Unique Identification Authority of India (UIDAI)

3rd Floor, Tower II, Jeevan Bharati Building,
Connaught Circus, New Delhi 110001

AADHAAR REGISTERED DEVICES

TECHNICAL SPECIFICATION - VERSION 2.0 (DRAFT)

JUNE 2016

Contents

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 provides online authentication using demographic and biometric data.

1.1 Aadhaar Authentication at a Glance

Aadhaar authentication is the process wherein Aadhaar Number, along with other attributes, including biometrics, are submitted online to the Aadhaar system for its verification on the basis of information or data or documents available with it. During the authentication transaction, the resident's record is first selected using the Aadhaar Number and then the demographic/biometric inputs are matched against the stored data which was provided by the resident during enrolment/update process.

For latest documentation on Aadhaar authentication, see http://authportal.uidai.gov.in

1.2 Target Audience and Pre-Requisites

This is a technical document and is targeted primarily at biometric device manufacturers/providers who want to build registered devices as per this specification for Aadhaar authentication ecosystem.

This document assumes that readers are fully familiar with Aadhaar authentication model, related terminology, and authentication API technology details. Before reading this document, readers must read the Aadhaar authentication API specification available at http://uidai.gov.in/images/FrontPageUpdates/aadhaar_authentication_api_2_0.pdf

IMPORTANT NOTE: In this document, term "**Device Provider**" used to refer to a device manufacturer or any agency who has partnership with the manufacturer to manage device certification and related security aspects of registered devices. Device provider should be an entity registered in India and is responsible for STQC certification, device key management (as per this spec), and any security or other responsibilities set forth by UDIAI as part of device provider ecosystem rules.

2 Registered Devices

This chapter describes the specification in detail for registered devices for biometric device providers and also provides details on registration flow before these can be used with larger host devices.

2.1 Understanding Public Devices

Before understanding registered devices and the need for it, it is important to understand how public devices work.

Following is the sequence of typical operations using the public devices required to conduct Aadhaar biometric authentication:

1. AUA/Sub-AUA provided application starts in host machine (any application such as banking used for providing resident services).
2. Application captures Aadhaar number and other essential business transaction data (such as amount in the case of payment).
3. When ready for biometric input capture, application prompts the Aadhaar holder for capturing biometrics (fingerprint or iris).
4. When the device driver detects a good image, it provides the captured good image back to host application.
5. Application with the help of "Biometric SDK" does additional processing for FMR/FIR/IIR/FID creation.
6. Using the FMR/FIR/IIR/FID inputs (one or more) application then creates the PID block (see Aadhaar Authentication API 2.0 specification for details) and does rest of the authentication steps to complete the transaction.

Several security measures are taken to ensure strong transaction security and end to end traceability even in public devices. These security measures fall into prevention and traceability. These include deploying signed applications, host and operator authentication by AUA, usage of multi-factor authentication, resident SMS/Email alerts on authentication, biometric locking, encryption/signing of sensitive data, and so on.

