Advanced Card Systems Ltd.
Card & Reader Technologies

ACS WalletMate

ACS WalletMate User Guide V1.01

Subject to change without prior notice

Table of Contents

List of Figures

List of Tables

No table of figures entries found.

1.0. Introduction

Thank you for purchasing ACS WalletMate with Advanced Card Systems Limited.

ACS WalletMate Mobile Wallet NFC Reader, developed by Advanced Card Systems Ltd., is one of the Google Smart Tap capable terminals which is capable to read Google Wallet Pass. WalletMate is also Apple VAS certified, and capable of reading Apple Wallet Pass.

This document will guide you on applying, and creating Apple Wallet and Google Wallet Passes, and introduce the workflow to use WalletMate.

WalletMate is VAS-only, it shall be used for reading VAS data, but not for payment transactions.

Please visit our website for the most updated information on our products.

http://www.acs.com.hk/

2.0. Apple Wallet Passes Documentation Guide

Apple Wallet Passes documentation can be found below on its website of Apple.

Apple Wallet Passes documentation:

https://developer.apple.com/documentation/walletpasses

You will need to enroll in Apple Developer Program before you can design, create, distribute, and update Apple Passes. Please refer to *Enroll in Apple developer's program for Apple Pass generation* for steps to enroll and create an Apple Pass certificate.

3.0. Google Smart Tap Documentation Guide

Google Smart Tap documentation can be found below website:

https://developers.google.com/wallet/smart-tap/

Google Wallet Generic Pass documentation overview:

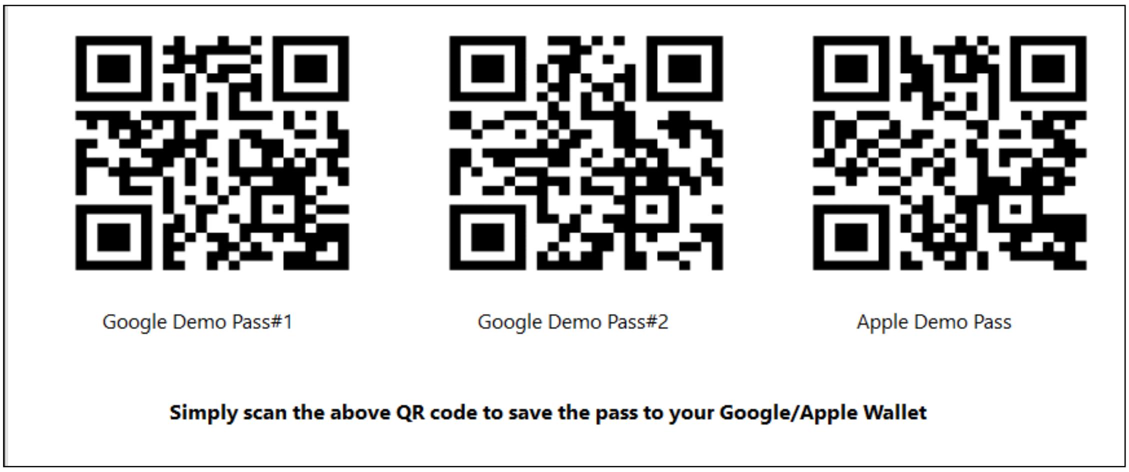
https://developers.google.com/wallet/generic?authuser=1

There are some confidential materials of Google Smart Tap, such as information on Cryptography and Key management which we are not allowed to share or discuss until we are authorized to do so. If you are interested to access the link and know more about it, you need to request access with **Google Wallet Support Team** <google-wallet-passes-support@google.com>, you can reach them by email.

https://developers.google.com/wallet/smart-tap/guides/implementation/key-management?hl=en&authuser=1

Remarks:
We (ACS) as a terminal provider, will be responsible to generate key pair, each key pair contains one Public Key and one Raw Long Time Private Key (LTPK) for your real application, for details, please refer to Get Your Key Pair For Smart Tap.

4.0. Preparation



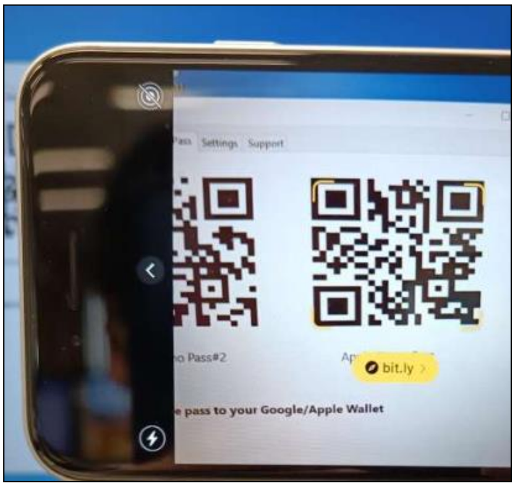
Simply scan the above QR code to save the pass to your Google/Apple Wallet

Figure 1: ACS Test Pass

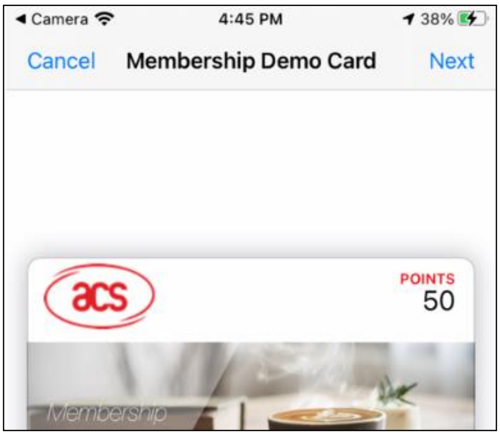
4.1.1. Download the ACS Apple Test Pass in Apple Wallet

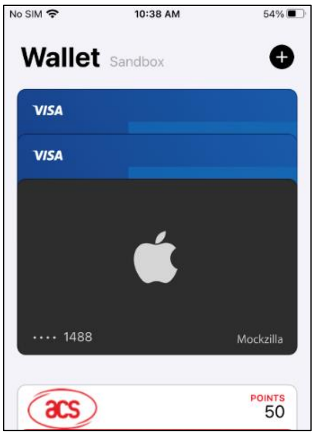
This section provides a simple step-by-step procedure on how to download ACS Apple Test Pass in Apple Wallet.

1. Open *Camera* on iPhone, scan the *Apple Demo Pass* QR Code in Figure 1
2. Click *bit.ly*



3. If you can't get bit.ly, please set the default browser to Safari
4. Click *Next* to save the ACS Test Pass to Apple Wallet.

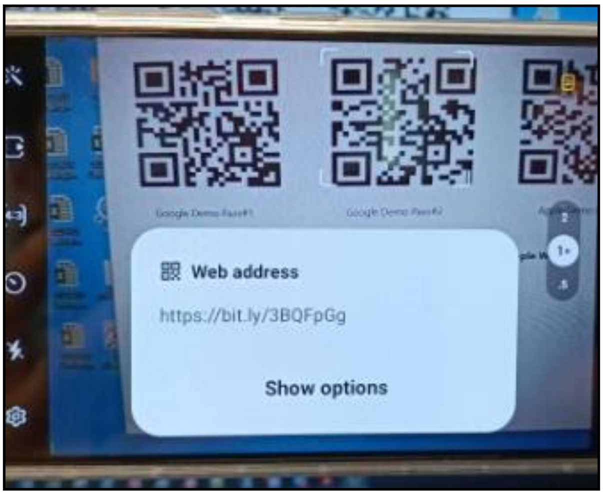




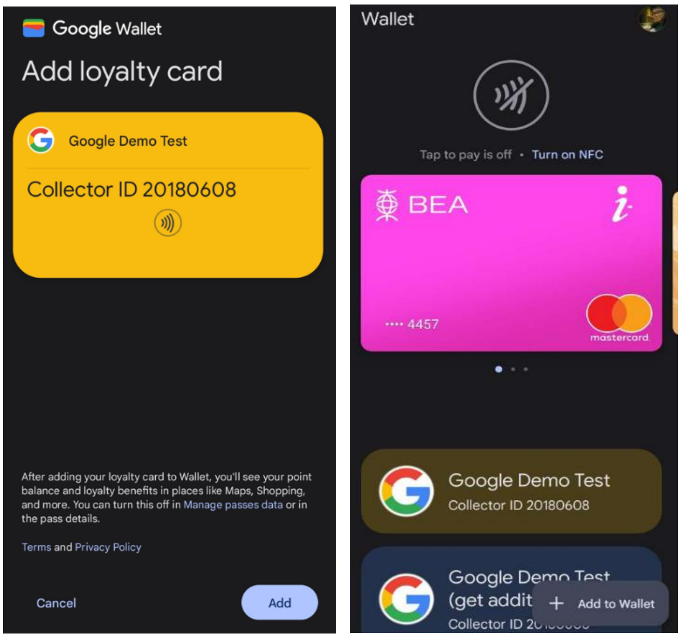
4.1.2. Download the Google Test Pass in Google Wallet

This section provides a simple step-by-step procedure on how to download **Google Test Pass** in Google Wallet.

1. Launch *Camera* on an Android phone or download the *QR code scanning app*, scan the *Google Demo Pass#1* QR Code (the Basic loyalty demo pass, expected payload 2018) or *Google Demo Pass#2* QR Code (a Long loyalty demo pass which get additional data, expected payload {"data-0":"data_here0", ... "data-49":"data_here49"}) in Figure 1
2. You can save both Demo Pass if needed
3. Click the *Web address*



4. Click *Add* to save the Google Test Pass in Google Wallet.



After adding your loyalty card to Wallet, you'll see your point balance and loyalty benefits in places like Maps, Shopping, and more. You can turn this off in Manage passes data or in the pass details.

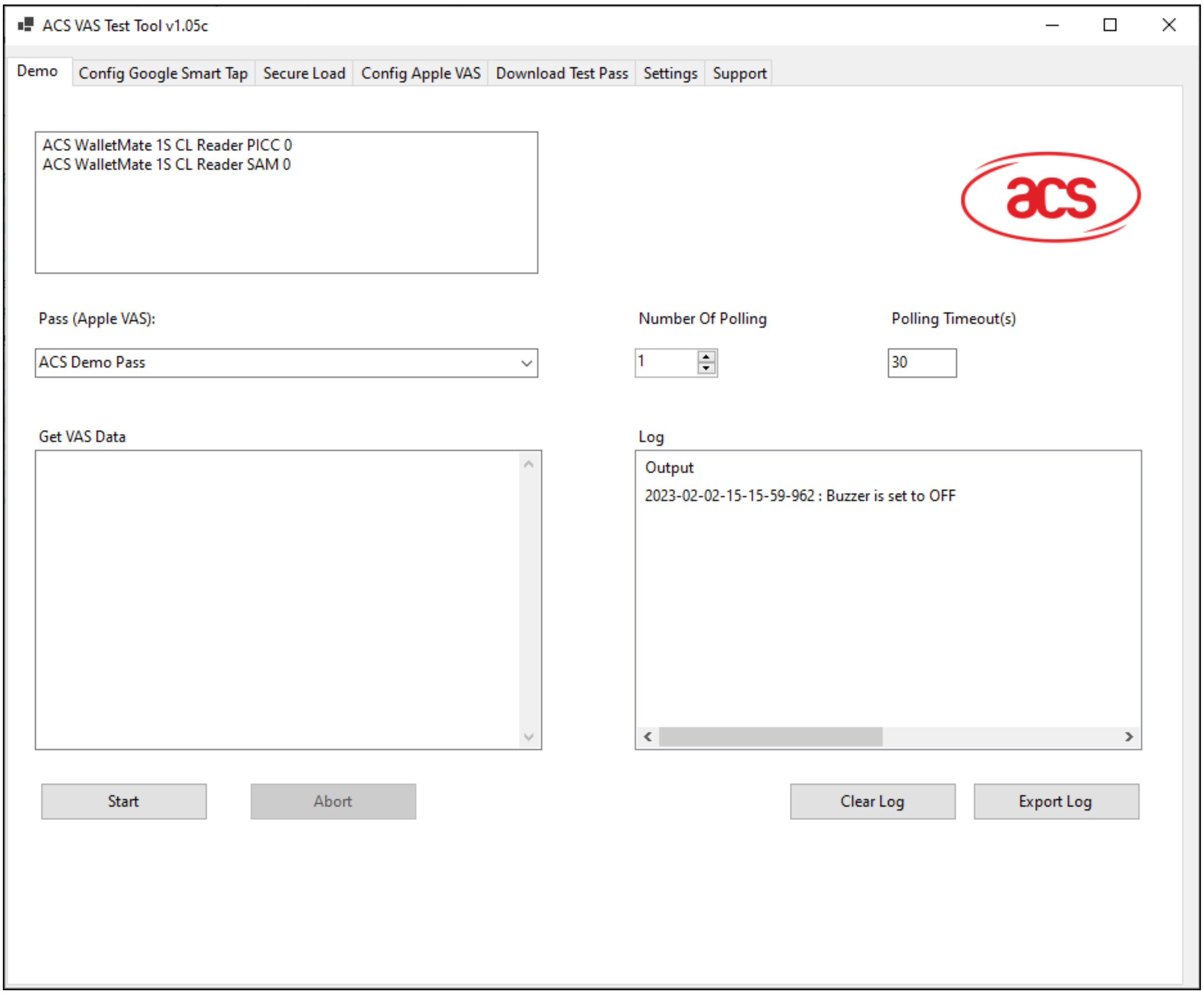
5.0. ACS VAS Test Tool

ACS VAS Test Tool is a tool developed by Advanced Card Systems Ltd. (ACS), intended to test NFC-enabled Apple Pay Passes and Google Pay Passes with WalletMate. Currently, ACS VAS Test Tool is supported in Windows, Linux, Android and macOS platforms. If you would like to use the ACS VAS Test Tool for various development needs, please contact our salesmen or fill out the form on our website https://www.acs.com.hk/en/sales-enquiry/.

5.1. ACS VAS Test Tool (Windows)

5.1.1. Launch the ACS VAS Test Tool

1. Download and install the most updated WalletMate Driver from the ACS Official website https://www.acs.com.hk/
2. Unzip the file *ACS_VAS_Test_Tool_Windows-X.XX* which provided by our salesman.
3. Connect your WalletMate Mobile Wallet NFC Reader to your PC via USB cable.
4. Navigate to ACS_VAS_Test_Tool_Windows-X.XX/exe/, double-click TestVasTool.exe to launch the application. The C# Source code is located inside ACS_VAS_Test_Tool_Windows-X.XX/SourceCode/
5. If there is a pop-up dialog asking to install missing Microsoft Windows Components, please download and install them.
6. The following screen shall appear if the WalletMate Mobile Wallet NFC Reader had been successfully connected and recognized by the PC, its name will appear in the upper top left corner.



