Unique Identification Authority of India (UIDAI)
Government of India (GoI)
Bangla Sahib Road, Behind Kali Mandir, Gole Market
New Delhi 110001

AADHAAR E-KYC

API SPECIFICATION - VERSION 2.5

AUGUST 2019

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 this API and avoid the cost of repeated KYC, the cost of paper handling and storage, and the risk of forged documents.

1.1 Target Audience and Pre-Requisites

This is a technical document that is targeted at software professionals who are incorporating the Aadhaar e-KYC API into their applications.

Readers should also read the following related documents for complete understanding.

1. Aadhaar Authentication API - http://uidai.gov.in/images/resource/aadhaar_authentication_api_2_5.pdf
2. Aadhaar Registered Devices Specification - https://uidai.gov.in/images/resource/Aadhaar_Registered_Devices_2_0_4.pdf
3. Aadhaar Request OTP API - http://uidai.gov.in/images/resource/aadhaar_otp_request_api_2_5.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.

Note: All further references to AUA in the rest of this document automatically refer to KUA. From a contract perspective, only KUA needs to have a contract with UIDAI.

1.3 Legal Framework

The Aadhaar (Targeted Delivery of Financial and Other Subsidies, Benefits and Services) Act 2016 was published in gazette notification on March 26, 2016. The Act is to provide for, as a good governance, efficient, transparent, and targeted delivery of subsidies, benefits and services to Aadhaar number holders. A gazette notification was issued by Central Government on 12th July 2016 to establish UIDAI as an Authority and operationalize certain provisions of Aadhaar Act 2016. Authentication regulations are also published under this Act. These documents specify legal framework for authentication usage, AUA/ASA engagements, audits, and other details. Detailed partner documents are also published. These documents are available athttps://uidai.gov.in/ecosystem/authentication-devices-documents/authentication-documents.html.

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 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, Government subsidy programs, Passport, PAN card, etc. Considering the large number of Aadhaar number holders in India and the ability to uniquely authenticate all Aadhaar number holders, more and more services are accepting Aadhaar for their service delivery.

Traditionally all “Know Your Customer (KYC)” 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.

2.2 Limited e-KYC

This allows agencies to do paperless KYC process without access to Aadhaar number thus significantly enhancing the privacy within Aadhaar system.

UIDAI will categorize all AUAs into two categories – “Global AUAs” and “Local AUAs”. Once this scheme is fully implemented, ONLY Global AUAs will have access to e-KYC with Aadhaar number, while all other agencies will only have access to “Limited KYC with masked Aadhaar Number”.

a. **Global AUAs**: UIDAI from time to time will evaluate AUAs based on the laws governing them and categorize them as “Global AUAs”. Only such agencies will have access to Full e-KYC (with Aadhaar number) and will have the ability to store Aadhaar number within their system.
b. **Local AUAs**: All AUAs who are not categorized under “Global AUAs” will automatically be categorized as “Local AUAs”. Such entities will ONLY have

access to “Limited KYC” and will NOT be allowed to store Aadhaar number within their systems. Since every agency using authentication and Limited KYC can get agency specific UID Token, that can be used within their systems to uniquely identify their customers. UIDAI reserves the right to determine, in addition to UID Token, what demographic fields need to be shared with the Local AUAs depending upon its need.

2.3 Aadhaar e-KYC API Usage

The e-KYC API (Full or Limited KYC) 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 ASA 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 ASA 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

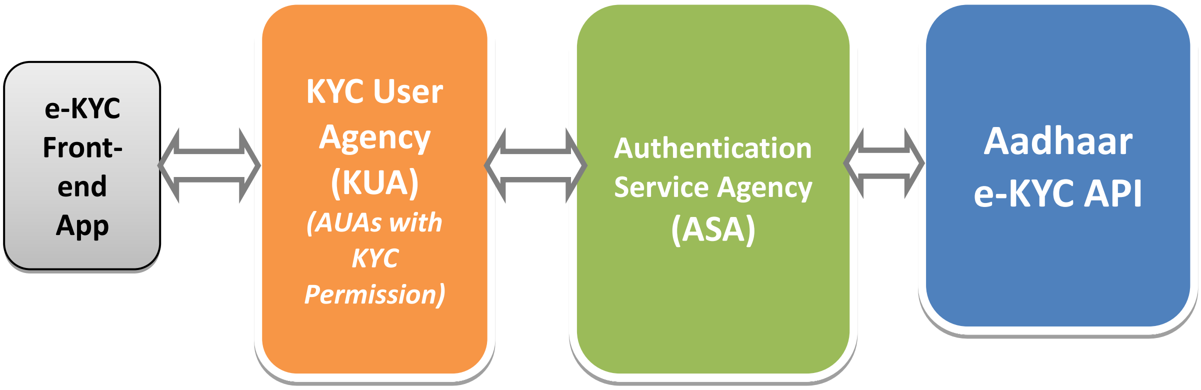
The Aadhaar e-KYC API provides a convenient mechanism for agencies to offer an electronic, paper-less KYC experience to Aadhaar number holders eliminating insecure and costly paper process that exist today.

