TCap Fingerprint Capturing API & Verification Service Integration

General Summary

The purpose of this document is to describe the technical aspects of the TCap Fingerprint Capturing API, developed to leverage secured fingerprint. The document also focuses on the fingerprint verification service provided by Bangladesh Election Commission (BEC) to verify the identity of Bangladeshi citizen. The target audience of this document is the application developers intended to invoke the capturing interface as well as to consume the services.

Prepared By
Bangladesh Election Commission
Nirbachan Bhaban
Agargoan, Dhaka
Bangladesh

Contents

Revision History

TCap Version	Description	Date
v-1.12	1. App v-1.12 released. Integrated support for Futronic and Abetree/Secugen fingerprint capture device	April 26, 2016
v-1.13	1. App v-1.13 released. Integrated support for DigitalPersona	May 18, 2016
v-1.14	1. App v-1.14 released. Integrated support for Morpho fingerprint capture device	May 24, 2016
v-1.15	1. Updated dependency DLL for SecuGen. This will enable support for later versions of the SecuGen devices	May 31, 2016
v-1.16	1. Added support for Futronic FS82HC 2. Updated dependency DLL for SecuGen. This will enable support for later versions of the SecuGen devices	July 22, 2019

Overview

The *TCap Fingerprint API* (and hence *TCap*) is developed to serve as a biometric capturing interface to be integrated in Biometric SIM Registration System employed by all telecommunication companies in Bangladesh. *TCap* provides encoded fingerprint data within device to ensure the absolute security of the user's biometric properties.

TCap currently supports Windows platform and developed in .Net Framework 3.5. Applications can access the managed library to capture encoded finger data from supported fingerprint capture devices. The encoded finger data can only be decoded at Bangladesh Election Commission (BEC) through appropriate fingerprint verification service to verify citizen's identity.

1. Installation

1.1. TCap Package

The *TCap* is distributed as a Dynamic-link library (or DLL). *TCap.dll* is the core DLL to encapsulate all the operations. *TCap.dll* has several other dependencies provided also in the same folder. All the dependencies must be provided in the application directory along with the *TCap.dll*. This will enable the TCap.dll to access and load its dependencies as needed.

1.2. Supported Hardware

TCap.dll supports the following fingerprint scanners:

Abetree Pro 20
Futronic FS80H, FS82HC
Safran Morpho MSO 1300 E2
SecuGen Hamster Pro 20
Digital Persona (Crossmatch) U.are.U

1.3. Preparing Environment

TCap is tested on Windows 7, 8 and 10 both 32-bit and 64-bit editions. The targeted .Net Framework[1] version is 3.5 which means applications that are developed in .Net Framework 3.5 or above can access the API. However the applications must provide the designated access token. Please find the API reference at *API Reference*.

[1] Windows 7 has pre-installed .Net Framework 3.5 whereas other versions of Windows will require it to be installed manually

2. TCap Architecture

The following is a pictorial description of the work flow of the API and the subsequent section describes the components in details.