ACS WalletMate 1S CL Reader PICC 0
ACS WalletMate 1S CL Reader SAM 0

Pass (Apple VAS): ACS Demo Pass

Number Of Polling: 1

Polling Timeout(s): 30

Get VAS Data

Log

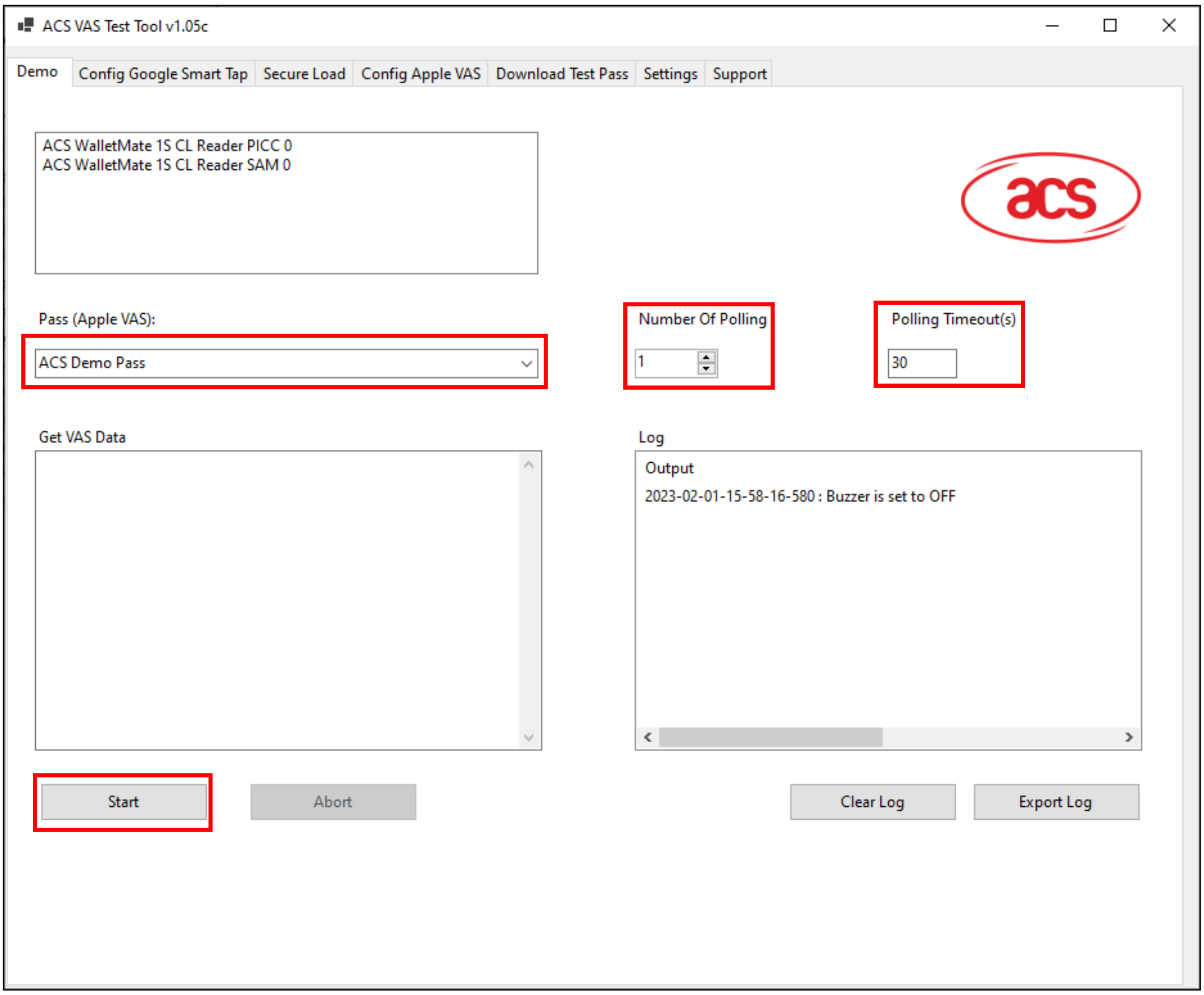
```
Output
2023-02-02-15-15-59-962 : Buzzer is set to OFF
```

Figure 2: ACS VAS Test Tool (Windows)

5.1.2. Using ACS VAS Test Tool with ACS Test Apple Pass

This section provides simple step-by-step procedures on how to test ACS Test Pass stored in Apple Wallet with ACS VAS Test Tool.

1. Go to *Demo* Tab.
2. Adjust the *Number Of Polling* and *Polling Timeout(s)* for each poll, select *ACS Demo Pass*, and then press *Start*. The passTypeIdentifier & Private Key for ACS Demo Pass is hard-coded into the demo.



ACS VAS Test Tool v1.05c

ACS WalletMate 1S CL Reader PICC 0
ACS WalletMate 1S CL Reader SAM 0

Pass (Apple VAS): ACS Demo Pass

Number Of Polling: 1

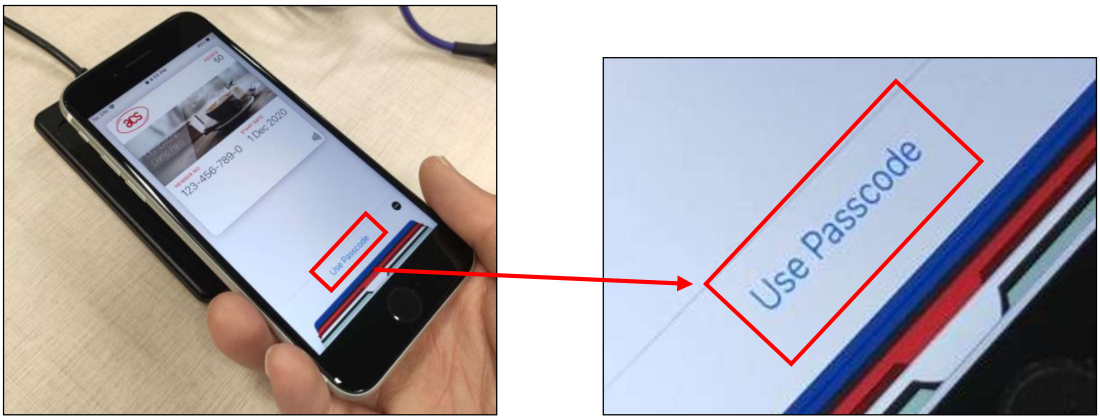
Polling Timeout(s): 30

Get VAS Data

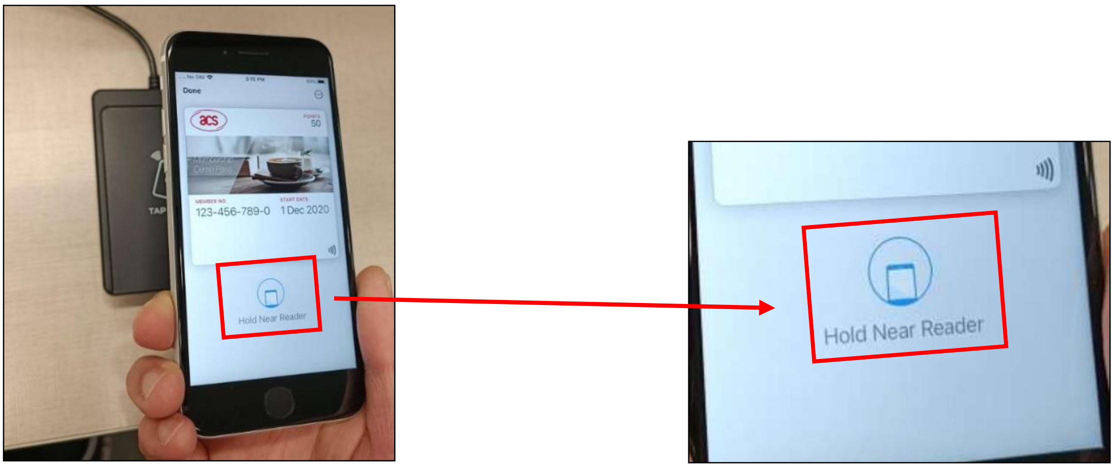
Log

```
Output
2023-02-01-15-58-16-580 : Buzzer is set to OFF
```

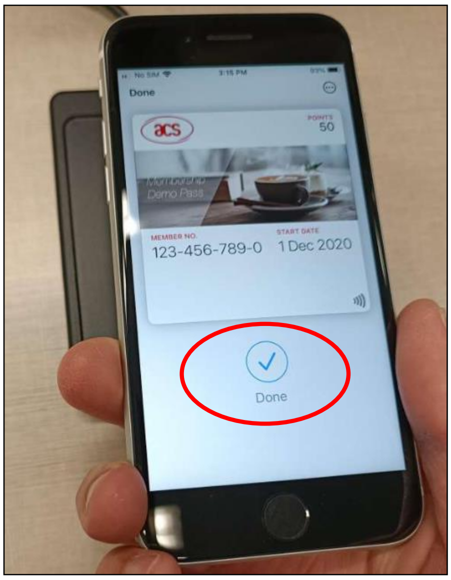
3. Tap the iPhone / Apple Watch on top of WalletMate Mobile Wallet NFC Reader.
4. ACS Test Pass will pop up. If your iPhone is password/Touch ID/Face ID protected, there shall be a prompt asking you to unlock the Pass.



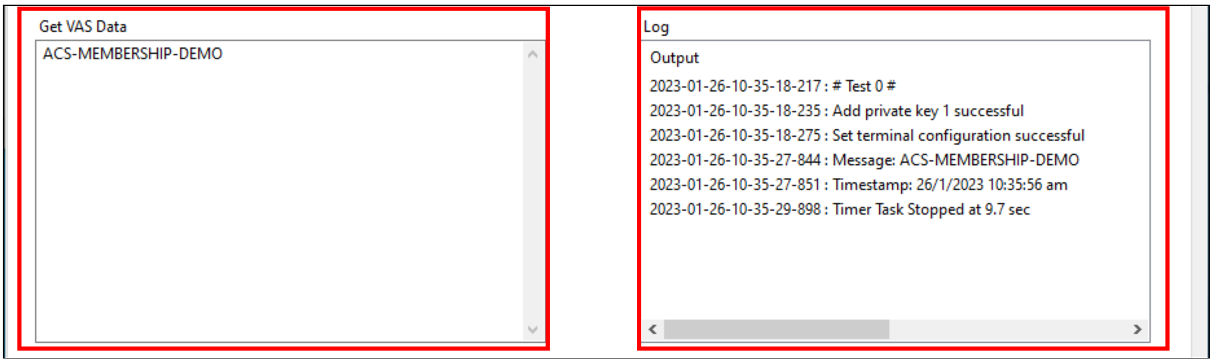
5. Tap the iPhone / Apple Watch on top of WalletMate Mobile Wallet NFC Reader again.



6. A tick mark should be shown on the screen, and the pass shall disappear.



7. Payload message and timestamp is shown in message box.


Get VAS Data

```
ACS-MEMBERSHIP-DEMO
```

Log

Output

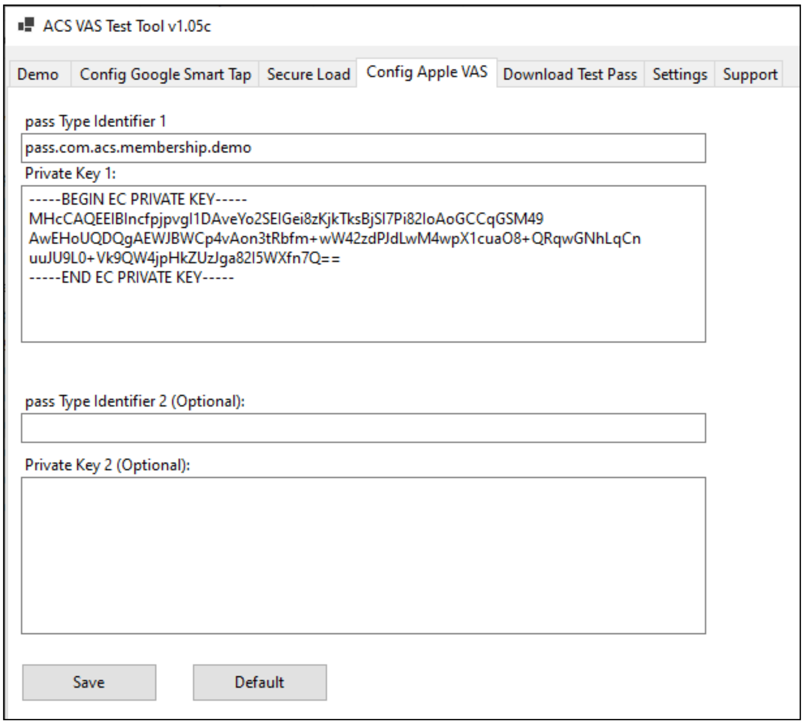
```
2023-01-26-10-35-18-217 : # Test 0 #
2023-01-26-10-35-18-235 : Add private key 1 successful
2023-01-26-10-35-18-275 : Set terminal configuration successful
2023-01-26-10-35-27-844 : Message: ACS-MEMBERSHIP-DEMO
2023-01-26-10-35-27-851 : Timestamp: 26/1/2023 10:35:56 am
2023-01-26-10-35-29-898 : Timer Task Stopped at 9.7 sec
```

5.1.3. Test Custom NFC-Enabled Apple Pass

This section provides a simple step-by-step procedure on how to use ACS VAS Test Tool with your custom NFC-Enabled Pass. If you had created your custom test pass with Apple, you should follow the guidance of this section.

You may refer to Apple VAS to learn more about the pre-requires of designing your own Apple Pass.

1. Go to the *Config Apple VAS* Tab and fill in the *pass Type Identifier 1* & *Private Key 1* (Be ware of the syntax) fields.
2. If you have two custom test passes wanted to test, please paste the 2nd passTypeIdentifier & Private Key in *pass Type Identifier 2* & *Private Key 2* respectively.
3. Press *Save*



pass Type Identifier 1

pass.com.acs.membership.demo

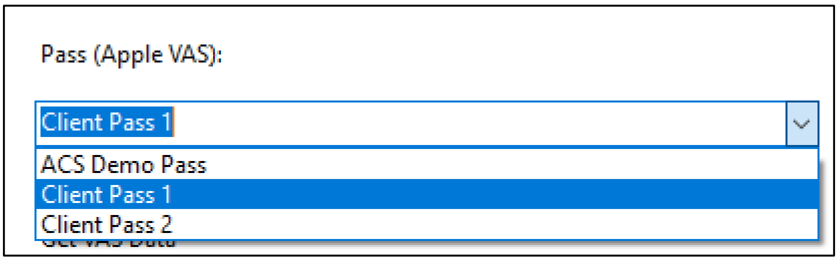
Private Key 1:

```
-----BEGIN EC PRIVATE KEY-----
MHcCAQEEIBlncfpjpvgl1DAveYo2SElGei8zKjkTksBjSI7Pi82IoAoGCCqGSM49
AwEHoUQDQgAEWJBWCp4vAon3tRbfm+wW42zdPJdLwM4wpX1cuaO8+QRqwGNhLqCn
uuJU9L0+Vk9QW4jpHkZUzJga82I5WXfn7Q==
-----END EC PRIVATE KEY-----
```

pass Type Identifier 2 (Optional):

Private Key 2 (Optional):

4. Go to *Demo* Tab, select *Client Pass 1* or *Client Pass 2* and press *Start*. Client Pass 1 and Client Pass 2 refers to the passTypeIdentifier 1, passTypeIdentifier 2 and theirs corresponding private key in *Config Apple VAS* Tab.

