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 E-KYC

API SPECIFICATION - VERSION 1.0 (DRAFT)

AUGUST 2012

Table of Contents

1. Introduction

The Unique Identification Authority of India (UIDAI) has been established with the mandate of providing a Unique Identification Number (Aadhaar) to all residents of India. The UIDAI also provides the service of online authentication of identity on the basis of demographic and biometric data.

Verification of the Proof of Identity (PoI) and Proof of Address (PoA) is a key requirement for access to financial products (payment products, bank accounts, insurance products, market products, etc.), SIM cards for mobile telephony, and access to various Central, State, and Local Government services. Today, customers provide physical PoI and PoA documents. Aadhaar is already a valid PoI and PoA document for various services in the Financial, Telecom, and Government domains. In addition, the UIDAI now also proposes to provide an e-KYC service, through which the KYC process can be performed electronically. As part of the e-KYC process, the resident authorizes UIDAI (through Aadhaar authentication) to provide their basic demographic data for PoI and PoA along with their photograph (digitally signed) to service providers.

Service providers can provide a paperless KYC experience by using e-KYC and avoid the cost of repeated KYC, the cost of paper handling and storage, and the risk of forged documents. Service providers may access the Aadhaar e-KYC service from UIDAI through the e-KYC API specified in this document.

1.1 Target Audience and Pre-Requisites

This is a technical document that is targeted at software professionals who are working in the technology domain, and are interested in incorporating the Aadhaar e-KYC 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

In addition, readers are highly encouraged to read the following documents to understand the overall system:

1. UIDAI Strategy Overview - http://uidai.gov.in/UID_PDF/Front_Page_Articles/Documents/Strategy_Overview-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

1.2 Terminology

Readers are expected to be familiar with the general terminology used in Aadhaar authentication such as AUA, ASA, etc. before reading this section.

KYC User Agency (KUA): KUAs are AUAs that are eligible for the e-KYC service.

KYC Service Agency (KSA): KSAs are ASAs that are eligible to provide access to the e-KYC service through their network.

Note: All further references to AUA in the rest of this document automatically refer to KUA and similarly all references to ASA refer to KSA.

1.3 Legal Framework

UIDAI will develop necessary legal framework and processes around the Aadhaar e-KYC service. These documents will also specify KUA/KSA eligibility criteria, registration process, and the operating model.

1.4 Objective of this document

This document provides Aadhaar e-KYC API specifications. It contains details including API data format, protocol, and security specifications.

2. Understanding Aadhaar e-KYC service

This chapter describes Aadhaar e-KYC 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 the 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 e-KYC API Usage

The e-KYC API can be used (only with the explicit authorization of the resident) by an agency to obtain latest resident demographic data and photo data from UIDAI. There are primarily two scenarios under which this API may be used:

1. **New customer/beneficiary**:
    a. In this case, KUA should use capture resident authentication data, invoke e-KYC API through a KSA network;
    b. The KYC data returned within the response of the e-KYC API is digitally signed by UIDAI and can be used for electronic audit at a later stage; and
    c. Using the resident data obtained through this KYC API, the agency can service the customer.

2. **Existing customer/beneficiary**
    a. In this case, KUA should use capture resident authentication data, invoke e-KYC API through a KSA network;
    b. The KYC data returned within the response of the KYC API is digitally signed by UIDAI and can be used for electronic audit;
    c. Since the resident is already a customer/beneficiary, the agency can use a simple workflow to approve the Aadhaar linkage by comparing data retrieved through the e-KYC API against what is on record (in paper or electronic form); and
    d. Once verified, the existing customer/beneficiary record can be linked to the Aadhaar number.

For both scenarios, the same e-KYC API is used to obtain the KYC data after successful resident authentication. Technical details for invoking the API are provided in subsequent chapters of this document.

2.3 Conclusion

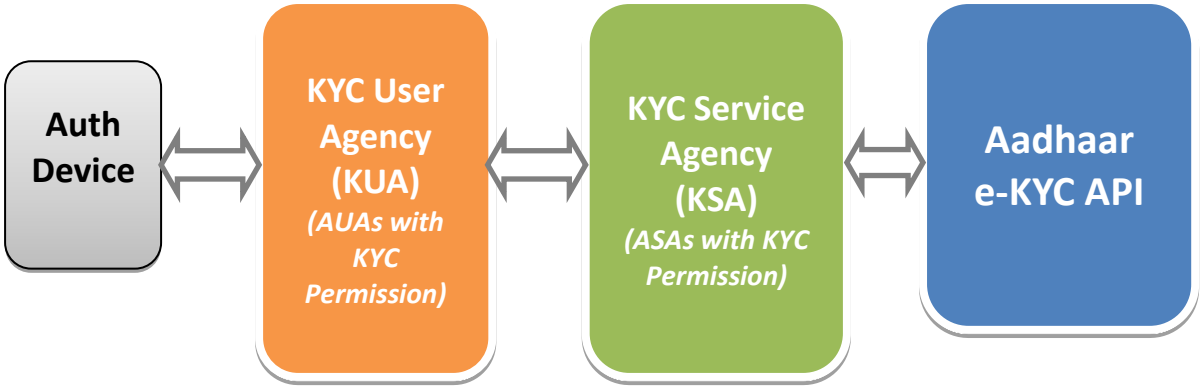
The Aadhaar e-KYC API provides a convenient mechanism for agencies to offer an electronic, paper-less KYC experience to Aadhaar holders. The e-KYC service provides simplicity to the resident, while providing cost-savings from managing and processing paper documents to the KUA.

