eSign API Specifications

Version 3.0

18 Jan 2019

Controller of Certifying Authorities

Ministry of Electronics and Information Technology

Document Control

Document Name	eSign API Specifications
Status	Release
Version	3.0
Release date	26.12.2018
Last update	22.02.2019
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 II & III certificates, 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 prescribed e-KYC guidelines

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 based on e-KYC response and a confirmation by CA, 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.

The prescribed modes of user verification may be online or offline. The 3.x version will be for offline verification and 2.x version will be for online verification In the case of offline user verification, the e-KYC service will be provided by CA and one time registration of user is required. Both 2.x and 3.x versions of API are designed for applying Digital Signature based on the response received from e-KYC service after online authentication of eSign user.

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 for offline verification. This includes 3.x API Data format, protocol and other related specifications.

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 or eKYC user or subscriber" 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, 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 and eSign

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. 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 integrated 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/eSign user id shall mean the unique identification such as username/number/id maintained by e-KYC provider to uniquely identify user;

e-KYC provider shall mean any e-KYC provider listed in e-Authentication Guidelines. eKYC provider is responsible for eKYC user management and authentication eSign user. In case CA maintains eSign User Accounts of subscribers/eSign user, the security and privacy will be applicable as per the provisions specified under IT Act.

'**OTP'** shall mean one-time password either sent to or generated on the eSign User's cell phone for the purpose of authentication, including SMS OTP, Time based OTP (TOTP), or any other secure OTP bound token generation methods;

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).

