Workgroup: Network Working Group
Internet-Draft:
draft-hallambaker-mesh-ceremonies
Published: 27 July 2020
Intended Status: Informational
Expires: 28 January 2021
Authors: P. M. Hallam-Baker
ThresholdSecrets.com

Mathematical Mesh 3.0 Part XIII: Mesh Ceremonies

Abstract

Ceremonies are security protocols that involve human participants as principal actors. Ceremonies for onboarding devices, establishing trust between parties and obtaining multi-factor authenticated responses from users are presented and analyzed with particular application to the Mathematical Mesh.

https://mailarchive.ietf.org/arch/browse/mathmesh/Discussion of this draft should take place on the MathMesh mailing list (mathmesh@ietf.org), which is archived at .

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The consideration of ceremonies as an aspect of network protocol design was introduced by Carl Ellison in 2007 [Ellison]. Since then, consideration of ceremony design has provided a bridge between security practitioners focused on network protocol and human-computer interaction.

While the design of ceremonies is naturally connected to the design of the user experience, the former represents an abstraction of the latter. For example, the description ceremony might require that the user be able to distinguish between two states but not how this distinction is represented in the user experience.

Failure to consider ceremony design in protocol design frequently leads to the user being consider able to avoid security breaches through clairvoyance. Consider the commonly given but unactionable advice that users 'take care' when opening email attachments. On what basis is the user supposed to exercise caution when standard SMTP email provides no means for determining the authenticity of a message?

Formalizing the interactions of users in a protocol allows the designers to consider if the expectations being put on the users are likely to be met. It is easy for Web site operators to exhort users to use a strong, unique password, to change it frequently and not write it down. But there is not the slightest chance that users will follow this advice except on rare occasions because it is utterly unreasonable to expect them to remember a different password for each of the hundreds of services they use.

Ceremonies formalize the interactions of humans with machines, but humans are not Turing machines. They do not interact in precise ways; they misunderstand information they are provided with and they fail to follow instructions. It is essential that ceremonies be designed with these constraints in mind.

This document describes the ceremonies that are used to establish trust in the Mesh:

- Onboarding devices to a Mesh profile

- Contact endorsement and exchange

- Authenticated response interactions

While these particular ceremonies were designed to meet the design requirements of the Mesh, most are based on pre-existing interaction patterns that are widely used but not necessarily considered as a ceremony.

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

2.3. Related Specifications

2.4. Implementation Status

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

3. Ceremony Contexts

A Mesh ceremony provides an abstract description of the interactions between users and devices. A Ceremony context describes the specific means by which the abstract ceremony is realized.

The ceremony context may be considered the equivalent of the physical layer. Just as IP packets may be transferred using Ethernet or WiFi, the short codes used in many of the onboarding ceremonies may be exchanged using QR codes, Bluetooth, Near Field Communication or any other infrastructure that provides the necessary affordances.

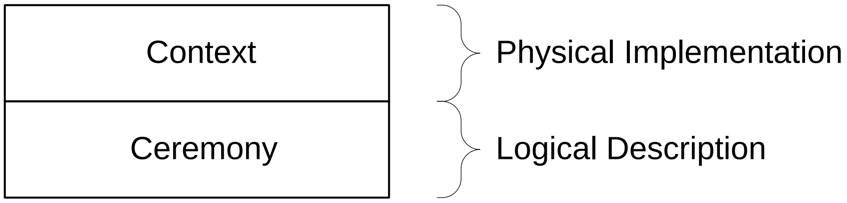

Figure 1

3.1. Users

It is assumed that users are of average technical skill or less and that they are unwilling to read any instructions or follow any procedure more complex than that required to purchase the target device.

The fact that a user is acting in an administrative role with respect to onboarding a device does not mean that they should be assumed to have administrative privileges on the machine they are using to perform that function.

3.2. Devices

The term 'device' is used to refer to any machine involved in the ceremony that is capable of communicating. Back at the dawn of the Internet age, every device connected to the Internet was at least the size of a washing machine that one or more users would interact with by means of a terminal device with a keyboard and either a display or a printer.

The range of affordances provided by modern devices is much broader. Today's desktop workstation provides essentially the same display, input an network affordances as those of a 'Personal Computer' sold in the mid-1990s. At the other end of the device capability spectrum, a 'smart' light bulb may offer only its light as a potential output and no inputs whatsoever.

Accessibility is an important consideration in contemporary systems design. Many users cannot use a traditional keyboard or display. In the interests of clarity, we refer to user text input devices as 'keyboards' and text output devices as 'displays' while recognizing that these **MAY** be realized using other technologies.

We recognize the following categories of device:

Static Computer A server or personal computer which is connected to a wired network (e.g. Ethernet). A static computer provides a keyboard and display, but not (necessarily) a camera.

Mobile Computer A laptop, tablet or smartphone device which supports use of a wireless network (e.g. WiFi, Cellular). A mobile computer provides a keyboard, display and a camera.

Device with Display A device which contains an embedded computer with a display affordance and some form of network communication capability (e.g. WiFi, Thread, Zigbee, Z-Wave, etc.) but not a keyboard or a camera.

Black Box Device A device which contains an embedded computer with some form of network communication capability (e.g. WiFi, Thread, Zigbee, Z-Wave, etc.) that does not provide input or output affordances to the user.

These capabilities are summarized as follows:

Class	Keyboard?	Display?	Camera?	Network Configuration
Static Computer	Yes	Yes	No	Automatic
Mobile Computer	Yes	Yes	Yes	Required
	No	No	No	Required

Class	**Keyboard?**	**Display?**	**Camera?**	**Network Configuration**
Device with Display				
Black Box Device	No	No	No	Required

Table 1

While there is a tendency for IoT devices to become more elaborate with succeeding generations, the expansion in the scope of IoT devices more than compensates for this effect. Thus just as there are more 8-bit CPUs manufactured today than at any point in history, the number of devices in the 'black box device' category will almost certainly increase over time rather than vanish.

3.3. Connection Codes

A Connection Code is a compact data object, typically 50 characters or less that is passed from one party to another during a ceremony.

Connection Codes **MAY** take many forms according to the capabilities of the devices involved.

*QR Code

*Serial communication using visible or Infra-Red light.

*Near Field Communications

3.4. Networks

IoT devices don't necessarily support IP

Local network config - sufficient to connect to the Mesh to bootstrap VPN.

