eSign API Specifications

Version 1.0

July 2015

Controller of Certifying Authorities
Department of Electronics and Information Technology
Ministry of Communications and Information Technology

Document Control

Document Name	eSign API Specifications
Status	Release
Version	1.0
Last update	15 July 2015
Document Owner	Controller of Certifying Authorities, India

Table of Contents

1. Introduction

Information Technology Act, 2000 grants legal recognition to electronic records and electronic signatures. IT Act,2000 provides that where any law requires that information or any other matter shall be authenticated by affixing signature then notwithstanding anything contained in the law, such requirement shall be deemed to be fulfilled if such information is authenticated by means of electronic signatures affixed in a manner prescribed by the Central Government. Under the IT Act, 2000, 'Electronic signatures' means authentication of an electronic record by a subscriber by means of electronic technique specified in second schedule and includes Digital signatures. Digital Signature means authentication of any electronic record by a subscriber by means of procedure specified in Section 3 of the IT Act, 2000.

The Controller exercises supervision over activities of Certifying Authorities and certifies public keys of certifying authorities. The Certifying Authorities are granted licence under the IT Act, 2000 by the Controller to issue Digital Signature Certificates. Any person can make an application to Certifying Authority for issue of an Electronic signature Certificate in such form as may be prescribed by the Central Government. For issuance of Digital Signature Certificates, the applicant's Personal identity, address and other details to be included in the DSC need to be verified by CAs against an identity document. For class III, physical presence of the individual is also required. Digital signatures are widely used for authentication in the electronic environment. The cost of verification individual's identity and address and also the secure storage of private keys are the stumbling block in the widespread usage of Digital Signature in the electronic environment.

X.509 Certificate Policy for India PKI states that the certificates will confirm that the information in the application provided by the subscriber does not conflict with the information in well-recognized consumer databases. The database of individual's information maintained by Unique Identification Authority of India (UIDAI) is deemed as authentic information by Government.

[1]The Unique Identification Authority of India (UIDAI) has been established with the mandate of providing a Unique Identification Number (Aadhaar Number) to all residents of India. During enrolment, the following data is collected:

1. Demographic details such as the name of the resident, address, date of birth, and gender;
2. Biometric details such as the fingerprints, iris scans, and photograph; and
3. Optional fields for communication of such as the mobile number and email address.

The UIDAI offers an authentication service that makes it possible for residents to authenticate their identity biometrically through presentation of their fingerprints or non-biometrically using a One Time Password (OTP) sent to the registered mobile phone or e-mail address

Verification of the Proof of Identity (PoI) and Proof of Address (PoA) is a pre-requisite for issuance of Digital Signature Certificates by Certifying Authorities. As part of the e-KYC process, the resident authorizes UIDAI (through Aadhaar authentication using either biometric/OTP) to provide their demographic data along with their photograph (digitally signed and encrypted) to service providers.

[1]https://eaadhaar.uidai.gov.in/

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. The real-time e-KYC service makes it possible for service providers to provide instant service delivery to residents, which otherwise would have taken a few days for activation based on the verification of KYC documents, digitization, etc.

The Government has introduced ***Electronic Signature or Electronic Authentication Technique and Procedure Rules, 2015*** in which the technique known as "e-authentication technique using Aadhaar e-KYC services" has been introduced to eliminate stumbling block in the widespread usage of Digital Signature. This service is termed as "eSign Service".

e-Sign facilitates digitally signing a document by an Aadhaar holder using an Online Service. While authentication of the signer is carried out using eKYC of Aadhaar, the signature on the document is carried out on a backend server, which is the e-Sign provider. The service can be run by a trusted third party service provider, like Certifying Authority. To begin with the trusted third party service shall be offered only by Certifying Authorities. The eSign is an integrated service that facilitates issuing a Signature Certificate and performing Signing of requested data by authenticating AADHAAR holder. The eSign Service shall be implemented in line with e-authentication guidelines issued by Controller. The certificate issued through eSign service will have a limited validity period and is only for one-time signing of requested data, in a single session.

1.1. Target Audience

This is a technical document and is targeted at Application Service Providers who require signing of digital document(s) in their application.

1.2. Objective of the document

This document provides eSign Service API specification. This includes API Data format, protocol and other related specifications.

1.3. Terminology

Application Service Provider (ASP): An organization or an entity using eSign service as part of their application to digitally sign the content. Examples include Government Departments, Banks and other public or private organizations. Currently there is no process of registration of ASP. ASP may contact the ESP (eSign Service Provider) directly to avail the service within its framework.

End-User: An Individual using the application of ASP and represents himself/herself for signing the document under the legal framework. For the purposes of KYC with UIDAI, the end-user shall also be the 'resident' holding the AADHAAR number. For the purposes of DSC by the CA, the end-user shall also be the 'applicant/subscriber for digital certificate', under the scope of IT Act.

eSign Service Provider (ESP): An organization or an entity providing eSign service. ESP is a "Trusted Third Party", as per the definitions of Second Schedule of Information Technology Act. ESP must be a registered KYC User Agency (KUA) with UIDAI. ESP will facilitate subscriber's key pair-generation, storing of key pairs on hardware security module and creation of digital signature. ESP can be a

Licensed Certifying Authority (CA), by themselves, or must be having an arrangement / integration with a CA for the purpose of obtaining Signature Certificate for the generated Key-pair.

Certifying Authority (CA): An organization or an entity licensed under CCA for issuance of Digital Certificate and carrying out allied CA operations.

UIDAI: An authority established by Government of India to provide unique identity to all Indian residents. It also runs the eKYC authentication service for the registered KYC User Agency (KUA).