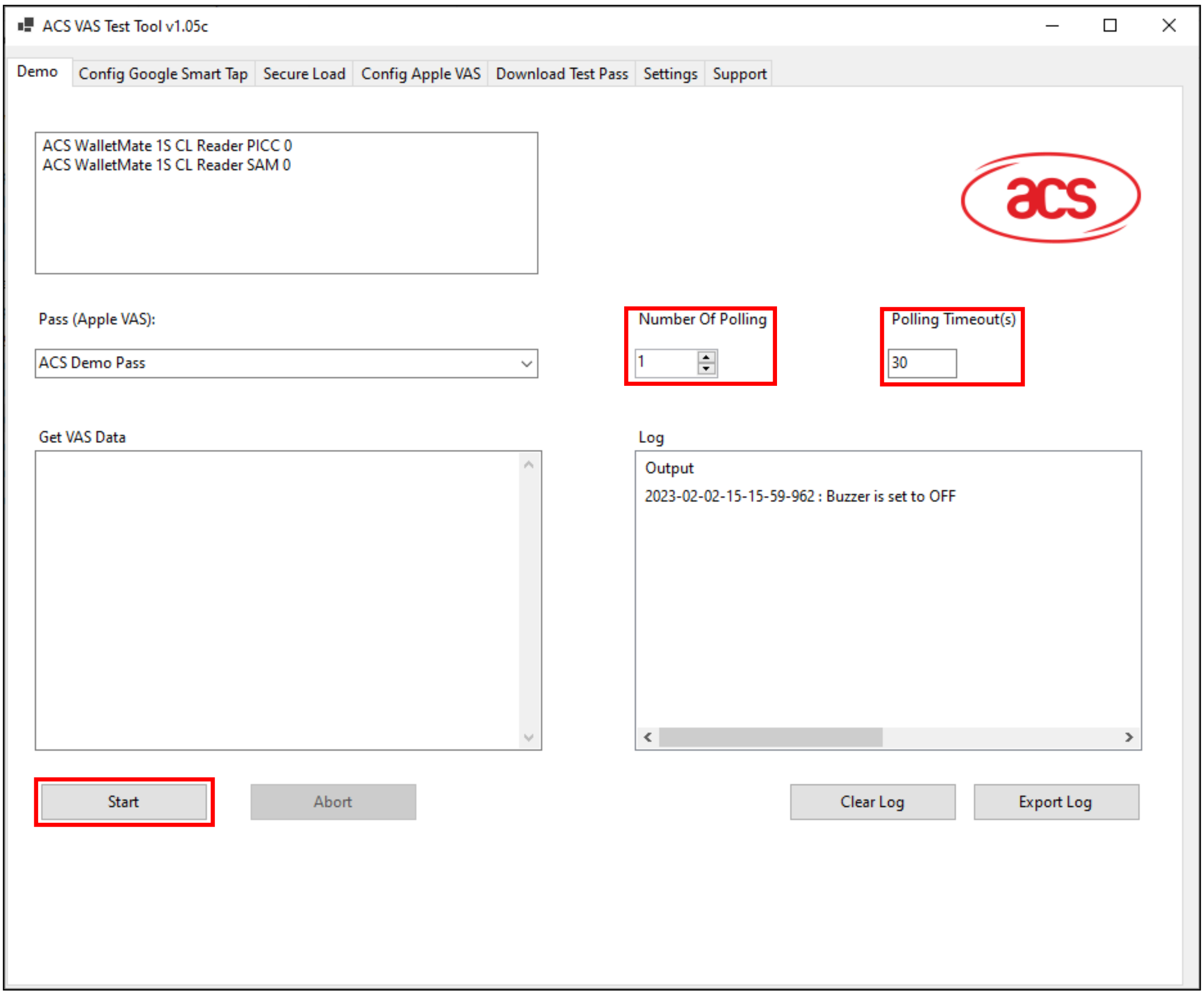


5. Then follow the rest procedures on Using ACS VAS Test Tool with ACS Test Apple Pass, you should be able to retrieve the information stored on your own Custom NFC-Enabled Pass.

5.1.4. Using ACS VAS Test Tool with Google Test Pass

This section provides simple step-by-step procedures on how to test ACS Test Pass stored in Google Wallet with ACS VAS Test Tool.

1. Go to *Demo* Tab.
2. You may adjust *the Number Of Polling* and *Polling Timeout(s)* for each poll, press *Start*. Leave the *Config Google Smart Tap* and *Secure Load* tabs in the default setting if you wish to test with Google Demo Test Pass.



ACS WalletMate 1S CL Reader PICC 0
ACS WalletMate 1S CL Reader SAM 0

Pass (Apple VAS): ACS Demo Pass

Number Of Polling: 1

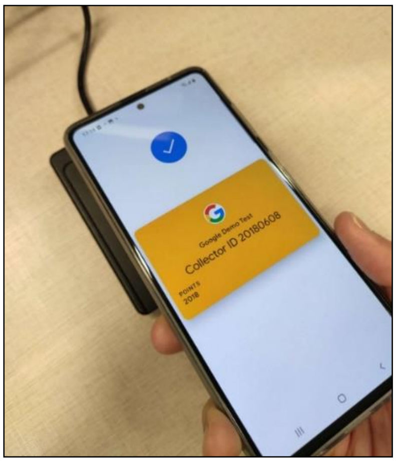
Polling Timeout(s): 30

Get VAS Data

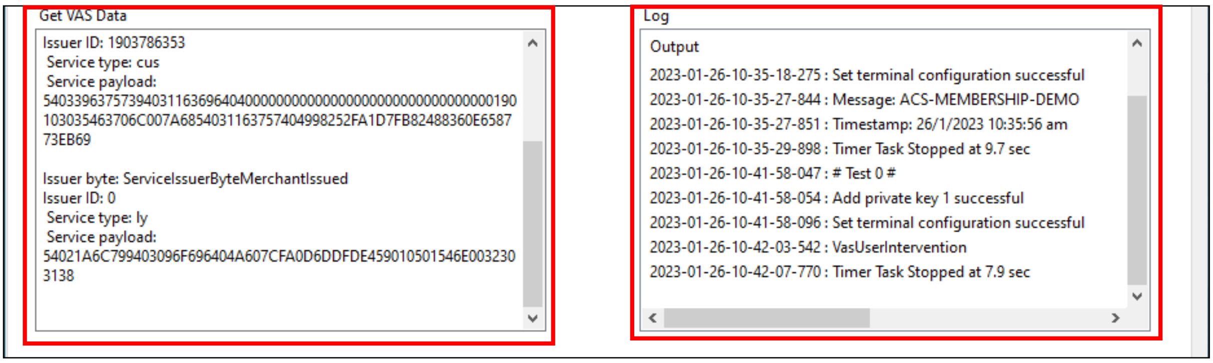
Log

```
Output
2023-02-02-15-15-59-962 : Buzzer is set to OFF
```

3. Unlock the screen, turn on the NFC on your android device, and tap the android mobile phone on top of WalletMate Mobile Wallet NFC Reader, the Google Demo Test Pass shall pop up immediately.



4. Payload message and timestamp is shown in the “Demo” Tab message box.



Get VAS Data

```
Issuer ID: 1903786353
 Service type: cus
 Service payload:
54033963757394031163696404000000000000000000000000000000000190
103035463706C007A6854031163757404998252FA1D7FB82488360E6587
73EB69

Issuer byte: ServiceIssuerByteMerchantIssued
Issuer ID: 0
 Service type: ly
 Service payload:
54021A6C799403096F696404A607CFA0D6DDFDE459010501546E003230
3138
```

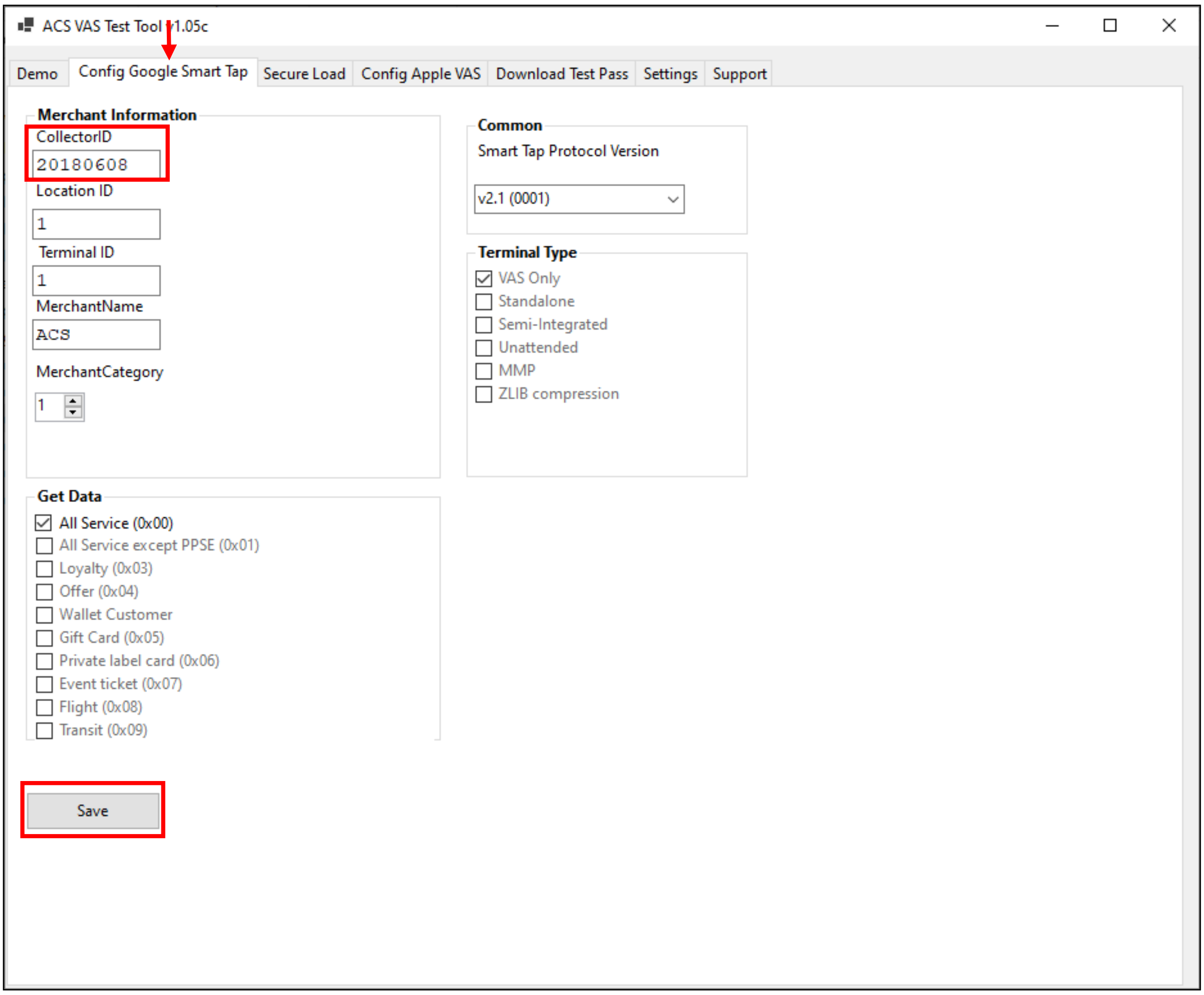
Log

```
Output
2023-01-26-10-35-18-275 : Set terminal configuration successful
2023-01-26-10-35-27-844 : Message: ACS-MEMBERSHIP-DEMO
2023-01-26-10-35-27-851 : Timestamp: 26/1/2023 10:35:56 am
2023-01-26-10-35-29-898 : Timer Task Stopped at 9.7 sec
2023-01-26-10-41-58-047 : # Test 0 #
2023-01-26-10-41-58-054 : Add private key 1 successful
2023-01-26-10-41-58-096 : Set terminal configuration successful
2023-01-26-10-42-03-542 : VasUserIntervention
2023-01-26-10-42-07-770 : Timer Task Stopped at 7.9 sec
```

5.1.5. Test Custom NFC-Enabled Google Pass

If you want to develop your NFC-Enabled Pass with Google and had generate the encrypted LTPK and Public Key, and had uploaded the Public Key into your Google Wallet API Issuer Account, you can follow the below guidelines and test your pass with ACS VAS Test Tool.