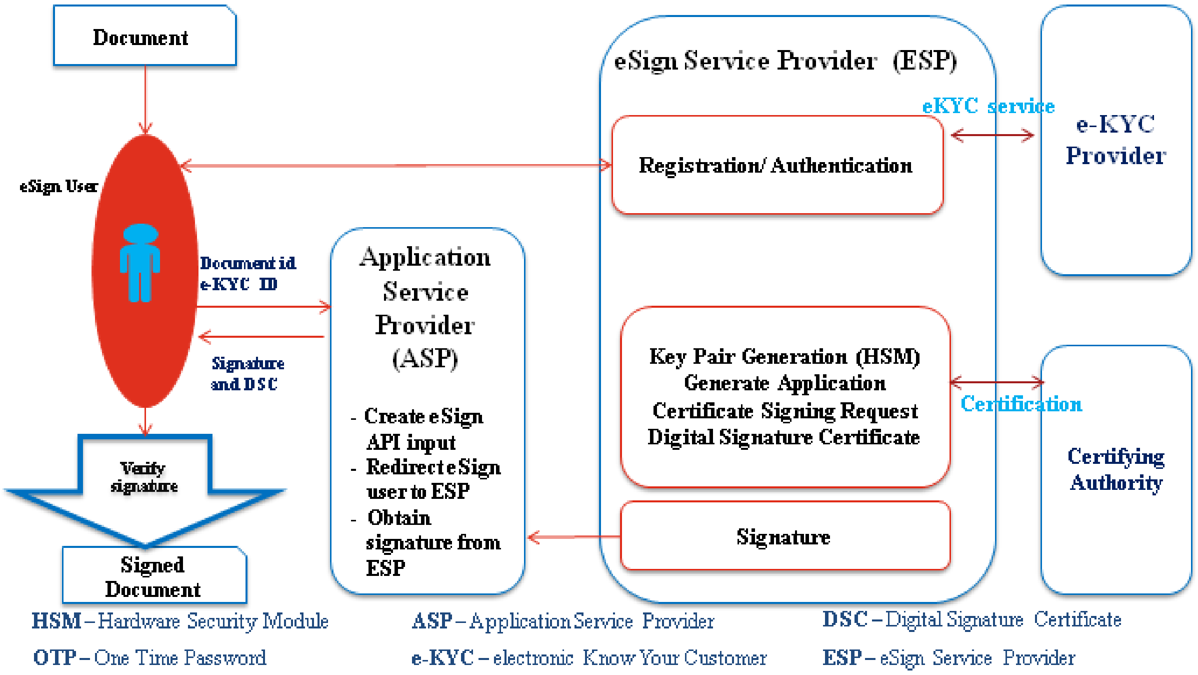
1.4. Legal Framework

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



Document
eSign User
Document id
e-KYC ID
Signature and DSC
Verify signature
Signed Document
Application Service Provider (ASP)
- Create eSign API input
- Redirect eSign user to ESP
- Obtain signature from ESP
eSign Service Provider (ESP)
eKYC service
Registration/ Authentication
e-KYC Provider
Key Pair Generation (HSM)
Generate Application
Certificate Signing Request
Digital Signature Certificate
Certification
Certifying Authority
Signature
HSM – Hardware Security Module
OTP – One Time Password
ASP – Application Service Provider
e-KYC – electronic Know Your Customer
DSC – Digital Signature Certificate
ESP – eSign Service Provider

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 has been carryout and the digitally signed e-KYC response is made available to ESP. The authentication needs to be carried out independent of section 3

The suggested method for obtaining authenticated e-KYC response is

> ESP facilitates authentication of eSign user by calling authentication URL of eKYC provider. The e-KYC response will be received by ESP and performs eSign on the eSign request received from ASP within permissible time limit.

3.1. eSign - Usage scenarios

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

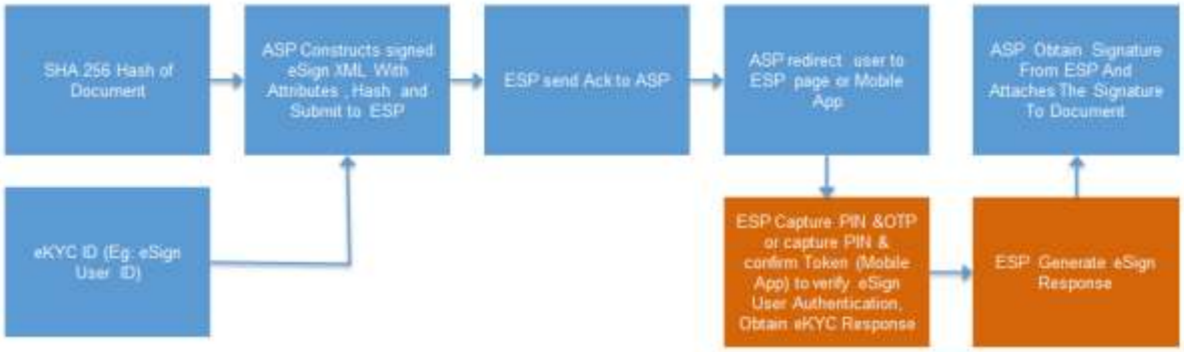
ASP provides eSign facility to public should integrate with all other ESPs within one month after on-boarding with first ESP.

The eSign service API can be used in the scenario where ASP initiates eSign request and ESP authenticates user for eKYC before eSign through eKYC provider.

3.1.1. eSign using e-KYC made by ESP

eSign 3.0 uses asynchronous API for request and response. ASP calls the ESP signing request API, later (post signature authorization by subscriber) ESP will call back ASP and provide the signature status and data.