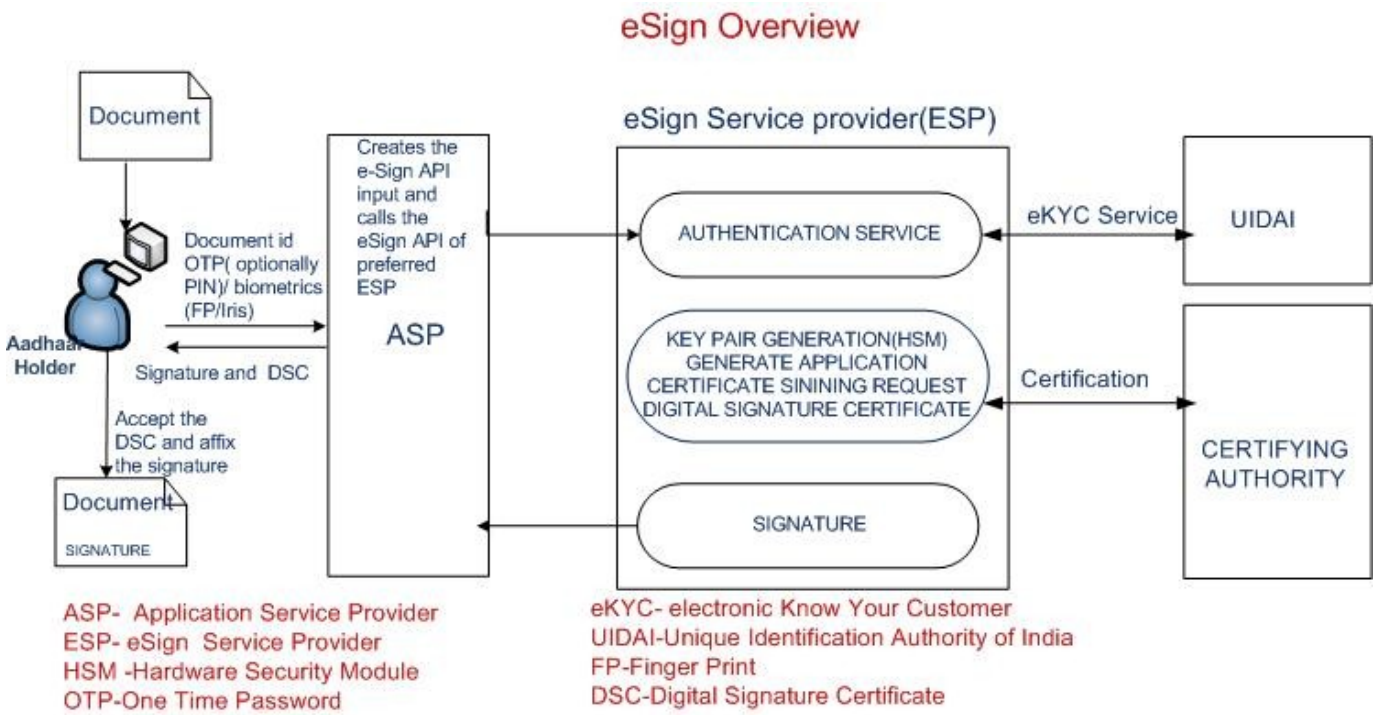
1.4. Legal Framework

eSign service will operate under the provisions of the Second Schedule of Information Technology Act, 2000 (e-authentication technique using Aadhaar e-KYC services) as notified vide (notification details)

2. Understanding eSign Service

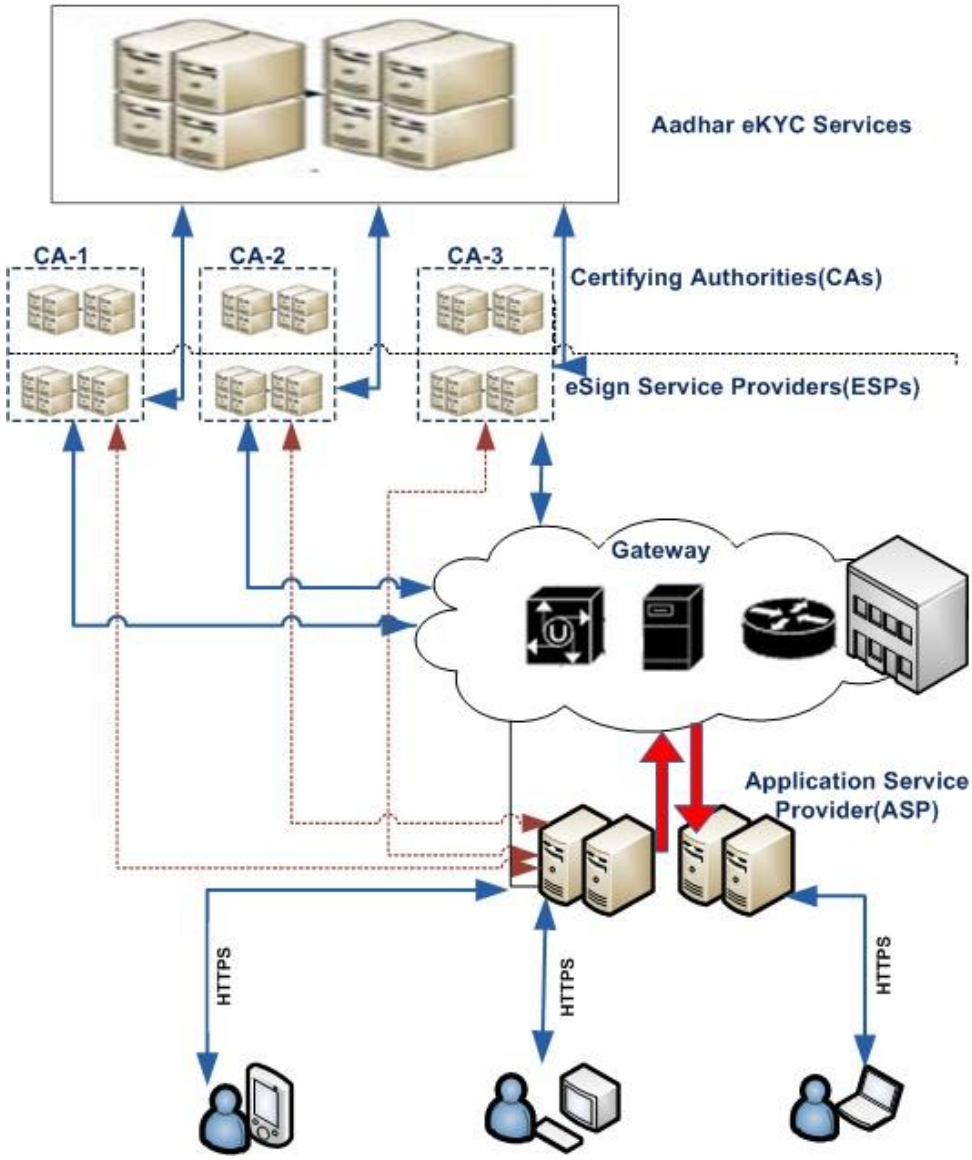
This chapter describes eSign Service, some of the envisioned usage scenarios, and working details. Technical details follow in subsequent chapters.