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.


e-KYC Front-end App
KYC User Agency (KUA)
(AUAs with KYC Permission)
Authentication Service Agency (ASA)
Aadhaar e-KYC API

1. KUA application captures Aadhaar number (or Virtual ID or UID Token or Encrypted Aadhaar Number in future) + biometric/OTP of resident and forms the encrypted PID block (see Authentication API 2.5 for details)
2. KUA forms the Auth XML using the PID block, signs it, uses that to form final e-KYC input XML and sends to ASA (if this is delegated to ASA, ASA also could do the input XML creation and signing)
3. ASA 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. E-KYC response (containing demographic data and photograph), by default, is encrypted with KUA public key
    - If KUA key is NOT available within CIDR, ASA public key will be used provided ASA is approved to do so.
    - If "de" attribute is used in input XML to delegate decryption to ASA (this can be done at transaction level), then ASA key will be used to encrypt response, provided ASA is approved to do so. This facility is subject to UIDAI approval.
6. ASA sends the response back to KUA enabling paperless electronic KYC.

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

3.2 API Protocol

Aadhaar e-KYC service is exposed as stateless service over HTTPS.

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 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 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 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 "**2.5**".

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. When VID, UID Token, or encrypted Aadhaar number (future) is used, pass "0" and "0" for these.

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.

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="" ra="" rc="" lr="" de="" pfr="">
  <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 “2.5””.
- **ra** – (mandatory) Resident authentication type. Valid values are “F”, “I”, “O”, “P” or any combination of these. 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.
- **rc** – (mandatory) Represents resident’s explicit consent for accessing the resident’s identity and address data from Aadhaar system. Only valid value is “Y”. Without explicit consent of the Aadhaar number holder application should not call this API.
- **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 ASA. If this flag is set to “Y”, then ASA public key will be used to encrypt e-KYC response XML instead of KUA key provided ASA is allowed to do so. This facility is subject to UIDAI approval.
    - **This is OPTIONAL attribute and hence should be used ONLY when KUA requires to change the default option based on ASA setup. This option works only if ASA is approved to do decryption**.
    - By default, KUA public key is always used to encrypt e-KYC response.
    - If KUA key is NOT available in CIDR, ASA key will be used to encrypt provided ASA 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 ASA they use for e-KYC service.
- **pfr** – (optional) Print format request flag for retrieving E-Aadhaar document in PDF format as part of response . Only valid values are "Y" and "N". If "Y" is passed the print format is returned in the response in addition to XML.

Element: **Rad** (mandatory)

This element contains base64 encoded Auth XML for resident. Authentication input XML must be fully compliant to Aadhaar Authentication API specification. In the case of registered devices, “wadh” value within PID block must be set. See important note below.

> 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:”**.

The e-KYC request XML may be digitally signed for message integrity and non-repudiation purposes. **Digital Signature at e-KYC XML level is optional**.

<u>IMPORTANT NOTE:</u>

- In the case of registered devices (not needed for public devices), KUA application MUST create the “wadh” value as below and use it while forming PID block. UIDAI eKYC server will validate the wadh value within PID block and if not valid, reject the API call with an error.

```
wadh = SHA-256(ts+ver+ts+ra+rc+lr+de+pfr)
```

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/) is fully encrypted using KUA public key (or ASA public key if KUA delegates it to ASA).

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", "ASA" or "" based on whose key was used to encrypt. 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 ASA to have audit capability even when the actual response is encrypted with KUA key**.

