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 (FINAL)

JANUARY 2014

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. Note that authentication AUA and sub-AUA automatically becomes KUA and Sub-KUA in e-KYC. From a contract perspective, only KUA needs to have a contract with UIDAI (quite like authentication) and sub-KUAs are entities under KUA.

IMPORTANT NOTE: All data sharing downstream from KSA to KUA to their sub-organizations must be done explicitly through "informed resident consent". No resident data must be shared with other organizations without explicit consent of the resident. If engaging sub-KUAs then KUA must sign data protection, security, and other contractual obligations with each sub-KUA.

1.3 Legal Framework

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

1.4 Objective of this document

This document provides Aadhaar e-KYC API technical 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.

All biometric/OTP (single or multi-factor) authentication schemes are valid for e-KYC service too.

2.2 Eliminating Photo copies and Costly, Insecure Paperwork

Aadhaar is now a valid Proof of ID (PoI) and proof of Address (PoA) for most services is fast being the key document for banking, telco, insurance, Govt subsidy programs, Passport, PAN card, etc. Considering the large number of Aadhaar holders in India and the ability to uniquely authenticate all Aadhaar holders, more and more services are accepting Aadhaar for their service delivery.

Traditionally all Know Your Customer processes and verification of PoI and PoA are done using copies of PoI/PoA documents. It is commonplace to provide self-attested photocopies of these documents every time a bank account is opened, SIM card issued, insurance is purchased, etc.

Aadhaar e-KYC service eliminates the need for the resident to provide photo copy of Aadhaar letter and instead resident can simply authenticate and authorize UIDAI to share the Aadhaar letter data in electronic and secure (encrypted and digitally signed) fashion instead of leaving paper copies of the identity document everywhere.

Eliminating paper verification and storage removes fraud, fake document usage, paper storage cost, manual audit cost, etc and makes entire process seamless, auditable, and secure. And most importantly this allows services such as bank account opening etc done using a mobile handheld in rural environments without worrying about the authenticity of papers and trustworthiness of front end touch points.

2.3 Aadhaar e-KYC API Usage

The e-KYC API can be used (ONLY with the explicit authorization of the resident via Aadhaar biometric/OTP authentication) by an agency (KUA) to obtain electronic copy of Aadhaar letter. 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. Electronic copy of Aadhaar letter returned as part of the e-KYC API response is encrypted and digitally signed by UIDAI and can be used for electronic audit at a later stage; and
    c. This eliminates collecting photocopy of Aadhaar letter from resident. Using the electronic Aadhaar letter data obtained through this e-KYC API, the agency can create new customer account and 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. Electronic copy of Aadhaar letter returned as part of the e-KYC API response is encrypted and digitally signed by UIDAI and can be used for electronic audit at a later stage;
    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 within UA database (in paper or electronic form); and
    d. Once verified, the existing customer/beneficiary record can be linked to the Aadhaar number and transaction trail can be stored for audit.

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

2.4 Conclusion

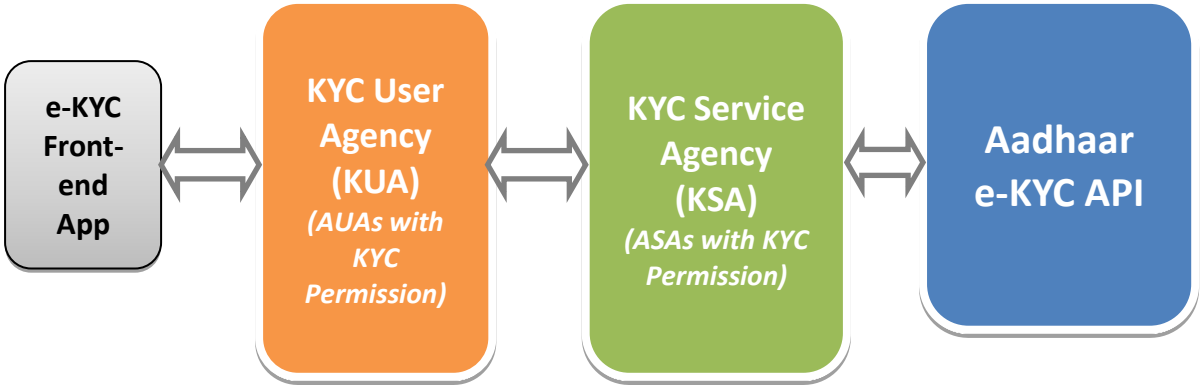
The Aadhaar e-KYC API provides a convenient mechanism for agencies to offer an electronic, paper-less KYC experience to Aadhaar holders eliminating insecure and costly paper process that exist today. The e-KYC service provides simplicity to the resident, while providing cost-savings from managing and 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 e-KYC API call from left to right and back.



