eSign API Specifications

Version 2.0

April 2017

Controller of Certifying Authorities
Ministry of Electronics and Information Technology

Document Control

Document Name	eSign API Specifications
Status	Release
Version	2.0
Release date	28.04.2017
Last update	28.04.2017
Document Owner	Controller of Certifying Authorities, India

Table of Contents

1. Introduction 1

1.1. Target Audience 2

1.2. Objective of the document 2

1.3. Terminology 2

1.4. Legal Framework 3

2. Understanding eSign Service 3

2.1. eSign Service at a glance 3

3. eSign Service API 4

3.1. eSign - Usage scenarios 4

3.1.1. eSign using ASP provided KYC data 4

3.1.2. eSign using e-KYC made by ESP 5

3.2. API Protocol - eSign Service 6

3.3. eSign API: Input Data Format - eSign Service 7

3.3.1. eSign XML structure 7

3.3.1.1. Element Details 7

3.3.2. OrgDetails XML structure 10

3.4. eSign API: Response Data Format - eSign Service 11

3.4.1. Element Details 11

3.4.2. Error Codes 12

Change History 14

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 e-KYC providers will be used for eSign . The accepted e-KYC providers are listed in the e-authentication guidelines.

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.

e-KYC 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 eSign Users which otherwise would have taken a few days for activation based on the verification of KYC documents, digitization, etc.

ESP and ASP have to make sure that mechanisms implemented for authentication of individuals adhere to the guidelines of e-KYC provider

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

e-Sign facilitates digitally signing a document by an eSign user using an Online Service. While authentication of the signer is carried out using e-KYC , the signature on the document is carried out on a backend server, which is the e-Sign provider. The 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 on basis of authenticated e-KYC response. 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.

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. eSign is designed for applying Digital Signature based on the response received from e-KYC service.

1.3. Terminology

"eSign" or "eSign Service" is an online Electronic Signature Service in which the key pair generation, certification of the public key by the CA and digital signature creation for electronic document are facilitated by the eSign online Electronic Signature Service provider instantaneously within a single online service based on successful authentication of individual using e-KYC services

"eSign User" is an Individual requesting for eSign online Electronic Signature Service of eSign Service provider. This individual shall be using the application of ASP and represents himself/herself for signing the document under the legal framework. For the purposes of DSC by the CA, the eSign user shall also be the 'applicant/subscriber for digital certificate', under the scope of IT Act.

"e-KYC" means the transfer of digitally signed demographic data such as Name, Address, Date of Birth, Gender, Mobile number, Email address, photograph etc of an individual. collected and verified by e-KYC provider on successful authentication of same individual

"**response code**" is the identification number maintained by e-KYC provider to identify the authentication

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.

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 will facilitate subscriber's key pair-generation, storing of key pairs on hardware security module and creation of digital signature. ESP must be/ 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.

e-KYC Number' shall mean the unique identification number maintained by e-KYC provider;

e-KYC provider shall mean any e-KYC provider listed in e-Authentication Guidelines

'**OTP**' shall mean one time password sent to the eSign User's cell phone for the purpose of authentication;