Wired Supports automatic configuration of the local network context via DHCP

WiFi Requires an SSID to be specified and in some cases a password.

Non-IP A large range of wired and wireless communication protocols that provide local communication. Includes RS485, CANBUS, Zigbee, etc.

3.4.1. Wired Network Configuration

Wired networks such as Ethernet provide automated network configuration via DHCP.

Can use this network as-is or as a bootstrap to establish a connection through a VPN.

3.4.2. WiFi Configuration

WiFi networks support DHCP but this only acts after a device has connected to the WiFi network by specifying the correct SSID and providing the necessary credentials.

It is therefore necessary for an onboarding process to initialize the WiFi settings before making use of the Internet.

A secondary consideration is the need to update the WiFi settings on devices after the WiFi network configuration is changed or if the device is moved from one network operated by the user to another.

To support these requirements we anticipate the use of at least three separate WiFi SSIDs types:

The Public Network SSID The SSID used by the network that connects to the external Internet.

The Device Hailing SSID An SSID that is monitored by a target device that is not connected to a Mesh to allow it to receive inbound connection initiation requests.

The Mesh Hailing SSID An SSID that is monitored by a WiFi access point to allow devices previously connected to a Mesh to reconnect.

This approach allows a device that has no preconfigured WiFi network configuration to be onboarded to a user's personal Mesh. Once accepted, the device can then connect to any WiFi network connected to the user's personal Mesh listening on the Mesh hailing SSID.

Support for this configuration **MAY** be deployed at the WiFi access point(s) for the network or by deploying a separate parallel WiFi access point dedicated to serving hailing requests.

[Diagram: WiFi with hailing access point]

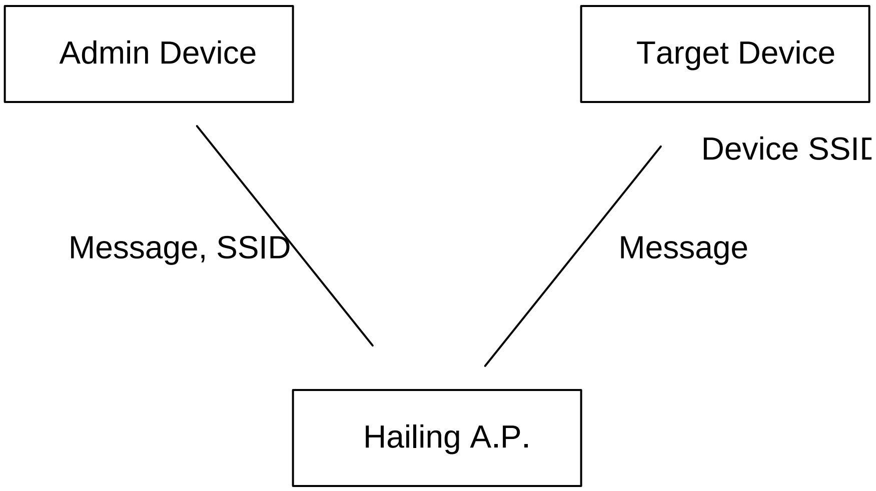

Figure 2

3.4.3. Non-IP Configuration

Configuration of non-IP networks is similar to that for WiFi with the important exception that some form of network gateway will be required to bridge the networks.

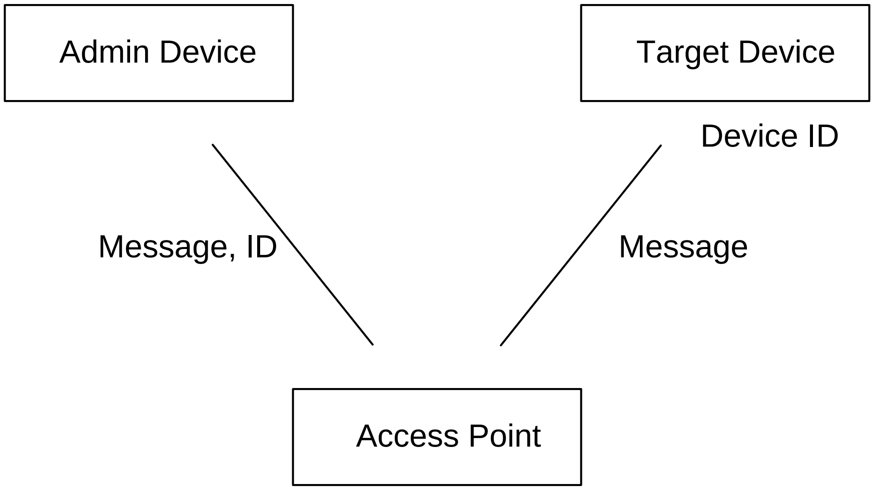

Figure 3

4. Device Onboarding Ceremonies

Devices

Target device The device being onboarded

Administration device A device that has the ability to authorize onboarding of the target device and cause the necessary credentials.

Capture device A device that has the ability to capture a connection code from a target device.

Combination device
A device that combines the administration and capture device roles.

Objectives

*Provide bootstrap network connectivity to the target device.

*Provision administrative axiom of trust to the target device.

*Establish trustworthy private keys on the target device.

*Provision credentials to the target device.

*The exchange of credentials **MUST** be mutually authenticated such that credentials are issued to a device if and only if it is the intended target device and it has received the intended administrative axiom of trust.

4.1. Networked Computing Device

4.1.1. Fingerprint Comparison

Primary application: Target device is a static computer. Administration and target devices are in close proximity.