1. Navigates to *Config Google Smart Tap* Tab, input the Collector ID, and press *Save*.



Merchant Information

CollectorID: 20180608
Location ID: 1
Terminal ID: 1
MerchantName: ACS
MerchantCategory: 1

Common

Smart Tap Protocol Version: v2.1 (0001)

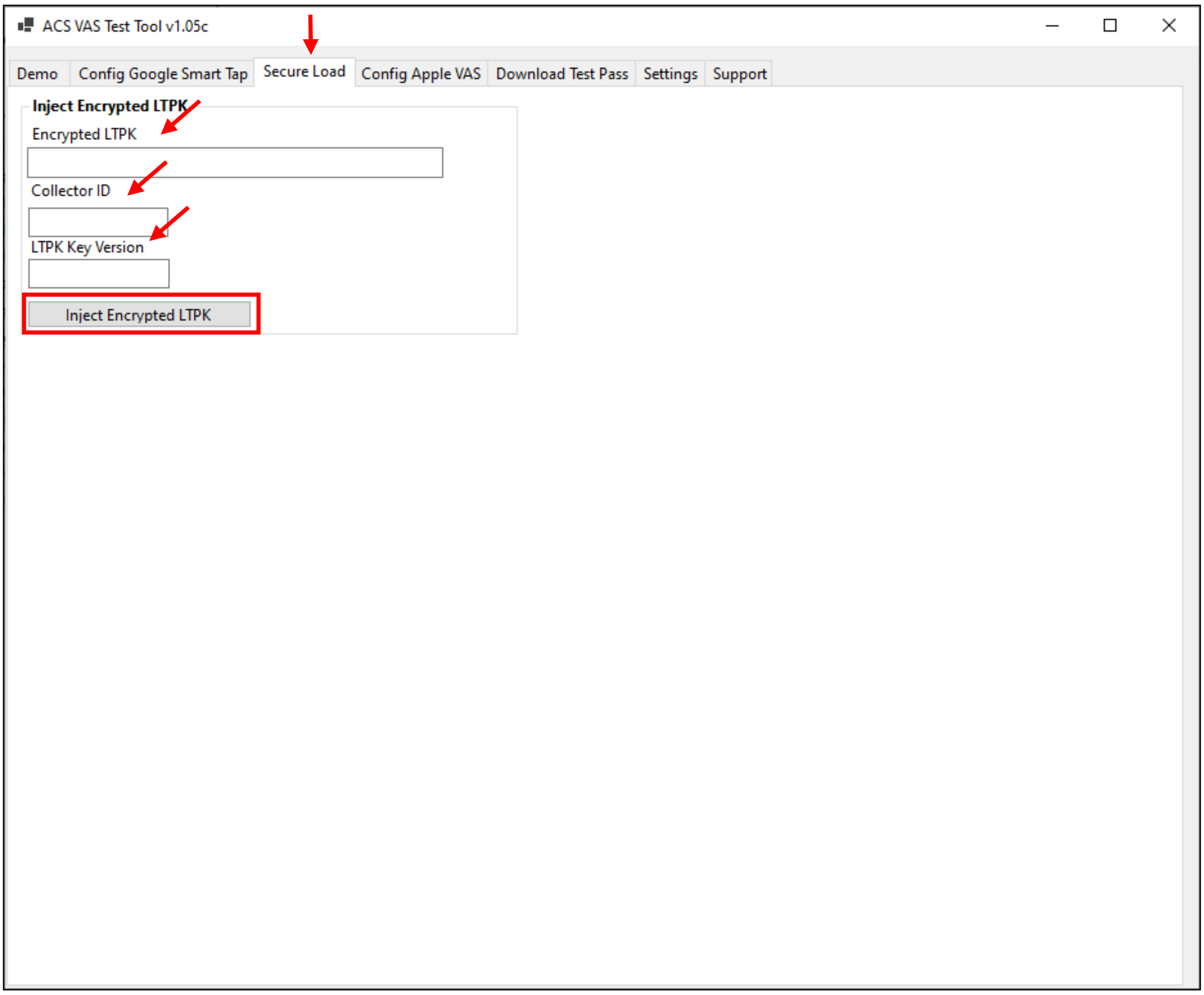
Terminal Type

☑ VAS Only
□ Standalone
□ Semi-Integrated
□ Unattended
□ MMP
□ ZLIB compression

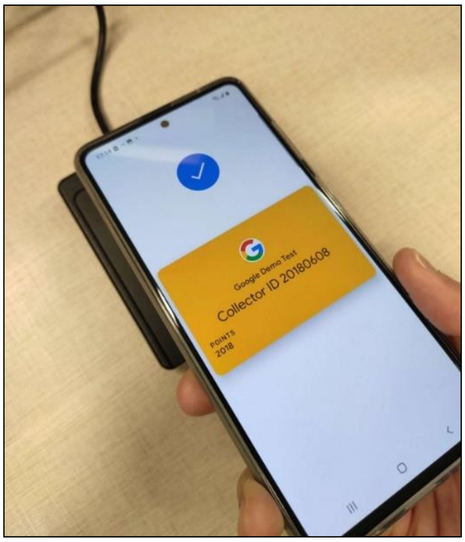
Get Data

☑ All Service (0x00)
□ All Service except PPSE (0x01)
□ Loyalty (0x03)
□ Offer (0x04)
□ Wallet Customer
□ Gift Card (0x05)
□ Private label card (0x06)
□ Event ticket (0x07)
□ Flight (0x08)
□ Transit (0x09)

2. Go to the *Secure Load* Tab, fill in the Encrypted LTPK (Refer to Get Your Key Pair For Smart Tap), together with your Collector ID and LTPK Key Version, and then press *Inject Encrypted LTPK*.



3. Then follow the rest procedures on Using ACS VAS Test Tool with Google Test Pass, you should be able to retrieve the information stored on your own Custom NFC-Enabled Pass.

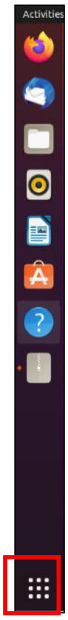


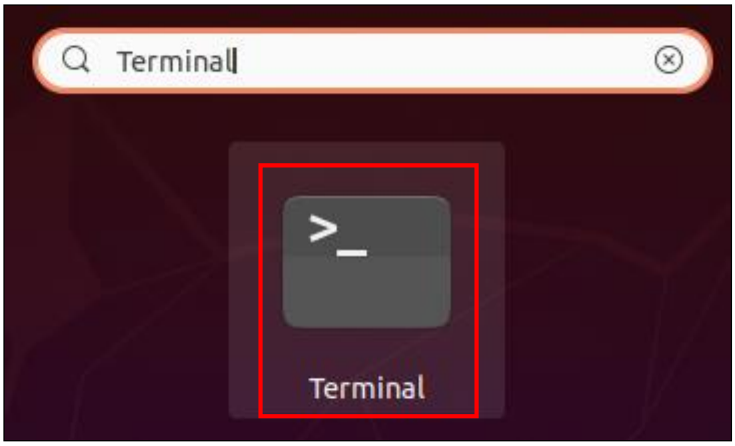
5.2. ACS VAS Test Tool (Linux)

5.2.1. Launch the ACS VAS Test Tool

The following instructions are written for Ubuntu only. If you wish to operate ACS VAS Test Tool in another Linux platform, please contact our sales representative and we will provide you with further assistants.

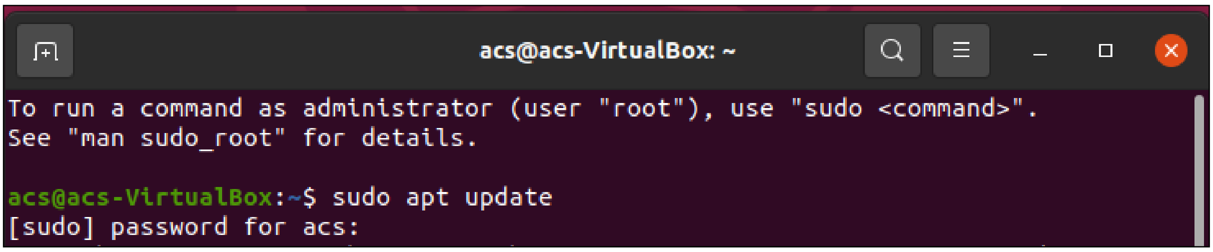
1. Download and install the most updated WalletMate Driver from the ACS Official website https://www.acs.com.hk/
2. Open *Terminal* in Linux (Ctrl + Alt + T)





3. Updates the package lists for upgrades on Linux, and runs the command in the terminal. A message will be prompted and ask for the password, please type in the log-in password of your current Linux login account and press *Enter*.

```
sudo apt update
```



```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

acs@acs-VirtualBox:~$ sudo apt update
[sudo] password for acs:
```

4. Install the dependencies for ACS VAS Test Tool and press *Enter*.

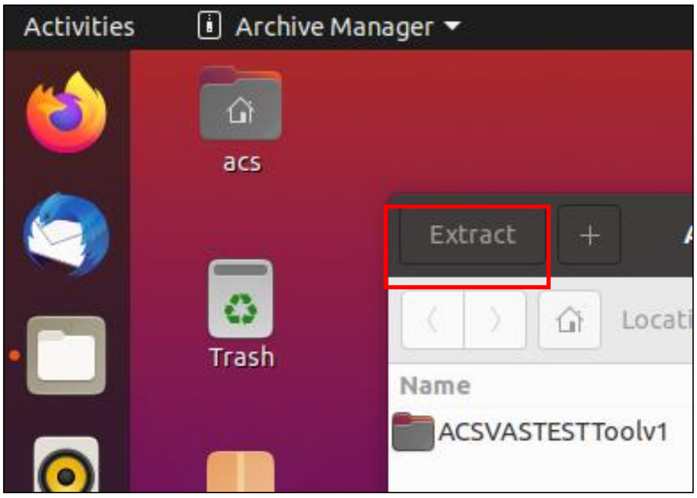
- Focal (20.04) or below

```
sudo apt-get install pcscd pcsc-tools libssl-dev qt5-default -y
```

- Jammy (22.04)

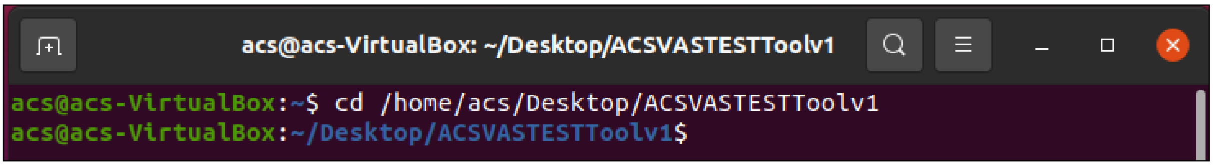
```
sudo apt-get install pcscd pcsc-tools libssl-dev qtbase5-dev qtchooser qt5-
qmake qtbase5-dev-tools -y
```

5. Unzip the zip file ACS_VAS_Test_Tool_Linux-X.XX into any directory in Linux.

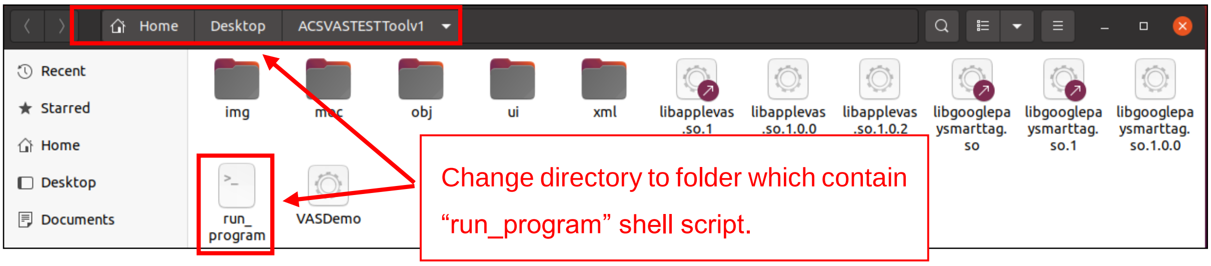


6. Open *Terminal* again, and change the directory (cd) to the folder which contains the “run_program” shell script. If the extracted file is on the desktop, the command will be (replace space in the directory to “\” symbol):

```
cd /home/YOUR_ACCOUNT_NAME/Desktop/ACSVASTESTToolv1
```



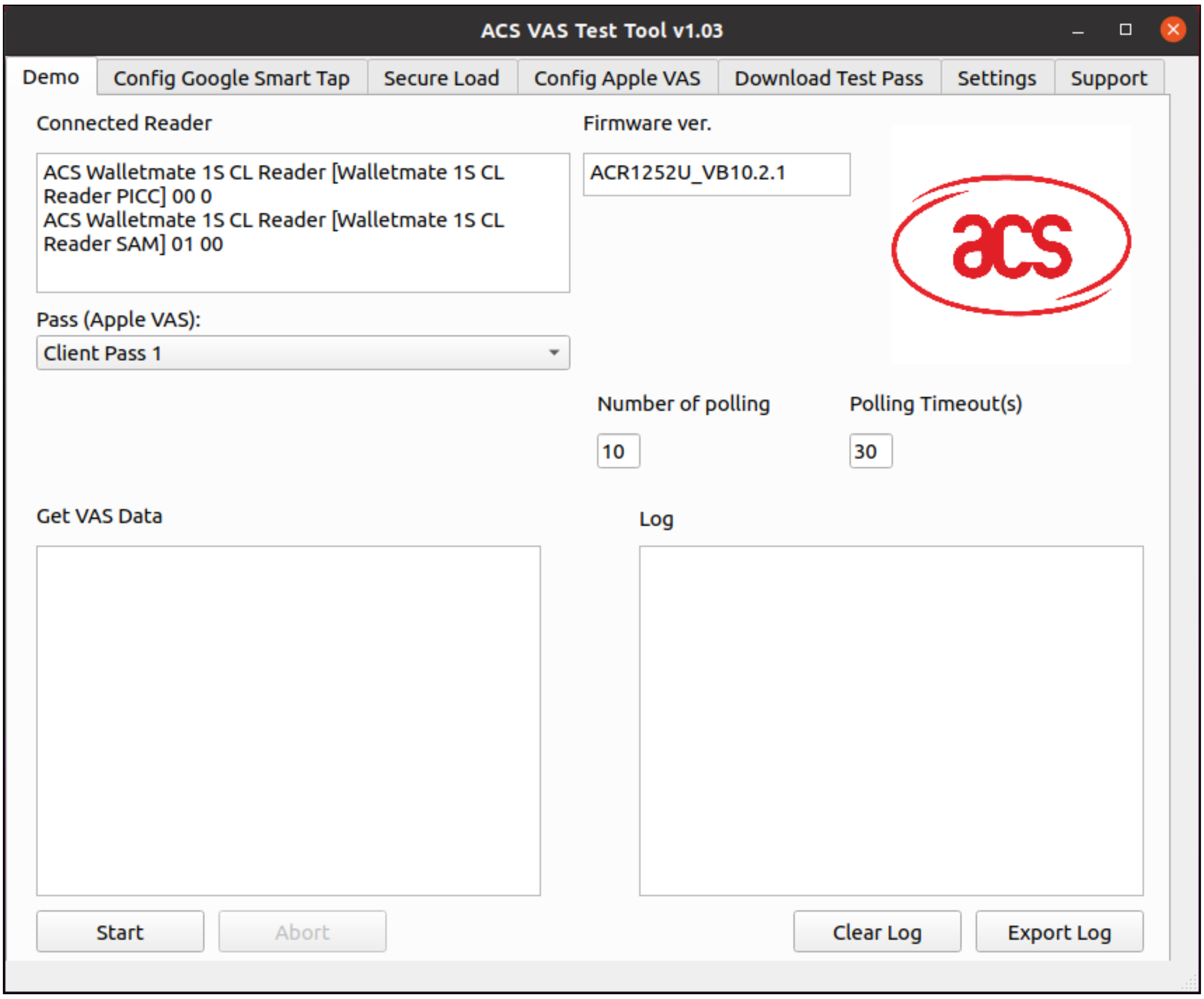
```
acs@acs-VirtualBox: ~/Desktop/ACSVASTESTToolv1
acs@acs-VirtualBox:~$ cd /home/acs/Desktop/ACSVASTESTToolv1
acs@acs-VirtualBox:~/Desktop/ACSVASTESTToolv1$
```



Change directory to folder which contain “run_program” shell script.

