

Unique Identification Authority of India
Planning Commission, Govt. of India (GoI),
3rd Floor, Tower II,
Jeevan Bharati Building,
Connaught Circus,
New Delhi 110001



AADHAAR BEST FINGER DETECTION

API SPECIFICATION - VERSION 1.0 (DRAFT)

DECEMBER 2011

Table of Contents

1. INTRODUCTION	3
1.1 BEST FINGER.....	3
1.2 TARGET AUDIENCE AND PRE-REQUISITES	3
1.3 OBJECTIVE OF THIS DOCUMENT.....	3
2. BEST FINGER DETECTION API	4
2.1 BEST FINGER DETECTION PROCESS	4
2.2 API PROTOCOL.....	5
2.2.1 <i>Element Details</i>	6
2.3 BEST FINGER DETECTION API: INPUT DATA FORMAT	6
2.3.1 <i>Element Details</i>	7
2.4 BEST FINGER DETECTION API: RESPONSE DATA FORMAT	11
2.4.1 <i>Element Details</i>	11
3. APPENDIX	14
3.1 RELATED PUBLICATIONS	14

1. Introduction

The Unique Identification Authority of India (UIDAI) has been created, with the mandate of providing a Unique Identity (Aadhaar) to all Indian residents. The UIDAI proposes to provide online authentication using demographic and biometric data.

Aadhaar “*authentication*” means the process wherein Aadhaar Number, along with other attributes, including biometrics, are submitted to the Central Identities Data Repository (CIDR) for its verification on the basis of information or data or documents available with it. UIDAI will provide an online service to support this process. Aadhaar authentication service only responds with a “yes/no” and no personal identity information is returned as part of the response.

1.1 Best Finger

When authenticating a resident using any single finger, the accuracy or the chances of being matched would be different due to differences in quality across all his/her fingers. This variation may also be present due to the manner in which the resident normally interacts with a typical fingerprint scanner and the different fingers may inherently have different amount of identifying information depending on the size of the finger and the commonness of the pattern it carries. It may thus be useful to appraise each resident of finger providing the best accuracy and successful matching results. We shall refer to this finger with best accuracy as the **best finger**. Resident may possess one or more best fingers. This knowledge allows the resident to provide his/her best finger(s) during authentication thereby increasing the chances of successful match.

1.2 Target Audience and Pre-Requisites

This is a technical document and is targeted at software professionals working in technology domain and interested in incorporating Aadhaar authentication into their applications. Readers must be fully familiar with Aadhaar Authentication API 1.5 http://uidai.gov.in/images/FrontPageUpdates/aadhaar_authentication_api_1_5_rev1_1.pdf before reading this document.

1.3 Objective of this document

This document provides Aadhaar Best Finger Detection API (Application Programming Interface) specification. It contains details including API data format, protocol, and security specifications.

2. Best Finger Detection API

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

2.1 Best Finger Detection Process

AUAs are expected to develop BFD application as part of their Aadhaar biometric authentication enabled applications. BFD application should use this API and is used to categorise resident's authentication readiness. In addition, this helps the resident understand how Aadhaar biometric authentication work and which of their fingers are best usable for such purposes. A sample Java application is made available by UIDAI.

BFD application should do the following:

1. Capture one finger at a time. During the capture of fingerprints by the BFD application, all captured fingerprint images are subjected to image check to measure NFIQ.
2. Up to three attempts are carried out so that good quality images are captured for each finger. Best quality image of three attempts would be used for BFD purpose.
3. Once the best attempt is captured for all fingers, application forms the input XML for the BFD API as specified in this document.
4. Application invokes the BFD API through AUA server (similar to authentication).
5. Based on the response, provide a printed receipt to the resident indicating the ranking for each finger.

AUA devices using biometric authentication implementing this BFD API should have the user interface for capture and sending as described below:

1. Resident/operator needs to clearly know which finger to capture and should be visible on screen.
2. There must be options to rescan after the capture is complete.
3. Capture high quality fingers (NFIQ 1 & 2) up to 3 attempts and application must pick up highest NFIQ image.
4. Application should remember the finger position because it has to be sent along side NFIQ for every finger as part of BFD API input.
5. Application should send only 10 Templates and not images.
6. Provide for exception where resident may not have all ten fingers.
7. Provide an option to buffer the transaction in case the connectivity is not there and replay from database.
8. Application must not keep any unencrypted data that involves resident information including biometrics.
9. BFD application must provide a printed receipt UI indicating BFD output details preferably with a picture of the hand
10. Buffered/online transaction receipt needs to be provided to resident.
11. Receipt needs to indicate colour code, rank as per response for each response.