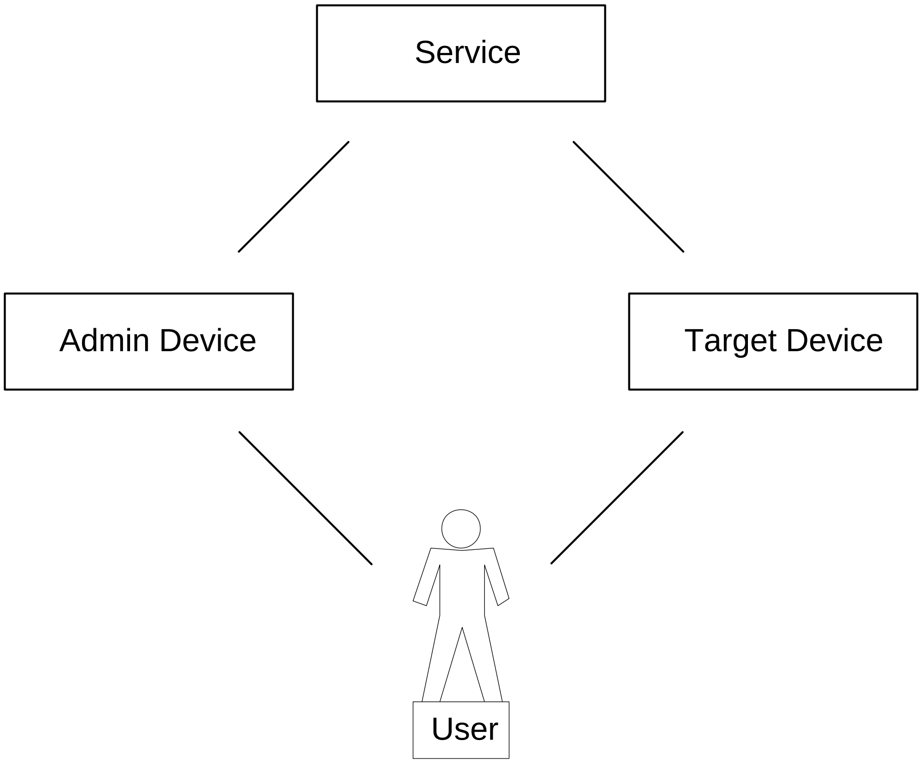

Figure 4

Target Device

Prompts user to enter account address and optional PIN

User on [Target Device] Enters account address into target device

Target Device Requests device connection to indicated Mesh Service account address

Mesh Service Returns connection verification code to Target Device

Posts connection request to indicated account

Target Device Presents connection verification code to user

Administration Device Synchronizes account to Mesh Service, receives pending connection request

[Optional] Prompts user for attention

User [Administration Device] Reviews pending connection requests on administration device

Verifies that connection codes match, rejects request if they do not match

Accepts request

Administration Device Posts result of connection request to Mesh Service

Mesh Service Appends results to for collection spool.

Target Device Requests results of connection request

Mesh Service Returns results

Target Device Decrypts result and completes configuration.

4.1.2. Out of band one-time code

Primary application: Target device is a static computer. Administration and target devices are not in close proximity.

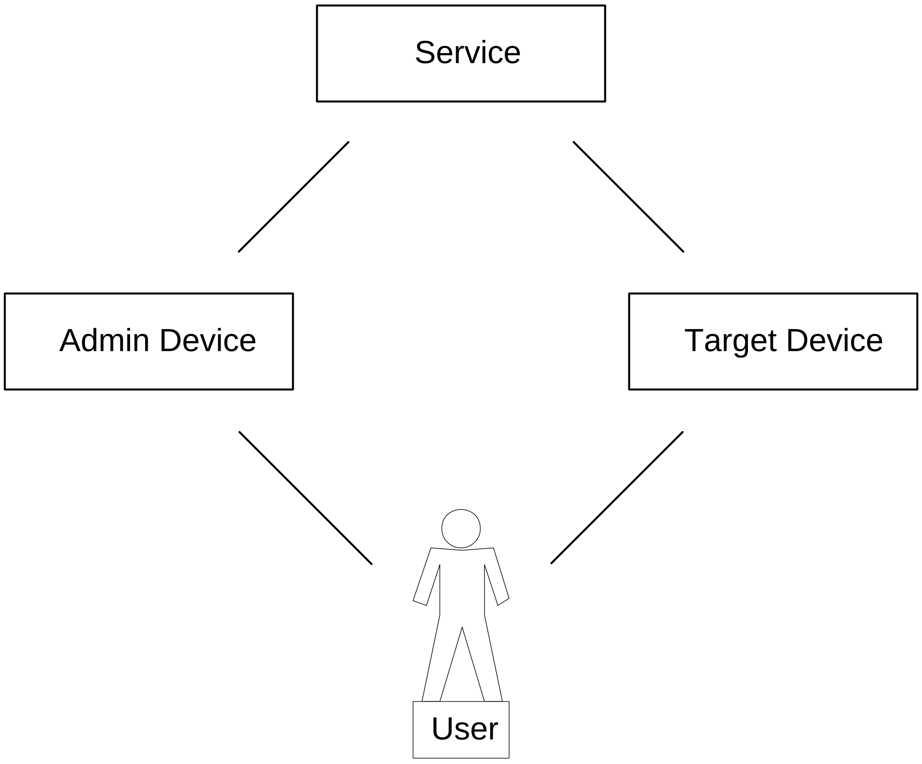

Figure 5

Chief difference is that the

User on [Administration Device] Requests PIN code

Administration Device Generates PIN code

Reports PIN code to user

Posts to administration catalog to allow other administration devices to use code for verifying connection request

Mesh Service Receives administration catalog entry.

Target Device Prompts user to enter account address and optional PIN

User on [Target Device] Enters account address and PIN into target device

Target Device Requests device connection to indicated Mesh Service account address using PIN for authentication.

Mesh Service Returns connection verification code to Target Device

Posts connection request to indicated account

Target Device Presents connection verification code to user

Administration Device Synchronizes account to Mesh Service, receives pending connection request

Verifies one-time code has been correctly specified, has not been expired or previously used.

If so, accepts connection request as pre-approved

Posts result of connection request to Mesh Service

Mesh Service Appends results to for collection spool.

Target Device Requests results of connection request

Mesh Service Returns results

Target Device Decrypts result and completes configuration.

4.2. Network bootstrap

Target device does not initially have network capability.

