**Software Requirements Specification Template**



**FIDO UAF Multi-factor Authentication Requirements Specification**



Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **document ID** | **Description** | **Reviewed By** | **Approved By & Date** | **Release Date** | **Effective Date** |
| FMA\_SRC\_01 | Software Requirements Document for FIDO UAF MFA | Pandurang Kamat |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document Distribution**: Persistent Systems Ltd.

**Document Owner**: CTO Org

Software Requirements Specification

FIDO UAF Multi-factor Authentication

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Description** | **Author** | **Reviewed By** | **Approved By & Date** |
| **0.1** | **First Draft** | **Sudhakar Shenoy** | **Pandurang Kamat** |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Contents

[1 Introduction 5](#_Toc446165709)

[2 Scope 5](#_Toc446165710)

[3 Definitions, Acronyms and Abbreviations 5](#_Toc446165711)

[4 References 5](#_Toc446165712)

[5 Overall Description 6](#_Toc446165713)

[6 Details 7](#_Toc446165714)

[7 Timelines 8](#_Toc446165715)

# Introduction

This document provides the Software Requirements Specification for the FIDO UAF Multi-factor Authenticator. The FIDO UAF is a protocol specification for password-less user registration and authorization. The protocol design ensures that the private user data such as password and keys used for registration does not leave the user device. This also means that the user data is not available on the servers and user data is safe even if the server is compromised. The protocol is built to ensure that user does not have to enter password multiple times. It can also increase the security levels based on the risk of the transaction. E.g. login to a portal is considered as low risk as against transferring $10,000. This provides flexibility to increase/decrease security level based on risk levels of the transaction.

# Scope

The scope of this document is define the high level requirements for the FIDO UAF MFA as per the current discussions. The scope can be expanded based on subsequent roadmap discussions.

# Definitions, Acronyms and Abbreviations

**FIDO** – Fast Identity Online

**UAF** – Universal Authentication Framework

**U2F** – Universal 2nd Factor

**MFA** – Multi-factor Authentication

**RSA** – Rivest-Shamir-Adleman Cryptographic Algorithm

**EC** – Elliptical Curve

**ASM** – Authenticator Specific Module

**AAID – Authenticator Attestation ID**

*A unique identifier assigned to a model, class or batch of FIDO Authenticators that all share the same characteristics, and which a Relying Party can use to look up an Attestation Public Key and Authenticator Metadata for the device.*

# References

## Project Glossary

* FIDO UAF Protocol Specification v1.0 (<https://fidoalliance.org/specs/fido-uaf-v1.0-ps-20141208/fido-uaf-reg-v1.0-ps-20141208.html>) Author – FIDO Alliance
* FIDO UAF Registry of Predefined Values (<https://fidoalliance.org/specs/fido-uaf-v1.0-ps-20141208/fido-uaf-reg-v1.0-ps-20141208.html> ) Author – FIDO Alliance
* FIDO Technical Glossary (<https://fidoalliance.org/specs/fido-uaf-v1.0-ps-20141208/fido-glossary-v1.0-ps-20141208.html>) Author – FIDO Alliance

# Overall Description

## Product Perspective

## Product Functions

## User Characteristics

## Constraints

## Assumptions and Dependencies

The current requirements has dependency on the available open source implementation of FIDO UAF from eBay.

## Requirements Subsets

### FIDO Server

FIDO Server is a protocol compliant implementation of FIDO UAF. The Server should be able to perform the following functionality:-

* Generate Registration Request
* Validate Registration Request Response
* Store Registration Information (preferably in DB)
* Generate Authentication Request
* Validate Authentication Request Response
* Update Attestation information using Metadata Service.
* Deregister

### FIDO Client (iOS)

### FIDO Client (Android)

### Server Administration (optional)

# Details

## Functionality

### Discovery of Authenticators

* The app should be able to discover all possible user verification methods available on the device.
* The discovered authenticator list should be made available for various FIDO operations.
* The user should be presented appropriate user verification method based on the above discovered methods and policy shared by server during Registration and Auth operations.

### Registration

* In this operation, the Client device registers itself with the FIDO Server.
* The attestation key is used to sign the data shared by server.
* The client also generates a key pair for server to register the device.

### Authentication

* In this operation, the client device authenticates itself with the FIDO server

### Transaction Confirmation

* Based on risk involved, some of the transactions may request enhanced security.
* Server will display user verification, along with transaction details.
* The client will approve/reject the transaction based on user verification method available.

### Deregistration

* Deregister a device from the FIDO server

## Security

* Ensure that all the private keys never leave the end user device.
* The private keys stored on the end user device are stored in secure manner and is not accessible to other apps on the device.

## Design Constraints

## Interfaces

* DB Interface to store data
* App Interface to key store
* Metadata Service Interface

# Timelines

## Use case Timelines

* Basic Flow - Registration (ETA - )

This flow will use the current ebay implementation as-is with addition of sign and verify functionality. The Registration request will have only Accepted AAIDs in the policy. The user registration will be stored in hashmaps and no DB will be there as permanent storage.

Components – FIDO Server, FIDO Client (iOS/Android?)