Logging information related to each of the attempts will help UIDAI to characterise and analyze such data. BFD server at UIDAI end processes incoming requests as described below:

1. Fingers are ranked in descending order based on the matching score.
 - a) Fingers with matching score above a particular “high” threshold are ranked “Green” with appropriate score.
 - b) Fingers with matching scores that lie between “high” and “low” thresholds are ranked “Yellow”. Yellow fingers are ranked below green fingers.
 - c) Fingers whose matching score is below “low” threshold are ranked as “Red” indicating these fingers cannot be used during authentication.
2. BFD application feedback helps resident to clearly identify which of his/her fingers are good for authentication. Resident is expected to use his/her fingers in the order of rank starting from 1 while doing authentication.

NOTE: Above is a high level logic and UIDAI may decide to enhance this logic at the backend from time to time. Change of such logic will not have any impact on the actual API input/output signatures. It is provided for general understanding.

2.2 API Protocol

Aadhaar Best Finger Detection (henceforth referred as BFD) 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 these services. 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 for Aadhaar BFD service:

```
https://<host>/bfd/<ver>/<ac>/<uid[0]>/<uid[1]>/
```

API input data should be sent to this URL as XML document using Content-Type “application/xml” or “text/xml”.



For security reason data collected for Aadhaar BFD service must not be stored in the devices or log files. It's essential for SA and AUA to maintain audit records for all the request metadata along with the response.

For all SSL communication the agencies should automatically validate the Aadhaar certificates. Also on every SSL handshake agencies should validate the revocation list online.

2.2.1 Element Details

host – Aadhaar BFD server address. Actual production server address will be provided to ASAs and AUAs. Note that production servers can only be accessed through secure leased lines. For development and testing purposes, public URL “*auth.uidai.gov.in*” can be used. Application at AUA/ASA server should ensure that actual URL is configurable.

Next part of the URL “bfd” indicates that this is a Best Finger Detection API call instead of regular authentication API call. Ensure that this is provided.

ver – BFD API version (optional). If not provided, URL points to current version. UIDAI may host multiple versions for supporting gradual migration. As of this specification, only valid version for production use is “1.0”.

ac – A unique code for the AUA which is assigned by UIDAI. This is an alpha-numeric string having maximum length 10. (A default value “public” is available for testing.)

uid[0] and **uid[1]** – First 2 digits of Aadhaar Number. Used for load-balancing.

For all valid responses, HTTP response code 200 is used. All application error codes are encapsulated in response XML element. In the case of connection and other server errors, standard HTTP error response codes are used (4xx codes such as 403, 404, etc.). HTTP automatic redirects also should be handled by ASA/AUA server.



ASA server must set REMOTE_ADDR header of HTTP with ASA server’s static IP address. Authentication sub-system system allows only white listed IP addresses coming through a secure private network.

2.3 Best Finger Detection API: Input Data Format

Aadhaar BFD service uses XML as the data format for input and output. To avoid sending unnecessary data, do not pass any optional attribute or element unless its value is different from default value. Any bad data or extra data will be rejected.

Following is the XML data format for BFD API:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Bfd uid="" tid="" ac="" sa="" ver="" txn="" lk="">
  <Skey ci="">encrypted and encoded session key</Skey>
  <Data>encrypted and then encoded block</Data>
  <Hmac>SHA-256 Hash of Pid XML, encrypted and then encoded</Hmac>
  <Signature>Digital signature of AUA</Signature>
</Bfd>
```

“Data” element contains “Rbd” (Resident Biometric Data) element which is a base-64 encoded encrypted block. Complete “Data” block should be encrypted at the time of capture on the capture device. But, encoding (base-64) of “Data” block and packaging it with enveloping XML under “Bfd” element can either be done on the device or on the AUA server based on the AUA needs. Device capability, protocol between devices and AUA server, and data format used between devices and AUA server, etc. should be considered for making that choice.

Following is the format for “Rbd” element:

```
<Rbd ts="" ver="">
  <Meta fdc="" apc="">
    <Locn lat="" lng="" vtc="" subdist="" dist="" state="" pc=""/>
  </Meta>
  <Bios>
    <Bio nfiq="" na="" pos="">encoded biometric</Bio>
  </Bios>
</Rbd>
```

For Public devices, data should be encrypted with a dynamic session key using **AES-256** symmetric algorithm (AES/ECB/PKCS7Padding). Session key, in turn, is encrypted with **2048-bit UIDAI public key** using asymmetric algorithm (RSA/ECB/PKCS1Padding). Reference implementation demonstrates this in detail. Session key must not be reused across transactions. This is same as what is used for Authentication API.

2.3.1 Element Details

Element: **Bfd** (mandatory)

- Root element of the input XML for BFD service.

Attributes:

- **uid** – (mandatory) Aadhaar Number of the resident
- **tid** – (mandatory) For Registered devices, send its unique Terminal ID. For Public devices, value should be passed as “public”.
- **ac** – (mandatory) A unique code for the AUA which is assigned by UIDAI during AUA registration process. This is an alpha-numeric string having maximum length 10. (A Default value “public” is available only for testing.)
- **sa** – (mandatory) A unique “Sub-AUA” code. AUAs are expected to manage these codes within their system and ensure uniqueness within their system. This allows auditing and business intelligence to be provided at SA level. If AUA and SA are same agency, use value of “ac” for this attribute. This is an alpha-numeric string having maximum length 10.
- **ver** – (mandatory) version of the API. Currently only valid value is “1.0”.
- **txn** – (optional) AUA specific transaction identifier. AUA can choose to pass this as part of input. This is returned as part of response as is. This is very useful for linking transactions full round trip across systems. This is an alpha-numeric string of maximum length 50. Only supported characters are A-Z, a-z, 0-9, period,

comma, hyphen, backward & forward slash, left & right parenthesis, and colon. No other characters are supported. It is highly recommended that AUAs use this attribute for correlating requests with responses for auditing and verification.

- **lk** – (mandatory) A valid “License Key” assigned to the AUA. Administration portal of UIDAI will provide a mechanism for AUA administrator to generate license keys and use it within the authentication. This is an alpha-numeric string of length up to 64 characters.

Element: Skey (mandatory for Public devices)

- Value of this element is base-64 encoded value of encrypted 256-bit AES session key. Session key should be dynamically generated and must not be reused across transactions. See next chapter for encryption details.

Attributes:

- **ci** – (mandatory) Public key certificate identifier using which “skey” was encrypted. UIDAI may have multiple public keys in field at the same time. Value of this attribute is the certificate expiration date in the format “YYYYMMDD”. Expiry date of the certificate can be obtained from the certificate itself.

Element: Data (mandatory)

- Contains the encrypted “Rbd” element in base-64 encoding. See “Rbd” element definition later.

Element: Hmac (mandatory)

- Devices which is constructing the “Rbd” element must perform the following to provide the Hmac value:
 - After forming Rbd XML, compute SHA-256 hash of Rbd XML string
 - Then encrypt using session key (skey)
 - Then encode using base-64 encoding (as described earlier, encoding can be done on the AUA server when forming final BFD request XML)

Element: Signature (mandatory)

- In order to ensure that BFD XML was not tampered during its transit between AUA servers and UIDAI Servers, it is required that BFD XML be digitally signed by AUA. X.509 certificate used by the AUA for the purposes of digital signature should have been issued by a well-know Certification Authority, and should be included in <Signature> element of the request.

UIDAI server will validate each request by verifying the digital signature. In addition, server will validate that one of the “O” elements in the X.509 certificate’s subject contains the AUA’s organization name.

Element: Rbd (mandatory) – Resident Biometric Data element.

Attributes:

- **ts** – (mandatory) Timestamp at the time of capture of BFD input. This is in format “YYYY-MM-DDThh:mm:ss” (derived from ISO 8601). Time zone should not be specified and is automatically defaulted to IST (UTC +5.30).