7. Run the command in the Terminal and the application should be run.

```
./run_program
```



Connected Reader:
ACS Walletmate 1S CL Reader [Walletmate 1S CL Reader PICC] 00 0
ACS Walletmate 1S CL Reader [Walletmate 1S CL Reader SAM] 01 00

Firmware ver.: ACR1252U_VB10.2.1

Pass (Apple VAS): Client Pass 1

Number of polling: 10

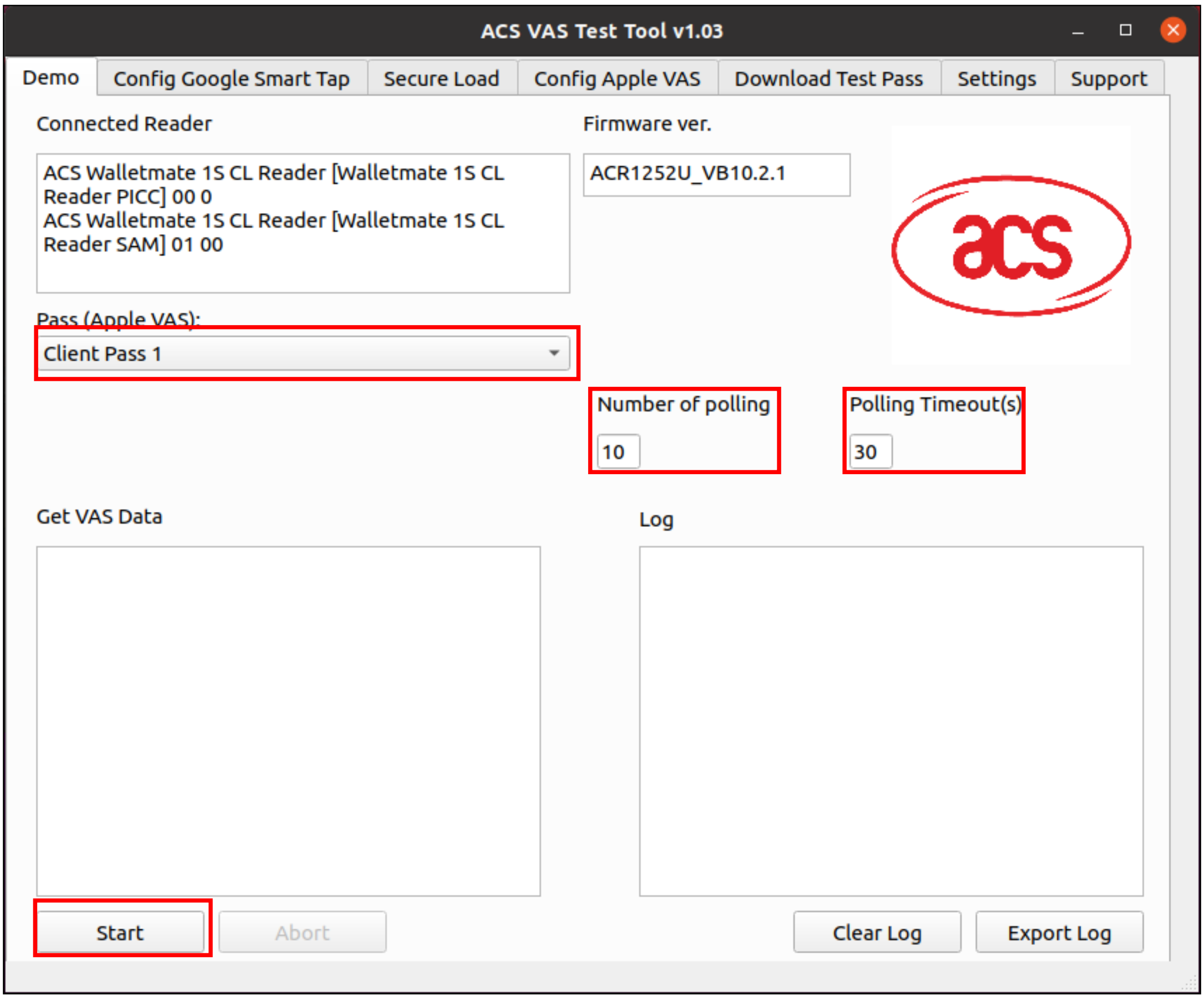
Polling Timeout(s): 30

Figure 3: ACS VAS Test Tool (Linux)

5.2.2. Using ACS VAS Test Tool with ACS Test Apple Pass

This section provides simple step-by-step procedures on how to test ACS Test Pass stored in Apple Wallet with ACS VAS Test Tool.

1. Go to *Demo* Tab.
2. Adjust the *Number Of Polling* and *Polling Timeout(s*) for each poll, select *Client Pass 1*, and then press *Start*. The passTypeIdentifier & Private Key for ACS Demo Pass is hard-coded into the demo



Connected Reader
ACS Walletmate 1S CL Reader [Walletmate 1S CL Reader PICC] 00 0
ACS Walletmate 1S CL Reader [Walletmate 1S CL Reader SAM] 01 00

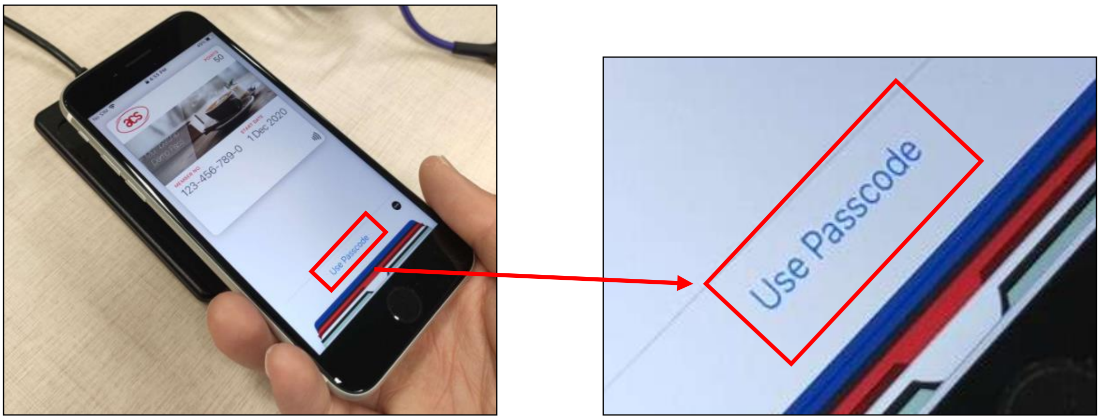
Firmware ver.: ACR1252U_VB10.2.1

Pass (Apple VAS): Client Pass 1

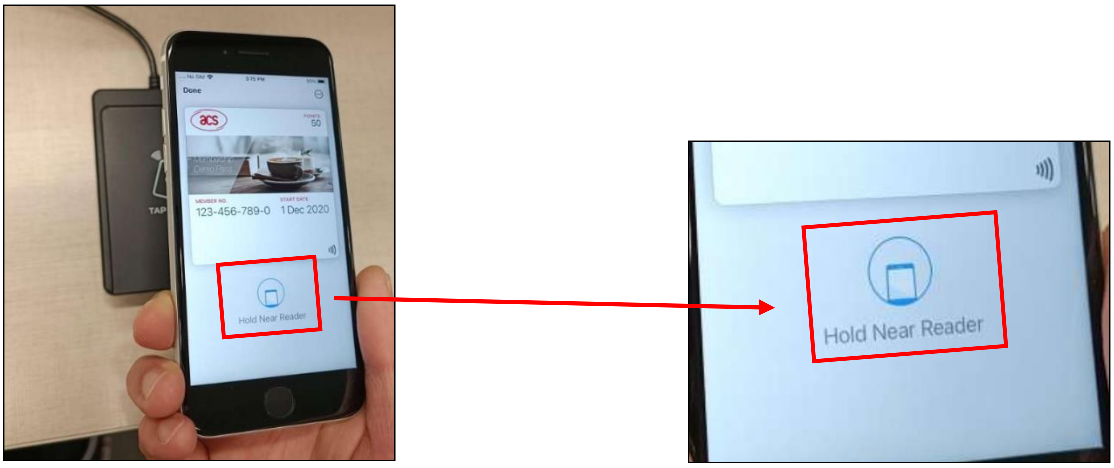
Number of polling: 10

Polling Timeout(s): 30

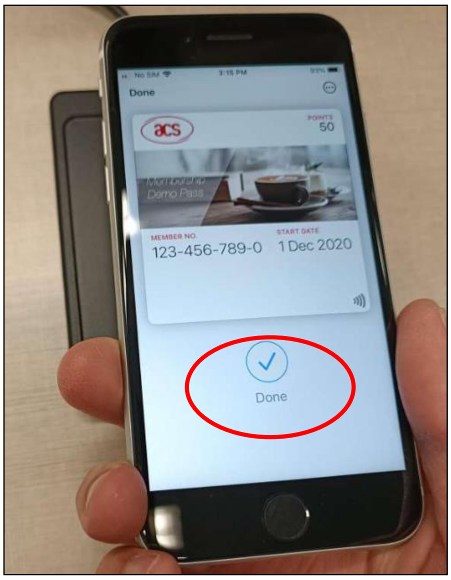
3. Tap the iPhone / Apple Watch on top of WalletMate Mobile Wallet NFC Reader.
4. ACS Test Pass will pop up. If your iPhone is password/Touch ID/Face ID protected, there shall be a prompt asking you to unlock the Pass.



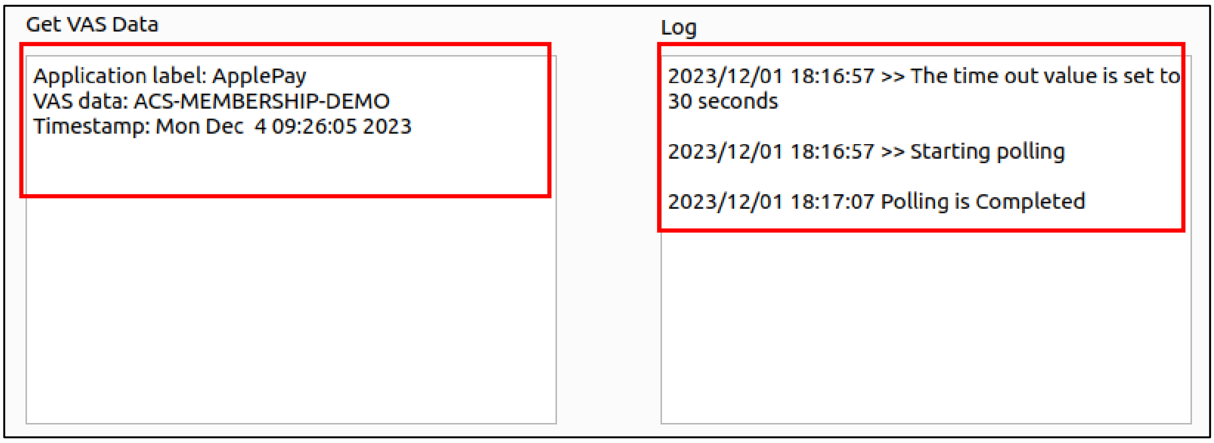
5. Tap the iPhone / Apple Watch on top of WalletMate Mobile Wallet NFC Reader again.



6. A tick mark should be shown on the screen, and the pass shall disappear.



7. Payload message and timestamp is shown in message box.



Get VAS Data

```
Application label: ApplePay
VAS data: ACS-MEMBERSHIP-DEMO
Timestamp: Mon Dec  4 09:26:05 2023
```

Log

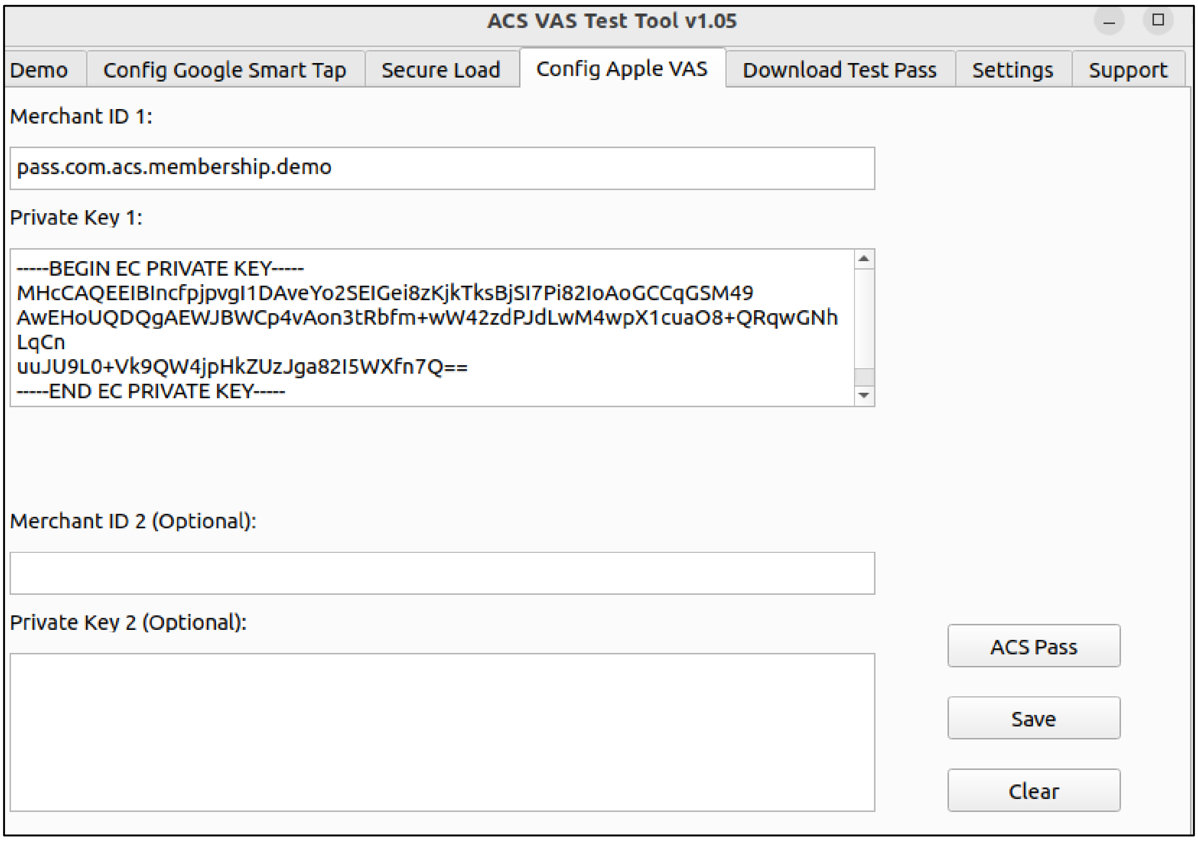
```
2023/12/01 18:16:57 >> The time out value is set to
30 seconds

2023/12/01 18:16:57 >> Starting polling

2023/12/01 18:17:07 Polling is Completed
```

5.2.3. Test Custom NFC-Enabled Apple Pass

This section provides a simple step-by-step procedure on how to use ACS VAS Test Tool with your custom NFC-Enabled Pass. If you had created your custom test pass with Apple, you should follow the guidance of this section.

