A Review on Unified Payment Interface [UPI]

Kalpesh Dinesh Mishra

Student, Department of MCA, Vivekanand Education Society's Institute of Technology, Maharashtra, India

Abstract - *This UPI enable the user to transfer the money from one account to another account using the simple VPA(Virtual Payment Address).It is more Secure,easy,cheap and more user friendly.*

Keywords: **UPI, Security, Instant-transfer, Digital India, Cashless-India, Common Librarypages, Device binding.**

1.INTRODUCTION

UPI (Unified Payment Interfaces) is a mechanism in which fund transfer takes places between two bank account. Using UPI we are not required to give the bank account details for the fund transfer through the UPI system.

1.1 Advantages of UPI:-

Using the UPI application we can add the multiple bank account in the same UPI applications .but this facility was not available in the IMPS (Immediate Payment Services). Means it contains single account in the single application same bank.

To send the money through the UPI, you don't need to know about the bank of the recipient. It is necessary in the case of IMPS. To use the IMPS, you need the bank account number and IFSC code of the recipient.
You can't pay for online shopping through the IMPS. But UPI gives the easiest way of online payment.

Virtual Payment Address (VPA)

The UPI payment system does not use the bank account details of the recipient. But, there should be an accurate identification of the money recipient. Ultimately, all this convenience is fruitful if the money goes in right hands.

So, every user of the UPI apps must have a unique ID. This unique ID is called as the Virtual Payment Address (VPA).This virtual payment address is used for transaction of the money

For example vivek@icici, Rajan@bob, Sohan@axis.

1.2 Security of UPI Payment.

The united payment interface is as much secure as internet banking or mobile banking. To transfer the money through the UPI app, one has to go through the two-factor authentication.
To open UPI app, you have to give a PIN (application passwords).
To transfer the money, again you have to enter the MPIN or UPI PIN.
All other transactions also go through these two two-factor authentication.
Common Library Page in the Applications:-

2.Register Common Library Pages to the UPI Applications

Register With Common Library
The PSP Mobile App(Upi Applications) needs to execute the following steps to register itself with Common Library.

2.1 Get Challenge

Call the Get Challenge service to receive a challenge from Common Library with help of keyword "Initial" and Device ID.

2.2 Get Token

Send the challenge to PSP server to call the Get Token service of UPI(which is actually the ListKeys API with specific values in Creds block) to receive a valid token. This token should be stored at PSP Mobile App end to ensure secure communication with Common Library.

2.3 Register App

Call the Register App service to verify and register the token in the Common Library.

The signatures of these services have been described below.
Get Challenge Services

PSP Mobile app should call the Get Challenge service to receive a challenge from Common Library.
Input of get challenges services

Parameter Name	Data Type	Description
Device Id	String	This is a mandatory parameter specifying the current device id.
Type	String	This is a mandatory parameter specifying the intention for getting challenge value. Valid values are - “initial” for registration and “rotate” for rotation.

Output of get Challenge Services

Output of this service would be a String containing the Challenge. PSP Mobile app should pass the same to PSP server to receive a valid token from UPI. If Common Library fails to create a challenge for this PSP app, it would return null.

Register App Service

After receiving a valid token from UPI, PSP Mobile App should store it. Subsequently, it should call the Register App service to validate the token and register the PSP App.

Input of register App service

Parameter Name	Data Type	Description
App Id	String	This is a mandatory parameter specifying the current app id.
Mobile	String	This is a mandatory parameter specifying the mobile number of the user which has been verified by the PSP App.
Device Id	String	This is a mandatory parameter specifying the current device id.
Hmac	String	Hmac is a mandatory field which should be prepared following the steps below. a. Concatenate App Id, Mobile number and Device Id with the separator “|”. b. Create a hash of the concatenated string using SHA-256 algorithm. c. Encrypt the hash with the token as key using AES-256 algorithm. d. Populate Hmac with the encrypted string.

Output of register App service
Output of this service would be a Boolean value stating whether the registration is successful or not.

