFjr7F49uoaS1JiDNcbYdJWVWjFta7mKiuCiLtLn89
DliADowWzVMY87E84bRnUrEZBnXxmLbNUR2FRaNn-D72-1BL_EYicfqUEqeLWk59d
UHW_73_J5LliOb4CDII95cWMzMH3O5vZ2eWxWPT8b1NB4YAcHaVOh4ZvpaFnmbk-d
1F7TyiZo2bIXsNUZAEEyn2QJa_BSJyJx9nEHLZZRQKR6gOU6xXi5IgNrAzqTrQzIT
ICs-5X6nRvfImenfGkiYtkFdZpq8mFdWAGTk-aWhDAPBxTChBVMU2naxPjIcp4cqu
yNYY2faW9sDa7-296gKasHwG1IpbPghXDRgaS8c_OZU9d3RbrfJbw_R5JnU5tO9jd
W7ITfWf0aCS21rdJCH8pHPUm3kbAz-OpWC87NkmdhpvSXn5hG4BkkMLKgc2XGGg0r
eM2gq4AFqdIuVXFhLZ_SZaMgoZlxZ6R-zKu06njyduJVtE2p7-OQhB_biSecuVeVH
M4KMS5LUwN2q_KsQ8YfoJ_UQVUXiWBC4Tyf6AJifOIVQjR-P_vgi_w_gUU2e33xCM
vEUJuZBAE8FPgt5fbbzl38QzzXXrW9WI7ezk3jOBTgKWelFAnhX8eX5R3BQ9aBeqw
iYQhWkF9i2YfBrWk8y5vKWJXCNaa9GwKe6DO3O5YsKx4zsWU19SWrufb2vysLhnJ-
3nXuO25RAHgN__BRL8KC1zV_7QuP21vwIZjsaqbGvChGjVKI_OCLgAG7-rKryEJN4
jGF0m0p_kMIwsjU3SrXvpX9frRaP2Gmwa70_QHODgLSgA-UggN6Y8z_GALKWawd2J
18bYU5yRXiaLc8htuvNqtKIJsTAK_iICyc4Yv3lL8OTfg0ThsNvXzIVXKFI8H75Vd
E1LUoVy0DD7QkepBA0OqzBQCRYb_JXaxoZ8enZUCjoNBTzbL_2kDF6oY7JUCR4JzK
3HT2Sj5KVJvuxu0Ez0_9wZBs1HlIsbJhNN5fPNAV497R1ywTKV4571yZRSmHEbfjs
-O_5mYZa4cgEaYa2VOrAHOXb5yh6xzihToXjdiVr5U_YXlFyZ9yStH4SGFg2fR7qa
2QoS0EQrtDS73VvQHynNljnBTLrHWRwy2yYEiki1aJgyDm2Y73A_CAp1GW5eYFA2X
_BTTGXKDRh-D1Wxq3pjiVLM1Gr2c8SC9puQXPTZB2-q2hVid16yJMfT-4l3vLoEV8
I0taADOnSNkXdVDiJx1fLB61Iao9hsIJF1Bd3EeX6j2MXA"
      ],
    [{
        "enc":"A256CBC",
        "kid":"EBQG-6MHH-4MKO-A3BW-4QAT-5G4V-KGPN",
        "Salt":"4rEUyqJfh9JNQxTMWQn_kw",
        "recipients":[{
            "kid":"MBSK-2Y6G-DK6P-T6TU-OSLD-GCFW-MPHG",
            "epk":{
              "PublicKeyECDH":{
                "crv":"X448",
                "Public":"RMvPIefL9-VCMhjeqEDHtoypp8zhJEeaJFA7y
_2NS6X0WiDrDK_zy2Qzq3G14wSoqfB7vk3OvteA"}},
            "wmk":"2QrhwRJV3-vzfCX72DIp5t5d-i03JVpncvJlar6hJ1DD
Sk7HPsMCvA"}
          ]},
      "g0rrPHxEvZNl0mNnuVb_7QcRweV96um9vKhQBEU-ykzzey_wD5klgY_G
CYRVkA_riis8pe6e0dfeUNbYAOW6riyrayzYJKDtW8BifDTYZi9poWfRkSFgNP2nk
QxyAyeUVFBDrHVTsKLODxLZiFe19To5xy1-wer7iJ4ZBNrhZm2F381lNmwFiGpCrd
IakW1Gu5ND1rUMbAGh6wWjyLSomYVO9bunkafA7A4JzI_sg6BLMorOkjbz2k5sITQ
nQ2ba8h1ffeXOLF9CU36Hj_ub2n75kXoHy1hIbErC6scXPvqRKvXXklT4Ronq2ItY
Mj7D_a7Ap7vJHkbqPvlwR93mvJ3MuwQfNIPVYSli5XwEI7uJxCkOPxCtLC-GYinn3
Dvb5BAJ-2kiFQVGsQAaxP6woOHqj5V3inom9YZ6BMG29xaAXUB7PMySIcRxvIps3u
R4Mr2lZDEkB6h92WUog3R2mGwpIy1H3UIzzngRg_8nwx63v9m7cZ8AQFQvvcc487q
UK7rC1eJJKbwdIQlRNptBs6MLFewmKGoUMkWMWKJIhFeM33hVxxMz4REFgGUQaRm2
-booDuKwSOUOGLhd2oWW0fINnLGgiZDT2qdMlJwr8KUIE88GM65c47je_NeldLKjw
8_WxWIwv7QLYd4einZlswYtubzjTcUi8QCW8Xqu5KdiCjrMhm9VX1jklyWUi8D4eS
Z1QVnwmAAuAkVan61BFJPBDm2c7cZ7WgYs8XvaH2MUxYvI3GkwS1mtzSc1Q9-fLGA
```

```
0qvbesohMls7RQhDRUiHaF79s3SqrkY4Kt0dw0m2t255PTk0trVTTmVDs40cJXiA_
-xTCGiRWUoWGAi7pmxt8um1C9GhvRZGw_NSIaE_gDT9GNEAqX5zio_p56qaTzYKiW
9He2wOIN1XMLla8kWwLufUnqoJKyqSY1tFnuJcd9nU4aBjGf-ImwrhbBc6oGqzzWr
n_PYfMrQyW3qCC1iEexqpq9YBQX97D-MxL8XohsRXx7MSjVioBhi3G_5CatiHj0pH
YudnUNZVH-umi8MM39KguMMMk2BczErevI_GD31fKFKeC-W064SeXPbtWlmyV8x1E
3Xf55oMc2-cWXMWwZw-Ukr7gP5IkpJ7FdcgQAgzEJ4NUlxA2bamnjbWfmXB0vvqbs
9pxf7WcQCJPkXB5LTnhMYxJeO3lZojIMU1xK5YPe0pDzR58iJbD0A9pWYbKd40l2R
s8_I4VQ9cogYmwkeGcHrSLrqRf_ow0wd-vVYIOj7v_yTRQx3Eh4tNcLE4ylVWXtCf
icxa_kYCVT0dHXjXCmWmkNLBPrTBVc7j6Y_OLBLxdWasz19pWJuuMkZtDl_QN1Xd8
T8N30-vr1tV6F7hdpldNyKkaQ2YFyHKoEI4siu7s35RJ_6H0ExwcaTQXalsACnn7Q
i4zld1bCEkffvocV_uRDCFIQWS06KLFZdG5re0jsfCZe8I_EhJMhkEBtUHUtw1q8E
4KbPIHbeqZ8iZN1YFdDyOn5ie485Z6r5ga777rkgFDHO5LkuM43xqnvCEw3FNIHKB
pdjasFI2QuZbpMWDqRNVkzJpTyTBMOtjCDn4hi-YPSSGgbAy1lVozSC5KIPcZ9qQT
krf2zEu5-ETMvamYjZp-CiZiWansT_tcHAz7TGpwNpgyixkJeVJtY2CJdbgjgb3gq
VJMJZ_B9_GgzGv8WkNy8LfdP4HDzuPDV1dNRcMsNLroQEz6LlLvG2hN2eJLmzliYS
My9y8W4tsleXYBdmcM781wpfxlx0Z_Kc75CssJ8hXg5TuEQpIQwjoqbBNHh_0MFSt
eMbK1UqZ2exrRGdrmHJYk925PagRH4-HF0Ft4faQ6wgbsLq2xBDUaCbt6q2Umm4Km
99E2B2zZKR-oLU_6L0RGgMXY0l32W8aH18R37CAU8SPRJlKsaVjI0X940groC9Z8X
sjuiPrKpxJK2MWQiackcwH7rWhvWbcUccFi7blvgr7EjRrNmABh4bX4KS7fnwvquj
875B4zQ2j5emZtCAp7ihr_RtYlOVkqhrskzjH4DAI_MKXA9ZtyEZWezjRs0EyBBwz
3GV855L1qlLUJ4om8t7PUcWbrIaAHnrnsTxgC68-RggTf_iLdjph2TnYSjPLJgLq7
JpYXFUfyU1otkQF6Wc7PWLrNDxa7F4TjpVevO4I8fL9bkxauDC3xvU2vkxNtdBv7V
bXkHYwO4zEbNLVuLTrW_bfGhiiT1pVs1iexx-ux233tegsASdBISFmBg7j4i2cThV
2mYjWb5xa00Dt9aV6DIp6cRrJPnQu0yA-Vf0ifgQtwYNW72hWJvXXecE_jAjq95ky
drxpJHX9A2ODNYkQlF44DamyZLGhbn9cn4uoX_brLtphn2V8qMMhD9NcayVj_FLDo
4FN8NYaVCqrncl-BO4jFH0VzlvHN4DPFK6yKjKNKpFW_FHhh3PtzDYvWymA1tizTl
Ien3QR8eBgzoocEZGTKFjRbFSP_1IkSAMftIS3KyLxA74BzAQ2s01oUQGZ7apa1ei
1sZ5vApQn4O5JHuX_BLak9cdAVcjPYswlLi-d5dkGPDZmV_Tv8r8AaWC_34u5MrPn
9LEzDmi08_JPm4y5anNpITqUel1G7vd00nLHFntD1Wiligr8WReNCPTlUb_miPe9V
ex6FjFXqvJx0fqZUw95ibTkwsgVeZv5DCdHPSGMw8KVzmzj5taaxR_KNeNWuvPwOi
Hlk_k8HUZDbuuN6D8Sag0Dv2YqVF3-W0g-bjhV01HeI0Nk05YUSegv2TnMFWThbDi
xjYcvrGdD7gpYqI5bgTvwcjVFk6MJEfTNnUOPINRLWsXHkgQVhR3l6bTiWPn-C0KB
LTHoypudzbZZHs0urryxgdCI2Pwy0uT1TWsHsdRDEANCha0mGhQ82kclyjchTVajF
r4pYCmD7OjTNZSqHGJE0S3BSXnLpqQJH4bpiwlFjAZ53tY8W_OYKiyctuTF1aWWLD
2DoIDi-HXJHpxgeCMfHpGTNmwwAazo3uaRtPG-WT6GQQnI5eizvm95EgvWHLAXVwT
Cf1zpACXDXfNFZ0ml1zJaxeihkG_kqBh-iG6BNHNKudjpkFx1v_akgviZVTgRRHoH
dwvAIxEbv8FKPAllM9v5VcTZCqyFYXIu3y6znNcqDXZPuqmpabL1DRQtb8-rnCod8
RlYX5-ZnKjrk2yVyFX3CLBf1d3I5VVVNAHVJtUvA2Xqb3w"
      ],
    [{
        "enc":"A256CBC",
        "kid":"EBQK-73NN-Y6SF-V7O5-RFW6-LT46-ILTA",
        "Salt":"dBmXzbaps7tStwW6lyYKNA",
        "recipients":[{
            "kid":"MBSK-2Y6G-DK6P-T6TU-OSLD-GCFW-MPHG",
            "epk":{
              "PublicKeyECDH":{
                "crv":"X448",
                "Public":"QlxU0Yn3qvnLAXHZUEXwRXsQlE4sFn8uAdDkP
DiFBCnEkAiSqfphZjqQeicj4MxSe8XD0-6Y1CkA"}},
            "wmk":"55EumGK8r--mJO7yoO7C0PxQDrq9s8EmxPMQqJcEz1U1
X-aHdCe1wg"}
          ]},
      "rH4yVLgpJWWC4JBLk3qlokm-MDAaW-OY-MUaNuCN3R3BgjnVrrHALyUM
PGL6OrSsKRRS85OyKM8W4Nul78K5pDrCQD9vu8rIPKzJ2GoZVl_ZnJ9n_EEgCOnMB
Z8CM3-4IA7WW8S-ct1Tmh7W2xUDTmYk1B7RPz0gOY9IsGVInG3xAybw4Q62ML0OIP
bJhPd-v2yJ4TwVSPk9QY0Ih7-3v_6BHU2SlHQygMwuOkRQ5wnMFWxBaespX7QeVUp
woWdZwkhudzyg3ab2O2msU-HNqUU1HuHEnoC_TkkLRnQ5jsqpsCNvaATML8InFjp0
xiIUkY-w8azjbdTr_Ne37rji7HiTKGi9Ucc9oetLnceOgSkY5pV1zsuE4e1d-V3tK
1SGIFS_8l7UIVcn0L8vtLVzgrJgJgCo-tbCUezrdcranPKM4YSKQEipO1w8_vpjBK
JcThJJyrHqf15tg7gQpT9iWa8fwtxiDw4SsZrA_9uuSxOfC4oi6vrEqVkLqddXGhV
g_2_5V9cspGVJqvfdm4fphzA7uj4H0TyXACzOnb4hhTNUvRGR8bcN_8915RXcH5zB
QNHEsil2gTVwjg0u4mAVUHacC-BrjNL7pmmfCoLFsCgzPrtTsW8QyF3fCt7FTssdb
vzHKq5X8sc80YH3_u1qp4Pmhcun8aWqbW5Nq2BNBJY2aNyc7WK_dJW3WOTFKw49Oi
osHUeiy42PD-ZI7ANdRpPHN8CLtCZnREGy6fg0RhOjDh8Qa70P4HOkn9VaU6LhpOT
FZsSohqaSO2Mapzc47MTw8vnFEQy5nlaKrfxkcfbk3WE8XuEh_rD4MTSIUrTIDPoJ
vcVZEB3VMEB2-RNJIIA8iAUbfzPsN4H0lX4ok-qLhou1cS4RISrEgNVTPD6SfbyZe
2DgCWTzeRCyqQrHJ4fDr7wEpni0hu5JzvMkRZmOBLDq9QwLiZ3XrkBlhgSxZhSreF
Ki3bd20eTYPtxU8xT4w1CvsAAJBA5K_2TLge7wXM3yEzMKp9dRpP1eAqwD657uxrZ
O2HhnKwWZkVImSf-nbn1M9ES81jEsMB5rythupRth5arm7mTdEkfGdbbwUMn_dWxk
sJrcmXenfuRhrwyFe5WkukCoyDGIq3mycSsMfCt060F0U5IFy1r8RJ6mm52VtO7XT
-cD7ir67xbIR5D9t-x5HI2HXg4Acfosooq1Z7b932As4KbGfmjglVF8sWi8sZxPCg
Ywzr6Al72el8VaG8jhHwwNIZHLl8sQdD5AmlvGKPDFCWraYUARnAmUim06yAIP7RB
dFIbJOm-PusdwYS0l18bGVfFgdF0dDxedF7-o6eySt9AG9eavG0J_eiGdR-NZGYgE
T2-QwE51YLVY2oohL6Sh7UJG6TDaHU9eD18G0B9yEkq8vrXiuaLuWhAXyCV4YeiUO
ifVgX8irOOvgRVizqiindHXyxJOMjlcW5CST9O2fjE_2zk2c45SfXN0HAL-o0uR2S
dbTFMZSktuQrhw7_ONu4wlr_BLuXabNBTNEEMvef3_Jte_uiQyQAU6ZZG9qUBuq9k
ckOMQD2Unrix9D0C7Vlw0my36WAFm0ppQfdNuvJw5Q7mC3-hAMc1_SVpQLs1emo4u
IcRrtPzJJ9mXEJPiBg6vDKqoQxuLkxLSmWuKSlmdyKf5-sgE9EL5ZGKcz4s1eg9ra
BM5vP4DFOCLcujo3r83gxfr-KXN2HUikU7Al8tPsistAHcvTex12FDFpJ73Ktocvc
_g0v-0WmJrf6iS4pgFX3gGkVixI7q32ALkKVQbxQx2ZlR8xYt04LHhoaSBl5pEThx
B8NeimvmTgSrwF2KgbrpYbzCcsLuiWgQ14RHXVDsRfqx7rLWzEp-4lVhtzks9va4n
pJGvueiv4GEIT7LB2L4E-t8u56KWyGwtba9N_O2DWP7uB819uSxeUi7fZ7WycO-Hk
aFRH8njTfjm-jXckbmImg5lIVgH94jFhaUnqTW_3F6UARF6FKjacQwCiBQpYFAcJz
-nnNukfQ6RQ_QeXONT3_3xPrzRfn4092Au_fOIZNA8n9DnIzOt0cOeERnxyEox1rD
K_dlOq5TIGQF7p_krpK-T1m4ygSQbHYbxmK3MowcCHO-BRjkbfhuzUUsRTLmjauz_
ru3V7KQHE8ofX-Z8HYM-XVARrJnG5uEi9j4ZciWchJhGYEnGNyX8xCv1_Gr3hyzDv
dO8AGAuO05wTLHCD0x65HWNkCekrNNTAX_33jspnxcH-nB7muYdXGtF-hqVe5RhHk
poRX04X01vSezWJikMZzuQ0X-_e3gQr7-1aAVY59q2MP-m6CzYnhLLsKGE7liOMRL
4mbJkGdWnFv1HLYjoae6KgWkANIjwr87lY64lD4DHHcNwGuGC5Q4Ocyn9CpOIMAq1
H8yHWTUgrDDVA5Jcy0YBdbZVQf_9zbsCltWH4eBQXwbr1tWF00PCkwPZdqEWC94um
JfkBuZWx9b_yUU39wAVMgOazegTjGrTutXC04honzfy_uqE1BgQ8zyyFuYe79xyQc
7I5Lg1I8ZCXsN_RTeIo7WCP8o_3OUkbqNQtDwkpvKYk_JPcJs3mRsSNymBceThEk_
XGC9uIHmCISYj4l3T_ld8g51-kV1rcxRtWuqs7onhlD2W3d6GmNHqUdmYk2pNHMFp
ank9JlcoiMlayn-GS97F2zX5nldTN742DAE3ve3nSbePXhCfw3C8RuXpGLPIQQGBn
nKAyUWxhH6jSJTWXFHCUDd0PC7MqIt-boTyefzM5jAZN3DR13nxcxdXEZzQV12KHa
wWm81UGjubwJPzxli3BpFbuiSQ8toHDlknnDDzZOS_wMixJEuCrd7MlV1tU4XP7zX
5VhSxYLypzsPK5gFORT41iX84e0R610mkOeOmj6CHzRMy0iRZpAiNLEd1ZPLRFEDM
```

