UIDAI

Unique Identification Authority of India
Planning Commission, Govt. of India (GoI),
3rd Floor, Tower II,
Jeevan Bharati Building,
Connaught Circus,
New Delhi 110001

AADHAAR MOBILE UPDATE

API SPECIFICATION - VERSION 1.0

APRIL 2015

Table of Contents

1. Introduction

The Unique Identification Authority of India (UIDAI) has been created, with the mandate of providing a Unique Identity (Aadhaar) to all Indian residents. The UIDAI proposes to provide online authentication using demographic and biometric data.

For mobiles to be instruments of authentication for digital identity, they should by unique, authenticable and fulfil requirements of non-repudiation. A possible way of achieving the same would be to link the mobile numbers of users to their Aadhaar and use Aadhaar OTP as one of the user authentication scheme. Aadhaar system already offers a mechanism to keep mobile number linked to Aadhaar identity and use it via mobile OTP (One Time Pin) based authentication.

Use of Aadhaar linked mobile OTP authentication allows the following:

1. Enable remote and secure verification of an Aadhaar holder using mobile during mobile based services.
2. Enable use of mobiles as an identity instrument and trusted authentication factor that is attached to Aadhaar, thereby simplifying online access to public services.

In addition, registered mobile numbers are used for resident communication, providing UIDAI mobile application for features such as HOTP, biometric locking, etc.

Such large scale usage assumes that mobile number in Aadhaar database is up to date. In addition to currently existing update mechanisms (permanent update centres, self-service update portal, etc.), if UIDAI offers a mobile update API for "trusted" authentication user agencies (such as specific Government AUAs, banks, telcos, etc.), few lakhs of biometric update touch points can be made available to residents for easy linking and updating of their mobile number within Aadhaar system.

This document is the proposed biometric based Mobile Update API for use within trusted AUAs.

1.1 Target Audience and Pre-Requisites

This is a technical document and is targeted at software professionals working in technology domain and interested in incorporating Aadhaar Mobile Update API into their applications.

Readers must be fully familiar with following authentication documents published on UIDAI website (http://uidai.gov.in/auth) before reading this document.

1. Aadhaar Authentication Framework - http://uidai.gov.in/images/authDoc/d2_authentication_framework_v1.pdf
2. Aadhaar Authentication Operating Model - http://uidai.gov.in/images/authDoc/d3_1_operating_model_v1.pdf
3. Aadhaar Authentication API Specifications - http://uidai.gov.in/images/FrontPageUpdates/aadhaar_authentication_api_1_6.pdf

4. Aadhaar Request OTP API Specifications - http://uidai.gov.in/images/FrontPageUpdates/aadhaar_otp_request_api_1_6.pdf

In addition, readers are highly encouraged to read the following documents to get an overall understanding of Aadhaar system:

1. UIDAI Strategy Overview - http://uidai.gov.in/UID_PDF/Front_Page_Articles/Documents/Strategy_Overveiw-001.pdf
2. The Demographic Data Standards and verification procedure Committee Report - http://uidai.gov.in/UID_PDF/Committees/UID_DDSVP_Committee_Report_v1.0.pdf
3. The Biometrics Standards Committee Report - http://uidai.gov.in/UID_PDF/Committees/Biometrics_Standards_Committee_report.pdf
4. Aadhaar Enabled Service Delivery - http://uidai.gov.in/images/authDoc/whitepaper_aadhaarenabledservice_delivery.pdf
5. Aadhaar architecture white paper - http://uidai.gov.in/images/AadhaarTechnologyArchitecture_March2014.pdf

1.2 Terminology

Readers are expected to have knowledge on general terminology used in Aadhaar authentication such as AUA, ASA, etc. before reading this section. These are available in documents mentioned in previous section.

1.3 Legal Framework

Access to Mobile Update API by trusted AUAs will be controlled via API license keys. AUAs having the access to Mobile Update API will need to get into appropriate agreement with UIDAI based on processes around Aadhaar Mobile Update services.

1.4 Objective of this document

This document provides Aadhaar Mobile Update API specification. It contains details including API data format, protocol, and security specifications.

2. Understanding Mobile Update API

This chapter describes Aadhaar Mobile Update API, its background, and usage. Technical details related to the API are provided in subsequent chapters.

2.1 Aadhaar Authentication

Aadhaar authentication is the process wherein Aadhaar Number, along with other attributes, including biometrics, are submitted online to the CIDR for its verification on the basis of information or data or documents available with it.

During the authentication transaction, the resident's record is first selected using the Aadhaar Number and then the demographic/biometric inputs are matched against the stored data which was provided by the resident during enrolment/update process. Alternatively, authentication can also be carried out on the basis of the OTP sent to the registered mobile number.