AUAs can buffer BFD requests and send it to Aadhaar authentication server to support occasional lack of network connectivity on the field. Maximum time up to which requests can be queued (buffered) will be defined by UIDAI policy. During initial release, this will be configured to 24 hours. All requests with “ts” value older than this limit will be rejected.

- **ver** – (mandatory) version of the “Rbd” element. Currently only valid value is “1.0”. Notice that this is NOT same as BFD API version. If there is a change in the structure of “Rbd” XML, all the devices may not upgrade to latest software version. Since AUAs cannot inspect the “Rbd” contents, it is essential that “Rbd” XML be allowed to evolve independently of “Bfd” XML.

Element: Meta (optional)

- This element specifies metadata related to the device and transaction. This is highly recommended to ensure that, in future, when certification of the device and application comes in, it can easily be incorporated by AUAs.

Attributes:

- **fdc** – (optional) Fingerprint device code. This is a unique code provided for the fingerprint sensor-extractor combination. AUAs will have access to this code through AUA portal. **It is highly recommended that this is populated.**
- **apc** – (optional) Application code. This is a unique code provided to the application that is installed on the device and is used to conduct Aadhaar authentication. AUAs will have access to this code through AUA portal. **It is highly recommended that this is populated.**

Element: Locn (Optional)

- This element specifies metadata related to the location of the device. This is to support enhanced fraud detection and analytics.

Attributes:

- **lat** – (optional) Latitude of the location where the device is currently being used.
- **lng** – (optional) Longitude of the location where the device is currently being used.

- **vtc** – (optional) Village/Town/City code where the device is currently being used. This code corresponds to the codification used for all resident addresses throughout Aadhaar system.
- **subdist** – (optional) Sub District code where the device is currently being used. This code corresponds to the codification used for all resident addresses throughout Aadhaar system.
- **dist** – (optional) District code where the device is currently being used. This code corresponds to the codification used for all resident addresses throughout Aadhaar system.
- **state** – (optional) State code where the device is currently being used. This code corresponds to the codification used for all resident addresses throughout Aadhaar system.
- **pc** – (optional) Postal pin code where the device is currently being used. This code corresponds to the codification used for all resident addresses throughout Aadhaar system.

Element: Bios – (mandatory)

- This element can have one or many “Bio” elements carrying biometric records to be matched.

Element: Bio (at least 1 element mandatory)

- Base-64 encoded biometric record. This is always of type FMR (“Fingerprint Minutiae Record”).
- Number of “Bio” elements under “Bios” must be between 1 and 10.

Attributes:

- **nfiq** – (mandatory) NFIQ score of the finger image calculated during capture. For BFD service to work well, it is necessary that BFD application attempts to capture best NFIQ score for each finger.
- **na** – (mandatory) Number of attempts before this capture was taken. It is suggested that BFD applications provide at least 3 attempts to residents to obtain best NFIQ.
- **pos** – (mandatory) In order to reduce the unnecessary matching of biometric templates, it is mandatory that BFD request contain the biometric position along with biometric templates. This attribute can have following values:

LEFT_INDEX
LEFT_LITTLE
LEFT_MIDDLE
LEFT_RING
LEFT_THUMB
RIGHT_INDEX
RIGHT_LITTLE
RIGHT_MIDDLE
RIGHT_RING
RIGHT_THUMB



For BFD service to work well, all available fingers of the resident should be captured and sent. All fingers must be captured and used while calculating the match scores for each finger and detecting best fingers. It is also necessary that BFD application attempts to capture best NFIQ score for each finger.

2.4 Best Finger Detection API: Response Data Format

BFD API does not provide any identity data as part of the response. All it does is to match given input and respond with a matchability rank for each finger. Response XML is as follows:

```
<BfdRes code="" txn="" err="" ts="" actn="" msg="">
  <Ranks>
    <Rank pos="" clr="G|Y|R" val=""/>
  </Ranks>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod
        Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha256"/>
      <Reference URI="">
        <Transforms>
          <Transform
            Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha256"/>
            <DigestValue></DigestValue>
          </Reference>
        </SignedInfo>
      <SignatureValue></SignatureValue>
    </Signature>
  </BfdRes>
```

2.4.1 Element Details

Element: BfdRes

Attributes:

- **code** – unique alphanumeric “BFD response” code having maximum length 40.
- **txn** – AUA specific transaction identifier that was part of input.
- **ts** – Timestamp when the response is generated. This is of type XSD dateTime.
- **err** – Failure error code. If BFD request fails, this attribute provides any of the following codes:
 - **“300”** – Biometric data did not match
 - **“500”** – Invalid encryption
 - **“510”** – Invalid “Bfd” XML format
 - **“511”** – Invalid “Rbd” XML format

- “520” – Invalid device
- “530” – Invalid AUA code
- “540” – Invalid version
- “561” – Request expired (“Rbd->ts” value is older than *N* hours where *N* is a configured threshold in BFD server)
- “562” – Timestamp value is future time (value specified “Rbd->ts” is ahead of BFD server time beyond acceptable threshold)
- “563” – Duplicate request (this error occurs when exactly same BFD request was re-sent by AUA)
- “564” – HMAC Validation failed
- “565” – License key has expired
- “566” – Invalid license key
- “569” – Digital signature verification failed
- “570” – Invalid key info in digital signature (this means that certificate used for signing the BFD request is not valid – it is either expired, or does not belong to the AUA or is not created by a CA)
- “572” – Invalid biometric position
- “583” – Best finger detection not allowed as per license
- “800” – Invalid biometric data
- “811” – Missing biometric data in CIDR for the given Aadhaar number
- “940” – Unauthorized ASA channel
- “941” – Unspecified ASA channel
- “999” – Unknown error
- **actn** – Actionable feedback in case resident or operator needs to take specific actions. Valid action codes are:
 - “00” – No action necessary
 - “01” – Resident needs to re-enrol his/her biometrics with Aadhaar system
 - “02” – Sensor/extractor needs replacement/upgrade
 - “03” – Retry BFD test again
 - “99” – Other unclassified actions
- **msg** – Actionable feedback message in English in case resident or operator needs to take specific actions. BFD applications should display message to enable operator and resident take appropriate actions.

Element: Ranks

- Biometric matching ranks for each finger that was part of input for BFD service.

Element: Rank

- Rank element for each finger. Number of “Rank” elements will match the number of “Bio” elements in the input XML.
- This is provided in the case there are no errors.

Attributes:

- **pos** – Finger position for which matching rank is provided. Possible values are same as that of “pos” attribute within “Bio” element of input XML.
- **clr** – Color band of the rank based on biometric matching. Possible values are “Green (G)”, “Yellow (Y)”, or “Red (R)”. Green indicates high matchability, Yellow

indicates medium matchability, and Red indicates that those fingers may not match most times and should not be used for authentication.

- **val** – In the case “clr” is “G” or “Y”, then this optional attribute indicates a value between 1 and 10. This attribute indicates the order of preference in which resident should use his/her fingers for authentication. Value 1 indicates first choice and 10 indicates last choice. Note that fingers with “clr” attribute “R” will not have any value attribute since they cannot be used for authentication.

Element: **Signature**

- This is the root element of digital signature. This is same as what is used for Authentication API.
- For more details, refer: <http://www.w3.org/TR/xmlsig-core/>

3. Appendix

3.1 Related Publications

The following standards are applicable and related to the information in this document.

Aadhaar Authentication API 1.5 (Rev 1)	http://uidai.gov.in/images/FrontPageUpdates/aadhaar_authentication_api_1_5_rev1_1.pdf
Biometric Standards	http://uidai.gov.in/UID_PDF/Committees/Biometrics_Standards_Committee_report.pdf
Aadhaar biometric APIs	http://uidai.gov.in/UID_PDF/Working_Papers/Aadhaar_ABIS_API.pdf
Data Encryption Algorithm	ANXI X3.92
Banking—Retail Financial Services Symmetric Key Management	ANSI X9.24
Public Key Cryptography for the Financial Service Industry: Agreement of Symmetric Keys Using Discrete Cryptography	ANSI X9.42
Triple Data Encryption Algorithm: Modes of Operation	ANSI X9.52
Security Requirements for Cryptographic Modules	FIPS PUB 140-2
Information Technology – Security Techniques – Hash Functions	ISO 10118
Information Technology – Security Techniques – Key Management	ISO 11770
Information Technology – Security Techniques – Encryption Algorithms	ISO 18033
Biometric standards	ISO 19794-4, ISO 19794-6
Date and Time format standard	ISO_8601
XML Signature	http://www.w3.org/TR/xmlsig-core/