```
8TZZHBWKBiPlbf_60gMT5XcmWeY5E764WGNftlfDp4MRuOTua7KAOeqACzdKypPzl
PZUFA7WwUzygiEbZisinzkQIoep_TyynKyR8bLJFd_Pv8rxzG9uZsyOTBdIKqo0_-
8sv4z4e5k-Y4wt9Kfx3AXH58rKT9qpfPedNHGxbUPRxposSQIhMMNXFwB47_5K0Kt
Hz3qykaqjyCbZujOPucdWwO4m0fg5vwiWuddv35rl9kUXDFEW-SK223jneCrC3Dge
aR09Prek4H_vPdNECAvfeDFk3U6d71eh01mlLTsJjo_gIw"
      ],
    [{
        "enc":"A256CBC",
        "kid":"EBQD-2DHN-IDSY-TPRB-ZFN5-6UXR-AJPE",
        "Salt":"ACti9vLtuWjsv2shn-jhDA",
        "recipients":[{
            "kid":"MBSK-2Y6G-DK6P-T6TU-OSLD-GCFW-MPHG",
            "epk":{
              "PublicKeyECDH":{
                "crv":"X448",
                "Public":"wBc9syGLbSOIO179_Irl5VuHBTjgH8__Gcbt5
t8rP9Y56wNSpzsDTzNY8vHzW72TH0f6P0G16BEA"}},
            "wmk":"_9wsar-YambFELlbqeKizG81M-RBIGVl-3FBeNlqznaa
viihI2vSLA"}
          ]},
      "6w7ZHeK1CmbmuLLTBM9wKsBrUuDBh3ZssoRvsPOLT13PLfmNToUi7-0t
GCHQnoJSFzewfJj1NtfSSo0qMpMIP4jqY1a-2MStl7P8xE5UPUomHY_xHsLA8vBD_
01YIRU0V-8n4BAwBEiLddP7cClfnKiSlhqgFVNnrXmFiTDhJbIUMBBNmkB8kVsCwZ
G5pvpnBssxRIjl8w_RFtaJzq8vVUoHDxYZqvNdmnOiVi4ivNzzDvquFUOCTJmU7mF
IA8qzDcbmkfPdfimYhudRenhIFaGmPY26mtO3wHT_s3trbte2n-j4HOuDnH8cFs5O
SFub4WfbJ5AZGcCMoRmDDthQC60NigXVcGdK4Oa3UmfYc09oE0cdMXf0aFe01Z8oS
elPn6dLWx_uEezNuv7CGEWi3hGCdO0qybGkkxSLQdlndh_BFFCwo_yvfW8Btrs4Jd
ynB3NdA9OLHn3GETo6KTebwJnTubLruCeCMg9XvIhfuM-lQ131huhRHiZQISFoiDi
2WszPxqCADaVdrtI3fCrVNZAtIZWaW8WqFg3EESiH0h5osOaw-F27wm6zN3BnIxID
vZQ-8YtL-_Xm1MaOn8HS8M2TWn-fLlkioRkQVvqb0Vde0uk1iwIrYHc4nNJifBlG0
h0fV0rQ8OlygOPiCAtxYRwGD1G53ekwO0k598BXBKHE2nVq_V8Omvu6Q2EG56585I
ZxXcpMcJuv1BLiH9ImzMWRvfmh2PsUAtRnZN9P_CiOA3Q3x_BRCo0dTHOARVGmn3V
M-QkNzTbYQYmymxipmj2VBznfLKYzMaSSoplnH61eVd2RoYEZQa2yv3EiIvsJF2fd
Nw-ACWJ3qNXoPgYcn7HeImR_ytCgn1f4v7d-d13t1ododCWlt6LoKQBRCOw09a-T5
bKN3MKMRWA9uLd6kuEzOzT5yVq4IqBYuafJql2WxyJq9GkKtKII_Sv_A1WCqS_rlR
eSI_G5ZEUejrpyre_68WyKWjIejDjstCdXIRqb8V03DdQDXMssilTgaxjffPI6yxQ
ejrLOKC59sZCzHWeQ6wHIhAe7LRKo55wTgZcc8bBQ_k2KQgmtAybwPFFgISBRo62j
at_h8rTB0QU9i2-3Fq4J03JTPp0kDKkBe3rljQZ4feG8IhF7YnSCraqleBGGtIAIJ
w7cJ4iYcm8RUG4iXiPixQRUNSjIVhNWpPyFUVLrAmstZ7DRWH-16tcG4OYGBuW-7g
forQeR08Lc9_hnt0gVFGpefxtrGzUtkMqn0N8WIlnxzId2crbQp1IsSs2fxg0qc5G
hprmrQc5OpOYOWTvz8WGARUMYwsWqlEz04na9FFAclWk3yefnSz-X7PwyxataJ8Yn
VPkEW-AZ9GafU1VigdlK2WideWKZ298tfb8O_PCHy-ev1Zzv6f4hl5KkpVji9zuuZ
rLZ-vadRrFb409gmys4Zl-qpHjynMXWgvz_7TZKmy6p3_qNLasUaTK0PVeMC81xmn
8K1gsfI7-0uMNrNoUyDfigaX0v-GCTUOPl9xrm-8pHuO5GpLFhdvA3YOKZO7g7mdt
TaWPlx3_gJxdBsZFV8ia_LGMKHuaucvD-iGf0HFrwVloPF9nkQvKTOHLNYkSV3N-v
oJ8wuCGphe3A5UUjxa1hiAAybjCeV8s0V6jp8eRm4MqEjeFnUxM3GV8bj8SB70C0D
IHPuLuIIv9LIStSYPrBtCGyNrKzuBaie7GZ3B69kfZ4o-kNXSTdRAYM5BAbXcGb-R
hf6WAdA3isL9GUmBCndTqaEZp2FIHuyV8VVFIDzWri0yiAAU2gzNIHbn9igzEDjmz
ri_Ew429KPeujH2Yw0sxGRVx5I8lS4qNJUW5AxD5EwJkwW_L-6L4vEP8sYRDk52Or
```