2.2 Aadhaar Mobile Update API

Mobile update API is built as a layer on top of Aadhaar authentication. This API provides a convenient mechanism for agencies to offer an easy way for resident to update their mobile number within the Aadhaar system in a secure way using biometric authentication.

> For security reasons, when using public devices, mobile update API mandates an operator authentication. For registered devices, only resident authentication is sufficient. Operator is any person with Aadhaar number assigned to operate the biometric terminal by the AUA/Sub-AUA.

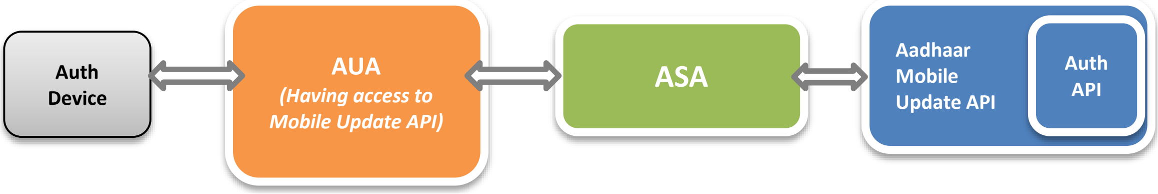
Rest of the document is technical in nature and is intended for software professionals working with applications wanting to enable their applications to support Aadhaar mobile update.

3. Aadhaar Mobile Update API

This chapter describes the API in detail including the flow, communication protocol, and data formats.

3.1 Mobile API Data Flow

Following the data flow of a typical Mobile Update.


Auth Device
AUA
(Having access to Mobile Update API)
ASA
Aadhaar Mobile Update API
Auth API

1. Front-end application initiates the mobile update transaction
2. If resident has obtained the verification code for new mobile (resident directly obtaining via SMS, resident portal, etc), then proceed to next step
    - Otherwise, front-end application invokes OTP Request API to send the verification (see later section on "Mobile Verification" for details)
    - When the OTP Request API is called, UIDAI server sends a verification code to resident's new mobile
3. If using public devices (for registered devices, this is optional), operator biometric authentication is mandated.
    - Either this API should capture Aadhaar number + biometric of operator (operator is **any person with Aadhaar number** assisting the resident to conduct update using the AUA/Sub-AUA application);
    - Or this API can send a valid operator authentication response code. If sending operator authentication response code, then it must be for a previous valid operator biometric authentication. Operator authentication must have been done within last ***n*** hours (currently UIDAI mandates within 4 hours for operator session for this API).
4. Mobile update front-end application captures Aadhaar number, consent, new mobile number, verification code, and biometrics of the resident. **In addition, optional attributes email ID and data sharing consent can also be captured and updated.**
5. Front-end device encrypts the respective PID blocks and sends to AUA server along with other necessary data required for mobile update API.
6. AUA server forms two separate Aadhaar authentication XMLs – one for resident and one for operator - using respective PID blocks.
7. Then AUA server uses that to form Mobile Update XML and signs it (if this is delegated to ASA, ASA also could form the Mobile Update XML and sign it) sends to ASA
8. ASA forwards the Mobile Update XML (if ASA forms the Mobile Update XML on behalf of AUA, ASA needs to form the Mobile Update XML, and sign it) to Aadhaar Mobile Update API
9. Aadhaar Mobile Update service within UIDAI CIDR authenticates the resident and operator using their respective authentication XMLs

10. If both resident and operator authentications are successful, UIDAI service creates the update request internally and responds with digitally signed response XML
11. ASA sends the response back to AUA for completing the transaction. If there was a failure, AUA mobile update application should show appropriate error and redo the transaction.

Mobile update updates resident UID master database within UIDAI CIDR. But, this update is done asynchronously and not instantaneous. Update will be reflected in resident master database for final use within 12 hours.

3.2 Mobile Verification

When updating a new mobile number, it is critical that the new number be verified before updating the UIDAI database. Verification is done as part of update. A verification code must be obtained and provided as part of mobile update API input. Verification code can be obtained by two ways:

- Aadhaar holder initiated – resident can obtain a verification code by sending an SMS from the new mobile to UIDAI or by going to resident portal.
- Application initiated – mobile update application can request a verification code by calling the Aadhaar OTP Request.

For residents who are not aware of how to obtain a verification code by themselves (via SMS, UIDAI mobile application, UIDAI resident portal, etc), application can call "Aadhaar OTP Request API 1.6" which, in turn, will send a verification code to the new mobile number which can be provided as part of mobile update API input.

See "**Aadhaar OTP Request API 1.6**" documentation for the URL and API parameters.