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

1.4. Legal Framework

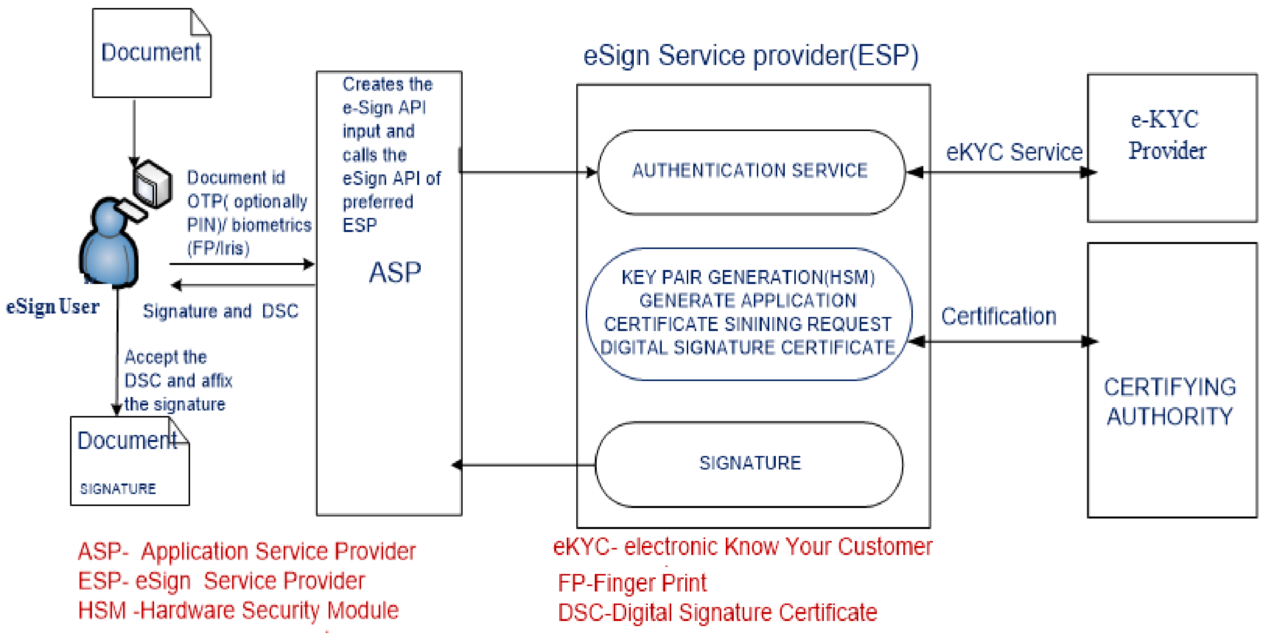
eSign service will operate under the provisions of the Second Schedule of Information Technology Act, 2000 (e-authentication technique using Aadhaar e-KYC services) as notified vide (notification details)

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

eSign Overview


Document
eSign Service provider(ESP)
Creates the e-Sign API input and calls the eSign API of preferred ESP
ASP
Document id OTP(optionally PIN)/ biometrics (FP/Iris)
eSign User
Signature and DSC
Accept the DSC and affix the signature
Document
SIGNATURE
AUTHENTICATION SERVICE
KEY PAIR GENERATION(HSM) GENERATE APPLICATION CERTIFICATE SINING REQUEST DIGITAL SIGNATURE CERTIFICATE
SIGNATURE
eKYC Service
e-KYC Provider
Certification
CERTIFYING AUTHORITY
ASP- Application Service Provider
ESP- eSign Service Provider
HSM -Hardware Security Module
eKYC- electronic Know Your Customer
FP-Finger Print
DSC-Digital Signature Certificate

3. eSign Service API

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

This API expects that authentication of the individual will be carried out by ESP or ASP and the digitally signed e-KYC data is made available to ESP. The authentication needs to be carried out independent of this API. However this API has given provision to carry response of e-KYC service depending on the e-KYC mode of authentication.

The suggested method for obtaining authenticated e-KYC response are

1. eSign user performs self-authentication via ASP application and send the authenticated e-KYC response to ESP via eSign API.
2. ESP facilitate authentication of eSign user by calling authentication URL of ESP by eSign user via ASP application. The e-KYC response will be received by ESP and performs eSign on the eSign request received from ASP after correlating e-KYC id and permissible time limit.

3.1. eSign - Usage scenarios

The API specifications remain common for all eSign Service provider. However, the parameters which will vary for each ESP are eSign Service URL and ASP ID (Unique User ID provided by the ESP).

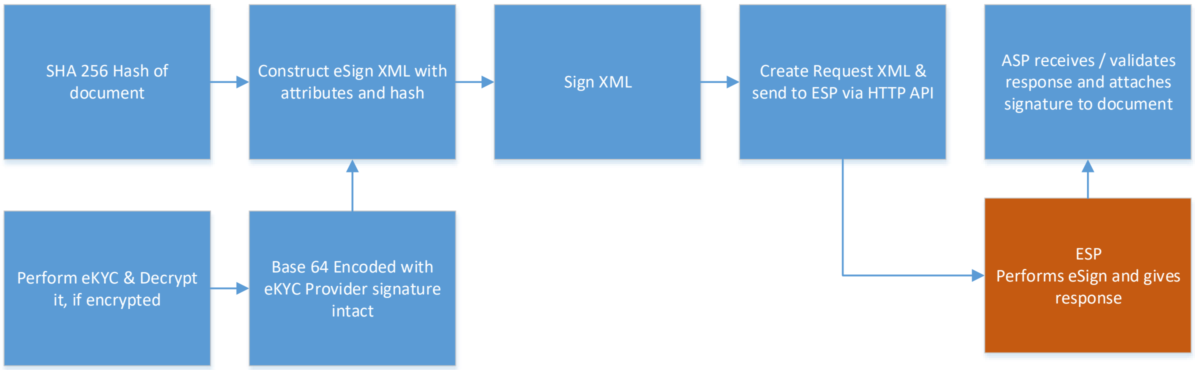
In case of multiple eSign provider, ASP shall have the parameters configurable for each request. The routing of requests to each APIs of ESP can be a round-robin, or a failure switchover, or an eSign user selection basis, or any other manner implemented by ASP.