```
51e_I9iCQNbOLDRW-HSdObtiLP8eoBL78wUS_qvz5chtrHX4cUUvOk24T-BLp5tsD
4dbaDC2GOqs8f-iiE2AjcTuKITAdes_7Ca__fYE8w8FNr4fOiWP81mTXoHEbjciRx
B35_yDjzrVNUmQK7N8onVebMOPgnfzNLm7vanF0fKqD56pbI6BIUDYLn2_q8RG4G-
yKfmsXHLqNEF1FFVQco1gu1HyGMD0axcT4foZPyv8QYTHvz9QljNPh2IdDvxAZnYV
9JtR9RSPftNdAcWSeHI3Vu2gd7Dja2EsOHS2va6flqUREoK0U0MTlsDgFRPNVLPVR
x114K44ufFLz9dKLqqE9kKpuLohUUO3p6o2oMYh4SxkwdVtzyLti66MvR1ta0MUg-
62htY-jlecoYpPe3fUJTVBTKhxPBuEJiJSuZAbmCLQyvm1bHbgONSSRt1NVDcKMfS
zwCjMB6aS12Z4FyJ5hn_zjtG2kGSI8w9riKCRsute1uFa-2fTjxXoSvb7zrrUSiU4
6oaYbUhhslbtl_d3doyAOZ-q-W4ZTC1FfFF0Z3ARFUiEZeFMEPKuBV8EHcalBWcKV
2TKAvTK5ZrhdRaGC6Lb7Okz0iiKVihaVx7kubK_MWW-6dFLRQ9Ve8mU8uet5Mp4_I
cmr5qQxtimawC9doF8ngyBDPsJJ7fa6kdlS3lSbW65qyXVW8hcBK1XHUVaNKoQfjD
gfFYuq7sq2n70xWX8qiI7MBXpDvOJpyf0gjm8kbRyFbPXQC6ZXQI3ogq8qAnl1_ot
GCx6qJYpyAnLdz00wlQ4JAasZKS6eSeHjM4vb8FHaXF4fRcr4VHMOURRJoQo2XL-d
2flfa7o9-BCaWNt-EX2W5Q9AzllufEsTgXBtOK5iC5rx0Yqmg_39vhuw1j6548u7F
zTbWUagkXg_yPFnPs2aNlXNIh-Qvj7OcYhv_znzhc9VXGdheuMNUjSpK7QPsF1l5a
hjQncodbfwMO1wYtaCC92nopf-cvXAqQtMyEWGICcdwq6FxLxa7YB7FEa7pNvMgYK
1rJWfLD4k5l2tMepW7euuS48i7mBYqRJdDaMZZ1ffWfkteHlk65P3i-O84bb9J-ZF
z7ycD7IziQndvRbmqhZodu7zY4AJgnMv585FIkYZWDWVlF8XpVnXFXN1ZfOb1z2ZB
vsHqc6m_y9euKhFW0XC8qXuXMMe6LzZGqlGPlWXkr_mEiToZVi5QIOr4A3DyAIKmq
kHBJ1MPDiNcECDttU1BbBsGrlC2vc4R9xY8WnTl5t-1zdZqyYFbB6iuVzDL81KnYt
fGO08JRUtBmoS_2IUAWuUzfBA8Z4tepyA-cxzZALgK29Bw"
      ]
    ]}}
```

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

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

4.2.2. SSH

SSH configuration profiles are described by entries in multiple catalogs

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

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

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

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

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

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

```
{
  "CatalogedApplicationSsh":{
    "ClientKey":{
      "Udf":"MAOY-3KTL-BIW6-BS7V-EANA-AZTL-5Q2Y",
      "PublicParameters":{
        "PublicKeyRSA":{
          "n":"wKWnEU6lrWi0oY05uOlefb0Lf1d0wBgBAJcoTKmabaZuIdv8G-
  495OJ-QyzRI8yRdqPMWFhUidHMpN9h_8D5mXXzFee34iULjfdWdteEM8M2XKGcKa2
  -8lQnAvxX4VX6ZfP0gUJ9xn7a0_j-5cnbMsKpchcG5CcOK5NbBwiA-GOwr28Z3dNm
  f73h6klWf5X7EgMiXxsTP0mpivyNOewrN6t0IM7P_3B6YJpk-SiVKVKCosK6RsmMs
  LvuV9yKh7K28Y2fSIlrzOS3AyZnoR_gUlqovVxsr-gwTjrQo_UDfRoJN_AGEtOsGY
  suGaqPFpXw-4QDSJtC2QqnW6wt7ovcaQ",
          "e":"AQAB",
          "kid":"MAOY-3KTL-BIW6-BS7V-EANA-AZTL-5Q2Y"}}},
    "Key":"MAOY-3KTL-BIW6-BS7V-EANA-AZTL-5Q2Y",
    "Grant":["web",
      "threshold"
      ],
    "EnvelopedEscrow":[[{
          "enc":"A256CBC",
          "kid":"EBQM-5LE4-Z26W-WCG6-OSZ3-CLO2-7DDM",
          "Salt":"v0Le6nYg2hisQRiYi-lRdQ",
          "recipients":[{
              "kid":"MBSK-2Y6G-DK6P-T6TU-OSLD-GCFW-MPHG",
              "epk":{
                "PublicKeyECDH":{
                  "crv":"X448",
                  "Public":"Xmlb80F5xDeLqrzGB1tAvvoV2AqUOFNImeVGW
  sdcUmht9zO1XOoTd5EZodFnr5N0RNaafrhOEsyA"}},
              "wmk":"wqcv5AGkFpMePLRoszuEG6uZ8lj--UOyF-uC-8c21MSN
  vrcvlksccQ"}
            ]},
        "kEXzFK9QHjuDjo0dxsG3xhpULXCder_shCN5rWeymF805vj748CXhkLb
  gd7VQu1_dfFAQcsmRM-VmeNLtqOpPbGS0KabYIbk91B_OYqYaAnPkc-dgMXsR33gI
  16w2bMmpPovGK0B0b1xTNj6WpbPZgn5aDrq5T6evtGPZ_q5oSei12oBpDE5T-oe39
  hallNxVAvN9FnepvHw20x9A844lCLXu9Z2U7XfFtqh7XNKFT-cz7pYMag8usEOu6V
  7TMJBWq1-sxoumN_cE_oFyZnMdEifaVvYEPRU29BERBzkRC8hoP1WlUC965ffRuqo
  veZzXp-4WrlXbJ_s9EI5DUxnXpjrfmoEOBfv6fVXvyjEstc3v1XbXnyYrHJi0YOF5
  wB0DS_1ydIR0x4Y85ZNLh9JfESVuQxMHeYpwnb6jj-RaEFoUMJOuqjnt4Bpg5zknO
  mcYTKme46JGyBpj_07KWP5WElVMMZRpdXv3AK2tpWOuUzikQ3UU3HWHUqqoVoumsm
  9LChaVIUOK2aGqrMXRbE1fXqffDcFn5EcOqDBjbO9DJ4R5CjdQvBqzCZI39ZBizPZ
  j2yAYPr2ewSfhBkDR1FG418gH2JlQfjjhtvfyGcq-opuWh3pOVaMx55pIaZu8yUH-
  1myKKy9hn8EJPMR7X4YflB6motQP-Uw8oVYn6NQN8_05HDlux8n_qgyaXu7dYH3gB
  6X2WMEmrqv48QpmrW_ftSVXTr0g8_1txHuY-xL8J_6tInEAs0MMiQZ14HvdD4Hnpa
  _2tGDfxLOx-OXItgrGgF_kUEPatIZX6px2A8SMhl8ZCpr5EppAuGzgHFfS8wNA8iL
  PT5Q__lQqtIrslS05jrqGuWTJlus7EkUJXxRapnZxasnvdGxCRoY3E-hLOb-Kn86I
  ayfkuG4lnmDEv7a-x5NIPuK_DmgL2DfgyhrxuJibvfUhjG9D3iWIhkLZMJlFHe1lK
  SqUo-9UZBqx2fNjZyXryPE6CKkXnICfTNax5qOd609SGh8E7YwKsczJcT4lJhIikf
  LVIENMu3Kco_PXxYg1XJvy4DToZ_i53S0rMdYFYjGuE7PH9eX9ah_pIPvD32hllAA
dOgcu8Ib2FCi7BxOS6FuRhr8FBZh-ZSsFuYzjIcBvt0Kg6zeRMridocp723Ni5d4E
nZFXbFibISMPI2edm1V5kR-Deo5zlZzqfkvFglHPR9dnFVkAGUwHuFvs84esOeVvn
8B8fnO5svk-9iBQnN9OspsUoJLVAh4S-_EGSxRePfTv2qIVC3I8pVQa2D4_UP9C2p
lz6rEkEhz2ThnUhKb6Mc6zl1kpjP00zDdTjRYpOVVdbzlCtsbBq8KM2Hy0h0RGC63
GuSWJw1BUV3DEL-WBiE5bsRX2BA9E5-MdtSDlM3Nwps5a6jIhEirEWhSP1wGHes8q
B7mBijuNzN7oTK0InK4Nbx8YmCQzQXAYbgGsUsZ1FY75VCRNqp8YIx7C4IXPloElT
fH7uXoU0SNBXbvLr_xntPFSiDKKf-A10nq7LvNSfdEGArLKX6jQLileNonHY3QjD2
zAdvWeQW5j_iabXUV2DMeEC9DaSuWYcVNpK16PZKrND0IS48ojZlHczk_VXbv5NR8
CBFqpkPSDCRkFLSWSP8mGsLiufihLr6Kc5wdPnckuYDCNQSagHR2UwwHnPqcwpFfF
ZkZf3uygVyuZymG_fbgThNI65oK3lm7HalCT2G_WrgZfeT3_eJBwB95QtQn-atbZS
Hsk81F2hRLyVSM_X9bqTcyWksFKEnGERNIDpw1V6Ge41U3YTsde8NzDdQru6gYlIZ
kqPfrV1zLgdIPe_kVd7QW-MMr2V2q66XCO115ivOiuCrNz2xytKsN1bXZonQ6YVZ7
B_dqwXZdzd-1Tun0iVfRyyIWOq436pNvyJI5xovAbuB1uwgvfNbC7xzkeUfg21dOz
f_e9hRHlMbbfv3BAMD6T4HtMkd20iVm4ehsImRrjPTOrnzaXP8wmT9IB4Usap4PpD
2K3yTnLT5VxEL8hq_dKrqHZGbHp3kNcqJ6DPoObQ7veISoZY2CWTPuFQ5d89Evxx8
4Gej07yMKKlOCvf4bwchDYSSjyj5rU2ICB9omSAlhg6E7-0DWOk32KK-ESjK4NNpg
mI4YlkHArct5PshPKuCDeuR8-Aa7lrqLNIS2w2ZdLsn0c9dk4Wnf8hxg0BKbkqF8F
Wmv_uOvVaYrrJJzYdNhMYaGcKtSL7qPG3LKQKEAu_NShgOrkCFR5x3JYP7SnAuT9X
homtf28tOnil8B1oHevWhWgq-HXeIkUEvTAPl3Nua0xNwXfLlk8UHEGtCaXD4ClIR
bA1od6QcbxyDG81wXBZxyXuh1wdGT6nacjXXFrRaAjAKKAkgmZ7LrNic5C6F7zsgX
2L5BRTtt6DNxHUZJhiUUpVNbc3ZGzNPh8WokgQpVsTlliYJl2d3dM9oWcOsVDGQ-n
DHe3Hwb-SX2_oEi_mzp21XgX7_QGl83_CTAnaQQKtVQTV5v3Vdul4gDvwmmuM3spl
XrsavUdZb2-O7AwW7739aF-Z1SAupVKFtblVWgJ_DW36gF-peeJNz0uUhn1h_9uFq
R7u1A1diUu2tW0k4SRRdpw2FDx5QfEgnlf3PYvq8DM1de5yiWb5gPx7sio3XWbQFA
4FRBcgZCP9gT_BXqowSF50MH5clcjVDp2DjU-_7iFKiL9gUtzt69pUtBWLt2uy-VI
1u1kZAwklbtUKB61esVLbcv7EwQBHAr1AwTFar5K7i2doBNxGneZvYNm153MRbhTM
6S-y_ZvaA19uAEjtD4Lkqz3ufaRBcylbr0aJgJxdwaXP9PjMl8S7e8i79omhWo2Cu
vTPjH0dBpBlSZlo30yPQXKZzTalYZeyDVD3oiNI4C1cTK08goCFXsSKocsBuydqZE
f3MR_uRBiy7wn7CMs4q6_vzxbjSEUndFwfSJZRo6yuMYiQb69ijMoMMv8U2pzBS7q
Bu-5hvcKfaiZqTYgmjCwDFe8w1jpwuZ76Vw8j1XxbWpfs8NCTMDxu6WCA0Mk7itZJ
EQn26pXsmsPyKNzxofpBoU2pnu8WE00pmXym8xWj3tM_XCX54x8s0QFiAhrPYeHjk
wfJ3wwu8hWVwC2-fAG85bHuZH08Qgry7-lzbQj4ChrjDzgodJmXmhWTEYQfkBGm-X
c7hkjCw_9OWzRJDKKUcU4wbHLp0DQfD6nU_fanV-1okO1Q"
      ]
    ],
  "LocalName":"ssh"}}
```

4.3. Bookmark

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

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