You may refer to Apple VAS to learn more about the pre-requires of designing your own Apple Pass.

1. Go to the *Config Apple VAS* Tab and fill in the *pass Merchant ID 1* & *Private Key 1* (Be ware of the syntax) fields.
2. If you have two custom test passes wanted to test, please paste the 2nd passTypeIdentifier & Private Key in *pass Merchant ID 2* & *Private Key 2* respectively.
3. Press *Save*



Merchant ID 1:

pass.com.acs.membership.demo

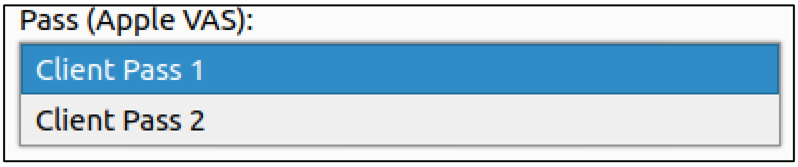
Private Key 1:

```
-----BEGIN EC PRIVATE KEY-----
MHcCAQEEIBIncfpjpvgI1DAveYo2SEIGei8zKjkTksBjSI7Pi82IoAoGCCqGSM49
AwEHoUQDQgAEWJBWCp4vAon3tRbfm+wW42zdPJdLwM4wpX1cuaO8+QRqwGNh
LqCn
uuJU9L0+Vk9QW4jpHkZUzJga82I5WXfn7Q==
-----END EC PRIVATE KEY-----
```

Merchant ID 2 (Optional):

Private Key 2 (Optional):

4. Go to *Demo* Tab, select *Client Pass 1* or *Client Pass 2* and press *Start*. Client Pass 1 and Client Pass 2 refers to the passTypeIdentifier 1, passTypeIdentifier 2 and theirs corresponding private key in *Config Apple VAS* Tab.



Pass (Apple VAS):

Client Pass 1

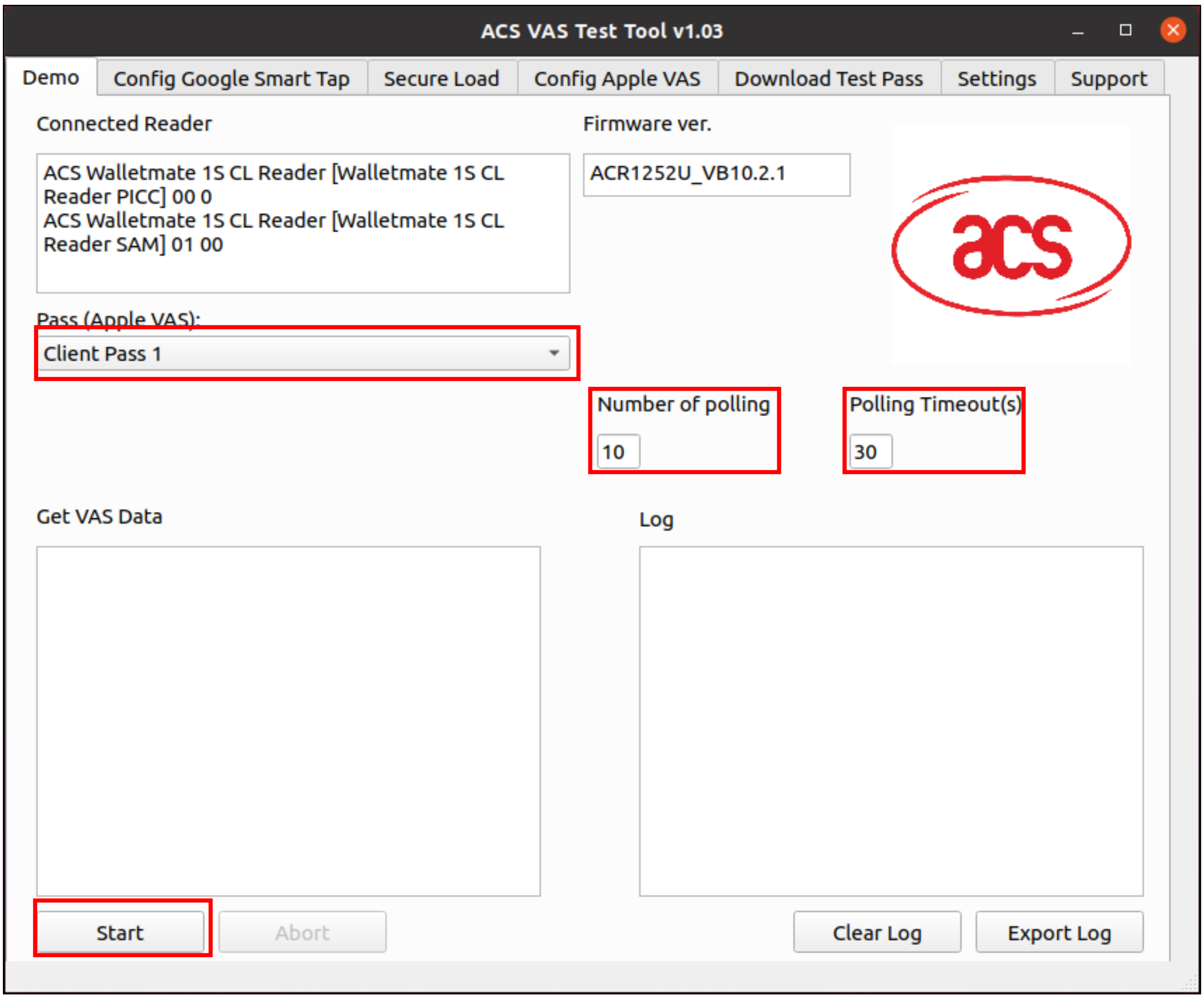
Client Pass 2

5. Then follow the rest procedures on Using ACS VAS Test Tool with ACS Test Apple Pass, you should be able to retrieve the information stored on your own Custom NFC-Enabled Pass.

5.2.4. Using ACS VAS Test Tool with Google Test Pass

This section provides simple step-by-step procedures on how to test ACS Test Pass stored in Google Wallet with ACS VAS Test Tool.

1. Go to *Demo* Tab.
2. You may adjust *the Number Of Polling* and *Polling Timeout(s)* for each poll, press *Start*. Leave the *Config Google Smart Tap* and *Secure Load* tabs in the default setting if you wish to test with Google Demo Test Pass



Connected Reader: ACS Walletmate 1S CL Reader [Walletmate 1S CL Reader PICC] 00 0
ACS Walletmate 1S CL Reader [Walletmate 1S CL Reader SAM] 01 00

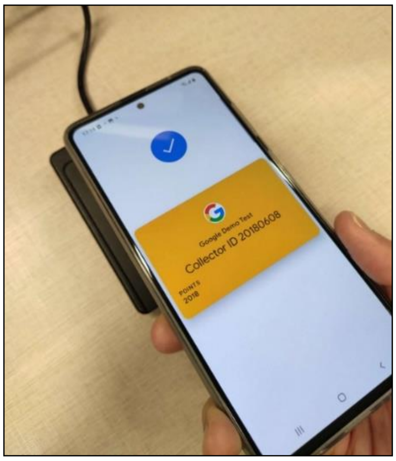
Firmware ver.: ACR1252U_VB10.2.1

Pass (Apple VAS): Client Pass 1

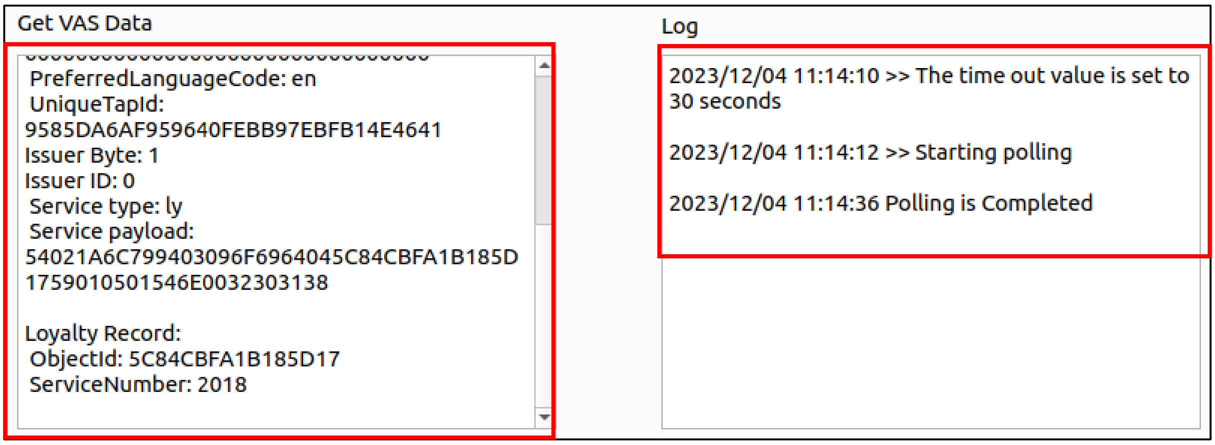
Number of polling: 10

Polling Timeout(s): 30

3. Unlock the screen, turn on the NFC on your android device, and tap the android mobile phone on top of WalletMate Mobile Wallet NFC Reader, the Google Demo Test Pass shall pop up immediately.



4. Payload message and timestamp is shown in the "Demo" Tab message box.



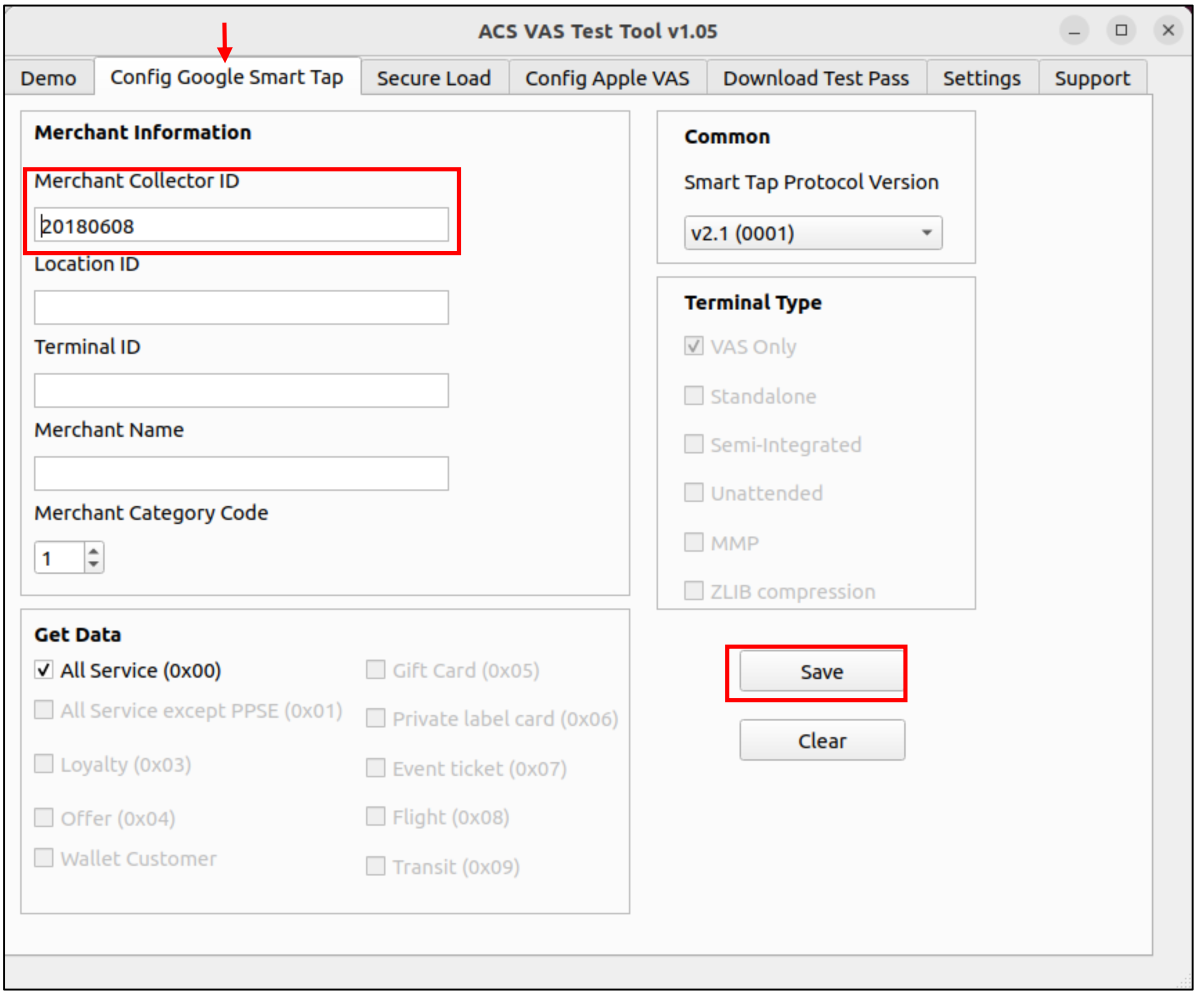
```
 PreferredLanguageCode: en
 UniqueTapId:
9585DA6AF959640FEBB97EBFB14E4641
Issuer Byte: 1
Issuer ID: 0
 Service type: ly
 Service payload:
54021A6C799403096F6964045C84CBFA1B185D
1759010501546E0032303138

Loyalty Record:
 ObjectId: 5C84CBFA1B185D17
 ServiceNumber: 2018
```

```
2023/12/04 11:14:10 >> The time out value is set to
30 seconds

2023/12/04 11:14:12 >> Starting polling

2023/12/04 11:14:36 Polling is Completed
```

5.2.5. Test Custom NFC-Enabled Google Pass

If you want to develop your NFC-Enabled Pass with Google and had generate the encrypted LTPK and Public Key, and had uploaded the Public Key into your Google Wallet API Issuer Account, you can follow the below guidelines and test your pass with ACS VAS Test Tool.