* Client invokes the “regRequest” Rest API.
* Server returns the default policy, server data, and challenge.
* Client generates Public/Private Key pair using (RSA/EC??) algorithm.
* Client signs the data using the private key and sends the assertion to server.
* The server verifies the client assertions.
* Stores the user public key against the user name and completes User Registration.
* Basic Flow – User Authentication ( ETA - )

Once user registration of device is completed, successive logins from the device will follow the User Authentication workflow. Following defines this workflow :-

* Client initiates the login using the registered device.
* This prompts “AuthRequest” API to be invoked.
* The server presents the server data, server challenge and policy configuration in the response.
* The client signs the data using the authenticator key and sends the response.
* The server verifies the response using the keys stored during registration operation.
* Enhanced Server ( ETA - )

The basic eBay server module is enhanced to support wider range of feature and additional verifications, which are currently stubbed in the open source code.

* Metadata Service
* Adding FCParams and channel binding.
* FCP Verification
* Dereg Flow

## FIDO Server

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Task Description** | **Priority** | **Owner** | **ETA** |
| Integration | Integrate Sign & Verify functionality with ebay code  (Understanding the code + adding the sign and verify functions which are already completed and tested separately) | High | Jasmira | 12 hours/ Complete (RSA+ECDSA) |
| Storage | Database connectivity  (Deciding which DB to use) | Medium | Jasmira | 9 hours/ Complete (connected to MySQL server 5.5 DB) |
|  | DB Schema Design  (come up with table and columns in DB) | Medium | Jasmira | 8 hours |
|  | StorageImpl using DB (add methods to store Registration Record, Login details, etc) | Medium | Jasmira | 9 hours/ Complete |
| Reg | Basic Reg flow For RSA Keys (IOS/Android Input) | High | Jasmira | Complete |
|  | Basic Reg flow For ECSDA Keys (IOS/Android Input) | High | Jasmira | Complete |
|  | Generate FCP | Medium | Jasmira | 7 hours/ Complete |
|  | Generate MatchCriteria | Medium | Jasmira | 9 hours |
|  | Verify FCP | Medium | Jasmira | 7 hours |
| Auth | Basic Auth flow for RSA keys(IOS/Android) | High | Jasmira | Complete for IOS/ Android 5 hours. |
|  | Basic Auth flow for ECDSA Keys (IOS/Android) | High | Jasmira | 7 hours |
|  | Get Public Key from Registration Record stored in DB instead of Hashmaps for auth validation | High | Jasmira | Complete |
|  | Match criteria class of ebay to include all policy related fields (userVerification, keyProtection, tcDisplay, authenticationAlgorithms, assertionSchemes) | Low | Jasmira | TBD |
| Dereg | Include DB to add and then delete user details | Low | Jasmira | TBD |

## FIDO Client (iOS)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Task Description** | **Priority** | **Owner** | **ETA** |
| iOS UAF client workspace setup | Develop iOS workspace add libraries for Network Management, JSON parsing, Data Model | High | Amogh | Completed |
|  | Add UAF client module to perform Registration and Authentication operations | High | Amogh | Completed |
|  | Add UAF ASM module to perform Registration and Authentication operations | High | Amogh | Completed |
| Reg Set up | Add network methods for Registration operations. Registration GET request and POST response | High | Amogh | Completed |
| Add Data models for registration JSON parsing and JSON builders | High | Amogh | 4 hours |
| Client module: Add Client registration module components | Develop UI for registration | High | Amogh | Completed |
| Methods for consuming registration network methods | High | Amogh | Completed |
| Consume ASM module methods for processing registration response | High | Amogh | Completed |
| Process ASM response and complete registration |  |  | 4 hours |
| ASM module: Add ASM registration module components | Methods for Client-ASM interaction and Touch ID ASM registration data models | High | Amogh | Completed |
| Touch ID user Verification+ UI | High | Amogh | Completed |
| RSA Secure Key Generation + Public key extraction + Key compatibility operations | High | Amogh | 2 hours |
| Build Assertions using server data + generated keys | High | Amogh | 8 hours |
| Attestation Operation : Extract RSA attestation private key from p12 file + Sign | High | Amogh | 2 hours |
| Build Registration Response using server data (FCP)+ assertions | High | Amogh | 6 hours |
| ASM module: Add ASM registration module component Enhancements | EC Secure Key Generation + Public key extraction + Key compatibility operations | High | Amogh | 6 hours |
|  | Attestation Operation : Extract RSA attestation private key from p12 file + Sign | High | Amogh | 6 hours |
| Registration Task Total ETA Hours |  |  |  | 38 Hours |
|  |  |  |  |  |
|  |  |  |  |  |

## FIDO Client (Android)

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Task Description** | **Owner** | **ETA** |
|  | Add ASM registration |  |  |
|  |  |  |  |
|  |  |  |  |

## Server Administration(Optional)

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Task Description** | **Owner** | **ETA** |
| Configuration Console | Provide a UI console for server parameter configuration |  |  |
|  | Add support for User Verification Configuration |  |  |
|  | Add support for Matcher Protection |  |  |
|  | Add support for Authenticator Attachment hints |  |  |
|  | Add support for Transaction confirmation display type |  |  |
|  | Add support for Authentication Algorithms |  |  |
|  | Add support for Public Key Representation formats |  |  |
| Support for multiple vendors | Support multiple vendor configuration |  |  |