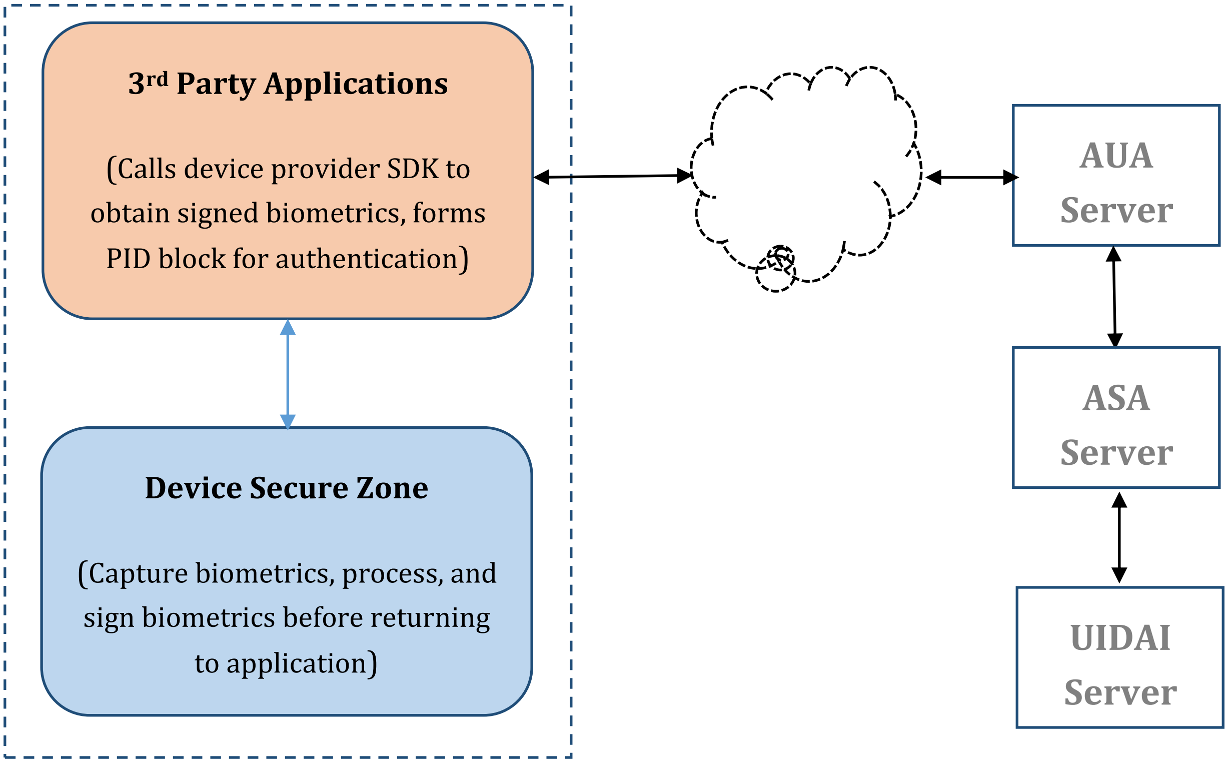
In the case of public devices, although above security measures are in place, there is still a technical possibility of having the biometric data captured in between sensor device and host machine if the device or host machine of AUA is compromised.

2.2 Understanding Registered Devices

Registered devices specification described in this document addresses the solution to eliminate the use of stored biometrics. It provides two key additional features compared to public devices:

1. **Device identification** – every device having a unique identifier allowing traceability, analytics, and fraud management.
2. **Eliminating use of stored biometrics –** biometric data is signed within the device using the provider key to ensure it is indeed captured live.

Following is the logical diagram of a registered device (*for illustration purposes only, actual HW design may differ*):


3rd Party Applications
(Calls device provider SDK to obtain signed biometrics, forms PID block for authentication)
Device Secure Zone
(Capture biometrics, process, and sign biometrics before returning to application)
AUA Server
ASA Server
UIDAI Server

There is no requirement for entire registered device to be physically separate unit. This is to ensure all devices (integrated and discrete) such as external devices connected to phones/laptops as well as biometric embedded phones, etc. can all act as registered devices.

UIDAI **does not mandate any specific hardware design** and device providers are expected to innovate appropriate form factors and SDKs for market use. Key design mandate is that registered devices **MUST securely sign the biometric data** and give it back to application for use within Aadhaar authentication.

Registered devices MUST ensure the following;

1. There should be no mechanism for any external program to provide stored biometrics and get it signed.
2. There should be no mechanism for external program/probe to obtain device private key used for signing the biometrics.

It is important to note that it is in device provider's interest to ensure the above two items are implemented securely since any compromise on these will result in fraudulent activities signed using the device provider key. As per IT Act it is essential for the key owners (device provider) to protect the signature key and take responsibility for any compromise.

NOTE: Rest of the document uses the term SDK to refer to device provider provided registered devices SDK that allows capture and processing of biometrics. This SDK then returns signed biometrics (using device private key within the registered devices secure zone) back to the calling application. All registered devices providers must provide SDKs for various supporting operating systems so that AUA applications can integrate easily.

Following is the sequence of typical operations using the registered devices required to conduct Aadhaar biometric authentication:

1. AUA/Sub-AUA provided application starts in host machine.
2. Application captures Aadhaar number (and any other business transaction data as required by the application).

3. When ready for biometric input capture, application calls the SDK (device provider provided registered devices SDK) to initiate the capture of biometrics.
    a. Application must pass PID timestamp to the SDK which is used during signing
4. When the SDK detects a good capture, it does necessary processing/extraction and provides the signed biometric record (FMR, FIR, IIR, FID) back to application.
5. Using the signed FMR/FIR/IIR/FID inputs (one or more) application then creates the PID block (see Aadhaar Authentication API 2.0 specification for details) and does rest of the authentication steps to complete the transaction.
6. SDK should also have APIs for applications to obtain "device provider ID" and "provider signed device public key cert", and "Device Code". Next section explains the details of signing and usage of these values.