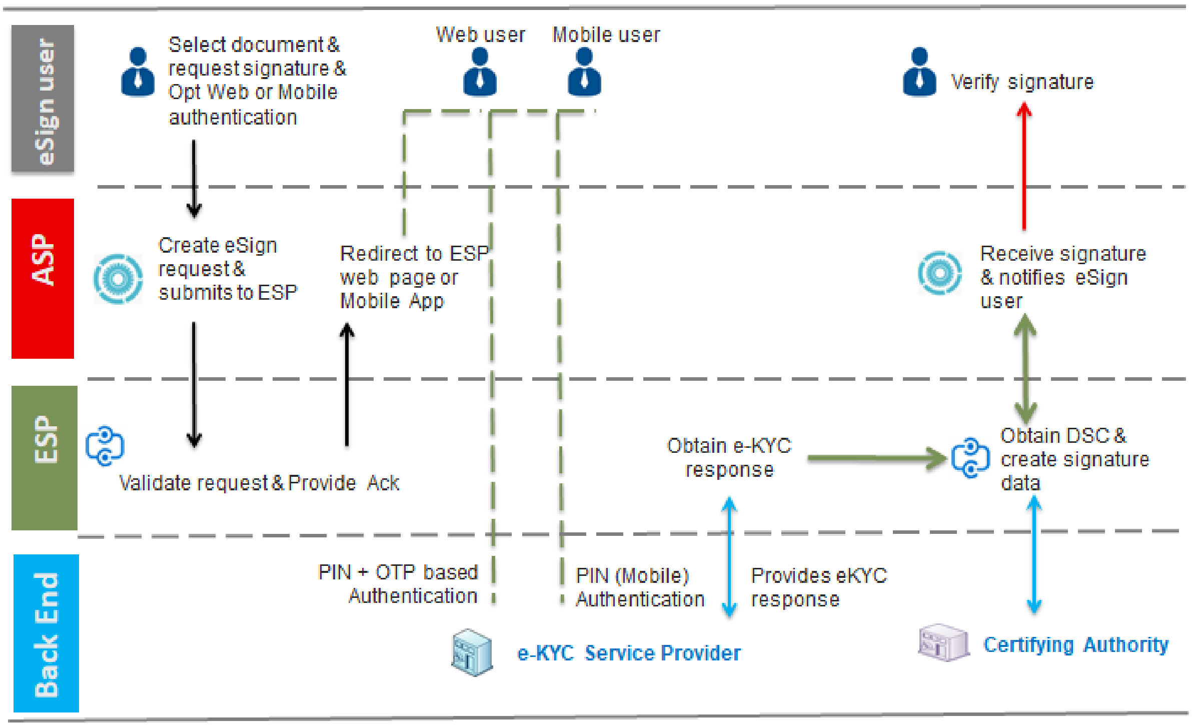
Flow of eSign process using this option:


SHA 256 Hash of Document
ASP Constructs signed eSign XML With Attributes , Hash and Submit to ESP
ESP send Ack to ASP
ASP redirect user to ESP page or Mobile App
ASP Obtain Signature From ESP And Attaches The Signature To Document
eKYC ID (Eg: eSign User ID)
ESP Capture PIN &OTP or capture PIN & confirm Token (Mobile App) to verify eSign User Authentication, Obtain eKYC Response
ESP Generate eSign Response

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 asks the eSign user id for certificate generation and signature.
4. ASP forms the input data for eSign API
5. ASP calls ESP's URL and submit 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 acknowledges the request back to ASP by providing an ack response with same txn ID. At this time ASP can close the connection to ESP.
6. ASP redirects the user to ESP’s authentication page. Alternatively, User can use ESP’s mobile app to authenticate. ASP shall suitably display necessary information.
    a. ESP displays e-authentication page (if web flow) or notifies on ESP mobile app to the eSign user.
    b. ESP performs authentication using OTP (SMS/TOTP for web flow or OTP bound token for ESP mobile app) along with PIN and get e-KYC information from e-KYC provider.
    c. ESP shows the document hash along with document information to eSign user.
    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.
    f. ESP signs the ‘document hash’
    g. ESP calls ASP’s response URL or redirects to response URL (which was part of eSign request) with signed XML response.
    h. If ASP has provided ‘redirectUrl’, ESP redirects the user back to ASP’s web page (web flow).
    i. In case response is not received by ASP or user session ends within ASP, ASP can check status of signing request using “checkStatus” API using the same txn ID of the request.
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. ASP shall provide a choice to user to obtain signed document via email, download, short URL sent via SMS, etc.