Requires code capture mechanism

4.2.1. Dynamic Code From Administration Device

Requires code capture capability on target device

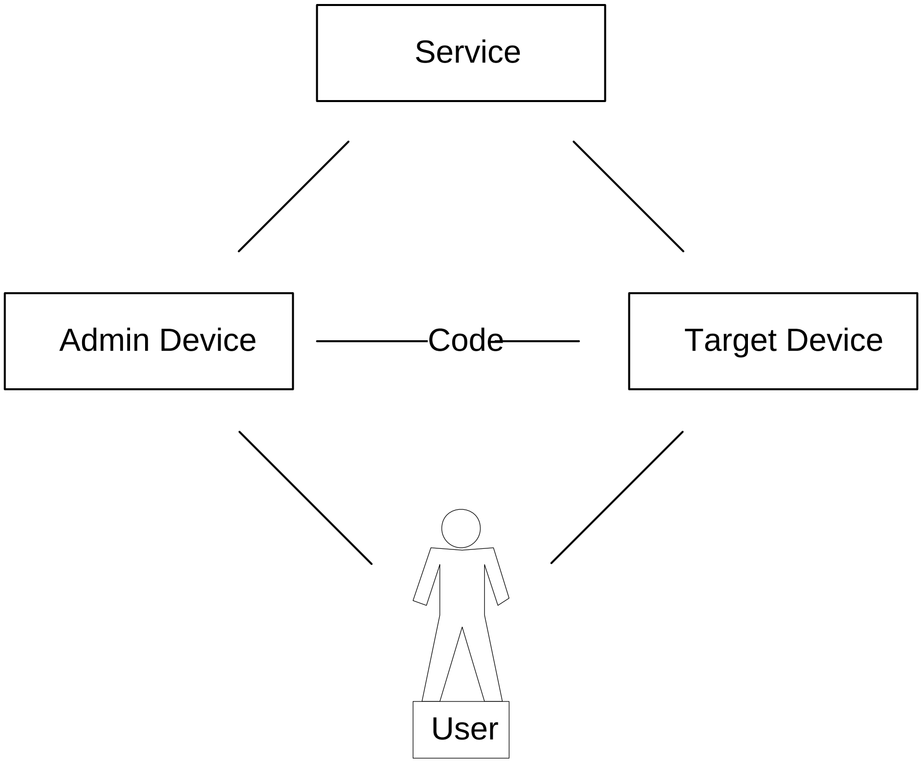
Service
Admin Device
Code
Target Device
User

Figure 6

User on Administration Device Requests device connection code display

User on Target device Scans code displayed on Administration device

Target Device Acquires connection code

Decodes local network bootstrap configuration and connection secret from connection code

Acquires access to bootstrap network

Posts connection request to service using connection secret for authentication.

Mesh Service Returns connection verification code to Target Device

Posts connection request to indicated account

Administration Device Synchronizes account to Mesh Service, receives pending connection request

Verifies one-time code has been correctly specified, has not been expired or previously used.

If so, accepts connection request as pre-approved

Posts result of connection request to Mesh Service

Mesh Service Appends results to for collection spool.

Target Device Requests results of connection request

Mesh Service Returns results

Target Device Decrypts result and completes configuration.

4.2.2. Code From Target Device

Requires code capture capability on administration device

Code may be dynamic or static.

Dynamic provides same security as for the Admnistration device display but requires the target device to have the display affordance.

Static avoids need for a display, the code is physically printed on the device itself. In this case the code is static meaning that the connection secret allowing anyone who has handled the device to hijack the connection attempt.

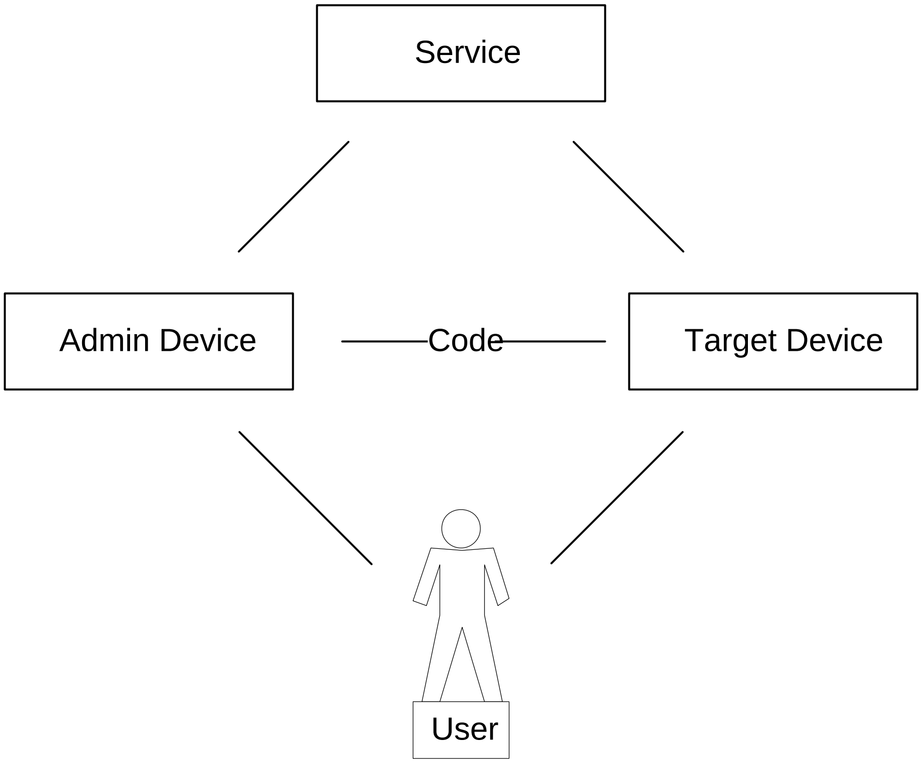

Figure 7

User on Target device Requests device connection code display

Target device Presents device connection code

User on Administration Device Scans code displayed on Target device

Administration Device Acquires connection code

Decodes connection secret and device bootstrap network configuration

Obtains device description and presents to user [*]

User on Administration Device Accepts connection

Administration Device Posts result of connection to device via device bootstrap network

Target Device Receives connection result from Administration device

Verifies that connection result is consistent with connection code posted.

The device description **MAY** be acquired from either

- *The device itself (via the Device bootstrap network)

- *The Internet

- *In either case the UDF digest of the connection secret is used to form the locator.

4.3. Proxy configuration

As before except that the administration functions are divided between the administration device and a separate capture device.