2.3 Signing Biometric Records

Providers of registered devices should:

1. Obtain the device provider ID from UDIAI.
2. Provide list of certified models (model code and other details).
3. Procure a CA certificate (refer 2.5 for supported algorithms) and get it signed by UIDAI. This would be the device provider key. Device providers can have one or more keys.
4. All devices should either generate an asymmetric key pair within the device (highly recommended) or securely inject the key pair. This would be the device key pair. Every physical instance of the device should have its own device key pair.
5. Device public key should be signed by one of the provider key. Refer section 2.5 for supported algorithms. Provider can sign the public key either within the device or on provider server over a secure channel.
6. Device provider MUST ensure each physical device has a unique code. Maximum length of the code is 32 characters when represented as string.

Note: Device public-private key generation/injection and signing of device public key with device provider key can be performed at any point of time during device's lifecycle. **However, the specific device key pair used to sign biometrics for the purposes of Aadhaar authentication should be used for UIDAI purposes only**.

Process of signing is described below:

1. SDK (device provider provided registered devices SDK) should provide an API for application developers to call whenever biometric capture is required.
    a. Application should mandatorily pass PID timestamp (*ts*) to the SDK every time biometric capture call is made.
2. SDK should also have APIs for applications to obtain "device provider ID (*Mi*)" and "device provider signed device public key cert (*Mc*)", and "Device Code (*Dc*)"
    a. *Mi* is used by the application to populate fpmi/irmi/fdmi (based on biometric modality) within the meta element of Authentication XML.
    b. *Mc* is used to populate fpmc/irmc/fdmc (based on biometric modality) within the meta element of Authentication XML.
    c. *Dc* is used to populate fdc/idc/cdc (based on biometric modality) within the meta element of Authentication XML. Device provider MUST ensure each physical device has a unique code. Maximum length of the code is 32 characters when represented as string.
3. When SDK is called, SDK should capture, process, and sign the biometric record (FMR, FIR, IIR, FID) using device key before returning to application. SDK may provide streams for preview to application.
4. When SDK returns the biometric data, it should also return "Biometric Signature (Bs)" for that data. SDK should use the following logic to sign the biometric record:
    a. `Bh = SHA-256(bio_record)` of each successful biometric capture.
    b. `Be = DSA(Bh+ts+Dc, Dpk)` where;
        - *`Dpk`* is the device private key
        - *`ts`* is the PID timestamp (in String representation) that is passed when calling the SDK from the application
        - *`Dc`* is the unique device code in String format

       Refer section 2.5 for supported signature algorithms.
    c. `Bs = base64(Be)`
    d. Return `Bs` to application along with bio record (FMR/FIR/IIR/FID).
5. Application can then form the PID block and Authentication XML as per Aadhaar Authentication API 2.0 specification to conduct biometric authentication.

2.4 Registration and Key Management

1. Device providers to register and obtain a device provider ID via UIDAI portal.
2. Device provider can register one or more public certificate procured from CA and get it signed by UIDAI. These are then used to sign the device public key certificate.
3. Device providers can rotate, revoke their keys via the UIDAI portal.

2.5 Certificates, Keys Policies

1. Below are the currently supported algorithms for digital signing.
    - SHA256withRSA (2048 bit key).
    - UIDAI may support ECDSA in future.
2. All device provider certificates should be procured from a certification authority (CA) as per Indian IT Act. (http://www.cca.gov.in/cca/?q=licensed_ca.html)
3. All device provider certificates should be class II or class III and X509 v3 compliant.
4. Organization attribute in the certificate's subject should match the device provider's name registered with UIDAI.
5. Device provider SHOULD have necessary server/backend infrastructure to sign the device public key, rotate device keys older than specified time as per UIDAI policy, and provide updates/fixed to their SDK.

2.6 Changes in Version 2.0 from Version 1.0

1.0	2.0
Pre-registration was required through register API	No registration API. Devices are implicitly registered on first use of authentication
Symmetric key management was necessary including key reset API	No symmetric key management and no new APIs
Complex for device providers in terms of device and key management	Simplified to allow on the field activation and use