2.1. eSign Service at a glance

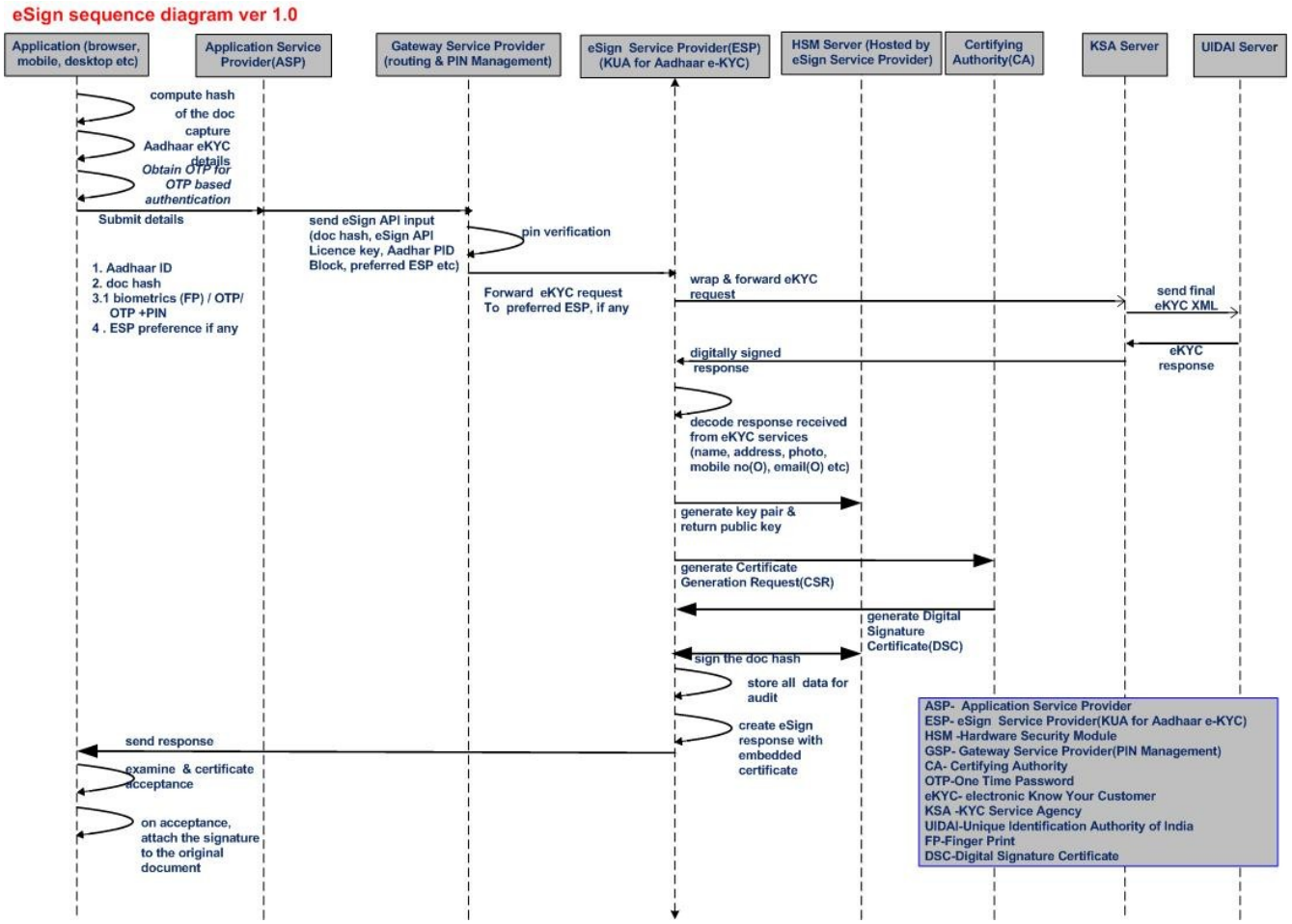


2.2. eSign framework

Following Diagram depicts the framework of eSign service.



API sequence diagram is given below:



3. eSign Service API

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

3.1. Usage scenarios

Option 1: Directly connecting to ESP
The API specifications remain common for all eSign Service provider. However, below are the things which will vary for each ESP.

1. eSign Service URL
2. ASP ID - Unique User ID provided by the ESP.

The eSign service API can be used in below based scenarios.

1. ASP using single eSign Service Provider
2. ASP using multiple eSign Service Provider

The usage of single eSign Service Provider is a straight forward case.

However, in case of multiple eSign provider, ASP shall have the above 3 parameters configurable for each request. The routing of requests to each API can be a round-robin, or a failure switchover, or an end-user selection basis, or any other manner implemented by ASP.

Option 2: Using a Gateway Service Provider
The API also allows the ASP to use a Gateway Service Provider. In such case, the gateway service provider will be having integration with one or more ASP and route the request accordingly.

The Gateway Service provider may also have additional validation process, where a one-time registration of end-user may happen, and a secure pin is provided to access the gateway. This will form a secure second factor protection in case of OTP based authentication.

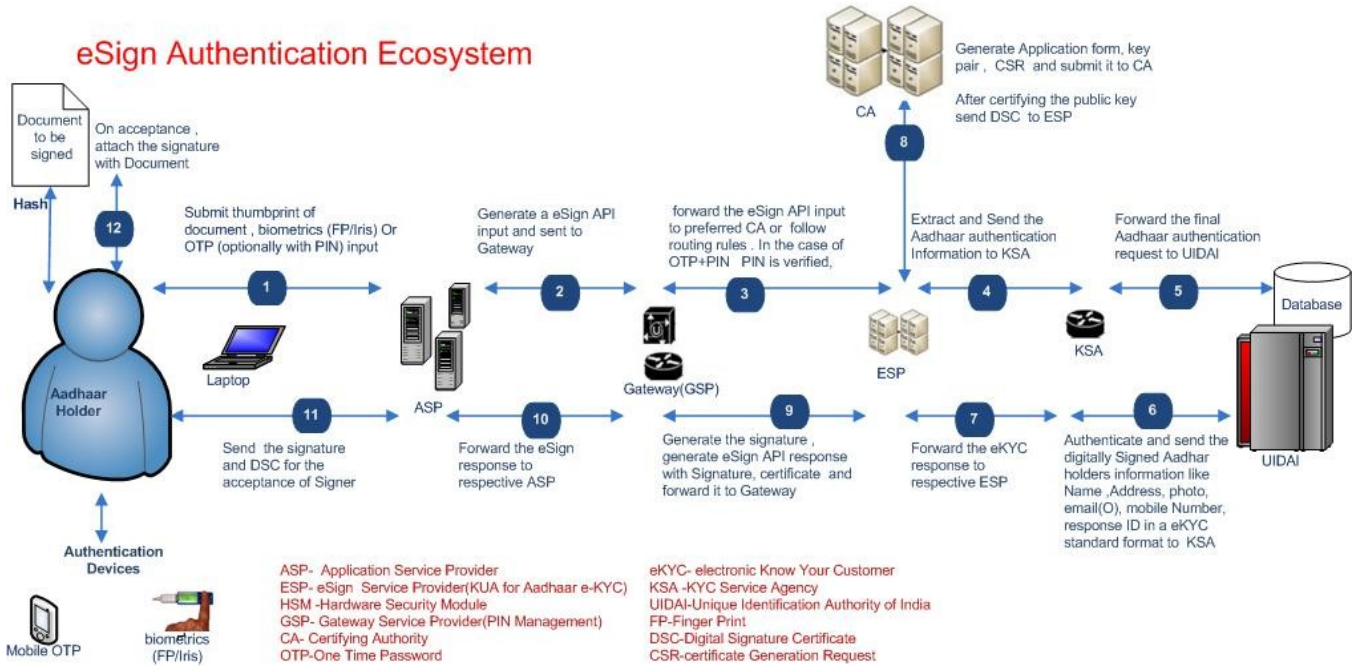
The Gateway Service provider should forwards the eSign API XML to ESPs without modifying the contents. The following functions are carried out at Gateway.

1) Verification of PIN, if present in the eSign API XML. If PIN verification fails, send a response back to ASP.
2) The routing of eSign API XML to CA in accordance with value in the "preferredCa ". If " preferredCa " is not present or service of ESP is not available, GSP may use their routing algorithms to select one ESP

The API specifications remain common for all the above cases.

3.1.1. eSign using Aadhaar Authentication

Following explains various scenarios and data flow. [1]



In this scenario:

1. ASP asks the end user to sign the document
2. ASP creates the document hash (to be signed) on the client side
3. ASP asks for AADHAAR number of the end-user.