```
{
  "CatalogedBookmark":{
    "Uri":"http://www.example.com",
    "Title":"site1",
    "LocalName":"Sites-1",
    "Uid":"ND3H-L37J-BWNZ-NERL-XWFR-ZC46-CW53"}}
```

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":"MB3T-WIPZ-JRCW-QZFM-SCQL-OVVO-AHO2",
    "Self":true,
    "Contact":{
      "ContactPerson":{
        "Id":"MB3T-WIPZ-JRCW-QZFM-SCQL-OVVO-AHO2",
        "Anchors":[{
            "Udf":"MB3T-WIPZ-JRCW-QZFM-SCQL-OVVO-AHO2",
            "Validation":"Self"}
          ],
        "NetworkAddresses":[{
            "Address":"alice@example.com",
            "EnvelopedProfileAccount":[{
                "EnvelopeId":"MB3T-WIPZ-JRCW-QZFM-SCQL-OVVO-AHO2",
                "dig":"S512",
                "ContentMetaData":"ewogICJVbmlxdWVJZCI6ICJNQjNULV
  dJUFotSlJDVy1RWkZNLVNDUUwtT1ZWTy1BSE8yIiwKICAiTWVzc2FnZVR5cGUiOiA
  iUHJvZmlsZVVzZXIiLAogICJjdHkiOiAiYXBwbGljYXRpb24vbW1tL29iamVjdCIs
  CiAgIkNyZWF0ZWQiOiAiMjAyMy0wNi0yOFQxNzowMDoxNVoifQ"},
              "ewogICJQcm9maWxlVXNlciI6IHsKICAgICJDb21tb25TaWduYX
  R1cmUiOiB7CiAgICAgICJVZGYiOiAiTURFMi1NS01JLTc3M1AtR0ozRi1ZWUFJLVV
  WQ0stT01LUyIsCiAgICAgICJQdWJsaWNQYXJhbWV0ZXJzIjogewogICAgICAgICJQ
  dWJsaWNLZXlFQ0RIIjogewogICAgICAgICAgImNydiI6ICJFZDQ0OCIsCiAgICAgI
  CAgICAiUHVibGljIjogImpSOXVyUGJvc2xvU1J1NThsa0tHOU80TDVCTnpxYkVzM2
  9xOEl6d0xxVTJReUdSazhrV0QKICBvazZPT3doWU9jUmdYZW90X2VWT0FHbUEifX1
  9LAogICAgIkFjY291bnRBZGRyZXNzIjogImFsaWNlQGV4YW1wbGUuY29tIiwKICAg
  ICJTZXJ2aWNlVWRmIjogIk1EM0UtRk42Vy0zRzQ1LVlRNDMtUVhZUi1DVTRYLVJLR
  zUiLAogICAgIkVzY3Jvd0VuY3J5cHRpb24iOiB7CiAgICAgICJVZGYiOiAiTUJTSy
  0yWTZHLURLNlAtVDZUVS1PU0xELUdDRlctTVBIRyIsCiAgICAgICJQdWJsaWNQYXJ
  hbWV0ZXJzIjogewogICAgICAgICJQdWJsaWNLZXlFQ0RIIjogewogICAgICAgICAg
  ImNydiI6ICJYNDQ4IiwKICAgICAgICAgICJQdWJsaWMiOiAiUjVXMFRZVXljRU1oR
  3V4R3VDa0JUTVlCMU1nS1owMzZ5MDUyWExWYk1zanhnZzlpREQteQogIFZZVk5lX3
  lVQ204UUd0cFN0XzhFYjNjQSJ9fX0sCiAgICAiQWRtaW5pc3RyYXRvclNpZ25hdHV
  yZSI6IHsKICAgICAgIlVkZiI6ICJNQkw1LUpTTjMtVjU2US00VUxZLUdZN1gtR00z
  Vi1LVlBaIiwKICAgICAgIlB1YmxpY1BhcmFtZXRlcnMiOiB7CiAgICAgICAgIlB1Y
  mxpY0tleUVDREgiOiB7CiAgICAgICAgICAiY3J2IjogIkVkNDQ4IiwKICAgICAgIC
  AgICJQdWJsaWMiOiAiWmM1bjlxNDgyRTVSdUhzSjRlZFdzcjc1YXh3elIzbVdibTV
  UNWxmbkJUUklBaWNhR1BvZwogIGJxYTU0eVNBN3NXamg0OTB4cnZyRXlhQSJ9fX0s
  CiAgICAiQ29tbW9uRW5jcnlwdGlvbiI6IHsKICAgICAgIlVkZiI6ICJNQlVGLVA3U
  zItV0ZFRi1EM01MLU9LQ0MtWFlPVC02U0xEIiwKICAgICAgIlB1YmxpY1BhcmFtZX
  RlcnMiOiB7CiAgICAgICAgIlB1YmxpY0tleUVDREgiOiB7CiAgICAgICAgICAiY3J
  2IjogIlg0NDgiLAogICAgICAgICAgIlB1YmxpYyI6ICI4U2hGY21OejVCVzluR09i
  dTdfdFVaRDVobTVhblM3VGFyNTNSWERjZzVheWlQcGI3TDh6CiAgVkMxbGpqSmVBd
  S1oazlUVU51eVhFN3NBIn19fSwKICAgICJDb21tb25BdXRoZW50aWNhdGlvbiI6IH
  sKICAgICAgIlVkZiI6ICJNRFlRLUpRQjItM0VPQy1ONFpELUZCSkUtSDNJWS1XTTZ
  WIiwKICAgICAgIlB1YmxpY1BhcmFtZXRlcnMiOiB7CiAgICAgICAgIlB1YmxpY0tl
  eUVDREgiOiB7CiAgICAgICAgICAiY3J2IjogIlg0NDgiLAogICAgICAgICAgIlB1Y
  mxpYyI6ICIyelBsMTBxbnpwR0hfMWlkREZhVk15RXltRWY2Wm1oTXR6cG1sR2ZaSF
```

```
oyc2lGQzBTSEk0CiAgd2FtZ2hzWVMzaEZMX3FYNm1PLVNRUXVBIn19fSwKICAgICJ
Qcm9maWxlU2lnbmF0dXJlIjogewogICAgICAiVWRmIjogIk1CM1QtV0lQWi1KUkNX
LVFaRk0tU0NRTC1PVlZPLUFITzIiLAogICAgICAiUHVibGljUGFyYW1ldGVycyI6I
HsKICAgICAgICAiUHVibGljS2V5RUNESCI6IHsKICAgICAgICAgICJjcnYiOiAiRW
Q0NDgiLAogICAgICAgICAgIlB1YmxpYyI6ICJvcjJYLXJQMHRhVFg2NkZ4WTY4UlI
0UjdHZmczcjYtTUljMzNRZUVnVTF3S2lfSFZLeWlCCiAgbktEWFpFZXhZdEVDMlpI
N0NOcVVDMlFBIn19fX19",
            {
              "signatures":[{
                  "alg":"S512",
                  "kid":"MB3T-WIPZ-JRCW-QZFM-SCQL-OVVO-AHO2",
                  "signature":"lhicUvvDwdI2cJGDmMDmEYhZIDOp0be5
IjblZGn0Uycnu3odE_h5jGOY3W58RlXBr_NHUwHfAbGAHcigqzKxUJGrM9MKXzgYF
5JUx7uHSN4qXpAcBPHHnU1qLepITOsRMoT92a3KmLGskrt9O2PlgBQA"}
                ],
              "PayloadDigest":"hg5Z9SBDuRlEje8R3-0KZFBNHW738w1k
0ZvF-nNJKZ4acEZKjmAwOzx3cbf6HXJoWy3eLs4BqYdhXQWkftir1g"}
            ],
          "Protocols":[{
              "Protocol":"mmm"}
            ]}
        ],
      "Sources":[{
          "Validation":"Self",
          "EnvelopedSource":[{
              "dig":"S512",
              "ContentMetaData":"ewogICJNZXNzYWdlVHlwZSI6ICJDb2
50YWN0UGVyc29uIiwKICAiY3R5IjogImFwcGxpY2F0aW9uL21tbS9vYmplY3QiLAo
gICJDcmVhdGVkIjogIjIwMjMtMDYtMjhUMTc6MDA6MTVaIn0"},
            "ewogICJDb250YWN0UGVyc29uIjogewogICAgIkFuY2hvcnMiOi
BbewogICAgICAgICJVZGYiOiAiTUIzVC1XSVBaLUpSQ1ctUVpGTS1TQ1FMLU9WVk8
tQUhPMiIsCiAgICAgICAgIlZhbGlkYXRpb24iOiAiU2VsZiJ9XSwKICAgICJOZXR3
b3JrQWRkcmVzc2VzIjogW3sKICAgICAgICAiQWRkcmVzcyI6ICJhbGljZUBleGFtc
GxlLmNvbSIsCiAgICAgICAgIkVudmVsb3BlZFByb2ZpbGVBY2NvdW50IjogW3sKIC
AgICAgICAgICAgIkVudmVsb3BlSWQiOiAiTUIzVC1XSVBaLUpSQ1ctUVpGTS1TQ1F
MLU9WVk8tQUhPMiIsCiAgICAgICAgICAgICJkaWciOiAiUzUxMiIsCiAgICAgICAg
ICAgICJDb250ZW50TWV0YURhdGEiOiAiZXdvZ0lDSlZibWx4ZFdWSlpDSTZJQ0pOU
WpOVUxWZEpVRm90U2xKRFZ5MQogIFJXa1pOTFZORFVVd3RUMVpXVHkxQlNFOHlJaX
dLSUNBaVRXVnpjMkZuWlZSNWNHVWlPaUFpVUhKdlptbHNaCiAgVlZ6WlhJaUxBb2d
JQ0pqZEhraU9pQWlZWEJ3YkdsallUNBiMjR2YlcxdEwyOWlhbVZqZENJc0NpQWdJ
a04KICB5WldGMFpXUWlPaUFpTWpBeU15MHdOaTB5T0ZReE56b3dNRG94TlZvaWZRI
n0sCiAgICAgICAgICAiZXdvZ0lDSlFjbTltYVd4bFZYTmxjaUk2SUhzS0lDQWdJQ0
pEYjIxdGIyNQogIFRhV2R1WVhSMWNtVWlPaUI3Q2lBZ0lDQWdJQ0pWWkdZaU9pQWl
UVVJGTWkxTlMwMUpMVGMzTTFBdFIwb3pSCiAgaTFaV1VGSkxWVldRMHN0VDAxTFV5
SXNDaUFnSUNBZ0lDSlFkV0pzYVdOUVlYSmhiV1YwWlhKeklqb2dld28KICBnSUNBZ
0lDQWdJQ0pRZFdKc2FXTkxaWGxGUTBSSUlqb2dld29nSUNBZ0lDQWdJQ0FnSW1OeW
RpSTZJQ0pGWgogIERRME9DSXNDaUFnSUNBZ0lDQWdJQ0FpVUhWaWJHbGpJam9nSW1
wU09YVnlVR0p2YzJ4dlUxSjFOVGhzYTB0CiAgSE9VODBURFZDVG5weFlrVnpNMjl4
T0VsNmQweHhWVEpSZVVkU2F6aHJWMFFLSUNCdmF6WlBUM2RvV1U5alUKICBtZFlaV
```

```
zkwWDJWV1QwRkhiVUVpZlgxOUxBb2dJQ0FnSWtGalkyOTFiblJCWkdSeVpYTnpJam
9nSW1Gc2FXTgogIGxRR1Y0WVcxd2JHVXVZMjl0SWl3S0lDQWdJQ0pUWlhKMmFXTmx
WV1JtSWpvZ0lrMUVNMFV0Ums0MlZ5MHpSCiAgelExTFZsUk5ETXRVVmhaVWkxRFZU
UllMVkpMUnpVaUxBb2dJQ0FnSWtWelkzSnZkMFZ1WTNKNWNIUnBiMjQKICBpT2lCN
0NpQWdJQ0FnSUNKVlpHWWlPaUFpVFVKVFN5MHlXVFpITFVSTE5sQXRWRFpVVlMxUF
UweEVMVWREUgogIGxjdFRWQklSeUlzQ2lBZ0lDQWdJQ0pRZFdKc2FXTlFZWEpoYld
WMFpYSnpJam9nZXdvZ0lDQWdJQ0FnSUNKCiAgUWRXSnNhV05MWlhsRlEwUklJam9n
ZXdvZ0lDQWdJQ0FnSUNBZ0ltTnlkaUk2SUNKWU5EUTRJaXdLSUNBZ0kKICBDQWdJQ
0FnSUNKUWRXSnNhV01pT2lBaVVqVlhNRlJaVlhsalJVMW9SM1Y0UjNWRGEwSlVUVm
xDTVUxblMxbwogIHdNelo1TURVeVdFeFdZazF6YW5oblp6bHBSRVF0ZVFvZ0lGWlp
WazVsWDNsVlEyMDRVVWQwY0ZOMFh6aEZZCiAgak5qUVNKOWZYMHNDaUFnSUNBaVFX
UnRhVzVwYzNSeVlYUnZjbE5wWjI1aGRIVnlaU0k2SUhzS0lDQWdJQ0EKICBnSWxWa
1ppSTZJQ0pOUWt3MUxVVcFRUak10VmpVMlVTMDBWVXhhTFVkWk4xZ3RSMDB6VmkxTF
ZsQmFJaXdLSQogIENBZ0lDQWdJbEIxWW14cFkxQmhjbUZ0WlhSbGNuTWlPaUI3Q2l
BZ0lDQWdJQ0FnSWxCMVlteHBZMHRsZVVWCiAgRFJFZ2lPaUI3Q2lBZ0lDQWdJQ0Fn
SUNBaVkzSjJJam9nSWtWa05EUTRJaXdLSUNBZ0lDQWdJQ0FnSUNKUWQKICBXSnNhV
01pT2lBaVdtTTFiamx4TkRneVJUVlNkVWh6U2pSbFpGZHpjamMxWVhoM2VsSXpiVm
RpYlRWU5XeAogIG1ia0pVVWtsQmFXTmhSMUJ2WndvZ0lHSnhZVFUwZVZZOQk4zTlh
hbWcwT1RCNGNuWnlSWGxoUVNKOWZYMHNDaUFnaUFnSUNBaVEyOXRiVzl1Ulc1amNu
bHdkR2x2YmlJNklIc0tJQ0FnSUNBZ0lsVmtaaUk2SUNKTlFsVkdMVkEKICAzVXpJd
FYwWkZSaTFFTTAxTUxVOUxRME10V0ZsUFZDMDJVMHhFSWl3S0lDQWdJQ0FnSWxCMV
lteHBZMUJoYwogIG1GdFpYUmxjbk1pT2lCN0NpQWdJQ0FnSUNBZ0lsQjFZbXhwWTB
0bGVVVkRSRWdpT2lCN0NpQWdJQ0FnSUNBCiAgZ0lDQWlZM0oySWpvZ0lsZzBORGdp
TEFvZ0lDQWdJQ0FnSUNBZ0lsQjFZbXhwWXlJNklDSTRVMmhHWTIxT2UKICBqVkNWe
mx1UjA5aWRUZGZkRlZhUkRWb2JUVmhibE0zVkdGeU5UTlNXRVJqWnpWaGVXbFFFjR0
kzVERoNkNpQQogIGdWa014YkdwcVNtVkJkUzFvYXpsVVZZVnNTFlVmhGTjNOQkluMTl
mU3dLSUNBZ0lDSkRiMjF0YjI1RmJtTnllSEZpaUk2SUhzS0lDQWdJQ0FnSWxWa1pp
SUNBZ0lDQWdJbEIxWW14cFkxQmhjbUZ0WlhSbGNuTWlPaUI3Q2lBZ0lDQWdJQ0FnS
mU3dLSUNBZ0lDQWdJbEIxWW14cFkxQmhjbUZ0WlhSbGNuTWlPaUI3Q2lBZ0lDQWdJ
1NrVXRTRE5KV1MxWFRUWldJaXdLSUNBZ0lDQWdJbEIxWW14cFkxQmhjbUZ0WlhSbG
NuTWlPaUI3QwogIGlBZ0lDQWdJQ0FnSWxCMVlteHBZMHRsZVVWRFJFZ2lPaUI3Q2l
BZ0lDQWdJQ0FnSUNBaVkzSjJJam9nSWxnCiAgME5EZ2lMQW9nSUNBZ0lDQWdJQ0Fn
SWxCMVlteHBZeUk2SUNJeWVsQnNNVEJ4Ym5wd1IwaGZNV2xrUkVaaFYKICBrMTVSW
Gx0UldZMW9UWFI2Y0cxc1IyWm1FTRm95YzJsR1F6QlRTRWswQ2lBZ2QyRnRaMm
h6V1ZNemFFWgogIE1YM0ZZTm0xUExWTlJVWFZCSW4xOWZTd0tJQ0FnSUNKUWNtOW1
hV3hsVTJsbmJtJtRjBkWEpsSWpvZ2V3b2dJCiAgQ0FnSUNBaVZZXUm1Jam9nSWsxQ00x
UXRWMGxRRV2kxS1VrTlhMVkZhUmswdFUwTlJUQzFQVmxaUExVRkVVekKICBpTEFvZ
0lDQWdJQ0FpVUhWaWJHbGpVR0Z5WVcxbGRHVnljeUk2SUhzS0lDQWdJQ0FnSUNBaV
VIVmliR2xqUwogIDJWNVJVTkVTQ0k2SUhzS0lDQWdJQ0FnSUNBZ0lDSmpjbllpT2l
BaVJXUTBORGdpTEFvZ0lDQWdJQ0FnSUNBZ0lsQjFZbXhwWXlJNklDSnZjakpaZ
TFhKUU1IUmhWRmcyTmtaNFdUWTRVbEkwVVpkSFptY3pjall0VFVsak0KICB6TlJaV
VZuVlRGM1MybGZTRlpMQ0NpQWdia3RFV0ZwWlpFaTFaRVZETWxxwSU4wTk9jVl
ZETWxGQkluMQogIDlmWDE5IiwKICAgICAgICAgIHsKICAgICAgICAgICAgInNpZ25
hdHVyZXMiOiBbewogICAgICAgICAgICAgImFsZyI6ICJTNTEyIiwKICAgICAg
ICAgICAgICAgICJraWQiOiAiTUIzVC1XSVBaLUpSQ1ctUVpGTS1TQ1FMLU9WVk8tQ
UhPMiIsCiAgICAgICAgICAgICAgICAic2lnbmF0dXJlIjogImxoaWNVdnZEd2RJMm
NKR0RtTURtRVloWklET3AwYmU1SWpibFpHbjBVeWNudTNvZEUKICBfaDVqR09ZM1c
1OFJsWEJyX05IVXdIZkFiR0FIY2lncXpLeFVKR3JNOU1LWHpnWUY1SlV4N3VIU040
cVhwQQogIGNCUEhIblUxcUxlcElUT3NSTW9UOTJhM0ttTEdza3J0OU8yUGxnQlFBI
n1dLAogICAgICAgICAgICAiUGF5bG9hZERpZ2VzdCI6ICJoZzVaOVNCRHVSbEVqZT
```

```
hSMy0wS1pGQk5IVzczOHcxazBadkYtbk5KS1o0YWMKICBFWktqbUF3T3p4M2NiZjZ
IWEpvV3kzZUxzNEJxWWRoWFFXa2Z0aXIxZyJ9XSwKICAgICAgICAiUHJvdG9jb2xz
IjogW3sKICAgICAgICAgICAgIlByb3RvY29sIjogIm1tbSJ9XX1dfX0",
            {
              "signatures":[{
                  "alg":"S512",
                  "kid":"MDE2-MKMI-773P-GJ3F-YYAI-UVCK-OMKS",
                  "signature":"rX3vdDg4XWLs91hy8okzai4Rw6nwBSGR
YxxgrCIM4lJmKRGM8MvFLm7x17qsZE-rI0pDuoyXu1AAiVcUtUpqYHFDHb7Rg7ApT
PDsgomlhVMT3UOaUPXJ1dCUAN3LMySBnMzOZ5wBvKHV35gb_-nu_iAA"}
                ],
              "PayloadDigest":"PJFh1hJR7p0Wt1dJFU1mOca6ZO51jMlz
abR5BiHDHvpLZpUWvqApCOr6U9A-qFO8qyiLPru6YRWgaKjEgyA3Jg"}
            ]}
        ]}}}}
```