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

1. ASP performs KYC of eSign user and sends eSign request to ESP
2. ASP initiates eSign request and ESP authenticates user for eKYC before eSign

3.1.1. eSign using ASP provided KYC data

Flow of eSign process using this option:


SHA 256 Hash of document
Construct eSign XML with attributes and hash
Sign XML
Create Request XML & send to ESP via HTTP API
ASP receives / validates response and attaches signature to document
Perform eKYC & Decrypt it, if encrypted
Base 64 Encoded with eKYC Provider signature intact
ESP
Performs eSign and gives response

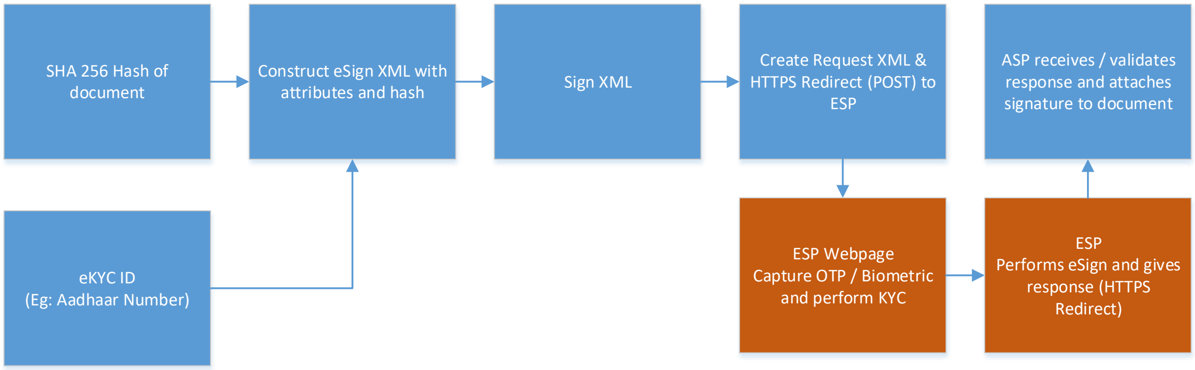
In this scenario:

1. ASP client application asks the eSign user to sign the document
2. ASP client application creates the document hash (to be signed) on the client side
3. ASP client application has / asks-for e-KYC identification Number
4. ASP client application performs e-KYC of user with allowed e-KYC provider.
5. ASP client application asks the eSign 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 verifies the Digital signature and Time of the e-KYC response corresponding to the e-KYC identification Number provided by eSign user
    c. ESP logs the transaction
    d. ESP creates a new key pair and CSR for eSign user.
    e. ESP calls the CA service and gets a Digital Signature Certificate for eSign user. The certificate will be a e-KYC class Digital Signature Certificate, which has eKYC number, Name of the eSign user, e-KYC response code, Authentication Type, and Time Stamp embedded.
    f. ESP signs the 'document hash' and provides to the ASP.
8. ASP receives the document signature and the eSign user's Digital Signature Certificate.
9. ASP client application attaches the signature to the document.
10. eSign user can accept or reject the signature and DSC

3.1.2. eSign using e-KYC made by ESP

Flow of eSign process using this option:

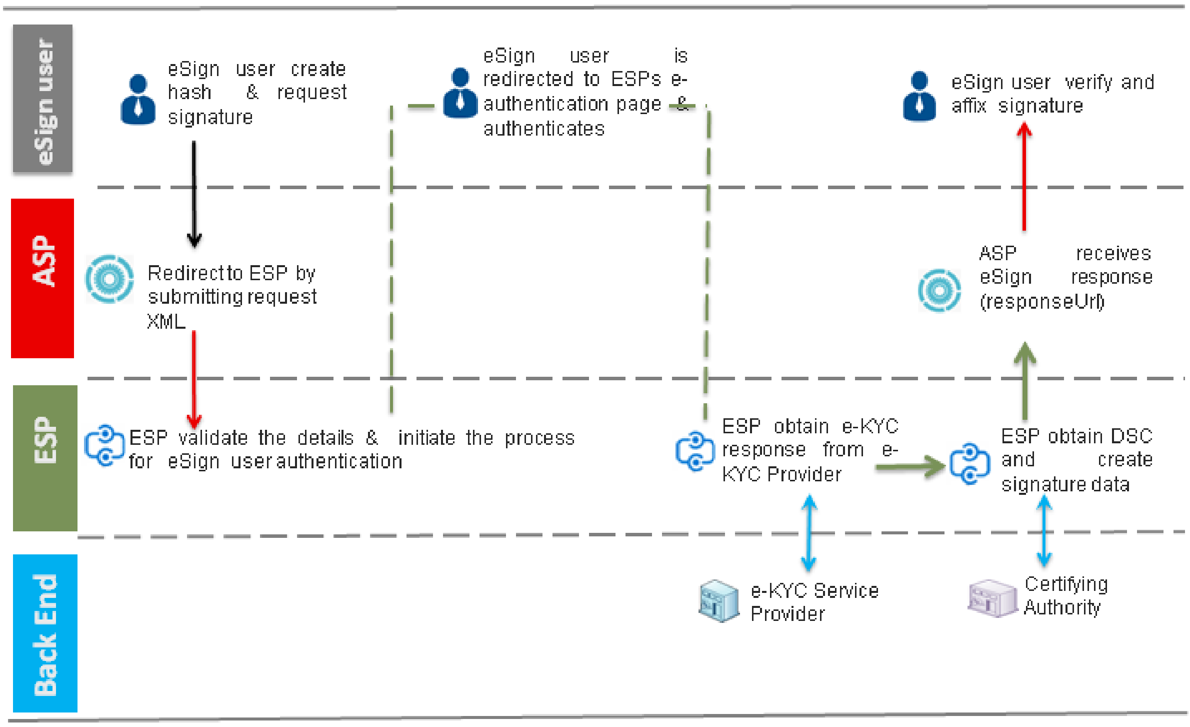

SHA 256 Hash of document
Construct eSign XML with attributes and hash
Sign XML
Create Request XML & HTTPS Redirect (POST) to ESP
ASP receives / validates response and attaches signature to document
eKYC ID
(Eg: Aadhaar Number)
ESP Webpage
Capture OTP / Biometric and perform KYC
ESP
Performs eSign and gives response (HTTPS Redirect)

In this scenario:

1. ASP client application asks eSign user to sign the document
2. ASP client application creates the document hash (to be signed) on the client side
3. ASP client application has / asks-for e-KYC identification Number
4. ASP client application asks the eSign user to provide consent for certificate generation and signature
5. ASP forms the input data for eSign API
6. ASP redirect to ESP application by submitting request XML
    a. ESP validates the calling application and the input.
    b. ESP verifies the Digital signature of ASP for eSign XML received
    c. ESP logs the transaction
    d. ESP redirects eSign user to e- authentication page
    e. ESP performs authentication and get e-KYC information from e-KYC provider
    f. ESP show the document hash along with document information to eSign user.
    g. ESP creates a new key pair and CSR for eSign user.
    h. ESP calls the CA service and gets a Digital Signature Certificate for eSign user. The certificate will be a e-KYC class Digital Signature Certificate, which has e-KYC number, Name of the eSign user, e-KYC response code, Authentication Type, and Time Stamp embedded.
    i. ESP signs the 'document hash' and provides response XML to the ASP by redirecting to ASP's response URL.

7. ASP receives the document signature and the eSign user's Digital Signature Certificate.
8. ASP client application attaches the signature to the document.
9. eSign user can accept or reject the signature and DSC

The web page flow for eSign using e-KYC made by ESP is as given below


eSign user
ASP
ESP
Back End
eSign user create hash & request signature
eSign user is redirected to ESPs e-authentication page & authenticates
eSign user verify and affix signature
Redirect to ESP by submitting request XML
ASP receives eSign response (responseUrl)
ESP validate the details & initiate the process for eSign user authentication
ESP obtain e-KYC response from e-KYC Provider
ESP obtain DSC and create signature data
e-KYC Service Provider
Certifying Authority

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 HTTPS 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 the requests and responses are digitally signed.