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


eSign user
ASP
ESP
Back End
Select document & request signature & Opt Web or Mobile authentication
Web user
Mobile user
Verify signature
Create eSign request & submits to ESP
Redirect to ESP web page or Mobile App
Receive signature & notifies eSign user
Validate request & Provide Ack
Obtain e-KYC response
Obtain DSC & create signature data
PIN + OTP based Authentication
PIN (Mobile) Authentication
Provides eKYC response
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 URL as HTTPS endpoint
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="" signerid="" ts="" txn="" maxWaitPeriod="" aspId="" responseUrl="" redirectUrl=""
signingAlgorithm="">
        <Docs>
                <InputHash id="" hashAlgorithm="" docInfo="" docUrl=""
                responseSigType="">Document Hash in Hex</InputHash>
        </Docs>
        <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 "3.0".
2.	signerid	Optional	Format: id@id-type.esp-id ASP collects the ID of the signer, along with ID type and ESP Name. ASP may make it intuitive for user to select their required ID type and then specify the value. Allowed ID Types: username, Mobile, PAN If mobile is the id-type, then mobile number should be same as in the eKYC XML. Allowed ESP ID: Unique Identifiers specified by CCA for each empanelled ESP. ASP should construct the signerid based on ID given by user and selected ID type and ESP. This information shall be used by ESP to validate and then prepopulate the username for the convenience. ESP should not allow modification of the username in their screen. If signerid is not present, ESP may facilitate the new signer id creation through eKYC provider, however

			authentication of user should be carried out before signing.
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. Should be unique for the given ASP-ESP combination for that calendar day.
5.	maxWaitPeriod	Mandatory	Expiry time in minutes. This is maximum wait time for the ESP to allow Signer to complete the signing. In case the user does not sign within ASP's expected duration, ESP should mark the transaction as error 'User timeout' error code. Default = 1440 minutes
6.	aspId	Mandatory	Organization ID of ASP
7.	responseUrl	Mandatory	ASP URL to receive the response from ESP. This should be a valid URI accessible from ESP system to make a call and submit the response XML packet using HTTP(S)-POST with Content-Type as application/xml. On success or failure including cancellation by user, ESP shall perform a background call to this response URL with 'eSign Response Format' which contains the status success/failure (status = 1/0).
8.	redirectUrl	Optional	ASP URL to redirect the user after completion of transaction. This is supported only in case where ASP uses redirection to ESP authentication page. If present, ESP shall redirect the user back to ASP's designated URL. Such redirection shall have a HTTP(S)-POST and Content-Type of 'application/x-www-form-urlencoded' with parameter of 'txnref' containing concatenated transaction ID and responseCode (separated with a pipe character) in base 64 encoding. txnref=Base64(transaction ID + "|" + responseCode)
9.	signingAlgorithm	Mandatory	This value represents the signature Algorithm. End user certificate generation (DSC) shall also be based on this algorithm. Allowed Values are: 1. ECDSA 2. RSA

Element Name: Docs

- Description: Contains one sub-element 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	The index number of the document. Should start with one. Maximum 5. Should be sequential. Shall not repeat.
2.	hashAlgorithm	Mandatory	Should be fixed to "SHA256"
3.	docInfo	Mandatory	Description for the respective document being signed, not more than 50 characters. docInfo should be strictly adhere to the content of document. Multiple documents of same type or different types should not be included in a single file. ESP shall display this information against docUrl, so that user can identify the same.
4.	docUrl	Mandatory	URL of the document. Should be a HTTP / HTTPS URL for the document, accessible by the signer during the transaction permitted duration (maxUserWait Time). ESP shall display this URL with hyperlink, so that user can access the document to view.
5.	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. raw 2. pkcs7

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

3.4. eSign: User Authentication Page

Once ASP submits the Request XML, ESP provides a ‘pending for completion’ (status=2) response which will contain the response code (as an acknowledgement). At this stage, ASP is expected to guide the user with proper information as under:

1. Redirect the user to the authentication page of the ESP.
2. Provide information to the user to authenticate over ESP’s mobile app. (ESP may also support push notification for mobile app users, and allowing to authenticate on mobile through eKYC provider)

In case of redirection (browser based flow), ESP shall expose a redirection URL with following specifications.