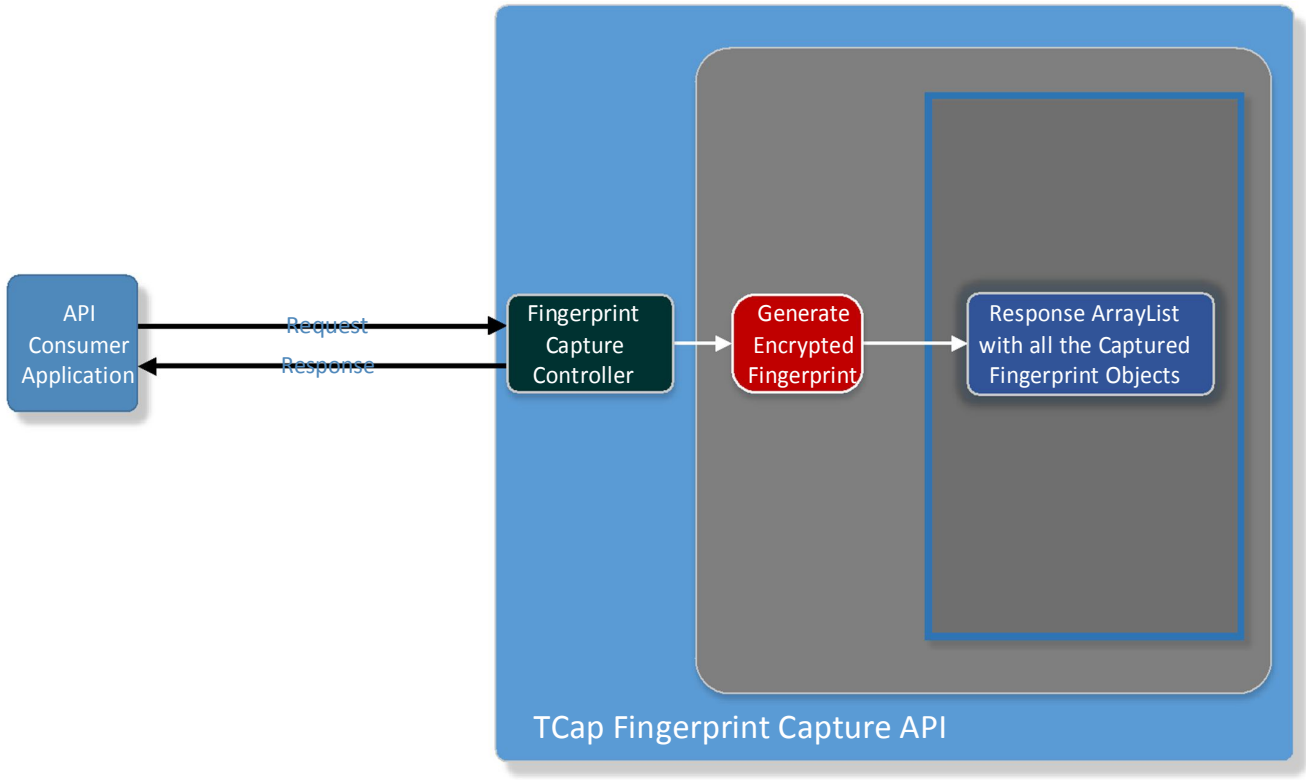

Fig: TCap Architecture

2.1. Request

- The API requests are standard C# method calls wrapped in *TCap* class within *TCap* namespace
- The application developers are required to provide a valid access token, minimum quality score in percentage and maximum try counter value to capture fingerprint. The token is managed by BTRC

2.2. Response

• The API response is an array list of captured fingerprint object(s) containing encoded fingerprint data, quality score along with finger code.