The usage of HTTPS shall ensure transport layer encryption, while digital signing of XML shall ensure integrity & authenticity of data.

Following is the URL format and the parameters for eSign service:

API URL	ESP shall expose two (2) URLs as under: 1. URL1: HTTPS API URL for 'preverified' requests. 2. URL2: HTTPS redirect URL for requests where ESP has to perform electronic KYC of eSign user.
Protocol	HTTPS
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. eSign API: Input Data Format - eSign Service

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

3.3.1. eSign XML structure

Following is the XML data format for eSign XML.

```
<Esign ver="" sc="" ts="" txn="" ekycMode="" ekycId="" ekycIdType=""
aspId="" AuthMode="" responseSigType="" preVerified=""
organizationFlag="" responseUrl="">
     <Docs>
          <InputHash  id=""  hashAlgorithm=""  docInfo="">Document
     Hash in Hex</InputHash>
     </Docs>
     <AspKycData>base-64 encoded digitally-signed ekyc XML as per
     e-KYC provider specifications</AspKycData>
     <OrgDetails>Base-64 encoded Organization XML with Authorized
     signatory's signature.</OrgDetails>
     <Signature>Digital signature of ASP</Signature>
</Esign>
```

3.3.1.1. Element Details

Element Name: Esign

- Description: Root element of the eSign 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 “2.0”.
2.	sc	mandatory	sc – (mandatory) ASP should have taken a clear consent from ‘eSign user’ to carry on eSign from their front ending application. This attribute represents signatory’s explicit consent is obtained by ASP for using the signatory’s identity and address data received from e-KYC provider to, 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”. Check box[Y/N*] *No by default
3.	ts	Mandatory	Request timestamp in ISO 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.	e-KYCMode	Mandatory	This represents the mode of e-KYC being used. The value can be any one out of below: 1. UIDAI = U
6.	e-KYCIdType	Mandatory	This represents the type of e-KYC ID being used. The value can be any one out of below: 1. Aadhaar = A
7.	ekycId	Mandatory	ekyc identity Number of the eSign user. The value can be any one out of below: 1. 12 digit Aadhaar Number
8.	aspId	Mandatory	Organization ID issued by ESP to the ASP
9.	AuthMode	Mandatory	Authentication Mode being used for e-KYC Authentication, either to be performed by ESP, or as already made by the ASP. Allowed values are: • OTP = 1 • Fingerprint = 2 (only for eKycMode = U) • IRIS = 3 (only for eKycMode = U) Class of eSign Certificate will be “OTP Class” for OTP based authentication. For Fingerprint and IRIS, the class will be “Biometric Class”. These will be as per definitions of “Class of Certificate” in India PKI-CP.
10.	responseSigType	Mandatory	This value represents the response signature type, where ASP can request for specific type of signature, like Raw or PKCS7. Allowed Values are: 1. rawrsa 2. pkcs7 Examples: responseSigType="rawrsa" responseSigType="pkcs7"
11.	preVerified	Mandatory	Represents whether ASP has already made a e-KYC transaction towards the eSign user. (Not older than 15 minutes) Allowed values are y and n representing yes & no. If Yes, the sub-element AspKycData is required. ESP/CA will verify the integrity by validating e-KYCProvider signature on AspKycData, and then use respective KYC information to generate Digital Signature Certificate.

			Else, ESP should authenticate the user and obtain KYC information from available eKycMode.
12.	organizationFlag	Optional	Represents that the request is for an Organization Certificate. Allowed values are y and n, representing yes or no. If yes, the sub-element OrgDetails is required. ESP shall operate a verified information of Organization and have the authorized signatory certificate mapped in the system. The OrgDetails XML signature should be verified against the certificate available at ESP end, for the data integrity and signature.
13.	responseUrl	Optional	This is mandatory, if preverified is set to 'no'. This should contain a valid HTTPS URL of the ASP, to which ESP has to redirect back to ASP with response XML.

Element Name: Docs

- Description: Contains minimum 1, maximum of 10 sub-elements with Document Hash
- Requirement of tag: Mandatory
- Value: Sub-elements
- Attributes: Not applicable

Element Name: InputHash

- 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: Table below

Sl No	Attribute	Required?	Value
1.	id	Mandatory	Contains running serial number for document hashes. This should start at 1, and should be sequential, maximum up-to 10. This will be the key to correlate the response signature of each document.
2.	hashAlgorithm	Mandatory	Should be fixed to "SHA256"
3.	docInfo	Mandatory	Description for the respective document being signed, not more than 50 characters.