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

4.5. Credential

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

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

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

4.6. Device

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

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

4.7. Network

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

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

4.8. Publication

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

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

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

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

Contact Used in the Static QR Code Contact Exchange interaction.

Profile Device Used in the Preconfigured Device Connection interaction.

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

>>>> Unfinished SchemaEntryPublication

Missing example 13

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":"NB57-PDLZ-LSIV-DJVF-OQGD-APGT-OZMO"}}
```

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
    = NCO5-AV7A-DZYY-A5JD-GGWI-KH3Y-OJND

EnvelopeID
    = MBDW-KFHR-OR66-U6CR-CCEV-N4DD-MXXQ

ResponseID
    = MACQ-IGXA-ZT5G-4GCI-CR4X-R5CS-Y4TY
```

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 AAKI-IIAD-GQ3H-JUY3-SXZN-PENW-PQ used to authenticate a an action of type Device:

```
pin = AAKI-IIAD-GQ3H-JUY3-SXZN-PENW-PQ
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)
    = AA6H-GDQF-3QDF-B7MB-GBAO-RJ4O-ZTPI
```

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

```
Account =  alice@example.com

PinId = UDFPresent (MAC (Account, saltedPINData))
    = ACVB-JSGA-EUN7-QIGS-XYWQ-G3OU-77IZ
```

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

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

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

7.4. EARL

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

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

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

8. Mesh Assertions

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

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

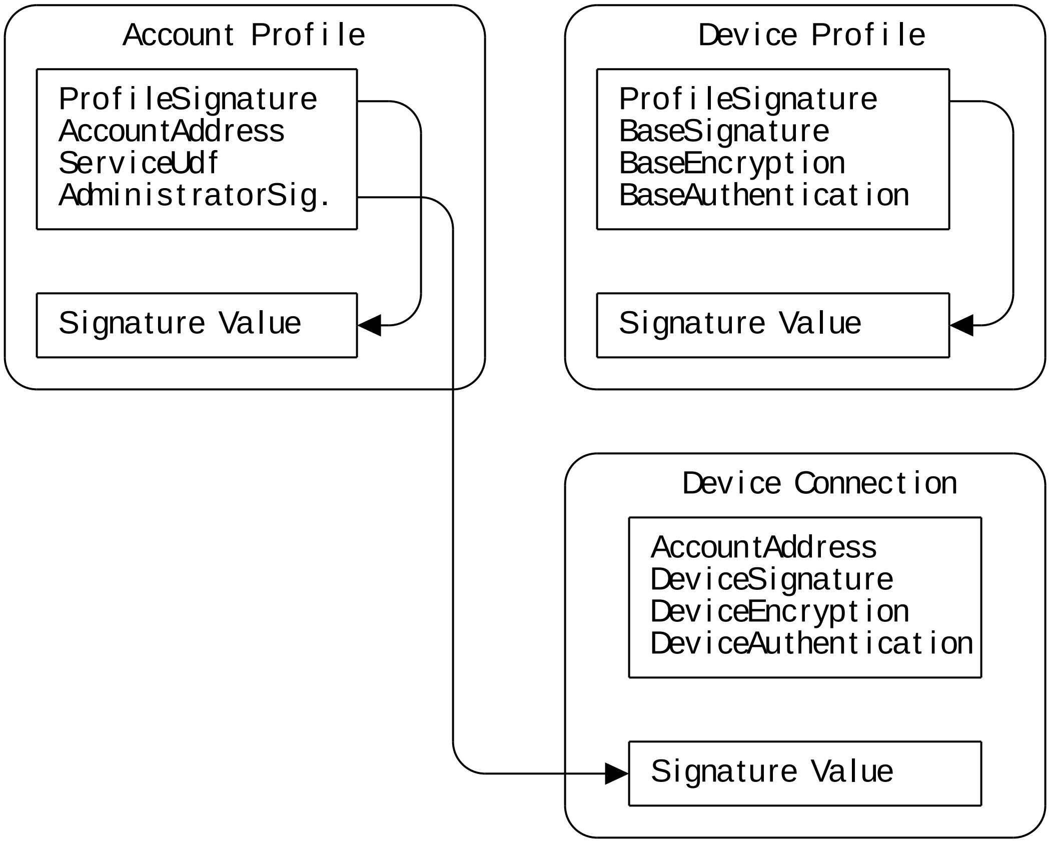

Figure 1: Profiles And Connections

8.1. Encoding

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

Identifier An identifier for the assertion.