3.3 Mobile Update API Protocol

Aadhaar Mobile Update service is exposed as stateless service over HTTPS. Usage of open data format in XML and widely used protocol such as HTTP allows easy adoption by the user agencies. To support strong end to end security and avoid request tampering and man-in-the-middle attacks, it is essential that encryption of data happens at the time of capture on the capture device.

Following is the URL format for Aadhaar Mobile Update service:

```
https://<host>/mou/<ver>/<ac>/<uid[0]>/<uid[1]>/<asalk>
```

API input data should be sent to this URL as XML document using Content-Type "application/xml" or "text/xml".

3.3.1 Element Details

host – Aadhaar Mobile Update API server address. Actual production server address will be provided to ASAs. Note that production servers can only be accessed through secure leased lines. ASA server should ensure that actual URL is configurable.

Next part of the URL “mou” indicates that this is a Mobile Update API call. Ensure that this is provided.

ver – Mobile Update API version (mandatory). UIDAI may host multiple versions for supporting gradual migration. As of this specification, default production version is “1.0”.

ac – A unique code for the AUA which is assigned by UIDAI. This is an alpha-numeric string having maximum length 10.

uid[0] and **uid[1]** – First 2 digits of Aadhaar Number. Used for load-balancing.

asalk – A valid ASA license key. ASAs must send one of their valid license keys at the end of the URL. It is important that license keys are maintained safely. When adding license key to the URL, ensure it is “URL encoded” to handle special characters. If ASA is not a valid ASA, appropriate error is returned.

For all valid responses, HTTP response code 200 is used. All application error codes are encapsulated in response XML element. In the case of connection and other server errors, standard HTTP error response codes are used (4xx codes such as 403, 404, etc.). HTTP automatic redirects also should be handled by ASA server.

ASA server must send one of their valid license keys as part of the URL (see details above). Mobile Update API is enabled only for valid ASAs and only for their registered static IP addresses coming through a secure private network.

3.4 Mobile Update API: Input Data Format

Aadhaar Mobile Update API uses XML as the data format for input and output. To avoid sending unnecessary data, do not pass any optional attribute or element unless its value is different from default value. Any bad data or extra data will be rejected.

Following is the XML data format for authentication API:

```
<Mou ver=“” ts=“” ra=“” rc=“” nmn=“” mvc=”” nem=”” dsc=””>
  <Rad>base64 encoded auth XML for resident</Rad>
  <Oad uid=””>base64 encoded auth request XML or previous valid auth
response code for operator</Oad>
</Mou>
```

3.4.1 Element Details

Element: **Mou** (mandatory)

Root element of the input XML for Mobile Update API

Attributes:

- **ver** – (mandatory) version of the Mobile Update API. Currently only valid value is "1.0".
- **ts** – (mandatory) Timestamp at the time of capture of resident authentication input. This value must match "ts" attribute of "PID" block of the resident authentication packet under "Rad" elements.
- **ra** – (mandatory) Resident authentication type. Valid values are "F", "I", and "FI". This should exactly match what is captured in the PID block of the resident authentication packet.
    - Devices that capture the resident authentication PID block, should determine value of this attribute based on what is captured.
    - For example, if resident authentication is based on fingerprint, then this should be "F", if resident authentication is based on iris, then this should have value "I", and if authentication is based on fingerprint and iris, then this should be "FI".
    - If this code and actual authentication factors do not match, appropriate error is returned.
- **rc** – (mandatory) Represents resident consent for accessing the resident data from Aadhaar system. Only valid value is "Y". Mobile Update front-end application must ensure it takes an explicit "resident consent" authorizing the AUA to conduct update transaction. Only if the resident has provided the consent (in the application UI, either in self-service mode or operator should prompt the resident and get consent), this should be populated as "Y". No other values are valid.
- **nmn** – (mandatory) New mobile number of the resident. Must be a valid 10 digit number without country code (+91 is assumed as country code).
- **mvc** – (mandatory) Verification code which was sent to resident's new mobile number (see section on "Mobile Verification" earlier in this document).
- **nem** – (optional) New email address of the resident. If provided, this must be a valid email address of the resident.
- **dsc** – (optional) Resident's data sharing consent. Valid values are "Y" or "N".
    - When residents are enrolled into Aadhaar, a data sharing consent is taken. Residents can choose to consent or not to consent.
    - UIDAI allows residents to subsequently update their consent via update services.
    - This API also allows residents to update their data sharing consent within Aadhaar system during mobile update.

Element: **Rad** (mandatory)

This element contains base64 encoded Auth XML for resident. Authentication input XML must be fully compliant to Aadhaar Authentication API specification.