API URL	ESP shall expose URL as HTTPS redirection page.
Protocol	HTTPS
Method	POST
Content-Type	application/x-www-form-urlencoded
Parameter name	txnref
Parameter Value	Concatenated transaction ID and responseCode (separated with a pipe character) in base 64 encoding.
Example format	txnref=Base64(transaction ID + “|” + responseCode)

3.5. eSign API: Response Data Format - eSign Service

Below is the response format of eSign Service API. This response shall be used in following situations:

1. Once the subscriber authorizes (or cancels or expire), ESP shall provide a completed response to the ASP on the responseUrl (status = 1/0).
2. ESP shall also respond to ‘Check Signing Status’ API call with this response format including ‘pending for completion’ statuses.

Note that, the API does not give any identity related data of the eSign user.

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

3.5.1. 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	Presence	Value
1.	ver	Mandatory	Should be set to 3.0
2.	status	Mandatory	In case of success, it will be "1" In case of failure, it will be "0" In case of pending for completion, it will be "2"
3.	ts	Mandatory	Will contain the response timestamp in ISO format.
4.	txn	Mandatory	The Transaction ID provided by ASP in the request.
5.	resCode	Mandatory	A unique response code provided by ESP. This is a unique id for the transaction (eSign user authentication & eSign request) provided by ESP. It shall make the transaction traceable, and ASP is expected to store this code in their audit log. The response code shall be maintained same for particular transaction. Being asynchronous, there may be need for providing the response multiple times including the acknowledgement stage and final response stage. All the responses shall carry same response code for the particular transaction.
6.	error	Optional	In case of failure, this will contain an error code. OR blank, in case of success.

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 raw(ECDSA) or PKCS7 (CMS) signature format as per the request XML.
- Attributes: Table Below

Sl No	Attribute	Presence	Value
1.	Id	Mandatory	Contains the corresponding ID to the Input Hash received
2.	sigHashAlgorithm	Mandatory	Should be fixed to “SHA256”
3.	error	Optional	In case of failure, this will contain an error code. OR blank, in case of success. ESP shall provide necessary option for signer to uncheck any document hash. Such unchecked document hash shall not be signed and shall be returned with an error called “User Rejected”.

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

3.6. eSign API: Check Signing Status - Request

This is an additional option for ASP to check the status of the transaction, in case necessary.

On a successful & timely flow, ESP will automatically call back the ASP’s responseUrl with necessary eSign response. However, in case of any need, ASP can call the signing status API and receive the response again.

ESP shall provide this service for minimum of 30 days from the date of transaction, for the ASP.

3.6.1. Request XML format

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

3.6.1.1. 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	Should be set to 3.0
2.	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.
3.	txn	Mandatory	Transaction ID of the ASP provided in original request.
4.	aspId	Mandatory	Organization ID of the ASP

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

This will respond with an eSign response data as defined in this document. The status attribute of the response will indicate the success or pending for completion.

4. eKYC Service requirements

CA shall implement a comprehensive eKYC service to fulfil the KYC requirements of eSign user.

Important points to consider:

1. eKYC system shall be a protected and shall not be exposed to any external services directly.
2. The access of eKYC information shall be on need basis for the services prescribed.
3. The access to such information by other services shall be bound by authentication of eSign user by two factors, namely the PIN and an OTP.
4. The information of PIN shall not be stored in plain text format. The authentication of PIN shall be always verified after compare against the stored value.
5. The PIN information in plain text shall not be part of any logs or data monitoring systems.

4.1. Functions of eKYC Service

eKYC Service shall operate with the minimum required functions.