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

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

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

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

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

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

8.2. Mesh Profiles

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

- Profiles describe credentials, they do not make identity statements

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

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

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

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

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

A Profile is valid if and only if:

*There is a SignatureKey field.

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

A profile has the status current if and only if:

*The Profile is valid

*Every Conditions clause in the profile is understood by the relying party and evaluates to true.

8.3. Mesh Connections

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

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

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

*The profile *P* is valid

*The AuthorityUdf field of the connection is consistent with the UDF of *P*

*The profile is signed under the key specified in the AdministrationKey field of *P*.

*Any conditions specified in the profile are met

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

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

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

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

8.4. Device Pre-configuration

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

- The Enveloped ProfileDevice.
- A ConnectionDevice assertion credentialing the device to the configuration provider Mesh Service.
- A ConnectionService assertion credentialing the device to the configuration provider Mesh Service.
- The secret seed used to create the ProfileDevice data.

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

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

```
{
 "DevicePreconfigurationPrivate":{
  "EnvelopedConnectionDevice":[{
    "dig":"S512",
    "ContentMetaData":"ewogICJNZXNzYWdlVHlwZSI6ICJDb25uZWN0aW
 9uRGV2aWNlIiwKICAiY3R5IjogImFwcGxpY2F0aW9uL21tbS9vYmplY3QiLAogICJ
 DcmVhdGVkIjogIjIwMjMtMDYtMjhUMTc6MDA6NTBaIn0"},
   "ewogICJDb25uZWN0aW9uRGV2aWNlIjogewogICAgIlNpZ25hdHVyZSI6IH
 sKICAgICAgIlVkZiI6ICJNQ1JQLTdPUVAtTFpFRy1LREhULVdUTlEtSlM0QS1QR0Z
 SIiwKICAgICAgIlB1YmxpY1BhcmFtZXRlcnMiOiB7CiAgICAgICAgIlB1YmxpY0tl
 eUVDREgiOiB7CiAgICAgICAgICAiY3J2IjogIkVkNDQ4IiwKICAgICAgICAgICJQd
 WJsaWMiOiAiVjFQdS1FQTE5Z1ZUOG5ibHMyeWgweUNmdUdENVRvaUhHeWY4czBsdj
 BBSVdocUdENzhvVAogIHBGRTIzUk5TRVdWczQtWFgtbnB3ekRFQSJ9fX0sCiAgICA
 iRW5jcnlwdGlvbiI6IHsKICAgICAgIlVkZiI6ICJNREY0LUNNQ1EtQktRWC0yUDZQ
 LVAySzItMjdVRC03QUdSIiwKICAgICAgIlB1YmxpY1BhcmFtZXRlcnMiOiB7CiAgI
 CAgICAgIlB1YmxpY0tleUVDREgiOiB7CiAgICAgICAgICAiY3J2IjogIlg0NDgiLA
 ogICAgICAgICAgIlB1YmxpYyI6ICJGOEI0ZW9YSXZ6X0txSGplckZybkRqbkdjR3J
 jVDlYOTM0MnQ2WEpNeS1VY1FXRUt6clVNCiAgRGE5STFTUUljV0xESGIyQUJtTUZK
 ZmFBIn19fSwKICAgICJBdXRoZW50aWNhdGlvbiI6IHsKICAgICAgIlVkZiI6ICJNR
 EY0LUNNQ1EtQktRWC0yUDZQLVAySzItMjdVRC03QUdSIiwKICAgICAgIlB1YmxpY1
 BhcmFtZXRlcnMiOiB7CiAgICAgICAgIlB1YmxpY0tleUVDREgiOiB7CiAgICAgICA
 gICAiY3J2IjogIlg0NDgiLAogICAgICAgICAgIlB1YmxpYyI6ICJGOEI0ZW9YSXZ6
 X0txSGplckZybkRqbkdjR3JjVDlYOTM0MnQ2WEpNeS1VY1FXRUt6clVNCiAgRGE5S
 TFTUUljV0xESGIyQUJtTUZKZmFBIn19fX19",
   {
    "signatures":[{
      "alg":"S512",
      "kid":"MAN2-ANEE-HJR4-I4T2-PKID-AZ4K-ESR4",
      "signature":"CUZYibn1PlgpCZuoqiKFZT1AKXDFe3EEmuSaTKoo
 TRH86oVdEeYNcVgzzn7sjFvO0TqDUvGK6saAcRsKDjTharVnLx3TnVHRmofUN5iqK
 yu9RwwC16bdvsOV4W7j7OhHrjC41Drp-MPozT3bHfaG7j0A"}
     ],
    "PayloadDigest":"VHfvwZ_1GC1Q40Q1-Qv2rZatLkVyEiGTjZi7-JMN
 dJ7QGF7My1VdsepgSoSc1Gslm8fvFz2NlKMPA-LCxlPovg"}
   ],
  "EnvelopedConnectionService":[{
    "dig":"S512",
    "ContentMetaData":"ewogICJNZXNzYWdlVHlwZSI6ICJDb25uZWN0aW
 9uU2VydmljZSIsCiAgImN0eSI6ICJhcHBsaWNhdGlvbi9tbW0vb2JqZWN0IiwKICA
 iQ3JlYXRlZCI6ICIyMDIzLTA2LTI4VDE3OjAwOjUwWiJ9"},
   "ewogICJDb25uZWN0aW9uU2VydmljZSI6IHsKICAgICJBdXRoZW50aWNhdG
 lvbiI6IHsKICAgICAgIlVkZiI6ICJNREY0LUNNQ1EtQktRWC0yUDZQLVAySzItMjd
 VRC03QUdSIiwKICAgICAgIlB1YmxpY1BhcmFtZXRlcnMiOiB7CiAgICAgICAgIlB1
 YmxpY0tleUVDREgiOiB7CiAgICAgICAgICAiY3J2IjogIlg0NDgiLAogICAgICAgI
 CAgIlB1YmxpYyI6ICJGOEI0ZW9YSXZ6X0txSGplckZybkRqbkdjR3JjVDlYOTM0Mn
 Q2WEpNeS1VY1FXRUt6clVNCiAgRGE5STFTUUljV0xESGIyQUJtTUZKZmFBIn19fX1
 9",
   {
    "signatures":[{
            "alg":"S512",
            "kid":"MAN2-ANEE-HJR4-I4T2-PKID-AZ4K-ESR4",
            "signature":"wYdQgPUrZcRuEvtZX55jZ-5aTnf5xN1TZ6pNHTD1
  0y3wlEXgKRP0KsMOLNRqZAo7eblK8NvsxVcAcQ9c8oWiqffZ6gDh1wH78XCreEeA_
  o62KkaYnN7rhcJ-4veoqc4Kz8du1xCRRjKDC0Wi7EqBfBsA"}
          ],
        "PayloadDigest":"YyVytW08cXC2UlAnQgNVkhi6_2ab5Gmy90WzFzBG
  -bDKpzRgbiK2vuaQpVlIRdZ5PYVeeO1QtGnu877s08E3Yg"}
      ],
    "PrivateKey":{
      "PrivateKeyUDF":{
        "PrivateValue":"ZAAQ-BIPI-QZV3-UXRZ-TRIO-S5XC-GEOJ-GELH-TXM
6-CBIN-QKF3-AWWC-HTOO-DJOG",
        "KeyType":"MeshProfileDevice"}},
    "ConnectUri":"mcu://maker@example.com/ECHI-CYLR-Y22Q-6OME-OAWV-
WIQD-YM",
    "EnvelopedProfileDevice":[{
        "EnvelopeId":"MBAI-IMKY-GI2T-D472-4VP5-SKRF-ZXYW",
        "dig":"S512",
        "ContentMetaData":"ewogICJVbmlxdWVJZCI6ICJNQkFJLUlNS1ktR0
  kyVC1ENDcyLTRWUDUtU0tSRi1aWFlXIiwKICAiTWVzc2FnZVR5cGUiOiAiUHJvZml
  sZURldmljZSIsCiAgImN0eSI6ICJhcHBsaWNhdGlvbi9tbW0vb2JqZWN0IiwKICAi
  Q3JlYXRlZCI6ICIyMDIzLTA2LTI4VDE3OjAwOjUwWiJ9"},
      "ewogICJQcm9maWxlRGV2aWNlIjogewogICAgIkVuY3J5cHRpb24iOiB7Ci
  AgICAgICJVZGYiOiAiTURGNC1DTUNRLUJLUVgtMlA2UC1QMksyLTI3VUQtN0FHUiI
  sCiAgICAgICJQdWJsaWNQYXJhbWV0ZXJzIjogewogICAgICAgICJQdWJsaWNLZXlF
  Q0RIIjogewogICAgICAgICAgImNydiI6ICJYNDQ4IiwKICAgICAgICAgICJQdWJsa
  WMiOiAiRjhCNGVvWEl2el9LcUhqZXJGcm5Eam5HY0dyY1Q5WDkzNDJ0NlhKTXktVW
  NRV0VLenJVTQogIERhOUkxU1FJY1dMREhiMkFCbU1GSmZhQSJ9fX0sCiAgICAiU2l
  nbmF0dXJlIjogewogICAgICAiVWRmIjogIk1DUlAtN09RUC1MWkVHLUtESFQtV1RO
  US1KUzRBLVBHRlIiLAogICAgICAiUHVibGljUGFyYW1ldGVycyI6IHsKICAgICAgI
  CAiUHVibGljS2V5RUNESCI6IHsKICAgICAgICAgICJjcnYiOiAiRWQ0NDgiLAogIC
  AgICAgICAgIlB1YmxpYyI6ICJWMVB1LUVBMTlnVlQ4bmJsczJ5aDB5Q2Z1R0Q1VG9
  pSEd5ZjhzMGx2MEFJV2hxR0Q3OG9UCiAgcEZFMjNSTlNFV1ZzNC1YWC1ucHd6REVB
  In19fSwKICAgICJBdXRoZW50aWNhdGlvbiI6IHsKICAgICAgIlVkZiI6ICJNQjNUL
  VlNTVYtV0FCVi1KM1lQLU9ZTEItWUFBSS1RRkQyIiwKICAgICAgIlB1YmxpY1Bhcm
  FtZXRlcnMiOiB7CiAgICAgICAgIlB1YmxpY0tleUVDREgiOiB7CiAgICAgICAgICA
  iY3J2IjogIlg0NDgiLAogICAgICAgICAgIlB1YmxpYyI6ICJBQ2ptd2NMUVVlOGk1
  UnVZUEZPSFhCTHlYRkN6RVcyb0lkUHh5Wk9mYlN2SEI0WUlEOUhqCiAgMHRsM0lad
  lJQdjlUc25Pc2hpZWRlMDJBIn19fSwKICAgICJQcm9maWxlU2lnbmF0dXJlIjogew
  ogICAgICAiVWRmIjogIk1CQUktSU1LWS1HSTJULUQ0NzItNFZQNS1TS1JGLVpYWVc
  iLAogICAgICAiUHVibGljUGFyYW1ldGVycyI6IHsKICAgICAgICAiUHVibGljS2V5
  RUNESCI6IHsKICAgICAgICAgICJjcnYiOiAiRWQ0NDgiLAogICAgICAgICAgIlB1Y
  mxpYyI6ICJkdTBFenhUdEZWa3BqWnphTi13LXBkRlFZMUpPWDRrQmlyX0pVUEdoQ3
  M3SzlELUpqTmV1CiAgNjdtMDcxR3AxSG9yZF9LeUJWSW8wWmtBIn19fX19",
      {
        "signatures":[{
            "alg":"S512",
            "kid":"MBAI-IMKY-GI2T-D472-4VP5-SKRF-ZXYW",
          "signature":"7ECNMccaM24RWFOoJJVU55zb1XfgLFisbc15VIVd
B_k6xeVGmpchZtgVrdeDjHgpesIqgJmbB7yAdPoIEusSu8pgY_Oc6BjMARJYAS9YW
bfE4UX2HBlWViyY7I2RUM0_VvcOgt6Fc3XMXCtvbsAG4gwA"}
        ],
      "PayloadDigest":"xalDY6zOXjHmqlXWGKty0QkSsPfDQ5tuDHY-Tm8R
e7UuxEfjvZnpTBtes9Etx86yn55D24AGSF199ejrRu8o5A"}
    ]}}
```