3. Open a Common Library Pages in UPI Application

Get Credential Service for capturing credential
PSP Mobile App should call the Get Credential Service to request Common Library to capture sensitive information like MPIN or OTP.
Input Of Get Credential Services

Parameter Name	Data Type	Description
keyCode	String	This is a mandatory parameter, specifying whether the NPCI or UIDAI public key is to be used. Valid values can be NPCI or UIDAI.
keyXmlPayload	String	This is a mandatory field containing the digitally signed XML payload received from list-Keys API of UPI. Response of list-keys API not modified

Controls	Json	This field specifies the schema for the credential(s) to capture, is in JSON Format. Common Library read this data and generates the Controls. The list of these credentials are to be obtained by the PSP app from ListAccountResponse API. Example is as below:
		{
		"CredAllowed": [{
		"type": "PIN",
		"subtype": "MPIN",
		"dtype": "NUM | ALPH",
		"dlength": "6"
		}, {
		"type": "OTP",
		"subtype": "OTP",
		"dType": "NUM | ALPH",
		"dLength": "6"
		}]
		}
		Based on the number of blocks in the JSON, one or more credential input control will be rendered by the common library.

configuration	Json	This is an optional parameter using which the PSPs can customize the UI displayed by the common library. A sample Json can be as below: {"payerBankName":"Indian Bank Ltd.","backgroundColor":"#FF9933", “color”: "#FF9933"}

Salt	Json	This is a mandatory parameter that captures different elements of the salt used for encryption. Following is an example of a salt Json:
		{
		"txnID":"2350406","txnAmount":"29.30" , "deviceId":"ABCDEF", "appId":"com.psp1.app", "mobileNumber":"9002050725", "payerAddr":"zeeshan.khan@sbi", "payeeAddr":"rohit.patekar@hdfc"}
		All the elements mentioned above would be used to create the cred block along with the credential.

Trust	String	Trust is a mandatory field which should be prepared following the steps below. a. Concatenate Transaction Amount, Transaction Id, Payer Address, Payee Address, App Id, Mobile number and Device Id with the separator "|". b. Create a hash of the concatenated string using SHA-256 algorithm. c. Encrypt the hash with the token as key using AES-256 algorithm. d. Populate Trust with the encrypted string.

payInfo	Json	Sample of this JSON parameter is as below:
		{
		"name": "payeeName",
		"value": "Name Of the Payee"
		},
		{
		"name": "note",
		"value": "Pay for collect"
		},
		{
		"name": "refId",
		"value": "1223423423"
		},
		{
		"name": "refUrl",
		"value": "https://psp1.com"
		},
		{
		"name": "account",
		"value": "122XXX423"
		}

languagePref	String	En_US

Output of the common library will be a HashMap<String, String>(for Android) with the following attributes. There can be more than one entries in the HashMap, based on the number of credentials being captured by the common library.

The key of this HashMap will have the credential name. This will use the name of the credential being captured. This should match with the "name" attribute in the control Json in the input.

The value will be a Json string containing the ki(key index) and the encrypted cred block. Example of the values in the HashMap is as per below. The keys of the HashMap will be the subtype (e.g. MPIN)

```
{
        "type": "PIN",
"subtype": "MPIN",
"data": {
                "code": "NPCI",
                "ki": "20150822",
"encryptedBase64String": "2.0|<encrypted base 64 encoded authentication data>"
}
}
```

4. Device binding in UPI Application for Security

The Device binding concept is used for the security purpose in the UPI Applications. The Following Concept is used for the Application register.

4.1 Telephony Manager

This concept is used to get the device id and IMEI (Device ID+ IMEI) of the mobile for the device binding concept.

4.2 Silent SMS Process

In this concept there is SMS sending process which is used to get the mobile number of the user.

4.3 Get Mobile Number PSP Server

In this Concept we send the (Device ID + IMEI) and the SMS to server. This server requires three parameter ie (Device ID + IMEI + SMS) as mentioned then this services response the Confirmation of user Mobile number.

4.4 Register user Webservices

Then we Call the PSP Server Name (Register User) if all data send by user in proper Format then it will send a message (Successfully Registered).this Message comes When you send all data which is necessary for server to proceed responces . Otherwise we will get an Exception of Error.

4.5 Login Page

Then it will send us on login page where user need to be enter the login pin which is entered during registration page.

4.6 Validate User Webservices

When user Entered the login pin then Validate User webservices call. If user send all the necessary data then it will send message Success and send to the main page of the application.

5.CONCLUSION

The UPI Application is very secure to transfer the money because it has two factor Authentication for user ie Application Password and MPIN for Common Library pages. UPI is very easy to transfer the money with the help of single virtual ID verifications ie (Benificiary virtual ID is valid or not).The main aim of this application is cashless transaction with secure transaction facilities.

6. REFERENCES

[1] www.npci.org.in/UPI_Documents.aspx
[2] http://razorpay.com/upi/
[3]https://en.wikipedia.org/wiki/Unified_Payments_Interface
[4]https://mypoolin.com/upi/
[5]http://www.favcounter.com/unified-payment-interface-mobile-payment/