Element: **Oad** (mandatory if using public devices)

This element contains base64 encoded authentication request XML for operator or a valid authentication response code for a previous operator authentication.

- If fresh operator authentication request is provided, authentication input XML must be fully compliant to Aadhaar Authentication API specification.
- **NOTE: Operator is not required to be authenticated every time along with resident mobile update.**
- If this element contains just the response code of a previous operator biometric authentication, then that authentication must have been performed in last ***n*** hours (currently UIDAI policy is set to 4 hours). Application must re-authenticate operator at least in every 4 hours.

Attributes:

- **uid** – (mandatory) Aadhaar number of the operator.

It is important to note that resident authentication XML (provided under “Rad” element) MUST have its “txn” attribute value starting with “UMN:R:” as the namespace. Similarly, operator authentication XML (provided under “Oad” element) MUST have its “txn” attribute value starting with “UMN:O:” as the namespace. Otherwise, this API will throw appropriate error indicating that the transaction value is invalid.

3.5 Mobile Update API: Response Data Format

Response XML for the Mobile Update API is as follows:

```
<MouRes ret=“” code=“” txn=“” ts=“” info=”” err=“” rerr=”” oerr=”” orc=””>
  <Rar>base64 encoded auth response XML for resident</Rar>
  <Oar>If request XML contained a new operator auth request
       under “Oad” element, then this element will contain
       base64 encoded auth response XML
  </Oar>
  <Signature />
</MouRes>
```

3.5.1 Element Details

Element: **MouRes**

Attributes:

- **ret** – this is the main Mobile Update API response. It is either “y” or “n”.
- **code** – unique alphanumeric response code for Mobile Update API having maximum length 40. AUA is expected to store this for future reference for handling

any disputes. Aadhaar Mobile Update server will retain Mobile Update trail only for a short period of time as per UIDAI policy.

- **txn** – Mobile Update API transaction identifier. This is exactly the same value that is sent within the request for the resident authentication.
- **ts** – Timestamp when the response is generated. This is of type XSD dateTime.
- **info** – Additional information regarding the original request. Currently this contains the following structure:
  ```
  {SHA-256 of mobile number, SHA-256 of email ID}
  ```
- **err** – Failure error code. If Mobile Update API fails ("ret" attribute value is "n"), this attribute provides any of the following codes (for latest updates on error codes, see https://developer.uidai.gov.in/site/api_err):
  - **"M-100"** – Resident authentication failed, see "Rar" element to see details of resident authentication error.
  - **"M-110"** – Operator authentication failed, see "Oar" element to see details of operator authentication error.
  - **"M-120"** – Authentication code that was sent under "Oad" is invalid. Operator must be re-authenticated.
  - **"M-121"** – Aadhaar number of the operator does not match with the earlier authentication request for which response code was provided.
  - **"M-200"** – Update service currently not available.
  - **"M-540"** – Invalid Mobile Update XML.
  - **"M-541"** – Invalid Mobile Update API version.
  - **"M-542"** – Invalid resident consent ("rc" attribute in "Mou" element).
  - **"M-543"** – Invalid timestamp ("ts" attribute in "Mou" element)
  - **"M-544"** – Invalid resident auth type ("ra" attribute in "Mou" element does not match what is in PID block within "Rad").
  - **"M-545"** – Resident has opted-out of this service.
  - **"M-546"** – Invalid mobile verification code.
  - **"M-547"** – Invalid email address ("nem" value has invalid format).
  - **"M-546"** – Invalid value for data sharing consent ("dsc" value is invalid).
  - **"M-551"** – Invalid "Txn" namespace either for "UMN:R" or for "UMN:O"
  - **"M-569"** – Digital signature verification failed for Mobile Update XML
  - **"M-570"** – Invalid key info in digital signature for Mobile Update XML (it is either expired, or does not belong to the AUA or is not created by a well-known Certification Authority)
  - **"M-600"** – AUA is invalid or not an authorized to call this API
  - **"M-999"** – Unknown error
- **rerr** – Error code (if any) of the resident authentication. Detail response is available within "Rar" element.
- **oerr** – Error code (if any) of the operator authentication. Detail response is available within "Oar" element.
- **orc** – Authentication response code for the operator. Detail response is available within "Rar" element.

Element: **Rar**

This element contains base64 encoded version of the entire authentication API response XML for the resident authentication.

Element: **Oar**

If request had a valid operator authentication input XML (under "Oad" element), then this element contains base64 encoded version of the entire authentication API response XML for the operator authentication. If request had operator auth response XML (under "Oad" element), then this element will NOT have any value.

Element: **Signature**

This is the root element of UIDAI's digital signature. This signature can be verified using UIDAI public key. Signature complies with W3C XML signature scheme.