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

9. Architecture

The Mesh architecture has four principal components:

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

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

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

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

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

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

9.1. Mesh Account

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

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

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

9.1.1. Account Profile

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

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

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

9.1.1.1. Creating a ProfileMaster

Creating a ProfileMaster comprises the steps of:

0. Creating a Master Signature key.

1. Creating an Online Signing Key

2. Signing the ProfileMaster using the Master Signature Key

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

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

9.1.1.2. Updating a ProfileMaster

Updating a ProfileMaster comprises the steps of:

0. Making the necessary changes.

1. Signing the ProfileMaster using the Master Signature Key

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

9.2. Device Management

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

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

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

9.2.1. The Device Catalog

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

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

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

9.2.2. Mesh Devices

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

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

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

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

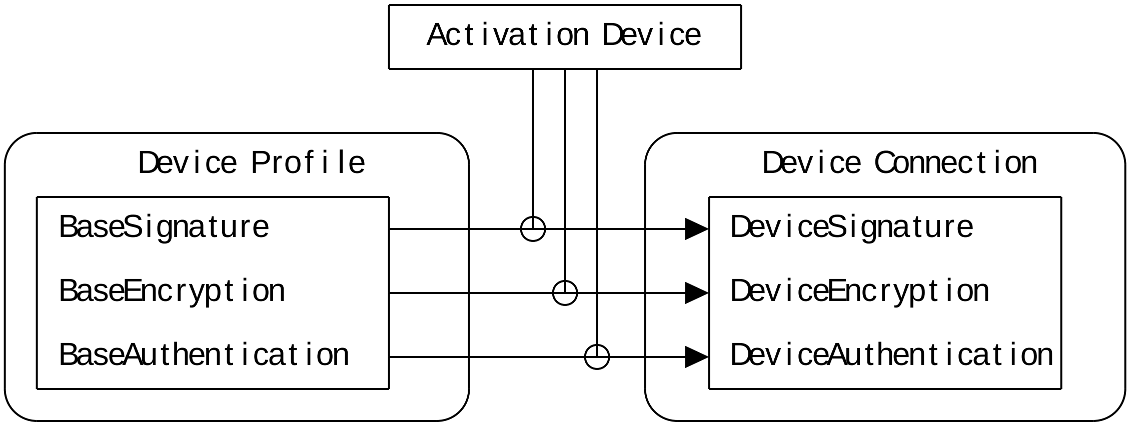

Figure 2: Mapping of Device Profile and Device Private to Device Connection Keys.

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

9.2.2.1. Creating a ProfileDevice

Creating a ProfileDevice comprises the steps of:

0. Creating the necessary key

1. Signing the ProfileDevice using the Master Signature Key

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

9.2.2.2. Connection to a Meh Account

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

0. Generating the PrivateDevice keys.

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

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

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

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

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

9.3. Mesh Services

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

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

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

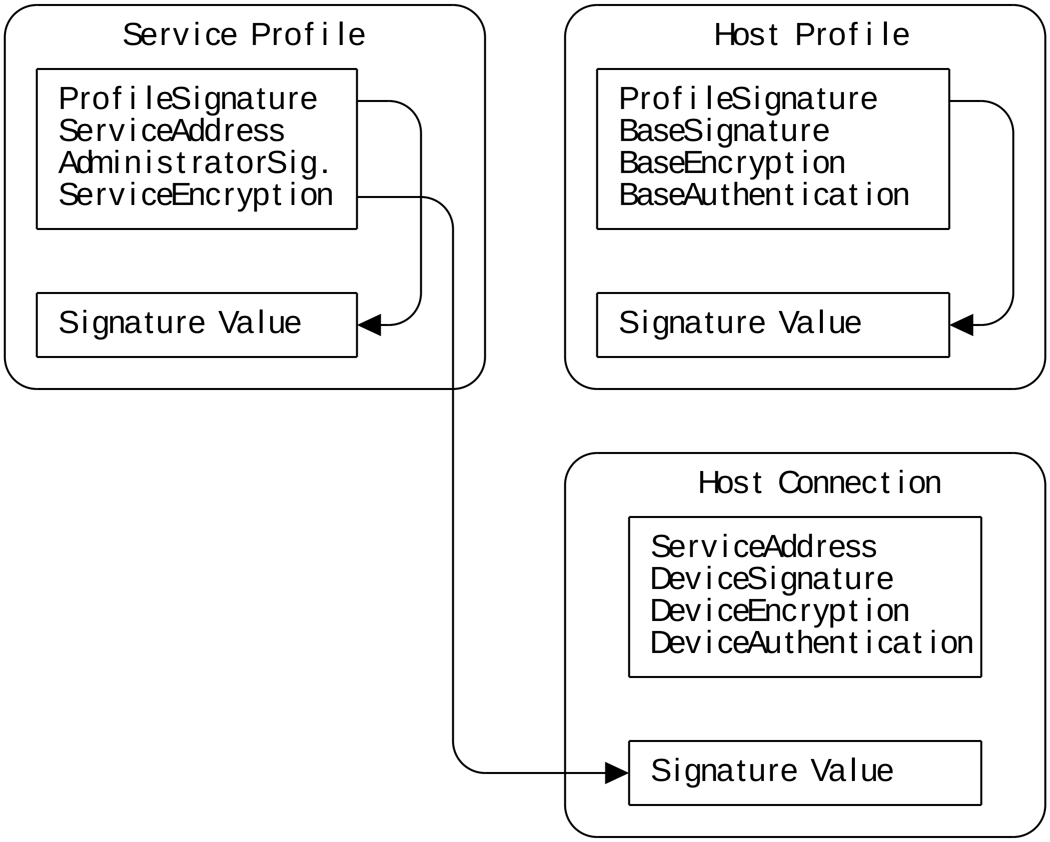

Figure 3: Service Profile and Delegated Host Assertion.

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

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

9.4. Mesh Messaging

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

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

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

9.4.1. Message Status

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

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

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

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

9.4.2. Four Corner Model

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

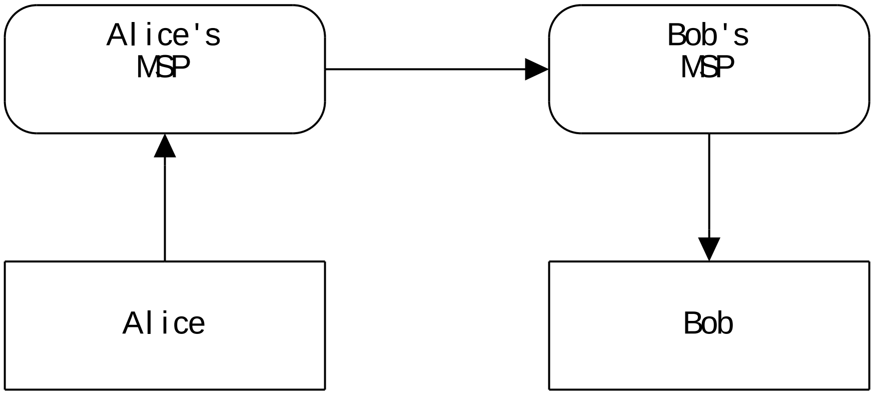

Figure 4: Four Corner Messaging Model

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

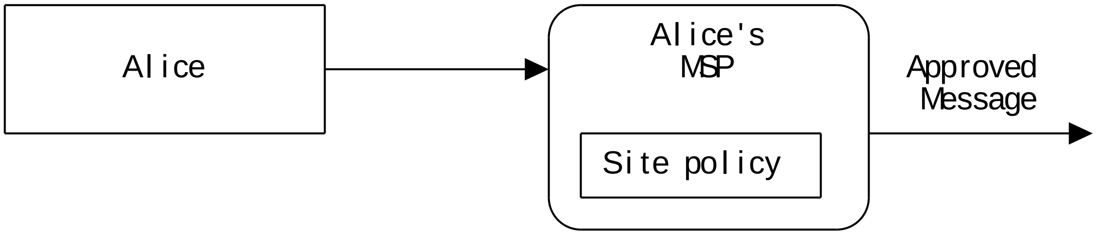

Figure 5: Performing Access Control on Outbound Messages

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

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

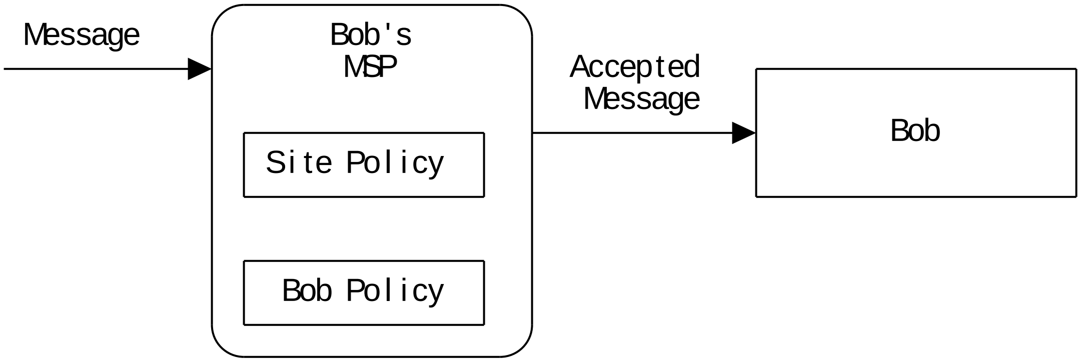

Figure 6: Performing Access Control on Inbound Messages

9.4.3. Traffic Analysis

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

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

10. Publications

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

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

10.1. Profile Device

10.2. Contact Exchange

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

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

11. Schema

11.1. Shared Classes

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

11.1.1. Classes describing keys

11.1.2. Structure: KeyData

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

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

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

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

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

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

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

11.1.3. Structure: KeyShare

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

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

11.1.4. Structure: CompositePrivate

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

11.2. Assertion classes

Classes that are derived from an assertion.

11.2.1. Structure: Assertion

Parent class from which all assertion classes are derived

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

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

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

11.2.2. Structure: Condition

Parent class from which all condition classes are derived.

[No fields]

11.2.3. Base Classes

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

11.2.4. Structure: Activation

Inherits: Assertion

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

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

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

11.2.5. Structure: ActivationEntry

Resource: String (Optional) Name of the activated resource

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

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

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

11.2.6. Mesh Profile Classes

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

11.2.7. Structure: Profile

Inherits: Assertion

Parent class from which all profile classes are derived

Description: String (Optional) Description of the profile

ProfileSignature: KeyData (Optional) The permanent signature key used to sign the profile itself. The UDF of the key is used as the permanent object identifier of the profile. Thus, by definition, the KeySignature value of a Profile does not change under any circumstance.

11.2.8. Structure: ProfileDevice

Inherits: Profile
Describes a mesh device.

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

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

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

11.2.9. Structure: ProfileAccount

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

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

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

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

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

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

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

11.2.10. Structure: ProfileUser

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

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

11.2.11. Structure: ProfileGroup

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

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

11.2.12. Structure: ProfileService

Inherits: Profile
Profile of a Mesh Service

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

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

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

11.2.13. Structure: ProfileMeshService

Inherits: ProfileService
Profile of a Mesh Service

[No fields]

11.2.14. Structure: ProfileHost

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

[No fields]

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: Connection

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

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

11.2.17. Structure: CallsignBinding

Inherits: Assertion The canonical form of the callsign.