The functions shall include:

1. Creation of eSign user account
2. Fetch eSign user information by ESP / CA systems (with user authentication)
3. Trigger OTP to the user
4. Mobile application based Access tokens
5. eSign user functionalities

4.2. Creation of eSign user account

eKYC system shall provide provision for online enrolment to eSign users and the same should be able access through ESP page or ASP applications. Such enrolment is bound by procedures and requirements defined under Identity Verification Guidelines. Username, Mobile number and PAN should be unique.

On successful enrolment of an eKYC User, following data eSign user information is recorded in eKYC user account. These fields are subject to verification against the prescribed 'Verified Source'. (Aadhaar Offline XML)

Aadhaar Offline XML shall be verified on its receipt for a valid digital signature by UIDAI.

Sl No	Field name	Description	Source	Additional Actions
1.	Username	eSign user Id in the format prescribed.	User entry	Should be Unique
2.	PIN	PIN of the user	User entry	Meet the requirements laid down in IVG
3.	Name	Name of the eKYC user	Aadhaar Offline XML	
4.	Mobile	Mobile Number of the user	User entry	This shall be checked with the Aadhaar Offline XML using the hashing process defined by UIDAI
5.	Email	Email ID of the user	User entry	If a value is present in Aadhaar Offline XML, the value entered by user shall be verified against it using hashing process defined by UIDAI. Else, ESP shall send an Email OTP and validate it for verification. This field is optional.
6.	Address	Address of the eKYC user	Aadhaar Offline XML	Shall be concatenated with the fields from Aadhaar Offline XML, to form an address, excluding State, Country and Postal Code.
7.	StateProvince	StateProvince of the eKYC user	Aadhaar Offline XML	
8.	Country	Country of the eKYC user	Aadhaar Offline XML	
9.	Postal Code	Postal Code of the eKYC user	Aadhaar Offline XML	
10.	Photograph	Photograph of the eKYC user	Aadhaar Offline XML	Shall be verified against Video, as prescribed under IVG
11.	DOB	DOB of the eKYC user	Aadhaar Offline XML	
12.	Gender	Gender of the eKYC user	Aadhaar Offline XML	
13.	PAN	PAN Number of eKYC user	User entry	CA should verify the PAN by the verification service provided by Income Tax.

				CA should preserve the evidence of verification with their digital signature. This field is optional.

4.3. Access to eKYC data

The eKYC user information shall be allowed to access for eSign process and DSC issuance. For access of such data for eSign process, ESP shall implement necessary rest API based eKYC request, as per the formats provided under this section.

The audit logs (both success & Failure) of eKYC user authentications shall be maintained by eKYC Provider with timestamp and user id. The maximum retries with failed authentication by a user (for specific transaction) shall be limited to 5 attempts.

Following is the URL format and the parameters for eKYC access:

API URL	ESP shall consume an 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.

4.3.1. eKYC request format

```
<eKycReq ver="" signerid="" ts="" txn="" otp="" mobileAccessToken="" pinhash="">
        <Signature>Digital signature of ESP</Signature>
</eKycReq>
```

4.3.1.1. Element Details

Element Name: eKycReq

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

Sl No	Attribute	Requirement	Value
1	ver	Mandatory	Should be set to 3.0
1	signerid	Mandatory	Signer ID entered by the user
2	ts	Mandatory	Will contain the request timestamp in ISO format.
3	txn	Mandatory	A unique transaction ID created by ESP system to request respective KYC data
4	otp	Optional	OTP entered by the user on the webpage. This should NOT be in a plain text. ESP shall implement any methods to replace plain text with encryption/hashing techniques. Shall be blank in case of mobile based authentication.
5	pinhash	Mandatory	PIN entered by the user. This shall be the hash of the PIN, further hashed after prefixing the txn value.

			pinhash = hash(txn + hash(pin))
6	mobileAccessToken	Optional	Access Token registered for the mobile App. This shall be present in the case user authenticates using the ESP mobile app (Mobile app should meet the requirements prescribed in this document.) Shall be blank in case of OTP based authentication.