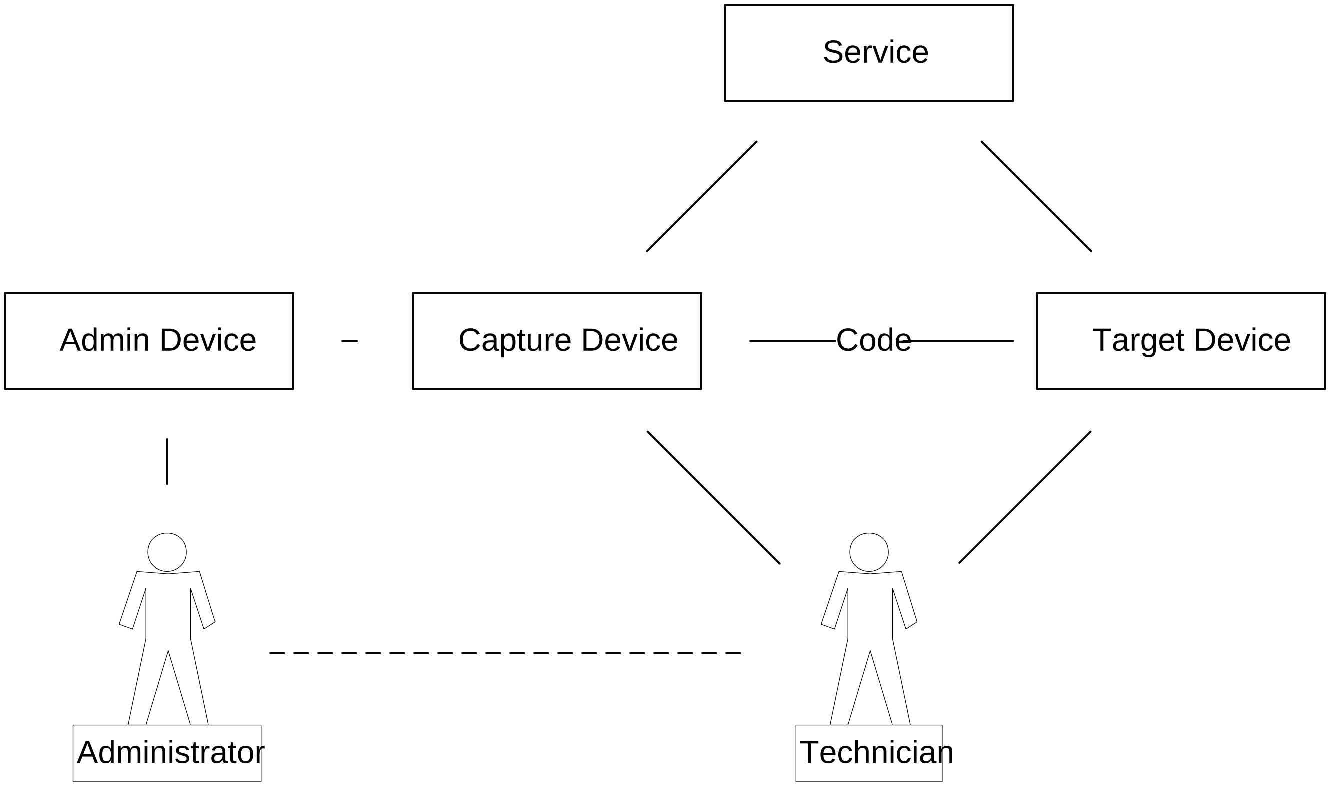

Figure 8

5. Contact Establishment Ceremonies

5.1. In Person

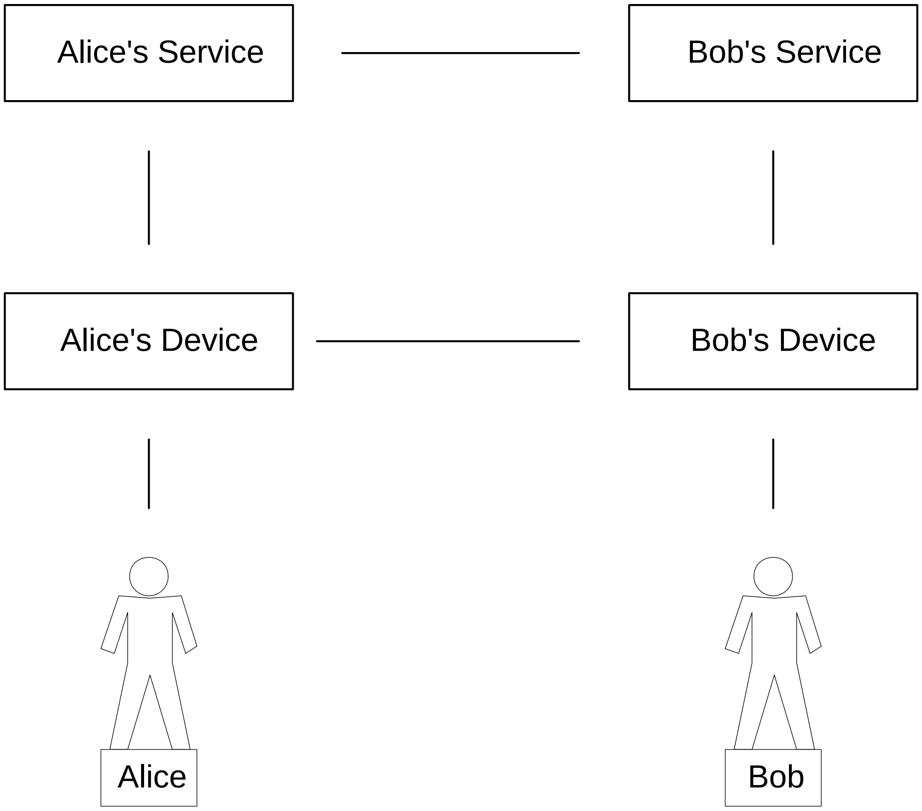

Figure 9

User Alice Opens contacts app, requests contact creation

Alice's Device Presents connection code

User Bob Scans connection code

Bob's Device Opens contacts app, acquires connection code

Decodes Connection Secret and Alice's account address

Posts connection request message to Bob's Mesh Service using connection secret as authenticator

Bob's Mesh Service Receives connection request from Bob

Forwards to Alice's account address

Alice's Mesh Service Receives connection request from Bob

Adds to Alice's inbound message spool

Alice's Device Synchronizes to Alice's Mesh Service

Receives message from Bob

Verifies that message is authenticated by connection secret, if not abort.

Presents Bob's contact information

User Alice Accepts Bob's contact

Alice's Device Generates contact response for Bob posts to Alice's Mesh Service using connection secret as authenticator

Alice's Mesh Service Forwards connection response to Bob's Mesh service

Bob's Mesh Service Receives connection response from Alice

Adds to Bob's inbound message spool.