1. Navigates to *Config Google Smart Tap* Tab, input the Merchant Collector ID, and press *Save*.



ACS VAS Test Tool v1.05

Merchant Information

Merchant Collector ID: 20180608

Location ID:

Terminal ID:

Merchant Name:

Merchant Category Code: 1

Common

Smart Tap Protocol Version: v2.1 (0001)

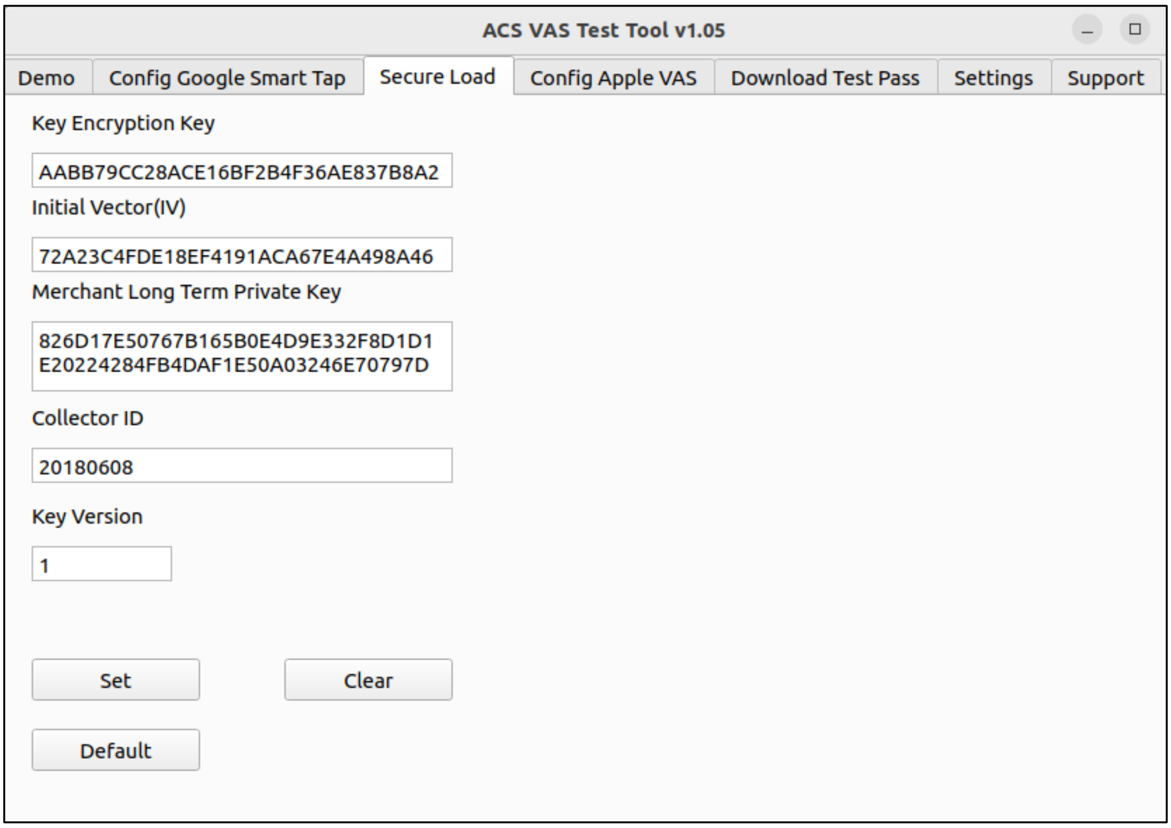
Terminal Type

☑ VAS Only
□ Standalone
□ Semi-Integrated
□ Unattended
□ MMP
□ ZLIB compression

Get Data

☑ All Service (0x00)
□ All Service except PPSE (0x01)
□ Loyalty (0x03)
□ Offer (0x04)
□ Wallet Customer
□ Gift Card (0x05)
□ Private label card (0x06)
□ Event ticket (0x07)
□ Flight (0x08)
□ Transit (0x09)

2. Go to the *Secure Load* Tab, fill in the Merchant Long Term Private Key (Refer to Get Your Key Pair For Smart Tap), together with your Collector ID and Key Version, and then press *Set*.



Key Encryption Key

AABB79CC28ACE16BF2B4F36AE837B8A2

Initial Vector(IV)

72A23C4FDE18EF4191ACA67E4A498A46

Merchant Long Term Private Key

826D17E50767B165B0E4D9E332F8D1D1
E20224284FB4DAF1E50A03246E70797D

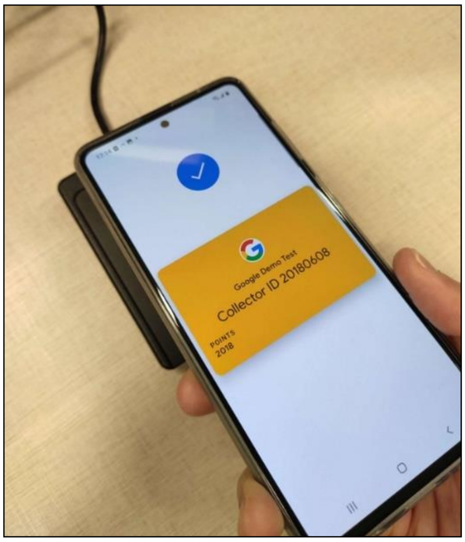
Collector ID

20180608

Key Version

1

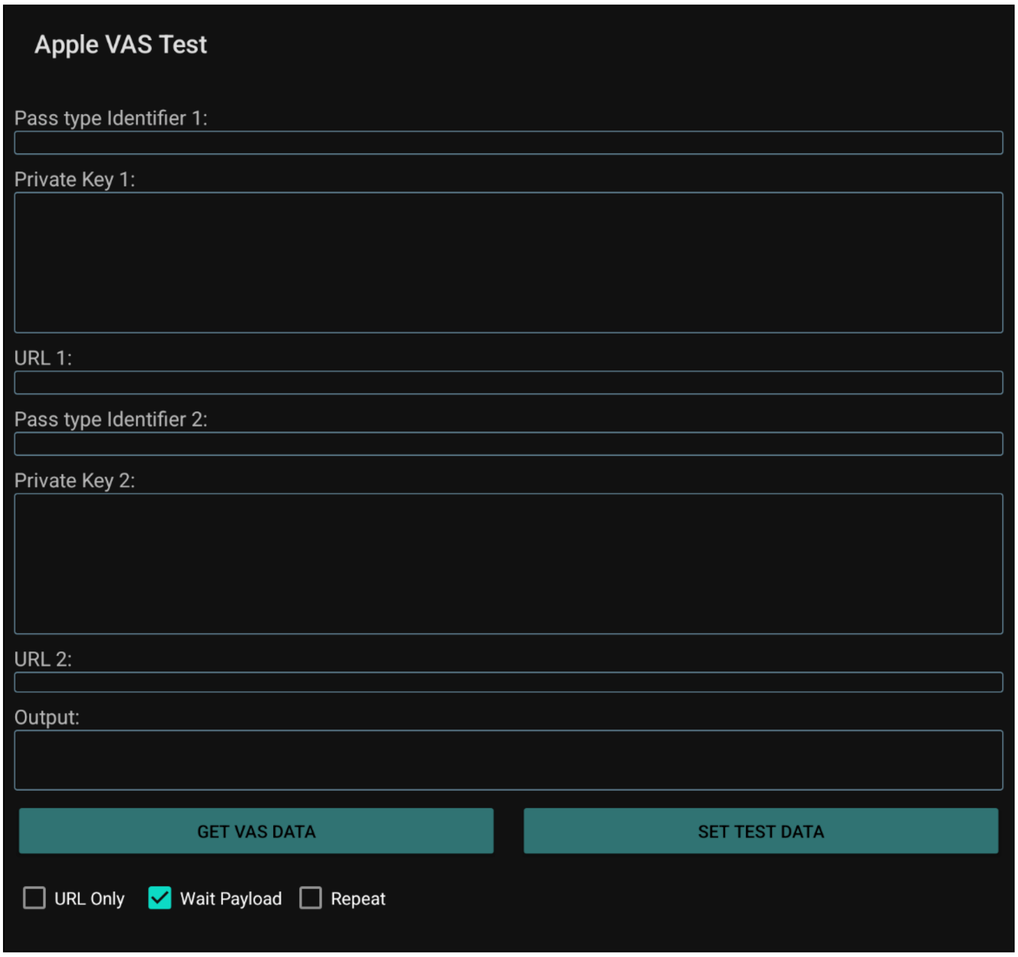
3. Then follow the rest procedures on Using ACS VAS Test Tool with Google Test Pass, you should be able to retrieve the information stored on your own Custom NFC-Enabled Pass.



5.3. ACS VAS Test Tool (Android)

5.3.1. Launch the Apple Vas Test App

1. Download the file *ACS_VAS_Test_Tool_Android-XXX* which provided by our salesman to the Android Mobile or Tablet.
2. Navigate to ACS_VAS_Test_Tool_Android-XXX/ACSVas API demo programs/TestAppleVas-XXX/, click *TestAppleVas-X.X.X.apk* to install the application. The JAVA Source code is located inside ACS_VAS_Test_Tool_Android-XXX/ACSVas API demo programs/TestAppleVas-XXX/app/src.
3. Connect your WalletMate Mobile Wallet NFC Reader to your Android Mobile or Tablet via USB cable.
4. Launch Apple VAS Test app
5. The following screen shall appear.



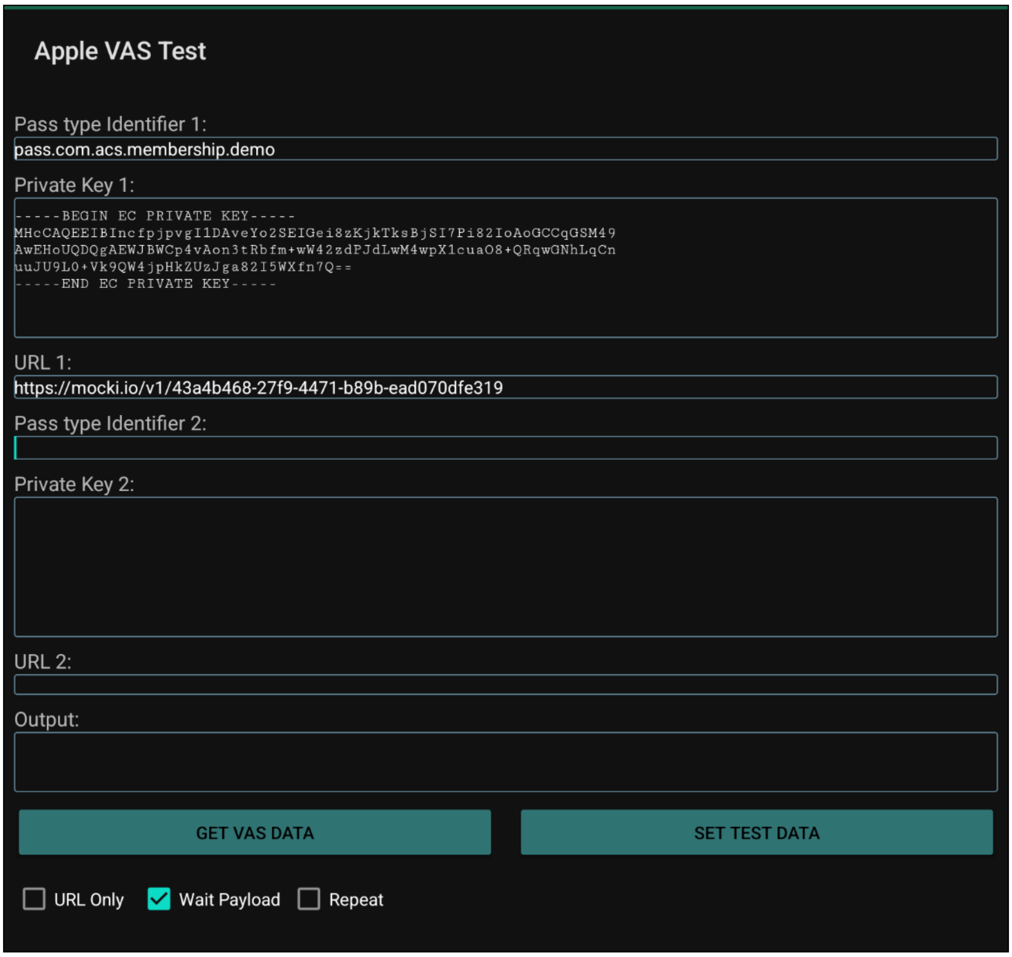
☐ URL Only ☑ Wait Payload ☐ Repeat

Figure 4: Apple VAS Test (Android)

5.3.2. Using ACS VAS Test Tool with ACS Test Apple Pass

This section provides simple step-by-step procedures on how to test ACS Test Pass stored in Apple Wallet with ACS VAS Test Tool.