4. ASP asks for Biometric Data (Fingerprint / IRIS) through a device approved by UIDAI
    a. Alternatively, ASP can ask for OTP received from UIDAI.
    b. OTP can be obtained by end-user through either by using UIDAI application, or by ASP calling the eSigin OTP service defined in this document.
5. ASP asks the end user to provide consent for certificate generation and signature
6. ASP forms the input data for eSign API
7. ASP calls the eSign API for Signing request
    a. ESP validates the calling application and the input.
    b. ESP forms the AADHAAR eKYC input as per AADHAAR API specification. The transaction ID shall be MD5 hash of document hash received along with Time Stamp received, to have correlation to eSign request. Eg: "UKC:" + md5(DocumentHashReceived + TimeStampReceived).
    c. ESP invokes AADHAAR eKYC API
    d. ESP logs the transaction
    e. ESP creates a new key pair and CSR for end-user.
    f. ESP calls the CA service and gets a certificate for end-user. The certificate will be a special class signature certificate, which has AADHAAR number, Name of the Aadhaar holder, eKYC response code, Authentication Type, and Time Stamp embedded.
    g. ESP signs the 'document hash' and provides to the end-user.
8. ASP receives the document signature and the end-user's public certificate.
9. ASP asks the end user to provide acceptability of certificate
10. On acceptance, ASP attaches the signature to the document.

3.2. API Protocol - eSign Service

eSign 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 and deployment of this service. 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 and the parameters for eSign service:

API URL	https://<host:port>/esign/<ver>/signdoc Where: • Host is the domain name or IP address of the ESP server. • Port is optional • <ver> indicates version of the API. Currently it must be "1.0" Example: • https://www.esp1.com/esign/1.0/signdoc • https://portal.esp2.com/esign/1.0/signdoc • https://www.esp3.com:8890/esign/1.0/signdoc
Method	POST
Content-Type	"application/xml"
Post data	A well-formed XML, as per the specifications provided in this document.

ASP is required to collect the necessary API URL from the respective ESP.

3.3. Authentication API: Input Data Format - eSign Service

eSign Service uses XML as the data format for input and output.

3.3.1. High level structure

Following is the XML data format for signing API:

```
<Esign ver="" sc="" ts="" txn="" aspId="" esignClass=""
preferredCa="" gatewayPin="" responseSigType="" >
      <Input>Document Hash in Hex</Input>
      <Aadhaar>base-64 encoded Aadhaar Auth XML as per UIDAI
      specifications</Aadhaar>
      <Signature>Digital signature of ASP</Signature>
</Esign>
```

3.3.1.1. Element Details

Element Name: Esign

- Description: Root element of the input xml
- Requirement of tag: Mandatory
- Value: Sub-elements
- Attributes: Table below

Sl No	Attribute	Required?	Value
1.	Ver	Mandatory	eSign version (mandatory). ESP may host multiple versions for supporting gradual migration. As of this specification, API version is “1.0”.
2.	sc	mandatory	sc – (mandatory) Represents signatory’s explicit consent for accessing the signatory’s identity and address data from Aadhaar system, generate and submit the electronic DSC application form to CA, creation of key pairs by ESP for signatory, submission of certificate to CA for certification , one time creation of signature on the hash along with this request, deletion of key pairs from the after applying signature . Only valid value is “Y”. **Sample** I hereby authorise ESP to 1. accessing my identity and address data through Aadhaar eKYC Services 2. Generate and submit the electronic DSC application form to CA. 3. create key pairs and submit of certificate Sigining Request for DSC Generation 4. one time creation of signature for this document/information I understand that the key pair will be deleted immediately after applying Signature

			Check box[Y/N*] *No by default
3.	ts	Mandatory	Request timestamp in ISO format. This should be same "ts" value as what is in PID block within Aadhaar authentication XML. See Aadhaar API specifications for details and format. The value should be in Indian Standard Time (IST), and should be within the range of maximum 30 minutes deviation to support out of sync server clocks.
4.	txn	Mandatory	Transaction ID of the ASP calling the API, this is logged and returned in the output for correlation
5.	aspId	Mandatory	Organization ID issued by ESP to the ASP
6.	esignClass	Mandatory	Class of eSign being requested. Allowed values are: • OTP Class = 1 • Bio Class = 2
7.	preferredCa	Optional	This should be blank while being sent directly to ESP. In case the request is sent to Gateway Service provider, this field can optionally have a unique code of particular CA. This helps in routing the request to specific CA, if requested by ASP. Allowed values are: • eMudhra = EMUDHRACA • (n)Code = NCODECA • Sify = SIFYCA
8.	gatewayPin	Optional	This should be blank while being sent directly to ESP. In case the request is sent to Gateway Service provider and authentication of user through OTP, the pin may be used. The value shall be SHA256 hash of the gatewayPIN concatenated with Time Stamp specified above. Eg: SHA256(SHA256(Gateway PIN) + Time Stamp)
9.	responseSigType	Optional	This value represents the response signature type, where ASP can request for specific type of signature, like Raw or PKCS7. The value can be comma separated values for multiple response types. In case the value is not defined, or blank, or the attribute is missing, all types of responses shall be given by ESP, as per the specification. Allowed Values are: 1. rawrsa

			2. pkcs7 Examples: responseSigType="rawrsa" responseSigType="rawrsa,pkcs7" responseSigType="" responseSigType="pkcs7"

Element Name: Input

- Description: Contains the value of Document Hash, which has to be signed.
- Requirement of tag: Mandatory
- Value: SHA256 hash value of the document in Hex format
- Attributes: Not applicable

Element Name: Aadhaar

- Description: Contains the end-user information and is based on AADHAAR Authentication API.
- Requirement of tag: Mandatory.
- Value: Base-64 encoded subset of Aadhaar Authentication XML, as per the UIDAI specifications for Auth XML. (Defined below)
- Attributes: Not applicable

Element Name: Signature

- Description: Contains the signature of ASP.
- Requirement of tag: Mandatory
- Value:
    - Signed value of Input XML, as per the W3C recommendation on XML Signature Syntax and Processing (Second Edition)
    - Refer http://www.w3.org/TR/xmldsig-core/ for more information
- Attributes: Not applicable

If resident does not provide this explicit consent, application SHOULD NOT process data using this API. ASP front-end application must ensure it takes an "explicit informed signatory's consent" authorizing the ESP to retrieve the resident data, DSC application form generation and submission, key-pair generation, CSR request to CA, Digital Signature on the hash submitted and key pair deletion after Digital Signature creation.

IMPORTANT NOTE: Digital Signature at eKYC XML level is mandatory .The eSign request XML should be digitally signed by ASP for authentication purposes.

3.3.2. Aadhaar Auth XML structure

Following is the XML data format for Aadhaar Auth XML version 1.6. Note that this XML may change based on Aadhaar specification. Currently e-Sign supports version 1.6.

```
<Auth uid="" tid="" ver="" txn="" >
      <Meta udc="" fdc="" idc="" pip="" lot="P" lov=""/>
      <Skey ci="">encrypted and encoded session key</Skey>
      <Data type="">encrypted PID block</Data>
      <Hmac>SHA-256 Hash of Pid block, encrypted and encoded</Hmac>
</Auth>
```

Details of Aadhaar authentication XML and its detail specifications are available in Aadhaar Authentication Specification document. Readers must refer to specification document available at http://uidai.gov.in/images/FrontPageUpdates/aadhaar_authentication_api_1_6.pdf

Additional notes for Auth XML format on Aadhaar Auth API document 1.6:

1. Attribute txn should contain the MD5 hash of the Document Hash plus Time Stamp.
2. Uses Element of Auth XML should not be present. This will be formed by ESP based on Class of eSign being requested.
3. Signature element of Auth XML should not be present.