Bob's device Synchronizes to Bob's Mesh Service

Receives connection response

Presents contact information to Bob

User Bob Accepts Alice's contact information

Bob's device Adds Alice's contact information to Bob's contacts catalog

Since it is easy to delete a contact entry from the catalog, users may opt to accept contact information without explicit user verification.

The application **SHOULD** capture the circumstances in which the contact was acquired including the time and place (if available). For example, if Alice meets Bob at a conference for which there is an entry in their calendar, their contacts app might make use of this information to label the connection.

As with any other type of connection, an in-person connection **MAY** be enrolled in a notary log to provide a fixed point in time.

5.2. Registration

Registration is essentially a variant of the In-Person contact exchange ceremony in which Bob establishes evidence of attendance at an event such as a conference by means of his connected mobile device.

The ceremony is identical to that of the In-Person contact exchange with the Roles 'Alice' and 'Alice's Device' replaced by 'Registrar' and 'Registrar's device' respectively.

Registration at one or mode physical events **MAY** be used by trusted third parties as the basis for providing endorsements according to their published Endorsement Policies and Practices Statements.

5.3. Remote

In the remote contact exchange scenario, Alice and Bob are not present in the same physical location and thus the risk of impersonation is inevitably increased. Despite this limitation, remote contact exchange allows participants to determine that they are interacting with the same person as in previous interactions. Which is sufficient for a wide number of purposes.

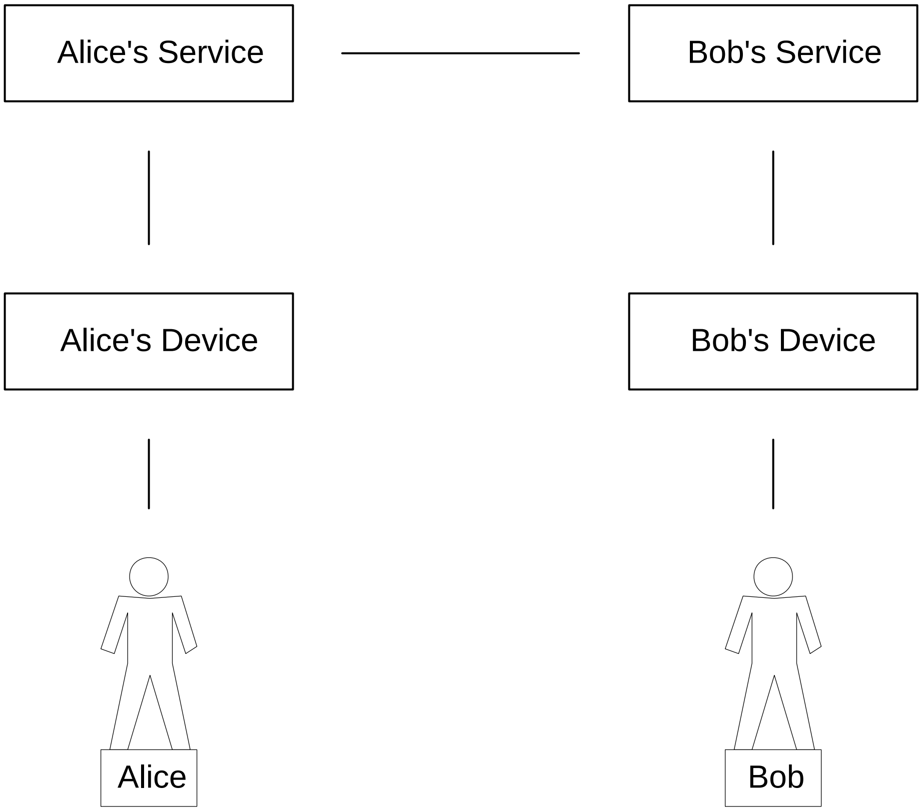

Figure 10

Alice Receives Bob's Account Address out of band

Opens contact management application

Requests remote contact establishment with the user at Bob's Account address.

Alice's Device Creates and signs the contact establishment request.

Alice's Service Receives contact establishment request

Signs request and forwards to Bob's Service.

Bob's Service Receives contact establishment request

Verifies signature of Alice's Service, rejects if invalid

Adds request to Bob's inbound spool.

Bob's Device Synchronizes to Bob's Service

Decrypts request

Verifies signature of Alice's request, rejects if invalid, rejects if insufficiently authorized according to policy (i.e. spam prevention).

Notifies Bob of pending request according to previously specified policy.

Bob Reviews request from Alice

Accepts or rejects request.

Bob's Device If reject, abort.

Otherwise add contact to Bob's contact catalog.

Create and sign contact request for Alice including proof of knowledge of request secret.

Bob's Service Receives contact establishment request

Signs request and forwards to Alice's Service.

Alice's Service Receives contact establishment request

Verifies signature of Bob's Service, rejects if invalid

Adds request to Alice's inbound spool.

Alice's Device Synchronizes to Alice's Service

Decrypts request

Verifies signature of Bob's request, rejects if invalid, rejects if insufficiently authorized according to policy (i.e. spam prevention).

Verifies proof of knowledge of request secret. If invalid, ignore.

Add's contact to Alice's contact catalog.

5.4. Trusted Third Party Endorsement

Trusted third party endorsement **MAY** be used to improve the reliability of either the in-person or remote contact establishment ceremonies.