The following is the capturing interface –

Fig: Fingerprint capturing interface

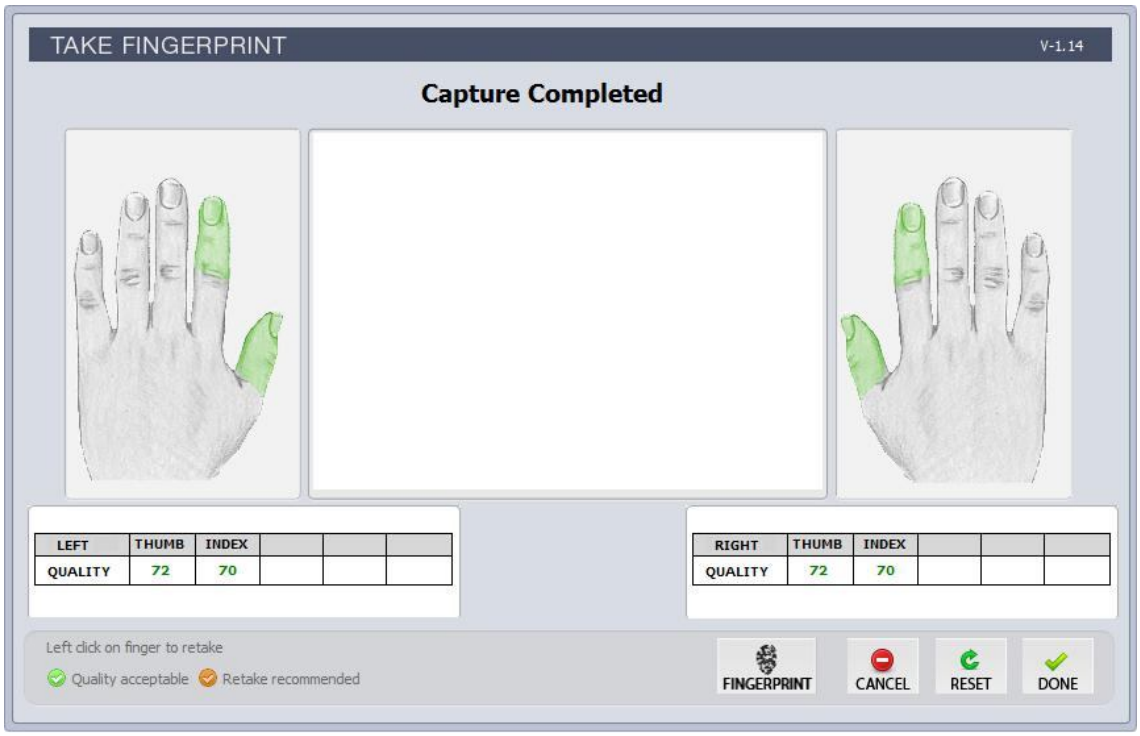

Fig: Fingerprint capturing interface [with score]

3. API Reference

3.1. API Definition

The following table briefly outlines the required method signatures and their usage. For further details, please refer to the sample application [*test_app*] provided in the *TCap_sample* folder.

Class/Method	Input	Output	Remarks
Class Name: TCap.TCap Method Signature: public void *SetConfig*(int[] captureList, int minQuality, int maxRetryCount, string accessToken)	• captureList • minQuality • maxRetryCount • accessToken		• *captureList* must contain an array list of finger code (integer) in the order they will be captured. Finger codes: ○ Right Thumb = 1 ○ Right Index = 2 ○ Left Thumb = 6 ○ Left Index = 7 • *minQuality* is an integer value that denotes the minimum score for accepting a capture image. The value should be within the range 1~100 • *maxRetryCount* is the maximum no. of tries the module will make to achieve the designated minimum quality. If no captured image can achieve the designated quality, the image with the highest quality will automatically be accepted • *accessToken* is a the secret key required to access the API's functionality. This should be collected from BTRC
Class Name: TCap.TCap Method Signature: public List<Fingerprint> *StartCapture*()		An array list of *Fingerprint* objects. *Fingerprint* contains: • finger code • encoded fingerprint data • fingerprint quality score	This method takes no input, but uses the configuration set by previous method to initiate fingerprint capture

3.2. Errors

The methods above may generate the following exceptions –

Source Method	Exception Message
SetConfig()	Max retry count is set beyond expected range
	At least one fingerprint must be captured
	Minimum quality is set beyond expected range
	Access token length incorrect
	Invalid access token provided
StartCapture()	Fingerprint capture failed

4. Verification Service Integration

BEC has launched fingerprint verification service to make sure the citizen is genuine against its claimed identity. BEC offers web service to registered authority/Mobile Network Operator (MNO). The service uses Simple Object Access Protocol (SOAP) to exchange information. The developer is required to collect the location of Web Services Description Language (WSDL) for the required service. The following is the method that serves fingerprint verification.

4.1. API Definition

#	Method	Input	Output	Remarks
1	loginServiceResponse login (loginServiceRequest request)	loginServiceRequest – a) String password b) String username	loginServiceResponse – a) String sessionUuid	*sessionUuid* • use the session UUID generated after successful login in subsequent verification request
2	verifyTypeTwoServiceResponse verifyByNidDobEncodedPrint (final verifyServiceRequest request)	verifyServiceRequest – a) requestHeader – I. String sessionUuid b) Date dateOfBirth c) fingerprints encodedFingerprints d) String nationalId	verifyTypeTwoServiceResponse – a) operationResult I. serviceError error II. Boolean success	*encodedFingerprints* • use the encoded fingerprint data provided by TCap *success* • true, when verification is successful for given data set • false, when verification fails

4.2. Errors

Error code	Description
1305	No biometric match found.