<u>Note</u>: As explained before, "KycRes" element is encrypted using the following logic:

1. By default, KUA public key is used to encrypt the AES key which had encrypted the response data.
2. If "de" attribute in input XML is set to "Y" and if KUA public key is not available in CIDR, ASA public key is used to encrypt, provided ASA is approved by UIDAI to do so.
3. If neither KUA nor ASA public keys are available in CIDR, an error is generated.
4. Please note the below new encryption process for OTP based e-KYC transactions which will be enabled by UIDAI in future under intimation to ecosystem partners. Availability of below process can be recognised from the eKYC header data through a new flag which will be intimated later.
    a. An 8 digit authenticator code (AC) will be sent to the resident mobile number.
    b. In CIDR, the Authenticator Code will be appended to the AES key (K0) and the SHA-256 of the combination (K1) is used to encrypt the response data. Please note K1=SHA256(K0+AC).
    c. K0 is encrypted with KUA/ASA public key as the case may be, and appended in the e-KYC response header.
    d. The decrypting agency may obtain the K0 after decrypting the header data with his private key (to get K0), append AC, compute SHA-256 of the combination and use the resulting string to decrypt the response data to obtain encoded KycRes element.

NOTE: New encryption process for OTP based KYC transactions will be done in future under intimation to all partner entities.

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="" tkn=””>
    <Poi name="" dob="" gender="" />
    <Poa co="" house="" street="" lm="" loc="" vtc=""
         subdist="" dist="" state="" country="" pc="" po=""/>
    <LData lang="" name="" co="" house="" street="" lm="" loc="" vtc=""
         subdist="" dist="" state="" country="" pc="" po=""/>
    <Pht>base64 encoded JPEG photo of the resident</Pht>
    <Prn type="pdf">base64 encoded signed Aadhaar letter for printing</Prn>
  </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 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**-514” – Invalid UID Token Used.
  - **“K**-515” – Invalid VID used.
  - **“K**-516” – Invalid ANCS Token used.
  - **“K**-517” – VID used is expired.
  - **“K**-519” – Invalid Authenticator Code.
  - **“K-540”** – Invalid KYC XML
  - **“K-541”** – Invalid e-KYC API version

- **"K-542"** – Invalid resident consent ("rc" 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. This feature is not implemented currently.
- "**K-546**" – Invalid value for "pfr" attribute
- "**K-547**" – Invalid value for "wadh" attribute within PID block
- **"K-550"**- Invalid Uses Attribute
- **"K-551"** – Invalid "Txn" namespace
- "**K-552**" – Invalid KUA License key
- "K-553" – KUA License key Expired.
- **"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-571" – Technical error while signing the eKYC response.
- **"K-600"** – AUA is invalid or not an authorized KUA
- **"K-601"** – ASA is invalid or not an authorized ASA
- **"K-602"** – KUA encryption key not available
- **"K-603"** – ASA encryption key not available
- "**K-604**" – ASA Signature not allowed
- **"K-605"** – Neither KUA nor ASA encryption key is available
- "**K-955**" – Technical Failure internal to UIDAI.
- "K-956" – Technical error while generating the PDF file.
- **"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** – Full Aadhaar Number in case of Full KYC or Masked Aadhaar number showing last 4 digits in case of Limited KYC.
- **tkn –** Agency specific UID token of the resident generated at UIDAI back end. This will be the same value available in info block of authentication API response**.**

Element: **Poi**

This element contains resident's name within Aadhaar system.

Attributes:

- **name** – Name of the resident

- **DoB/YoB –** Date of birth / Year of birth of the resident in DD-MM-YYYY/YYYY format respectively.
- **gender** – Gender of the resident. Valid values are M (male), F (female), and T (transgender)

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
- **country** – Country 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
- **country** – Country 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: **Prn**

This element contains base64 encoded e-Aadhaar PDF of the resident in line with the XML and according to Limited or Full KYC. This PDF is digitally signed. UID token or virtual ID will not be part of this printable format. This is useful for applications where a paper print is still needed. Application providers are highly encouraged to move away from the paper printing and instead store and use the digitally signed XML data which is part of the response.

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 Changes in Version 2.5 from Version 2.1

New (2.5)
XML/API Version changed from 2.1 to 2.5
Concept of Global and local AUAs introduced.
API allows VID and UID token in addition to the Aadhaar Number.
Response contain UID token field as tkn attribute in addition to masked/full UID.
Limited KYC is introduced for local KUAs with limited fields as approved by UIDAI.
Masked Aadhaar is provided in response to KYC request for Local KUAs.
New (future) encryption process for OTP based e-KYC transactions.