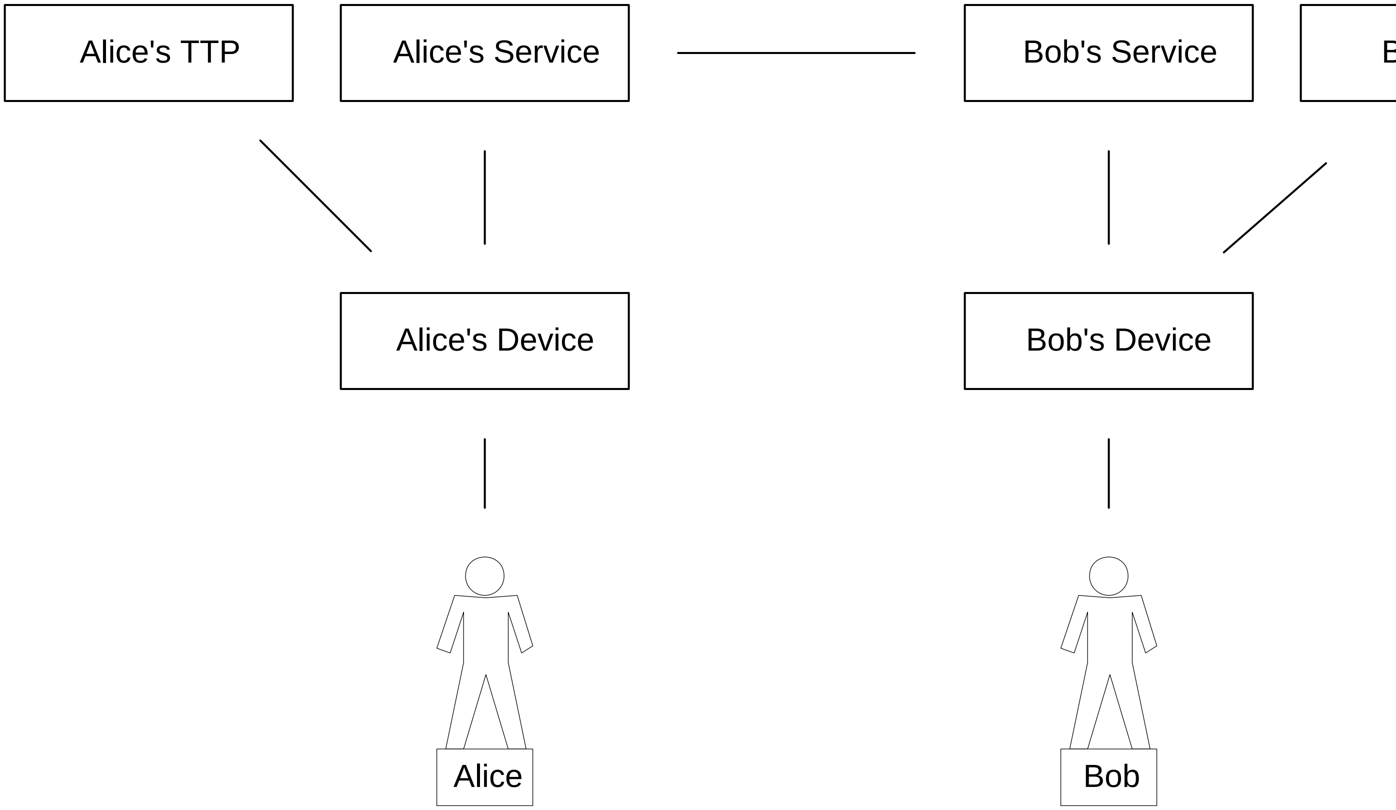

Figure 11

The ceremony mechanics are unaffected except at the point where the contact information is accepted when the information from one (or more) trusted third parties **MAY** be presented to the user to assist them in their decision to accept or reject the contact information.

Trusted Third Parties **MAY** provide an ongoing service, advising users of a change in the trustworthiness of contact data.

6. Authentication Ceremonies

Second factor authentication by means of entering a one time code is widely used despite the obvious limitations of this approach:

*A response code of four, six or even eight digits has a miniscule work factor compared to the industry benchmark of 2^{128} or greater.

*The process of presenting a code is vulnerable to Man in the Middle attack.

*Response codes are not bound to the context in which they are used and thus do not provide for non-repudiability.

Modern mobile devices are both ubiquitous and offer sufficient affordances to provide user experience that is more satisfactory and offers substantially greater security.

6.1. Confirmation

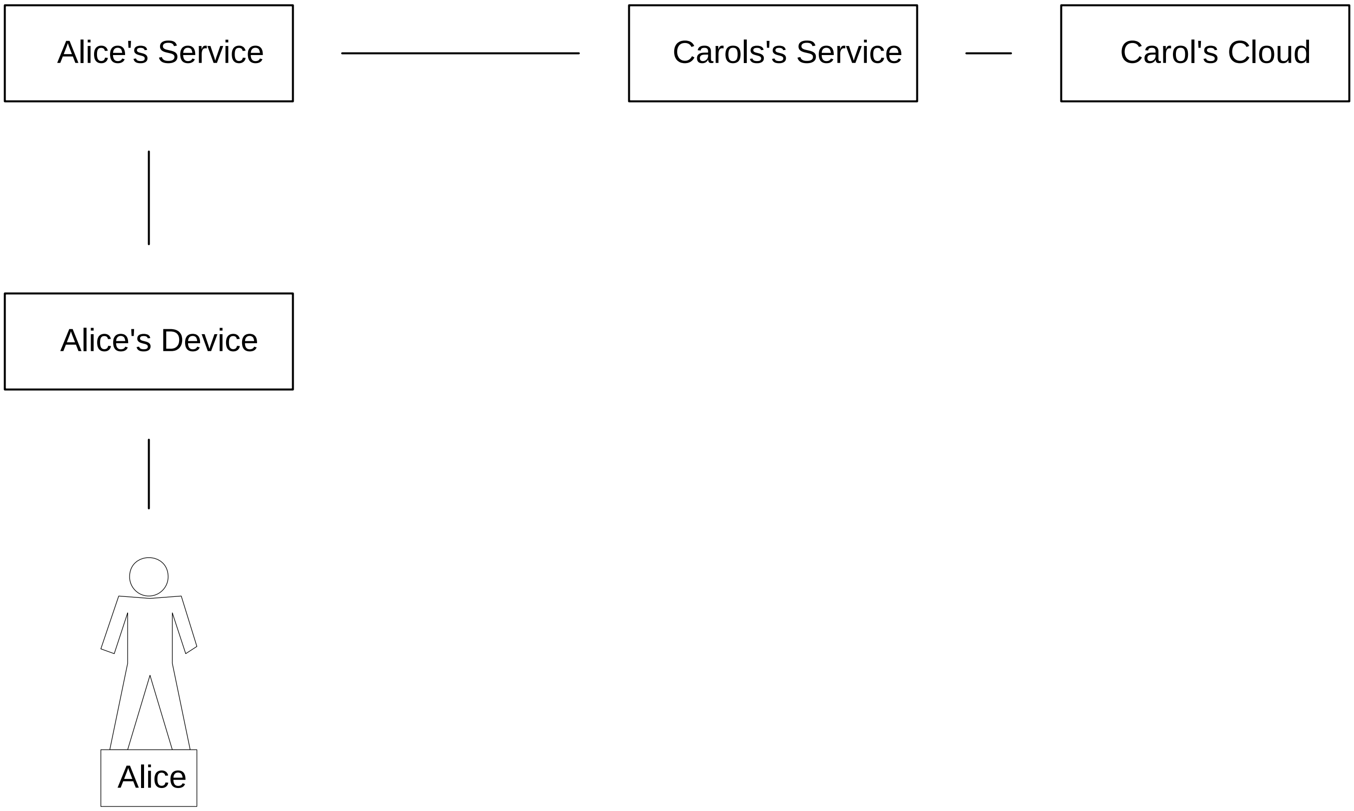

Figure 12

Alice Attempts action requiring confirmation at Carol's Cloud

Carol's Cloud Generates and signs confirmation request for Alice's Account Address