3.4. Authentication API: Response Data Format - eSign Service

Below is the response format of eSign Service API. Note that, the API does not give any identity related data of the end-user.

```
<EsignResp status="" ts="" txn="" resCode="" errCode="" errMsg="">
      <SignedData      sigHashAlgorithm="SHA256">Signature      data
corresponding to input document hash</SignedData>
      <UserX509Certificate>base64  value  of  end  user  certificate
(.cer)</UserX509Certificate>
      <Pkcs7Response>Consolidated   PKCS7   signature   with   CMS
data</Pkcs7Response>
      <AadhaarResp>base-64 encoded authentication response which is
contained within the eKYC response of UIDAI</AadhaarResp>
      <Signature>Signature of ESP</Signature>
</EsignResp>
```

ASP should make sure that the affixing of digital signature to document or storage of digital signature only after the signatory's approval of contents of certificate and signature.

3.4.1. Element Details

Element Name: EsignResp

- Description: This element is the root element of the response and contains the meta values.
- Value: Sub-elements
- Attributes: Table below

Sl No	Attribute	Value
1.	status	In case of success, it will be "1" In case of failure, it will be "0"
2.	ts	Will contain the response timestamp in ISO format.
3.	txn	The Transaction ID provided by ASP in the request.
4.	resCode	A unique response code provided by ESP. This is a unique id for

		the transaction provided by ESP. It shall make the transaction traceable, and ASP is expected to store this code in their audit log.
5.	errCode	In case of failure, this will contain the failure error code. In case of success, it will be “NA”
6.	errMsg	In case of failure, this will contain a descriptive message against the error code. In case of success, it will be “NA”

Element Name: SignedData

- Description: This element will contain the signed value of the hash data requested.
- Presence: Mandatory, if the responseSigType is indicating to respond raw RSA (PKCS#1) signature. Else not required.
- Value: SHA256 signed value. The hash is verified against the public key of the end-user before responding.
- Attributes: Table Below

Sl No	Attribute	Value
1.	sigHashAlgorithm	Should be fixed to “SHA256”

Element Name: UserX509Certificate

- Description: This element will contain the Base 64 value of the Certificate. No private key information shared. For manual verification, this value can be copied and saved as .cer file (With begin and end statements).
- Presence: Mandatory, if the responseSigType is indicating to respond raw RSA (PKCS#1) signature. Else not required
- Value: Base 64 value of end-user certificate (public).
- Attributes: Not Applicable

Element Name: Pkcs7Response

- Description: This element will contain the consolidated PKCS7 CMS data containing the signature data as well as User’s certificate. This can additionally contain the trust chain certificates for necessary validation. (But, ASP is expected to have the parent trust chain pre-configured for necessary validation.)
- Presence: Mandatory, if the responseSigType is indicating to respond PKCS#7 (CMS) signature. Else not required.
- Value: Base 64 value of PKCS7 CMS data.
- Attributes: Not Applicable

Element Name: AadhaarResp

- Description: This element contains base-64 encoded authentication response which is contained within the eKYC response of UIDAI.
- Presence: This shall be present if ESP is able to get any kind of response from UIDAI during eKYC request, irrespective of whether eSign succeeds or fails.
- Value: base-64 encoded Aadhaar Authentication response XML. This is contained within eKYC response. ESP after doing eKYC, can simply pass this back to ASP. This provide a mechanism for ASP to keep the audit of Aadhaar authentication and take advantage of the response meta data such as action codes, resident messages, etc. Base on this, ASP application may have to show messages (in case biometric auth fails for example) to end user to ensure smooth transaction flow.
- Attributes: Not Applicable

Element Name: Signature

- Description: This element will contain the signature of ESP, which can be used for verification by ASP and protect the response from any kind of tamper.
- Value:
    - Signed value of response XML, as per the W3C recommendation on XML Signature Syntax and Processing (Second Edition)
    - Refer http://www.w3.org/TR/xmldsig-core/ for more information
- Attributes: Not Applicable

3.4.2. Error Codes

The List of error codes are available at annexure 1. ASP can automate their application based on prominent errors, in order to ease the flow for end-user.

Any error from AADHAAR authentication will be responded with respective error codes defined by UIDAI. Please refer the latest error codes of UIDAI in Authentication API document.

4. OTP Generation Service API

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

It may also be noted that, even though UIDAI allows OTP over Email and Mobile, for the purpose of eSign, the **OTP will be delivered only to Mobile Number** of end user (If provided by end user during AADHAAR enrolment). In addition to SMS based OTPs, as and when UIDAI provide enhanced mobile OTP capabilities including HOTP/TOTP, etc., it shall be a valid mobile OTP for eSign purposes.

Initial authentication using login and password at ASP front-end application level is mandatory for the usage of OTP option for eSign.

4.1. API Protocol - OTP Generation Service

OTP Generation service is exposed as stateless service over HTTPS. This is a proxy service to Aadhaar OTP service. If Aadhaar holder does not have a valid mobile OTP, ASP can use this service to send a mobile OTP to Aadhaar holder's registered mobile. Note that UIDAI provides Aadhaar holders a mechanism to obtain OTP directly (via SMS, resident portal, mobile app, etc). If Aadhaar holder does not have a valid OTP in possession, ASP application should provide an option to "Request OTP" and execute that request via this API.

[1] https://eaadhaar.uidai.gov.in/

Following is the URL format and the parameters for OTP Generation service:

API URL	https://<host:port>/esign/<ver>/getotp

	Where: • Host is the domain name or IP address of the ESP server. • Port is optional • <ver> indicates version of the API. Currently it must be “1.0”
Method	POST
Content-Type	“application/xml”
Post data	A well-formed XML, as per the specifications provided in this document.

ASP is required to collect the necessary API URL from the respective ESP.

4.2. Supplementary API: Input Data Format - OTP Generation Service

OTP Generation Service uses XML as the data format for input and output.

4.2.1. High level structure

Following is the XML data format for signing API:

```
<OTP ts="" ver="" txn="" aspId="" uid="">
     <Signature>Digital signature of ASP</Signature>
</OTP>
```

4.2.1.1. Element Details

Element Name: OTP

- Description: Root element of the input xml
- Requirement of tag: Mandatory
- Value: Sub-elements
- Attributes: Table below

Sl No	Attribute	Required?	Value
1.	ts	Mandatory	Request timestamp in ISO format as per Aadhaar authentication API specification. The value should be in Indian Standard Time (IST), and should be within the range of maximum 30 minutes deviation to support out of sync server clocks.
2.	ver	Mandatory	OTP API version (mandatory). ESP may host multiple versions for supporting gradual migration. As of this specification, API version is “1.0”.
3.	txn	Mandatory	Transaction ID of the ASP calling the API, this is logged and returned in the output for correlation
4.	aspId	Mandatory	User ID issued by ESP to the ASP
5.	uid	Mandatory	Aadhaar number of the resident

Element Name: Signature

- Description: Contains the signature of ASP.
- Requirement of tag: Mandatory
- Value:

- Signed value of Input XML, as per the W3C recommendation on XML Signature Syntax and Processing (Second Edition)
- Refer http://www.w3.org/TR/xmldsig-core/ for more information