Element Name: AspKycData

- Description: Contains the Base 64 representation of digitally-signed e-KYC XML as per e-KYC provider specifications
- Requirement of tag: Mandatory, if preVerified attribute is set to YES. Else not required.
- Value: Base-64 encoded subset of KYC Data XML, received from e-KYC Provider. Please refer to respective e-KYC Provider specifications for the structure of XML. The digital signature of e-KYC provider should be verifiable by ESP.
- Attributes: Not applicable

Element Name: OrgDetails

- Description: Contains the Base 64 representation of Organization Details. ESP shall operate a verified information of Organization and have the authorized signatory certificate mapped in the system. The OrgDetails XML signature should be verified against the certificate available at ESP end, for the data integrity and signature.
- Requirement of tag: Mandatory, if organizationFlag attribute is set to YES. Else not required.
- Value: Base-64 encoded Organization XML with Authorized signatory's signature.
- 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 eSign user 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 e-KYC XML level is mandatory .The eSign request XML should be digitally signed by ASP for authentication purposes.

3.3.2. OrgDetails XML structure

Following is the representative XML data format for OrgDetails XML.

Note:

1. e-KYC Number mentioned here should match with the e-KYC Number in eSign element.
2. The name of applicant will be considered from here, instead of e-KYC Number, for the purpose of certificate generation.
3. All information in this XML, except PAN and Organization Unit are mandatory.

```
<OrganizationDetails>
      <PersonInfo>
             <Aadhaar></Aadhaar>
            <ApplicantName></ApplicantName>
            <Designation></Designation>
            <ApplicantPan></ApplicantPan>
      </PersonInfo>
      <OrgInfo>
            <OragnisationName></OragnisationName>
            <OrganisationUnit></OrganisationUnit>
            <OrganisationAddress>
                  <Street></Street>
                  <State></State>
```

```
                <Pincode></Pincode>
           </OrganisationAddress>
      </OrgInfo>
      <Signature>Digital  Signature  of  Authorized  Signatory  of
      corresponding organization, verified by ESP</Signature>
</OrganizationDetails>
```

3.4. eSign 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 eSign user.

```
<EsignResp status="" ts="" txn="" resCode="" errCode="" errMsg="">
      <UserX509Certificate>base64 value of eSign user certificate
(.cer)</UserX509Certificate>
      <Signatures>
           <DocSignature id="" sigHashAlgorithm="SHA256" error="">
                Signature data in raw (PKCS#1) or PKCS7 (CMS)
           signature as requested
           </DocSignature>
           .
           .
           <DocSignature id="" sigHashAlgorithm="SHA256" error="">
                Signature data in raw (PKCS#1) or PKCS7 (CMS)
           signature as requested
           </DocSignature>
</Signatures>
<Signature>Signature of ESP</Signature>
</EsignResp>
```

ASP should provide mechanism to verify and accept the contents of DSC.

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: UserX509Certificate

- Description: This element will contain the Base 64 value of the Certificate. No private key information is shared. For manual verification, this value can be copied and saved as .cer file (With begin and end statements - PEM Format).
- Presence: Mandatory, if success.
- Value: Base 64 value of eSign user certificate (public).
- Attributes: Not Applicable

Element Name: Signatures

- Description: This element contains the sub-elements of signatures corresponding to InputHash.
- Presence: Mandatory, if success.
- Value: Sub-elements.
- Attributes: Not Applicable

Element Name: DocSignature

- Description: This element will contain the signed value which will be verifiable against original document.
- Presence: Mandatory
- Value: Signed value in raw (PKCS#1) or PKCS7 (CMS) signature format as per the request XML.
- Attributes: Table Below

Sl No	Attribute	Value
1.	Id	Contains the corresponding ID to the Input Hash received
2.	sigHashAlgorithm	Should be fixed to “SHA256”
3.	error	In case of failure, this will contain the corresponding error

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 eSign user.

Annexure 1

eSign Service -error codes

Sl No	Error Code	Error message	Originator
1	ESP-901	Invalid Authentication Mode	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

Annexure 2

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

Change History

Change History			
Section	Ver	Date	Modification