Sends to Carol's Service

Carol's Service Countersigns confirmation request and forwards to Alice's Service

Alice's Service Verifies countersignature of Carol's service, if invalid, aborts

Adds confirmation request to Alice's inbound spool.

Alice's Device Synchronizes to Alice's Service

Discards messages that are unauthorized according to entries in contacts catalog

Decrypts confirmation request

Notifies Alice according to policy.

Alice Reads confirmation request

Responds with Accept or Reject.

Alice's Device Creates and signs/encrypts response including request data

Appends to confirmation catalog.

Forwards response to Alice's Service.

Alice's Service Countersigns response, forwards to Carol's Service

Carol's Service Verifies counter signature, rejects if invalid

Appends response to Carol's inbound spool.

Carol's Device Synchronizes to service.

Decrypts response, acts accordingly.

Waiting for the response from Alice is essentially equivalent to waiting for input from Alice

This description assumes that the devices poll the service to obtain notification of updates. Provision for push notifications will of course improve response.

6.2. Confirmation with Personal Presence

In certain situations we would like to require that Alice be physically present when responding to a confirmation request. For example, when opening a door or logging in to a workstation at a facility.

In these circumstances, a confirmation code is used to provide evidence that Alice is in the physical vicinity. Such codes may be presented by means of a QR code, Near Field Communications or any equivalent means. Noting of course that all proximity controls are inherently vulnerable to a relay attack.

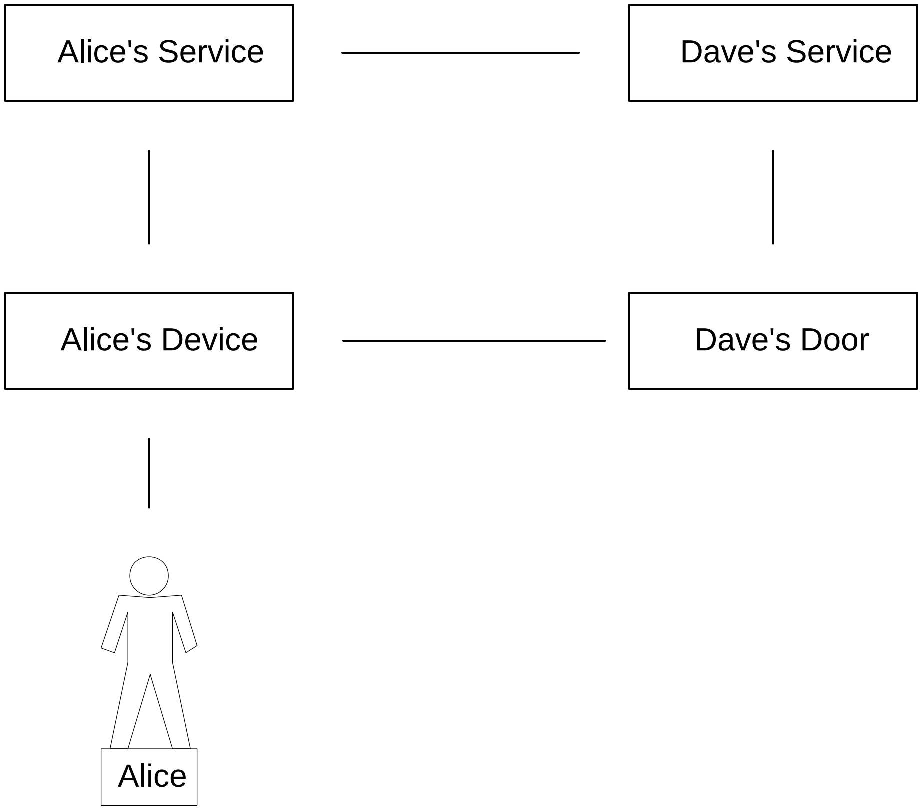

Figure 13

Alice Attempts action requiring confirmation with physical presence at Dave's Door

Dave's Door Generates unique connection code.

Generates and signs confirmation request for Alice's Account Address

Sends to Dave's Service

Presents connection code to Alice's Device via local channel

Alice's Device Reads Connection code

Synchronizes to Alice's Service, no message pending (yet), waits.

Dave's Service Countersigns confirmation request and forwards to Alice's Service

Alice's Service Verifies countersignature of Dave's service, if invalid, aborts

Adds confirmation request to Alice's inbound spool.

Alice's Device Synchronizes to Alice's Service

Discards messages that are unauthorized according to entries in contacts catalog

Decrypts confirmation request

Notifies Alice according to policy.

Alice Reads confirmation request

Responds with Accept or Reject.

Alice's Device Creates and signs/encrypts response including request data

Appends to confirmation catalog.

Forwards response to Alice's Service.

Alice's Service Countersigns response, forwards to Dave's Service

Dave's Service Verifies counter signature, rejects if invalid

Appends response to Dave's inbound spool.

Dave's Door Synchronizes to service.

Decrypts response, acts accordingly.

7. Security Considerations

8. IANA Considerations

This document requires no IANA actions.

9. Acknowledgements

10. Normative References

[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>.

11. Informative References

[draft-hallambaker-mesh-developer]

Hallam-Baker, P., "Mathematical Mesh: Reference Implementation", Work in Progress, Internet-Draft, draft-hallambaker-mesh-developer-09, 23 October 2019, <https://tools.ietf.org/html/draft-hallambaker-mesh-developer-09>.

[Ellison] Ellison, C., "Ceremony Design and Analysis", 2007.