- Attributes: Not applicable

4.3. Supplementary API: Response Data Format - OTP Generation Service

Below is the response format of OTP Generation Service API. As this is not a data sensitive API, and just a triggering API, this will not be a signed response.

```
<OTPResp status="" ts="" txn="" resCode="" errCode="" errMsg="">
     <AadhaarResp>base-64 encoded OTP API response received from
UIDAI</AadhaarResp>
     <Signature>Signature of ESP</Signature>
</OTPResp>
```

4.3.1. Element Details

Element Name: OTPResponse

- Description: This element is the root element of the response and contains the meta values.
- Value: No value will be specified, and will be a self-closing XML tag.
- Attributes: Table below

Sl No	Attribute	Value
1.	status	In case of success, it will be “1” In case of failure, it will be “0”
2.	ts	Will contain the response timestamp in ISO format.
3.	txn	The Transaction ID provided by ASP in the request.
4.	resCode	A unique response code provided by ESP. This is a unique id for the transaction provided by ESP. It shall make the transaction traceable, and ASP is expected to store this code in their audit log.
5.	errCode	In case of failure, this will contain the failure error code. In case of success, it will be “NA”
6.	errMsg	In case of failure, this will contain a descriptive message against the error code. In case of success, it will be “NA”

Element Name: AadhaarResp

- Description: This element contains base-64 encoded OTP response XML received from UIDAI.
- Value: base-64 encoded Aadhaar OTP response XML. This provide a mechanism for ASP to keep the audit of Aadhaar OTP API and take advantage of the response meta data such as action codes, resident messages, etc. Base on this, ASP application may have to show messages (in case of unverified or missing mobile number in Aadhaar database for example) to end user to ensure smooth transaction flow.

Element Name: Signature

- Description: This element will contain the signature of ESP, which can be used for verification by ASP and protect the response from any kind of tamper.
- Value:

- Signed value of response XML, as per the W3C recommendation on XML Signature Syntax and Processing (Second Edition)
- Refer http://www.w3.org/TR/xmldsig-core/ for more information

- Attributes: Not Applicable

4.3.2. Error Codes

The List of error codes are available at annexure 2. ASP can automate their application based on prominent errors, in order to ease the flow for end-user.

Any error from AADHAAR authentication will be responded with respective error codes defined by UIDAI. Please refer the latest error codes of UIDAI in OTP API document.

Annexure 1

eSign Service -error codes

Sl No	Error Code	Error message	Originator
1	ESP-901	Invalid eSign Class	ESP
2	ESP-902	Invalid ASP ID. It cannot be Empty	ESP
3	ESP-903	Invalid ASP ID. It may not exist or may be inactive.	ESP
4	ESP-905	Document Hash not received	ESP
5	ESP-906	Aadhaar cannot be Empty	ESP
6	ESP-907	Request Time Stamp cannot be Empty	ESP
7	ESP-908	Request Time Stamp is not valid. Please check the server time.	ESP
8	ESP-909	Transaction ID cannot be Empty	ESP
9	ESP-910	Duplicate Transaction ID for the given ASP.	ESP
10	ESP-911	Input XML Signature verification failed.	ESP
11	ESP-922	Invalid Signature on Input XML. Please use the corresponding certificate mapped with ESP.	ESP
12	ESP-991	ESP Database Connectivity Error	ESP
13	ESP-992	Input XML Parsing Error.	ESP
14	ESP-993	Error Parsing CA Response XML	ESP
15	ESP-994	Error from KSA Server	ESP
16	ESP-995	Unknown CIDR Error	ESP
17	ESP-996	Unable to parse UidData XML string	ESP
18	ESP-999	Unknown Error	ESP
19	K-100	Resident authentication failed	CIDR-KUA
20	K-200	Resident data currently not available	CIDR-KUA
21	K-540	Invalid KYC XML	CIDR-KUA
22	K-541	Invalid e-KYC API version	CIDR-KUA
23	K-542	Invalid resident consent (“rc” attribute in “Kyc” element)	CIDR-KUA
24	K-543	Invalid timestamp (“ts” attribute in “Kyc” element)	CIDR-KUA
25	K-544	Invalid resident auth type (“ra” attribute in “Kyc” element does not match what is in PID block)	CIDR-KUA
26	K-545	Resident has opted-out of this service	CIDR-KUA
27	K-550	nvalid Uses Attribute	CIDR-KUA
28	K-551	Invalid “Txn” namespace	CIDR-KUA
29	K-552	Invalid License key	CIDR-KUA
30	K-569	Digital signature verification failed for e-KYC XML	CIDR-KUA
31	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)	CIDR-KUA
32	K-600	AUA is invalid or not an authorized KUA	CIDR-KUA
33	K-601	ASA is invalid or not an authorized KSA	CIDR-KUA
34	K-602	KUA encryption key not available	CIDR-KUA
35	K-603	KSA encryption key not available	CIDR-KUA
36	K-604	KSA Signature not allowed	CIDR-KUA
37	K-605	Neither KUA key nor KSA encryption key are available	CIDR-KUA
38	K-955	Technical Failure	CIDR-KUA

39	K-999	Unknown error	CIDR-KUA
40	100	“Pi” (basic) attributes of demographic data did not match.	CIDR-AUA
41	200	“Pa” (address) attributes of demographic data did not match	CIDR-AUA
42	300	Biometric data did not match	CIDR-AUA
43	310	Duplicate fingers used	CIDR-AUA
44	311	Duplicate Irises used.	CIDR-AUA
45	312	FMR and FIR cannot be used in same transaction	CIDR-AUA
46	313	Single FIR record contains more than one finger	CIDR-AUA
47	314	Number of FMR/FIR should not exceed 10	CIDR-AUA
48	315	Number of IIR should not exceed 2	CIDR-AUA
49	400	Invalid OTP value	CIDR-AUA
50	401	Invalid TKN value	CIDR-AUA
51	500	Invalid encryption of Skey	CIDR-AUA
52	501	Invalid certificate identifier in “ci” attribute of “Skey”	CIDR-AUA
53	502	Invalid encryption of Pid	CIDR-AUA
54	503	Invalid encryption of Hmac	CIDR-AUA
55	504	Session key re-initiation required due to expiry or key out of sync	CIDR-AUA
56	505	Synchronized Key usage not allowed for the AUA	CIDR-AUA
57	510	Invalid Auth XML format	CIDR-AUA
58	511	Invalid PID XML format	CIDR-AUA
59	520	Invalid device	CIDR-AUA
60	521	Invalid FDC code under Meta tag	CIDR-AUA
61	522	Invalid IDC code under Meta tag	CIDR-AUA
62	530	Invalid authenticator code	CIDR-AUA
63	540	Invalid Auth XML version	CIDR-AUA
64	541	Invalid PID XML version	CIDR-AUA
65	542	AUA not authorized for ASA. This error will be returned if AUA and ASA do not have linking in the portal	CIDR-AUA
66	543	Sub-AUA not associated with “AUA”. This error will be returned if Sub-AUA specified in “sa” attribute is not added as “Sub-AUA” in portal	CIDR-AUA
67	550	Invalid “Uses” element attributes	CIDR-AUA
68	551	Invalid “tid” value for registered device	CIDR-AUA
69	552	Invalid registered device key, please reset	CIDR-AUA
70	553	Invalid registered device HOTP, please reset	CIDR-AUA
71	554	Invalid registered device encryption	CIDR-AUA
72	555	Mandatory reset required for registered device	CIDR-AUA
73	561	Request expired (“Pid->ts” value is older than N hours where N is a configured threshold in authentication server)	CIDR-AUA
74	562	Timestamp value is future time (value specified “Pid->ts” is ahead of authentication server time beyond acceptable threshold)	CIDR-AUA
75	563	Duplicate request (this error occurs when exactly same authentication request was re-sent by AUA)	CIDR-AUA
76	564	HMAC Validation failed	CIDR-AUA
77	565	AUA license has expired	CIDR-AUA
78	566	Invalid non-decryptable license key	CIDR-AUA
79	567	Invalid input (this error occurs when some unsupported characters were found in Indian language values, “lname” or “lav”)	CIDR-AUA