1. e-KYC front-end application 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, uses that to form final e-KYC input XML and sends to KSA (if this is delegated to KSA, KSA also could do the input XML creation and signing)
3. KSA forwards the KYC XML to Aadhaar e-KYC service
4. Aadhaar KYC service authenticates the resident and if successful responds with digitally signed and encrypted XML containing resident's latest demographic and photograph information
5. Demographic data and photograph in response, by default, is encrypted with KUA public key
   - If KUA key is NOT available within CIDR, KSA public key will be used provided KSA is approved to do so.
   - If "de" attribute is used in input XML to delegate decryption to KSA (this can be done at transaction level), then KSA key will be used to encrypt response, provided KSA is approved to do so (this option allows KUAs to dynamically delegate decryption to KSA based on their relationship and setup with KSA)
6. KSA sends the response back to KUA enabling paper-less electronic KYC. KUA/KSA (based on decryption setup) should keep the digitally signed XML as-is (equivalent to physical Aadhaar letter) for audit purposes.

Note: Digital signature in input (KUA or KSA) is independent of response data encryption. Input signature is used by UIDAI server to assert authenticity of the requesting agency where as response encryption is to protect resident data.

3.2 API Protocol

Aadhaar e-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 input PID data happens at the time of capture on the capture device.

Following is the URL format for Aadhaar e-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 PID data collected for Aadhaar e-KYC must NOT be stored on any device or server. It's essential for KSA and KUA to maintain audit records for all the authentication request metadata along with the response and protect the PII data.

3.2.1 Element Details

host – Aadhaar e-KYC API server address. Actual production server address will be provided to KSAs. Note that production servers can only be accessed through secure leased lines. KSA server should ensure that actual URL is configurable.

Next part of the URL "kyc" indicates that this is a e-KYC API call. Ensure that this is provided.

ver – e-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 (KUA and AUA codes are same since KUA is an AUA having access privilege to e-KYC service) 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**.

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). E-KYC API is enabled only for valid KSAs 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="" mec="" lr="" de="">
  <Rad>base64 encoded fully valid Auth XML for resident</Rad>
</Kyc>
```

3.3.1 Element Details

Element: **Kyc** (mandatory)
Root element of the input XML for e-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.
    - **If this value is not matching with PID ts, then, an error will be generated**.
    - Front-end application on the device must send the PID "ts" value to KUA server to ensure PID capture timestamp is used "as-is" within this XML. This is to ensure authentication input cannot be independently used for e-KYC later.
- **ra** – (mandatory) Resident authentication type. Valid values are "F", "I", "O", "FO", "IO", "FI", and "FIO". Front end e-KYC application that capture the resident authentication PID block, should determine value of this attribute based on what is captured. For example, if resident authentication uses fingerprints, then this

should be "F", if both fingerprint and OTP are used this should be "FO", and so on (see table below for all values). This and actual authentication factors within PID block do not match, an error is returned.

Authentication Factors Used	Value of "ra"
Fingerprints	F
Iris	I
OTP	O
Fingerprints and OTP	FO
Iris and OTP	IO
Fingerprints and Iris	FI
Fingerprints, Iris, and OTP	FIO

- **rc** – (mandatory) Represents resident's explicit consent for accessing the resident's identity and address data from Aadhaar system. Only valid value is "Y". **If resident does not provide this explicit consent, application SHOULD NOT access resident data using this API**.
- **mec** – (optional) Represents resident's explicit consent for accessing the mobile number and email address of the resident from Aadhaar system. Valid values are "Y" and "N". Default value is "N" (by default, this API does not return mobile and email data).
- **lr** - (optional) Flag indicating if AUA application require local language data in addition to English. Valid values are "Y" and "N". Default value is "N" (by default, this API does not return local Indian language data).
- **de** – (optional) Flag indicating if KUA is delegating decryption to KSA. If this flag is set to "Y", then KSA public key will be used to encrypt eKYC response XML instead of KUA key provided KSA is allowed to do so.
    - **This is OPTIONAL attribute and hence should be used ONLY when KUA requires to change the default option based on KSA setup. <u>This option works only if KSA is approved to do decryption</u>**.
    - By default, KUA public key is always used to encrypt e-KYC response.
    - If KUA key is NOT available in CIDR, KSA key will be used to encrypt provided KSA is authorized to do so.
    - A dynamic option of setting "de" attribute to "Y" allows KUA to make this choice at transaction level based on the KSA they use for e-KYC service.

E-KYC front-end application **must ensure it takes an "explicit informed resident consent"** authorizing the KUA to retrieve the resident data. E-KYC Application should not hard-code values for "rc" and "mec" under any circumstances and should ensure that both consents are taken explicitly through the application UI.

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 "UKC:" as the namespace for KYC API. Otherwise, this API will throw appropriate error indicating that the transaction value is invalid.

Any valid Authentication API version and features can be used while invoking e-KYC. Only restriction being that the prefix of "txn" attribute value of the authentication input XML (authentication namespace) must start with "UKC:".

IOMPORTANT NOTE: Digital Signature at eKYC XML level is optional

- The e-KYC request XML may be digitally signed for message integrity and non-repudiation purposes.

3.4 e-KYC API: Response Data Format

Resident data as part of the response based on successful authentication (thus resident authorizing UDIAI to share his/her data with the KUA/KSA) is fully encrypted using KUA public key (or KSA public key if KUA delegates it to KSA).

Response XML for the KYC API is as follows:

```
<Resp status="" ko="" ret="" code="" txn="" ts="" err="">encrypted and
base64 encoded "KycRes" element</Resp>
```

Element:

- **Resp** - container for keeping encrypted e-KYC response. Value of the "Resp" element is base64 encoded version of the encrypted "KycRes" element (see "KycRes" element description later).

Attributes:

- **status** - 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.
- **ko** – This attribute contains either value "KUA" or "KSA" or "". If response is encrypted with KUA key, this will have value "KUA", otherwise, if it is encrypted with KSA key, this will have value "KSA". If there were any errors (when "status" is "-1"), this attribute will have blank value.
- **ret, code, txn, ts, err** – These attributes are exactly same as what is inside the encrypted block. See "KycRes" element and its attribute descriptions below. **These attributes are also made available at this element for KSA to have audit capability even when the actual response is encrypted with KUA key**.

Note: As explained before, "KycRes" element is encrypted using the following logic:

1. By default, KUA public key is used to encrypt response data
2. If "de" attribute in input XML is set to "Y" or if KUA public key is not available in CIDR, KSA public key is used to encrypt, provided KSA is approved to do so.
3. If neither KUA nor KSA public keys are available in CIDR, an error is generated.

Once decoded and decrypted, "KycRes" has the following structure:

```
<KycRes ret="" code="" txn="" ts="" ttl="" actn="" err="">
  <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=""/>
    <LData lang="" name="" 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 e-KYC API having maximum length 40. AUA is expected to store this for future reference for handling any disputes. Aadhaar KYC server will retain e-KYC trail only for a short period of time as per UIDAI policy.
- **txn** – e-KYC API transaction identifier. This is exactly the same value that is sent within the request XML.
- **ts** – Timestamp when the response is generated. This is of type XSD dateTime.
- **ttl** – "*Time To Live*" for demographic data within AUA system. AUAs may not use the resident data obtained through this API beyond this time and should use this API to obtain latest resident data.
    - It is important to understand that demographic information changes from time to time (address change, mobile number change, etc.).
    - AUAs should build applications understanding the nature of this data and ensure that they use this API from time to time to obtain latest KYC data of the resident.
- **actn** – (optional). This attribute may or may not exist in response. This attribute will have specific action codes (published from time to time) meant for future purposes to be shown to resident/operator.
    - **This attribute MUST be sent to front-end application by KSA and KUA to ensure action and corresponding message is displayed to resident/operator**.

- **err** – Failure error code. If e-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 e-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-550"**- Invalid Uses Attribute
  - **"K-551"** – Invalid "Txn" namespace
  - "**K-552**" = Invalid License key
  - **"K-569"** – Digital signature verification failed for e-KYC XML
  - **"K-570"** – Invalid key info in digital signature for e-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-604**" – KSA Signature not allowed
  - **"K-605"** – Neither KUA key nor KSA encryption key are available
  - "**K-955**" – Technical Failure
  - **"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 DD-MM-YYYY format
- **gender** – Gender of the resident. Valid values are M (male), F (female), and T (transgender)
- **phone** – Mobile phone if any based on "mec" attribute
- **email** – Email address if any based on "mec" attribute

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: **LData**

This element contains resident's name and address in local Indian language which was used while last data update. This is returned only if "lr" attribute in the API input XML is set to "Y".

Attributes (all data in Indian local language):

- **lang** – Local language code (see table below)
- **name** – Name of the resident
- **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

Language	Language code
Assamese	01
Bengali	02
Gujarati	05
Hindi	06
Kannada	07
Malayalam	11
Manipuri	12
Marathi	13
Oriya	15
Punjabi	16
Tamil	20
Telugu	21
Urdu	22

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/