Mathematical Mesh: Web Application Binding

Mathematical Mesh Web Application

<series>draft-hallambaker-mesh-app-web

<status>informational

<stream>independent

<ipr>trust200902

<author>Phillip Hallam-Baker

<surname>Hallam-Baker

<initials>P. M.

<firstname>Phillip

<organization>Comodo Group Inc.

<email>[philliph@comodo.com](mailto:philliph@comodo.com)

<also>http://prismproof.org/Documents/draft-hallambaker-mesh-app-web.html

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. This document describes the use of the Mesh to store Web application information.

# 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.

## 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 <norm="RFC2119"/>.

## Related Specifications

The related specifications are described in the Mesh Architecture specification <norm="draft-hallambaker-mesh-architecture"/>

## Defined Terms

No terms of art are defined.

## Implementation Status

The implementation status of the reference code base is described in the companion document <info="draft-hallambaker-mesh-developer"/>.

# Introduction

The Mathematical Mesh is a personal PKI that permits a user to connect multiple devices to a ‘personal profile’ through which application information is shared between the connected devices. All Mesh communications are secured through a combination of end-to-end security to protect confidentiality and integrity and transport security to provide protection against traffic analysis.

A full description of the Mathematical Mesh architecture is to be found in <norm="draft-hallambaker-mesh-architecture"/>

This document describes a proposed design for a demonstration of using the Mesh to provide a cloud based password manager for connected Web browsers. The approach may be readily extended to support management of Web bookmarks.

<include=..\Examples\ExamplesWeb.md>

## Bookmark Management

The use of the Mesh to store bookmarks is an obvious extension to use of the Mesh as a password manage. The principal differences being that the privacy concerns are somewhat less critical than storing credentials and a bookmark file is likely to be considerably longer than a password file.

The principal design challenge in adding bookmarks is working out how to provide a convenient interface to help the user manage their bookmarks. A hierarchical list of folders quickly becomes cluttered.

# Application Schema

<include=..\Generated\SchemaWeb.md>

# Demonstration

A demonstration of using the Mesh to manage Web browser passwords is described.

The end goal in developing the Mesh application protocols is to encourage application providers to provide native support for the Mesh rendering extensions obsolete. Such implementation is likely to be best encouraged through provision of a reference library in C.

I propose implementation of a demonstration as follows:

* Platform Windows
* Browser: Chrome
* Approach:
  + Integration to browser features to be supported by platform independent extension module
  + Mesh integration to be provided by a platform specific executable written in C.

For initial testing / canned demo purposes, the Mesh integration module will be a ‘stub’ that access a data file at a defined location on disk that contains the PasswordProfilePrivate data structure. The task of synchronizing data with the Mesh will be performed using the Mesh profile management client.

Further development:

* Implementation of the production extension by modifying the platform specific executable.
* Support for macOS by implementing a Mac specific platform executable
* Support for Linux by implementing a Mac specific platform executable

This approach allows the platform specific extensions to be tailored to the cryptographic key management capabilities offered by each platform. For example, the use of a TPM to protect private keys on Windows or the Keyring mechanism on macOS.