80	568	Unsupported Language	CIDR-AUA
81	569	Digital signature verification failed (means that authentication request XML was modified after it was signed)	CIDR-AUA
82	570	Invalid key info in digital signature (this means that certificate used for signing the authentication request is not valid – it is either expired, or does not belong to the AUA or is not created by a well-known Certification Authority)	CIDR-AUA
83	571	PIN Requires reset (this error will be returned if resident is using the default PIN which needs to be reset before usage)	CIDR-AUA
84	572	Invalid biometric position	CIDR-AUA
85	573	Pi usage not allowed as per license	CIDR-AUA
86	574	a usage not allowed as per license	CIDR-AUA
87	575	fa usage not allowed as per license	CIDR-AUA
88	576	FMR usage not allowed as per license	CIDR-AUA
89	577	FIR usage not allowed as per license	CIDR-AUA
90	578	IIR usage not allowed as per license	CIDR-AUA
91	579	OTP usage not allowed as per license	CIDR-AUA
92	580	PIN usage not allowed as per license	CIDR-AUA
93	581	Fuzzy matching usage not allowed as per license	CIDR-AUA
94	582	Local language usage not allowed as per license	CIDR-AUA
95	584	Invalid pincode in LOV attribute under Meta tag	CIDR-AUA
96	585	Invalid geo-code in LOV attribute under Meta tag	CIDR-AUA
97	710	Missing “Pi” data as specified in “Uses”	CIDR-AUA
98	720	Missing “Pa” data as specified in “Uses”	CIDR-AUA
99	721	Missing “Pfa” data as specified in “Uses”	CIDR-AUA
100	730	Missing PIN data as specified in “Uses”	CIDR-AUA
101	740	Missing OTP data as specified in “Uses”	CIDR-AUA
102	800	Invalid biometric data	CIDR-AUA
103	810	Missing biometric data as specified in “Uses”	CIDR-AUA
104	811	Missing biometric data in CIDR for the given Aadhaar number	CIDR-AUA
105	812	Resident has not done “Best Finger Detection”. Application should initiate BFD application to help resident identify their best fingers. See Aadhaar Best Finger Detection API specification.	CIDR-AUA
106	820	Missing or empty value for “bt” attribute in “Uses” element	CIDR-AUA
107	821	Invalid value in the “bt” attribute of “Uses” element	CIDR-AUA
108	901	No authentication data found in the request (this corresponds to a scenario wherein none of the auth data – Demo, Pv, or Bios – is present)	CIDR-AUA
109	902	Invalid “dob” value in the “Pi” element (this corresponds to a scenarios wherein “dob” attribute is not of the format “YYYY” or “YYYY-MM-DD”, or the age of resident is not in valid range)	CIDR-AUA
110	910	Invalid “mv” value in the “Pi” element	CIDR-AUA
111	911	Invalid “mv” value in the “Pfa” element	CIDR-AUA
112	912	Invalid “ms” value	CIDR-AUA
113	913	Both “Pa” and “Pfa” are present in the authentication request (Pa and Pfa are mutually exclusive)	CIDR-AUA
114	940	Unauthorized ASA channel	CIDR-AUA
115	941	Unspecified ASA channel	CIDR-AUA

116	980	Unsupported option	CIDR-AUA
117	997	Invalid Aadhaar status (Aadhaar is not in authenticatable status)	CIDR-AUA
118	998	Invalid Aadhaar Number	CIDR-AUA
119	999	Unknown error	CIDR-AUA
120	E - 101	KYC XML Not Parsed Properly.	ASA-KSA
121	E - 102	Audit Logging in DB is failed for request.	ASA-KSA
122	E - 103	Audit Logging in DB is failed for response.	ASA-KSA
123	E - 104	Audit Logging in DB is failed for Error occurred.	ASA-KSA
124	E - 105	KYC XSD Validation Failed.	ASA-KSA
125	E - 106	KYC Request Signature Verification Failed.	ASA-KSA
126	E - 109	Blank Response Received from UIDAI	ASA-KSA
127	E - 110	Unable to Decrypt Response at KSA.	ASA-KSA
128	E - 111	KYC Response Signature Verification Failed.	ASA-KSA
129	E - 112	BFD XSD Validation Failed.	ASA-KSA
130	E - 113	BFD XSD Validation Failed.	ASA-KSA
131	E - 113	OTP XSD Validation Failed.	ASA-KSA
132	E - 114	OTP XSD Validation Failed.	ASA-KSA
133	E - 114	KYC Response XML Not Parsed Properly.	ASA-KSA
134	E - 115	KYC Response XML Not Parsed Properly.	ASA-KSA
135	E - 119	ASA/KSA is unable to connect to UIDAI server.	ASA-KSA
136	E - 119	ASA/KSA is unable to connect to UIDAI server.	ASA-KSA
137	E - 120	Auth XSD Validation Failed.	ASA-KSA
138	E - 121	Database audit logging in failed due to the duplicate transaction ID.	ASA-KSA
139	E - 123	BFD Request XML Not Parsed Properly.	ASA-KSA
140	E - 124	OTP Request XML Not Parsed Properly.	ASA-KSA
141	E - 125	BFD Request Signature Verification Failed	ASA-KSA
142	E - 126	OTP Request Signature Verification Failed	ASA-KSA
143	E - 129	BFD Response XML Not Parsed Properly.	ASA-KSA
144	E - 130	OTP Response XML Not Parsed Properly.	ASA-KSA
145	E - 132	Error during KYC Request Signature Verification.	ASA-KSA
146	E - 133	Error during KYC Response Signature Verification.	ASA-KSA
147	E - 136	Error during BFD Request Signature Verification.	ASA-KSA
148	E - 137	Error during OTP Request Signature Verification.	ASA-KSA
149	E - 138	Error during KYC XSD Validation.	ASA-KSA
150	E - 140	Error during BFD XSD Validation.	ASA-KSA
151	E - 141	Error during BFDOTP XSD Validation.	ASA-KSA
152	E - 143	Response Received is E	ASA-KSA
153	E - 144	BFD Response Signature Verification.	ASA-KSA
154	E- 199	KSA/ASA Internal Error.	ASA-KSA
155	E - 555	Duplicate Transaction Id Error.	ASA-KSA
156	E - 999	KSA Internal Error.	ASA-KSA