3. Aadhaar e-KYC API

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

3.1 e-KYC API Data Flow

Following the data flow of a typical KYC API call from left to right and back.



1. KYC front-end application (depicted as auth device in diagram above) captures Aadhaar number + biometric/OTP of resident and forms the encrypted PID block
2. KUA forms the Auth XML using the PID block, signs it, and uses that to form KYC XML and signs it (if this is delegated to KSA, KSA also could form the KYC XML and sign it) sends to KSA
3. KSA forwards the KYC XML (if KSA forms the KYC XML on behalf of KUA, KSA needs to form the KYC XML, and sign it) to Aadhaar KYC API
4. Aadhaar KYC service authenticates the resident and if successful responds with digitally signed and encrypted demographic and photograph in XML format
5. Demographic data and photograph in response is encrypted with either KSA or KUA public key (based on the setup at CIDR)
6. KSA sends the response back to KUA enabling paper-less electronic KYC

3.2 API Protocol

Aadhaar KYC 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 KYC service:

```
https://<host>/kyc/<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".

For security reason data collected for Aadhaar KYC must not be stored in the devices or log files. It's essential for ASA and AUA to maintain audit records for all the authentication request metadata along with the response.

3.2.1 Element Details

host – Aadhaar KYC 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 "kyc" indicates that this is a KYC API call. Ensure that this is provided.

ver – KYC API version (optional). If not provided, URL points to current version. 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 KSA, 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). KYC API is enabled only for valid ASAs and only for their registered static IP addresses coming through a secure private network.

3.3 e-KYC API: Input Data Format

Aadhaar KYC 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:

```
<Kyc ver="" ts="" ra="" rc="">
  <Rad>base64 encoded fully valid Auth XML for resident</Rad>
  <Signature/>
</Kyc>
```

3.3.1 Element Details

Element: **Kyc** (mandatory)
Root element of the input XML for KYC API

Attributes:

- **ver** – (mandatory) version of the KYC API. Currently only valid value is "1.0".
- **ts** – (mandatory) Timestamp at the time of capture of authentication input. This value must match "ts" attribute of "PID" block of the resident authentication packet under "Rad" element.
- **ra** – (mandatory) Resident authentication type. Valid values are "F", "I", "O", "FI", "FO", "IO", and "FIO". 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 OTP, then this should have value "O", and if authentication is based on fingerprint and OTP, then this should be "FO". 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". KYC front-end application must ensure it takes an explicit "resident consent" authorizing the AUA to retrieve the resident data. 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.

Element: **Rad** (mandatory)
This element contains base64 encoded Auth XML for resident. Authentication input XML must be fully compliant to Aadhaar Authentication API specification.

It is important to note that resident authentication XML (provided under "Rad" element) MUST have its "txn" attribute value starting with "UKR:" as the namespace for KYC API. Otherwise, this API will throw appropriate error indicating that the transaction value is invalid.

Element: **Signature** (mandatory)

- The request XML should be digitally signed for message integrity and non-repudiation purposes.
- Digital signing should always be performed by the entity that creates the final request XML
    - AUA can digitally sign after forming the API input XML. This is almost always the case. In such cases, AUA ensures the message security and integrity between AUA servers and its client applications.
    - ASA can digitally sign the request XML if it is a domain-specific aggregator and forms the request XML on behalf of the AUA. In such cases, ASA and AUA ensure the message security and integrity between their servers.
- Procuring digital signature certificates:
    - It should be procured from a valid certification authority as per Indian IT Act (see http://cca.gov.in/rw/pages/faqs.en.do#thecaslicensedbythecca)
    - Digital certificates have two parts:
        - X.509 certificate representing public key.
        - Private Key which is used for digital signing. Private Key should be stored securely and is the responsibility of the owner of the certificate to ensure that it is not compromised.
    - It should be a class II or class III certificate.
    - X.509 certificate contains information about the owner of the certificate; in this case it will be details of the person and the organization to which he/she belongs. UIDAI server checks to ensure that certificate belongs to the ASA or AUA organization. Hence, it is mandatory that "O" attribute of "Subject" in the X.509 certificate matches the name of the organization.
- Digital signing of request XML
    - XML digital signature algorithm as recommended by W3C.
    - Signature should include key info element that contains X.509 certificate details. This is needed for UIDAI server to validate the signer.
- Verification of digital signature by UIDAI servers. UIDAI server validates the signature in the following sequence:
    - Checks if the signature element is present. If not, it throws an error.
    - If signature element is present, then it validates if the certificate is issued by one of the valid certification authority. If not valid, throws error.
    - If it is a valid certificate, then it validates whether the "O" attribute in the X.509 certificate's subject matches the AUA organization name. If yes, proceeds with API logic.
    - If it does not match AUA organization name, it checks configuration to see if ASA is allowed to sign on behalf of that AUA. If not, throws error.

- If ASA is allowed to sign on behalf of that AUA, it checks whether the "O" element of the certificate matches with the organization name of the ASA. If not, throws error.
- If it matches, it proceeds with API logic.
- In future, UIDAI may choose to conduct additional validations against a white listed certificated within UIDAI database.

3.4 e-KYC API: Response Data Format

Since KYC API provide resident data based on authentication (hence authorizing UDIAI to share his/her data with the AUA), entire response is encrypted.

Response XML for the KYC API is as follows:

```
<Resp status="">encrypted and base64 encoded "KycRes" element</Resp>
```

Element Resp is just a container for keeping encrypted KYC data signed by UIDAI. This element contains just one attribute "status" which indicates high level status of the API call. It can have values "0" or "-1". If the status is "0", it means that the encrypted data contained within the "Resp" element is valid. If it contains "-1", it means the data should not be decrypted and used.

Value of the "Resp" element is base64 encoded version of the encrypted "KycRes" element. Note that "KysRes" is encrypted using either KSA public key or KUA public key based on the KSA/KUA setup on UIDAI server. Once decoded and decrypted, "KycRes" has the following structure:

```
<KycRes ret="" code="" txn="" err="" ts="">
  <Rar>base64 encoded fully valid Auth response XML for resident</Rar>
  <UidData uid="">
    <Poi name="" dob="" gender="" phone="" email=""/>
    <Poa co="" house="" street="" lm="" loc="" vtc=""
         subdist="" dist="" state="" pc="" po=""/>
    <Pht>base64 encoded JPEG photo of the resident</Pht>
  </UidData>
  <Signature />
</KycRes>
```

3.4.1 Element Details

Element: **KycRes**

Attributes:

- **ret** – this is the main KYC API response. It is either "y" or "n".
- **code** – unique alphanumeric response code for KYC API having maximum length 40. AUA is expected to store this for future reference for handling any disputes. Aadhaar KYC server will retain KYC trail only for a short period of time as per UIDAI policy.

- **txn** – KYC API transaction identifier. This is exactly the same value that is sent within the request.
- **ts** – Timestamp when the response is generated. This is of type XSD dateTime.
- **err** – Failure error code. If KYC 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):
  - **"K-100"** – Resident authentication failed
  - **"K-200"** – Resident data currently not available
  - **"K-540"** – Invalid KYC XML
  - **"K-541"** – Invalid KYC API version
  - **"K-542"** – Invalid resident consent ("rc" attribute in "Kyc" element)
  - **"K-543"** – Invalid timestamp ("ts" attribute in "Kyc" element)
  - **"K-544"** – Invalid resident auth type ("ra" attribute in "Kyc" element does not match what is in PID block)
  - **"K-545"** – Resident has opted-out of this service
  - **"K-551"** – Invalid "Txn" namespace
  - **"K-569"** – Digital signature verification failed for KYC XML (means that authentication request XML was modified after it was signed)
  - **"K-570"** – Invalid key info in digital signature for KYC XML (it is either expired, or does not belong to the AUA or is not created by a well-known Certification Authority)
  - **"K-600"** – AUA is invalid or not an authorized KUA
  - **"K-601"** – ASA is invalid or not an authorized KSA
  - **"K-602"** – KUA encryption key not available
  - **"K-603"** – KSA encryption key not available
  - **"K-999"** – Unknown error

Element: **Rar**

This element contains base64 encoded version of the entire authentication API response XML (AuthRes element – see Authentication API specification document) for the resident authentication.

Element: **UidData**

This element and its sub-elements contain demographic data and photograph of the resident as per Aadhaar system.

Attributes:

- **uid** – 12-digit Aadhaar number of the resident

Element: **Poi**

This element contains resident's name within Aadhaar system.

Attributes:

- **name** – Name of the resident
- **dob** – Date of birth of the resident in YYYY-MM-DD format

- **gender** – Gender of the resident. Valid values are M (male), F (female), and T (transgender)
- **phone** – Mobile phone if any
- **email** – Email address if any

Element: **Poa**

This element contains resident's address within Aadhaar system.

Attributes:

- **co** – "Care of" person's name if any
- **house** – House identifier if any
- **street** – Street name if any
- **lm** – Landmark if any
- **loc** – Locality if any
- **vtc** – Name of village or town or city
- **subdist** – Sub-District name
- **dist** – District name
- **state** – State name
- **pc** – Postal pin code
- **po** – Post Office name if any

Element: **Pht**

This element contains base64 encoded JPEG photo of the resident.

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.

For more details, refer: http://www.w3.org/TR/xmldsig-core/

4. Appendix

4.1 Related Publications

Demographic Data Standards	http://uidai.gov.in/UID_PDF/Committees/UID_DDSVP_Committee_Report_v1.0.pdf
Aadhaar Authentication API Specification	http://uidai.gov.in/images/FrontPageUpdates/aadhaar_authentication_api_1_6.pdf
XML Signature	http://www.w3.org/TR/xmldsig-core/