Element Name: Signature

- Description: This element will contain the signature of ESP, which can be used for verification by eKYC system.
- 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. eKYC response format

```
<eKycResp ver="" status="" signerid="" ts="" txn="" error="" respCode="" >
        <kycData name="" mobile="" email="" address="" stateProvince="" country=""
        postalCode=""  PAN=""  DOB="" Gender=""/>
        <Photo>base64 encoded JPEG photo of the eKYC account holder</Photo>
        <Signature>Digital signature</Signature>
</eKycResp >
```

Element Details

Element Name: eKycResp

- 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.	ver	Should be set to 3.0
1.	status	In case of success, it will be “1” In case of failure, it will be “0”
2.	signerid	Signer ID sent in the request.
3.	ts	Will contain the response timestamp in ISO format.
4.	txn	The Transaction ID provided in the request.
5.	error	In case of failure, this will contain a descriptive error message. OR blank, in case of success.
6.	respCode	Unique eKYC response code given by KYC system. This shall form as a permanent reference to the log towards traceability of the transaction.

Element Name: kycData

- Description: This element contains the KYC information.
- Value: Not Applicable
- Attributes: Table below

Sl No	Attribute	Presence	Value
1.	name	Mandatory	Name of the eKYC account holder.
2.	mobile	Mandatory	Mobile Number of the eKYC account holder

3.	email	Optional	Email ID of the eKYC account holder
4.	address	Mandatory	Address of the eKYC account holder
5.	stateProvince	Mandatory	State or the Province of the address
6.	country	Mandatory	Two-character ISO representation of the country. Eg: IN=India
7.	postalCode	Mandatory	Postal code of the address
8.	PAN	Optional	PAN Number of eKYC account holder
9.	DOB	Optional	DOB Number of eKYC account holder
10.	Gender	Optional	Gender of eKYC account holder

Element Name: Photo

- Description: This element will contain the Photo of eKyc account holder.
- Presence: Mandatory
- Value: Base 64 encoded JPEG photograph of the eKYC account holder.
- Attributes: Not Applicable

Element Name: Signature

- Description: This element will contain the signature of eKYC system, which can be used for verification by ESP 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

4.4. OTP Functionality

ESP should implement an internal secure API communication, in order to send OTP to the user.

Alternatively, ESP may also implement Time Based OTP (TOTP) functionality using compliant TOTP authenticators and/or ESPs own authenticator app through eKYC provider. This shall implement secure TOTP in compliance with RFC 6238. The number of steps to support client clock drifts shall not exceed + or – 1 step. It shall also support the time step of either 30 or 60 seconds only.

ESP can also implement ESP mobile app based authentication, as an alternate to OTP.

Important notes on SMS OTP functionality:

1. The architecture requires ESP system to request KYC system for sending OTP to requesting user.
2. Such communication shall include minimum of the user name in the request, and provide an acknowledgement to ESP system on successful trigger.
3. The response to ESP system shall not share the OTP.
4. The OTP shall be valid for maximum of 15 minutes, and shall not be logged in any place other than for validation of OTP in authentication request.
5. ESP shall implement necessary process to avoid more than one OTP trigger within a span of one minute, unless last OTP was successfully consumed.
6. OTP shall be send with purpose and the purpose should be part of audit logs.

4.5. Mobile application based Access tokens

Towards an improved user experience, ESP may offer mobile based authentication. The requirements for the same need to be fulfilled as under, in order to qualify for Mobile based authentication. Such qualified mobile app-based authentication shall be equivalent to OTP authentication. Thereby, any such signing process shall have PIN + Mobile app authentication to fulfil KYC data access.