Canonical: String (Optional)

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

CharacterPage: String (Optional) Specifies the page to which the Description"CharacterPageLatin"

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

TransferUdf: String (Optional) The profile to which the name has been transfered.

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

ServiceAddress: String (Optional) The Mesh service address.

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

11.2.18. Structure: Accreditation

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

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

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

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

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

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

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

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

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

11.2.19. Structure: ConnectionStripped

Asserts that a profile is connected to an account address.

Inherits: Connection
Stripped down connection assertion

Account: String (Optional) To be removed

11.2.20. Structure: ConnectionService

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

ProfileUdf: String (Optional) The account address

11.2.21. Structure: ConnectionDevice

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

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

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

11.2.22. Structure: ConnectionApplication

Inherits: Connection
Connection assertion stating that a particular device is

[No fields]

11.2.23. Structure: ConnectionGroup

Describes the connection of a member to a group.

Inherits: Connection

[No fields]

11.2.24. Structure: AccountHostAssignment

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

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

CallsignServiceProfile: ProfileAccount (Optional) Profile of the callsign registry used by the service.

11.2.25. Structure: ConnectionHost

Inherits: Connection
[No fields]

11.2.26. Activation Assertions

11.2.27. Structure: ActivationAccount

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

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

11.2.28. Structure: ActivationHost

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

Inherits: ActivationAccount
[No fields]

11.2.29. Structure: ActivationCommon

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

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

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

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

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

11.2.30. Structure: ActivationApplication

Inherits: Activation
[No fields]

11.2.31. Structure: ActivationApplicationSsh

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

11.2.32. Structure: ActivationApplicationMail

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

OpenpgpSign: KeyData (Optional) The OpenPGP signature key

OpenpgpEncrypt: KeyData (Optional) The OpenPGP encryption key

11.2.33. Structure: ActivationApplicationGroup

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

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

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

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

11.3. Application Data

11.3.1. Structure: ApplicationEntry

Identifier: String (Optional)

11.3.2. Structure: ApplicationEntrySsh

Inherits: ApplicationEntry

EnvelopedActivation: Enveloped (Optional)

11.3.3. Structure: ApplicationEntryGroup

Inherits: ApplicationEntry

EnvelopedActivation: Enveloped (Optional)

11.3.4. Structure: ApplicationEntryMail

Inherits: ApplicationEntry

EnvelopedActivation: Enveloped (Optional)

11.4. Data Structures

Classes describing data used in cataloged data.

11.4.1. Structure: Contact

Inherits: Assertion

Base class for contact entries.

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

Local: String (Optional) The local name.

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

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

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

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

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

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

11.4.2. Structure: Anchor

Trust anchor

Udf: String (Optional) The trust anchor.

Validation: String (Optional) The means of validation.

11.4.3. Structure: TaggedSource

Source from which contact information was obtained.

LocalName: String (Optional)

Short name for the contact information.

Validation: String (Optional) The means of validation.

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

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

11.4.4. Structure: ContactGroup

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

[No fields]

11.4.5. Structure: ContactPerson

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

11.4.6. Structure: ContactOrganization

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

11.4.7. Structure: OrganizationName

The name of an organization

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

RegisteredName: String (Optional) The registered name.

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

11.4.8. Structure: PersonName

The name of a natural person

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

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

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

First: String (Optional) First name.

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

Last: String (Optional) Last name.

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

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

11.4.9. Structure: NetworkAddress

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

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

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

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

EnvelopedProfileAccount: Enveloped (Optional) The account profile

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

11.4.10. Structure: NetworkProtocol

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

11.4.11. Structure: Role

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

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

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

11.4.12. Structure: Location

Appartment: String (Optional)
Street: String (Optional)

District: String (Optional)

Locality: String (Optional)

County: String (Optional)

11.4.13. Structure: Bookmark

Postcode: String (Optional)

Country: String (Optional)

Uri: String (Optional)

Title: String (Optional)

11.4.14. Structure: Reference

Role: String [0..Many]

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

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

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

11.4.15. Structure: Engagement

Key: String (Optional) Unique key.

Start: DateTime (Optional)

11.5. Catalog Entries

Finish: DateTime (Optional)

StartTravel: String (Optional)

11.5.1. Structure: CatalogedEntry

FinishTravel: String (Optional)

TimeZone: String (Optional)

Base class for cataloged Mesh data.

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

LocalName: String (Optional) User specified identifier.

Uid: String (Optional) Globaly unique identifier

11.5.2. Structure: CatalogedDevice

Inherits: CatalogedEntry

Public device entry, indexed under the device ID Hello

Updated: DateTime (Optional) Timestamp, allows

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

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

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

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

EnvelopedProfileDevice: Enveloped (Optional) The device profile

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

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

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

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

11.5.3. Structure: CatalogedSignature

Inherits: CatalogedEntry Cataloged Signature

[No fields]

11.5.4. Structure: CatalogedPublication

Inherits: CatalogedEntry A publication.

Id: String (Optional) Unique identifier code

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

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

NotOnOrAfter: DateTime (Optional) Epiration time (inclusive)

11.5.5. Structure: CatalogedCredential

Inherits: CatalogedEntry

Protocol: String (Optional) Specifies the client identification key

Service: String (Optional)

Username: String (Optional) Means of authenticating the host key

Password: String (Optional)

ClientAuthentication: KeyData [0..Many]

HostAuthentication: KeyData [0..Many]

11.5.6. Structure: CatalogedApplicationSsh

Inherits: CatalogedApplication The S/Mime encryption key

ClientKey: KeyData (Optional)

11.5.7. Structure: CatalogedNetwork

Inherits: CatalogedEntry

Protocol: String (Optional)

11.5.8. Structure: CatalogedContact

Service: String (Optional)

Username: String (Optional)

Password: String (Optional)

Inherits: CatalogedEntry Unique key.

Key: String (Optional)

Self: Boolean (Optional) If true, this catalog entry is for the user who created the catalog.

11.5.9. Structure: CatalogedAccess

Inherits: CatalogedEntry

[No fields]

11.5.10. Structure: Capability

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

Active: Boolean (Optional) The authentication mode: Device,

Issued: Integer (Optional) Account, PIN

Mode: String (Optional)

Udf: String (Optional) Identifies the authentication credential. For a device, this is the authentication key identifier, for an account, the profile identifier, for a PIN, the locator value of the PIN.

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

11.5.11. Structure: NullCapability

Inherits: Capability

[No fields]

11.5.12. Structure: AccessCapability

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

11.5.13. Structure: PublicationCapability

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

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

Data: Binary (Optional) The published document.

11.5.14. Structure: CryptographicCapability

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

11.5.15. Structure: CapabilityDecrypt

Inherits: CryptographicCapability
The corresponding key is a
decryption key

[No fields]

11.5.16. Structure: CapabilityDecryptPartial

Inherits: CapabilityDecrypt
The corresponding key is an encryption
key

[No fields]

11.5.17. Structure: CapabilityDecryptServiced

Inherits: CapabilityDecrypt
The corresponding key is an encryption
key

AuthenticationId: String (Optional)

UDF of trust root under which request to use a serviced capability must be authorized. [Only present for a serviced capability]

11.5.18. Structure: CapabilitySign

Inherits: CryptographicCapability
The corresponding key is an administration key

[No fields]

11.5.19. Structure: CapabilityKeyGenerate

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

[No fields]

11.5.20. Structure: CapabilityFairExchange

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

[No fields]

11.5.21. Structure: NamedService

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

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

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

11.5.22. Structure: ServiceAccessToken

Inherits: NamedService Session initiation token
Token: Binary (Optional)
SharedSecret: Binary (Optional) Session shared secret

11.5.23. Structure: CatalogedBookmark

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

Comments: String [0..Many]
User comments on bookmark entry

11.5.24. Structure: CatalogedTask

Inherits: CatalogedEntry Unique key.
EnvelopedTask: Enveloped (Optional)
Title: String (Optional) **11.5.25. Structure:**
Key: String (Optional) **CatalogedApplication**

Inherits: CatalogedEntry Enveloped keys for use with Application
Default: Integer (Optional)
Key: String (Optional) Escrow entries for the application.
Grant: String [0..Many]
Deny: String [0..Many] **11.5.26. Structure: CatalogedMember**
EnvelopedCapabilities: DareEnvelope [0..Many]
EnvelopedEscrow: Enveloped [0..Many]

ContactAddress: String (Optional)
MemberCapabilityId: String (Optional) **11.5.27. Structure:**
ServiceCapabilityId: String (Optional) **CatalogedGroup**
Inherits: CatalogedEntry

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

EnvelopedProfileGroup: Enveloped (Optional) The Mesh profile

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

11.5.28. Structure: CatalogedApplicationMail

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

11.5.29. Structure: CatalogedApplicationNetwork

Inherits: CatalogedApplication
[No fields]

11.5.30. Structure: MessageInvoice

Inherits: Message
[No fields]

11.5.31. Structure: CatalogedReceipt

Inherits: CatalogedEntry

[No fields]

11.5.32. Structure: CatalogedTicket

Inherits: CatalogedEntry

[No fields]

11.6. Publications

11.6.1. Structure: DevicePreconfigurationPublic

EnvelopedProfileDevice: Enveloped (Optional) The device profile

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

11.6.2. Structure: DevicePreconfigurationPrivate

Inherits: DevicePreconfigurationPublic

A data structure that is passed

EnvelopedConnectionDevice: Enveloped (Optional) The device connection

EnvelopedConnectionService: Enveloped (Optional) The device connection

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

11.7. Messages

11.7.1. Structure: Message

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

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

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

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

11.7.4. Structure: MessageValidated

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

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

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

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

11.7.5. Structure: MessagePin

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

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

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

11.7.6. Structure: RequestConnection

Connection request message. This message contains the information

Inherits: MessageValidated
AccountAddress: String (Optional)

11.7.7. Structure: AcknowledgeConnection

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

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

11.7.8. Structure: RespondConnection

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

Inherits: Message

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

CatalogedDevice: CatalogedDevice (Optional) The device information. MUST be present if the value of Result is "Accept". MUST be absent or null otherwise.

11.7.9. Structure: MessageContact

Inherits: MessageValidated 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.

11.7.10. Structure: GroupInvitation

Inherits: Message
Text: String (Optional) **11.7.11. Structure: RequestConfirmation**

Inherits: Message
Text: String (Optional) **11.7.12. Structure: ResponseConfirmation**

Inherits: Message
Request: Enveloped (Optional) **11.7.13. Structure: RequestTask**
Accept: Boolean (Optional)

Inherits: Message
[No fields]

11.7.14. Structure: MessageClaim

Inherits: Message
PublicationId: String (Optional) **11.7.15. Structure: ProcessResult**
ServiceAuthenticate: String (Optional)
DeviceAuthenticate: String (Optional) Report result of message
Expires: DateTime (Optional) processing.

Inherits: Message The error report code.
Success: Boolean (Optional)
ErrorReport: String (Optional) **11.7.16. Structure: ProcessResultNotSupported**

The message type is not supported.

Inherits: ProcessResult

[No fields]

11.7.17. Structure: ProcessResultNotFound

Inherits: ProcessResult

[No fields]

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-21, 23 October 2022, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-architecture-21>.

[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-02, 23 October 2022, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-callsign-02>.

[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-16, 23 October 2022, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-dare-16>.

[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-14, 23 October 2022, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-protocol-14>.

[draft-hallambaker-mesh-security]
Hallam-Baker, P., "Mathematical Mesh 3.0 Part IX Security Considerations", Work in Progress, Internet-Draft, draft-hallambaker-mesh-security-09, 20 April 2022, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-security-09>.

[draft-hallambaker-mesh-udf]
Hallam-Baker, P., "Mathematical Mesh 3.0 Part II: Uniform Data Fingerprint.", Work in Progress, Internet-Draft, draft-hallambaker-mesh-udf-17, 23 October 2022, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-udf-17>.

[draft-hallambaker-threshold]
Hallam-Baker, P., "Threshold Modes in Elliptic Curves", Work in Progress, Internet-Draft, draft-hallambaker-threshold-08, 23 October 2022, <https://datatracker.ietf.org/doc/html/draft-hallambaker-threshold-08>.

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

16. Informative References

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

[draft-irtf-cfrg-frost] Connolly, D., Komlo, C., Goldberg, I., and C. A. Wood, "Two-Round Threshold Schnorr Signatures with FROST", Work in Progress, Internet-Draft, draft-irtf-cfrg-frost-13, 8 May 2023, <https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-frost-13>.

[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>.