OTP Services -Error Codes

Sl No	Error Code	Error message	Originator
1	ESP-902	Invalid ASP ID. It cannot be Empty	ESP
2	ESP-903	Invalid ASP ID. It may not exist or may be inactive.	ESP
3	ESP-906	Aadhaar cannot be Empty	ESP
4	ESP-907	Request Time Stamp cannot be Empty	ESP
5	ESP-908	Request Time Stamp is not valid. Please check the server time.	ESP
6	ESP-909	Transaction ID cannot be Empty	ESP
7	ESP-910	Duplicate Transaction ID for the given ASP.	ESP
8	ESP-911	Input XML Signature verification failed.	ESP
9	ESP-922	Invalid Signature on Input XML. Please use the corresponding certificate mapped with ESP.	ESP
10	ESP-991	ESP Database Connectivity Error	ESP
11	ESP-992	Input XML Parsing Error.	ESP
12	ESP-994	Error from KSA Server	ESP
13	ESP-995	Unknown CIDR Error	ESP
14	ESP-999	Unknown Error	ESP
15	110	Aadhaar number does not have verified mobile/email	CIDR
16	111	Aadhaar number does not have verified mobile	CIDR
17	112	Aadhaar number does not have both email and mobile.	CIDR
18	510	Invalid "Otp" XML format	CIDR
19	520	Invalid device	CIDR
20	530	Invalid AUA code	CIDR
21	540	Invalid OTP XML version	CIDR
22	542	AUA not authorized for ASA. This error will be returned if AUA and ASA do not have linking in the portal	CIDR
23	543	Sub-AUA not associated with "AUA". This error will be returned if Sub-AUA specified in "sa" attribute is not added as "Sub-AUA" in portal	CIDR
24	565	AUA License key has expired or is invalid	CIDR
25	566	ASA license key has expired or is invalid	CIDR
26	569	Digital signature verification failed	CIDR
27	570	Invalid key info in digital signature (this means that certificate used for signing the OTP request is not valid - it is either expired, or does not belong to the AUA or is not created by a CA)	CIDR
28	940	Unauthorized ASA channel	CIDR
29	941	Unspecified ASA channel	CIDR
30	950	Could not generate and/or send OTP	CIDR
31	999	Unknown error	CIDR
32	E - 101	KYC XML Not Parsed Properly.	ASA-KSA
33	E - 102	Audit Logging in DB is failed for request.	ASA-KSA
34	E - 103	Audit Logging in DB is failed for response.	ASA-KSA
35	E - 104	Audit Logging in DB is failed for Error occurred.	ASA-KSA
36	E - 105	KYC XSD Validation Failed.	ASA-KSA

37	E - 106	KYC Request Signature Verification Failed.	ASA-KSA
38	E - 109	Blank Response Received from UIDAI	ASA-KSA
39	E - 110	Unable to Decrypt Response at KSA.	ASA-KSA
40	E - 111	KYC Response Signature Verification Failed.	ASA-KSA
41	E - 112	BFD XSD Validation Failed.	ASA-KSA
42	E - 113	BFD XSD Validation Failed.	ASA-KSA
43	E - 113	OTP XSD Validation Failed.	ASA-KSA
44	E - 114	OTP XSD Validation Failed.	ASA-KSA
45	E - 114	KYC Response XML Not Parsed Properly.	ASA-KSA
46	E - 115	KYC Response XML Not Parsed Properly.	ASA-KSA
47	E - 119	ASA/KSA is unable to connect to UIDAI server.	ASA-KSA
48	E - 119	ASA/KSA is unable to connect to UIDAI server.	ASA-KSA
49	E - 120	Auth XSD Validation Failed.	ASA-KSA
50	E - 121	Database audit logging in failed due to the duplicate transaction ID.	ASA-KSA
51	E - 123	BFD Request XML Not Parsed Properly.	ASA-KSA
52	E - 124	OTP Request XML Not Parsed Properly.	ASA-KSA
53	E - 125	BFD Request Signature Verification Failed	ASA-KSA
54	E - 126	OTP Request Signature Verification Failed	ASA-KSA
55	E - 129	BFD Response XML Not Parsed Properly.	ASA-KSA
56	E - 130	OTP Response XML Not Parsed Properly.	ASA-KSA
57	E - 132	Error during KYC Request Signature Verification.	ASA-KSA
58	E - 133	Error during KYC Response Signature Verification.	ASA-KSA
59	E - 136	Error during BFD Request Signature Verification.	ASA-KSA
60	E - 137	Error during OTP Request Signature Verification.	ASA-KSA
61	E - 138	Error during KYC XSD Validation.	ASA-KSA
62	E - 140	Error during BFD XSD Validation.	ASA-KSA
63	E - 141	Error during BFDOTP XSD Validation.	ASA-KSA
64	E - 143	Response Received is E	ASA-KSA
65	E - 144	BFD Response Signature Verification.	ASA-KSA
66	E- 199	KSA/ASA Internal Error.	ASA-KSA
67	E - 555	Duplicate Transaction Id Error.	ASA-KSA
68	E - 999	KSA Internal Error.	ASA-KSA

5. Change History

<table>
<tr><th colspan="4">Change History</th></tr>
<tr><td>Section</td><td>Ver</td><td>Date</td><td>Modification</td></tr>
<tr><td>3.1</td><td>1.0</td><td>19.05.2015</td><td>Option :2
(inserted) The Gateway Service provider should forwards the eSign API XML to ESPs without modifying the contents. The following functions are carried out Gateway.
1) Verification of PIN, if present in the eSign API XML. If PIN verification fails, send a response back to ASP.
2) The routing of eSign API XML to CA in accordance with value in the "preferredCa". If " preferredCa" is not present or service of ESP is not available, GSP may use their routing algorithms to select one ESP</td></tr>
<tr><td>3.3.1.1 Element Details</td><td>1.0</td><td>19.05.2015</td><td>preferredCa(Table, sl no 7)

(deleted)Gateway Service provider, shall validate this field and remove the value, before sending to ESP.
gatewayPin (Table, sl no 8)

(deleted)-In case the request is sent to Gateway Service provider, this field shall have the pin belonging to user, and be mandatory in case of OTP authentication.

(inserted) In case the request is sent to Gateway Service provider and authentication of user through OTP, the pin may be used</td></tr>
<tr><td>3.3.1 High level structure</td><td>1.0</td><td>15.07.2015</td><td>responseSigType=""
3.3.1.1 Element Details
NEW 9. responseSigType Optional

This value represents the response signature type, where ASP can request for specific type of signature, like Raw or PKCS7.

The value can be comma separated values for multiple response types.
In case the value is not defined, or blank, or the attribute is missing, all types of responses shall be given by ESP, as per the specification.

Allowed Values are:
1. rawrsa
2. pkcs7

Examples:
responseSigType="rawrsa"
responseSigType="rawrsa,pkcs7"
responseSigType=""
responseSigType="pkcs7"</td></tr>
<tr><td>3.4.1 Element Details</td><td>1.0</td><td>15.07.2015</td><td>Element Name: SignedData(addition)
<table><tr><td>Sl No</td><td>Attribute</td><td>Value</td></tr><tr><td>2.</td><td>sigHashAlgorithm</td><td>Should be fixed to “SHA256”</td></tr></table></td></tr>
</table>