Requirements to be fulfilled by such mobile applications:

1. Mobile app shall be owned and operated by ESP with complete control on its code, architecture, security and publishing requirements.
2. Mobile app shall support largely used Mobile operating systems. However, it shall not support any operating systems or its versions, which are known to have security issue or deprecation.
3. Mobile app shall have a secure architecture and undergo vulnerability assessments to avoid any exploitation.
4. User shall login to the mobile app using a secure procedure involvement PIN + OTP authentication. This first time usage shall have a secure layer to create and make a handshake with KYC server with generation of a unique Access Token.
5. Such access token shall be generated in the mobile device in a secure area / element supported by the platform, and shared with eKYC server for enrolment of the device against that eKYC user.
6. Access Token shall be marked for expiry in 3 months from its last successful usage. In case of expired access token, mobile app shall clear the local Access Token and force the user for fresh login.
7. System shall support multiple access Tokens against one eKYC user, towards supporting multiple mobile devices.
8. Subscriber portal shall provide necessary option for accessToken history and revocation of accessTokens.
9. Any signing transaction ‘waiting for user authentication’ shall be queued and shown separately on the mobile app. It is also recommended to show new signing transactions as a push notification.
10. User shall be able to open the mobile app (with or without a local sign in functionality) and confirm the signature with PIN authentication.
11. Mobile app may also support additional eSign user functions using same level of security required for eSign user portal.
12. Mobile app should be secure enough to avoid any kind of access breach, or any kind of hacks to gain direct access to the token and the eKYC server endpoint consumed by such mobile app.

4.6. Subscriber Functionalities

ESP shall offer a subscriber portal to meet the following requirements through eKYC provider.

1. PIN change functionality
2. Signing History
3. Request to update mobile number (Requires fresh Video verification)
4. Other modifications to user data

This portal shall implement single factor authentication including either PIN, or OTP, or Mobile app to login to the system.

The portal shall be secured and permit minimum of the requirements stated in this section. Any request for modifications to KYC data shall undergo necessary verification procedures laid down by CCA.

5. Error Codes

Below are the standard error codes for various types of failures. ASP application shall use the error codes to identity the cause of failure and display / take necessary action.

5.1. eSign Response

Error Code	Error Message
101	Invalid Request Format
102	Invalid Signer ID
103	Invalid Version
104	XML Signature validation failed
105	Invalid transaction ID
106	Invalid ASP ID
107	Invalid Digital Signature
108	Minimum one document is required
109	Request exceeds Maximum number of documents allowed
110	Invalid Timestamp
111	Invalid Maximum Wait Period
112	Duplicate Transaction ID
113	User Timeout. Maximum Wait Time expired.
114	Authentication failed. User credentials invalid.
199	Unknown error / Custom error from ESP

5.2. eSign Document Level Response

Error Code	Error Message
201	Invalid Document Hash
202	Invalid response signature type
203	Invalid document URL
204	Invalid document information
205	Invalid hash algorithm
206	Document cancelled by user
299	Unknown error / Custom error from ESP

5.3. eSign Status Check Response

Error Code	Error Message
301	Invalid Request Format
302	Transaction number not found
303	Invalid Version
399	Unknown error / Custom error from ESP

6. Change History

Change History			
Section	Ver	Date	Modification
3.3.1	3.0	18.01.2019	Request XML => "Signing Algorithm" parameter added
3.3.1.1	3.0	18.01.2019	Definition for " Signing Algorithm" added Definition change for "Response Signature Type"
4.3	3.0	18.01.2019	Maximum failed attempts specified (The audit logs ....limited to 5 attempts.)
5.1	3.0	18.01.2019	error message
3.3.1	3.0	22.02.2019	signerid row, under value Coloum, the following is inserted *"If mobile is the id-type, then mobile number should be same as in the eKYC XML"*
3.5	3.0	22.02.2019	In the XML Header, status="" is added and "err" replaced with "error
4.2	3.0	22.02.2019	In the eKYC response format, PAN="" DOB="" and Gender="" included
4.3.2	3.0	22.02.2019	In the response format description table (Element Name: kycData) PAN, DOB and Gender added