1. Click *SET TEST DATA*
2. Adjust the *URL* only, Wait Payload, Repeat for the polling, and then press *GET VAS DATA*.



Apple VAS Test

Pass type Identifier 1:
pass.com.acs.membership.demo

Private Key 1:

```
-----BEGIN EC PRIVATE KEY-----
MHcCAQEEIBIncfpjpvgI1DAveYo2SEIGei8zKjkTksBjSI7Pi82IoAoGCCqGSM49
AwEHoUQDQgAEWJBWCp4vAon3tRbfm+wW42zdPJdLwM4wpX1cuaO8+QRqwGNhLqCn
uuJU9L0+Vk9QW4jpHkZUzJga82I5WXfn7Q==
-----END EC PRIVATE KEY-----
```

URL 1:
https://mocki.io/v1/43a4b468-27f9-4471-b89b-ead070dfe319

Pass type Identifier 2:

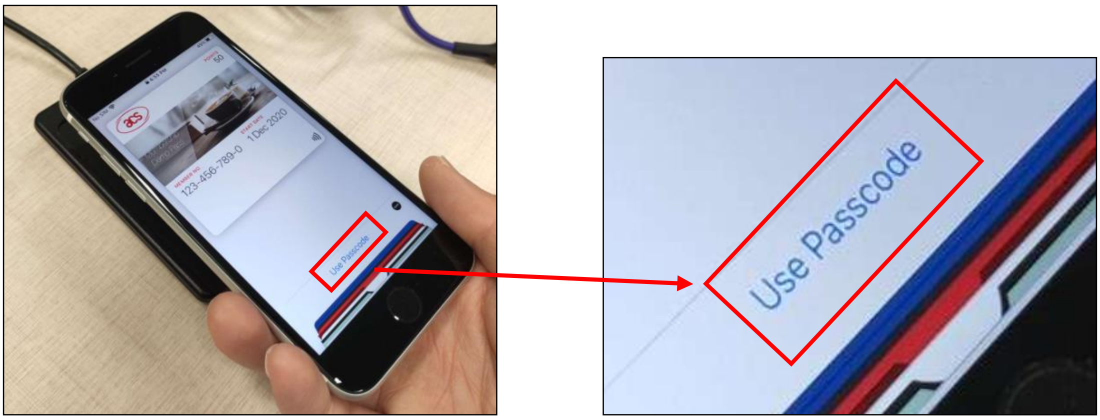
Private Key 2:

URL 2:

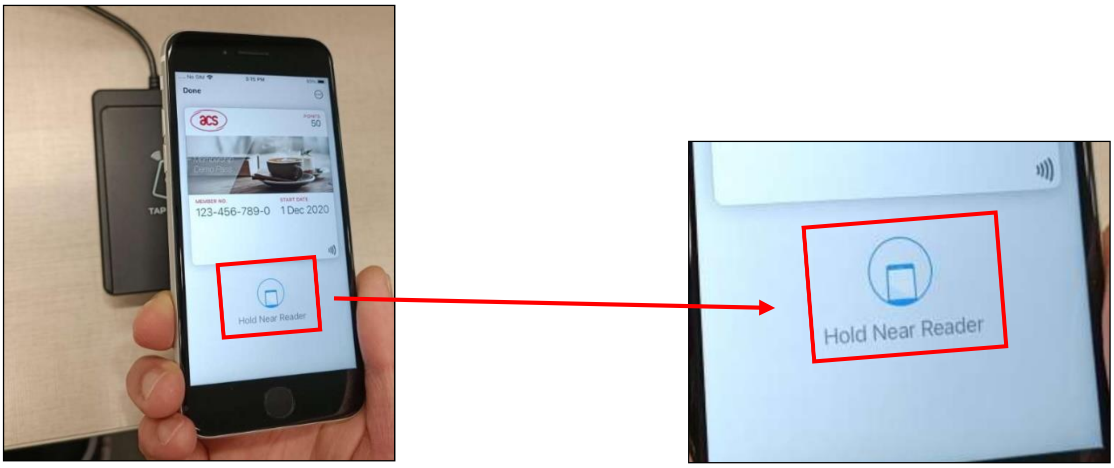
Output:

☐ URL Only ☑ Wait Payload ☐ Repeat

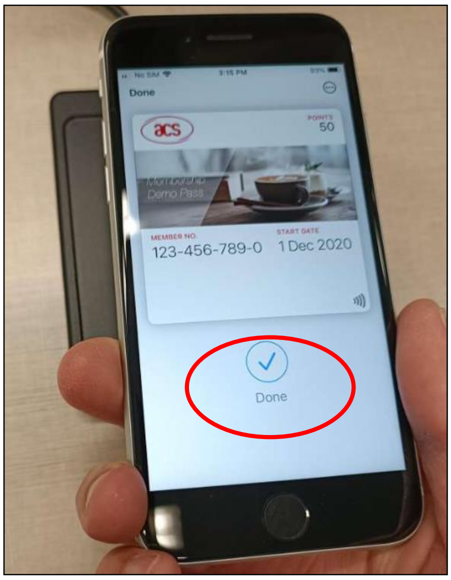
3. If there is a pop message, click *Allow*
4. Tap the iPhone / Apple Watch on top of WalletMate Mobile Wallet NFC Reader.
5. ACS Test Pass will pop up. If your iPhone is password/Touch ID/Face ID protected, there shall be a prompt asking you to unlock the Pass.



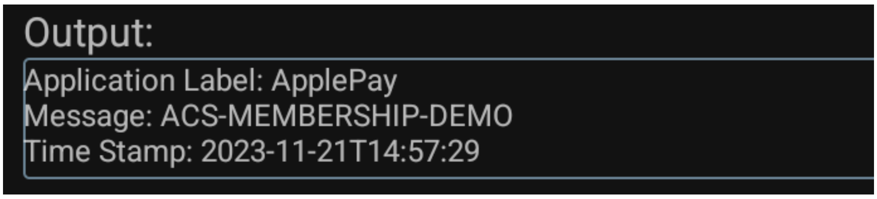
6. Tap the iPhone / Apple Watch on top of WalletMate Mobile Wallet NFC Reader again.



7. A tick mark should be shown on the screen, and the pass shall disappear.



8. Payload message and timestamp is shown in message box.



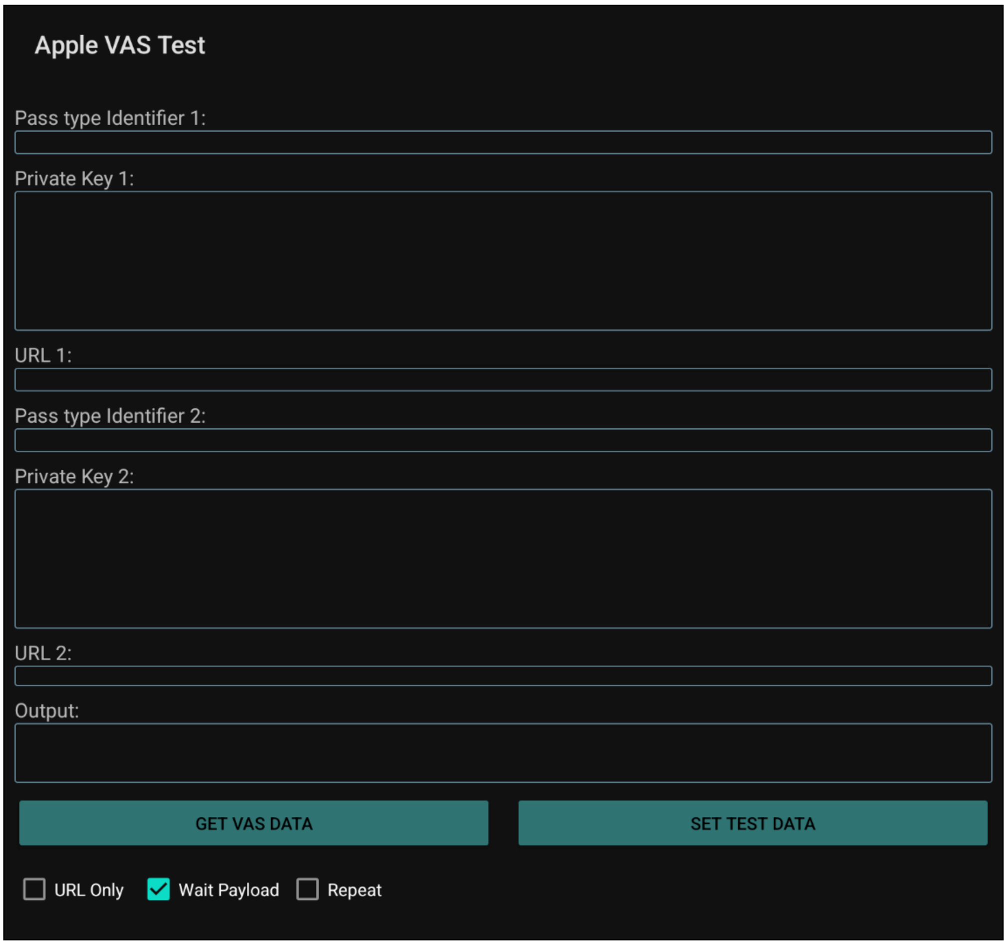
```
Output:
Application Label: ApplePay
Message: ACS-MEMBERSHIP-DEMO
Time Stamp: 2023-11-21T14:57:29
```

5.3.3. Test Custom NFC-Enabled Apple Pass

This section provides a simple step-by-step procedure on how to use ACS VAS Test Tool with your custom NFC-Enabled Pass. If you had created your custom test pass with Apple, you should follow the guidance of this section.

You may refer to Apple VAS to learn more about the pre-requires of designing your own Apple Pass.

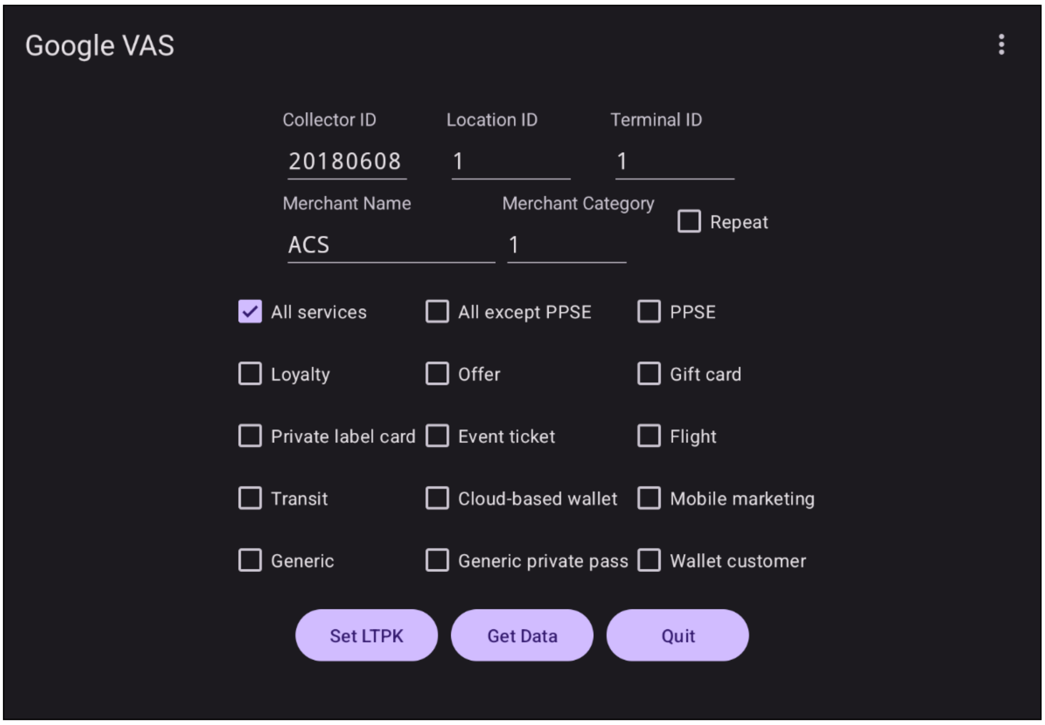
1. Fill in the *pass Type Identifier 1*, *Private Key 1* and URL 1(Be ware of the syntax) fields.
2. If you have two custom test passes wanted to test, please paste the 2nd passTypeIdentifier, Private Key and URL in *pass Type Identifier 2*, *Private Key 2* and *URL 2* respectively.
3. Press *SET TEST DATA*



4. If there is a pop message, click *Allow*
5. Then follow the rest procedures on Using ACS VAS Test Tool with ACS Test Apple Pass, you should be able to retrieve the information stored on your own Custom NFC-Enabled Pass.

5.3.4. Launch the TestGoogleVas App

1. Download the file *ACS_VAS_Test_Tool_Android-XXX* which provided by our salesman to the Android Mobile or Tablet.
2. Go to ACS_VAS_Test_Tool_Android-XXX/ACSVas API demo programs/TestGoogleVas-XXX/, click *TestGoogleVas-X.X.X.apk* to install the application.
   The JAVA Source code is located inside ACS_VAS_Test_Tool_Android-XXX/ACSVas API demo programs/TestGoogleVas-XXX/app/src.
3. Connect your WalletMate Mobile Wallet NFC Reader to your Android Mobile or Tablet via USB cable.
4. Launch TestGoogleVas app
5. The following screen shall appear.



Google VAS

Collector ID: 20180608
Location ID: 1
Terminal ID: 1
Merchant Name: ACS
Merchant Category: 1
☐ Repeat

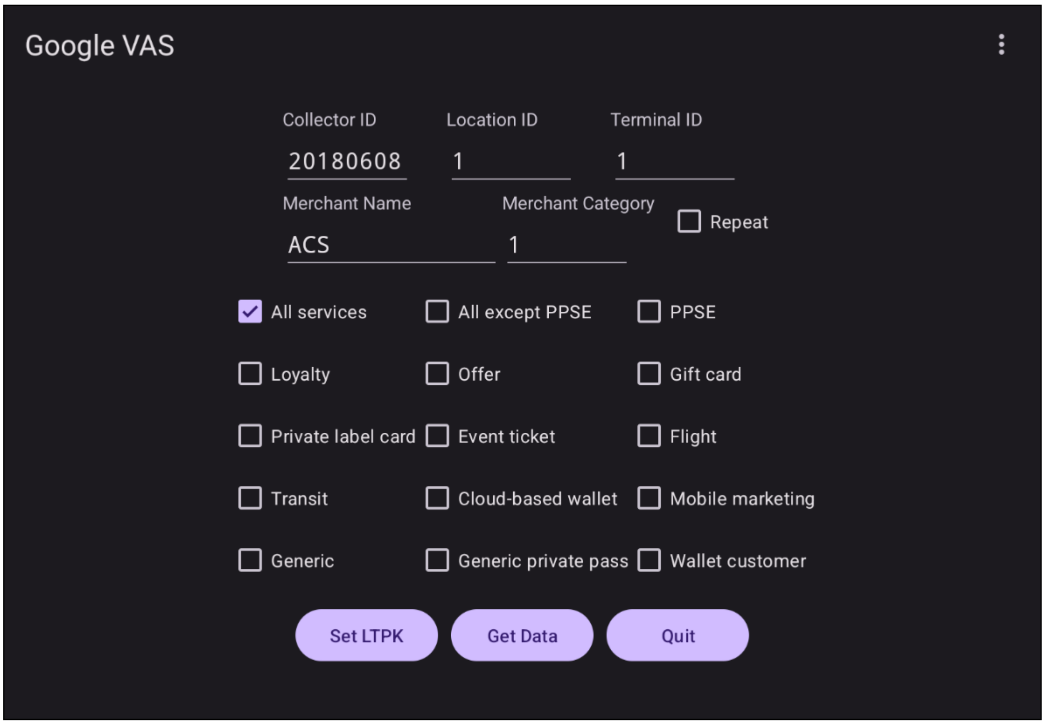
☑ All services ☐ All except PPSE ☐ PPSE
☐ Loyalty ☐ Offer ☐ Gift card
☐ Private label card ☐ Event ticket ☐ Flight
☐ Transit ☐ Cloud-based wallet ☐ Mobile marketing
☐ Generic ☐ Generic private pass ☐ Wallet customer

Figure 5: Google VAS (Android)

5.3.5. Using ACS VAS Test Tool with Google Test Pass

This section provides simple step-by-step procedures on how to test ACS Test Pass stored in Google Wallet with ACS VAS Test Tool.

1. You may adjust *Repeat* for the polling, press *Get Data*. Leave the default setting if you wish to test with Google Demo Test Pass



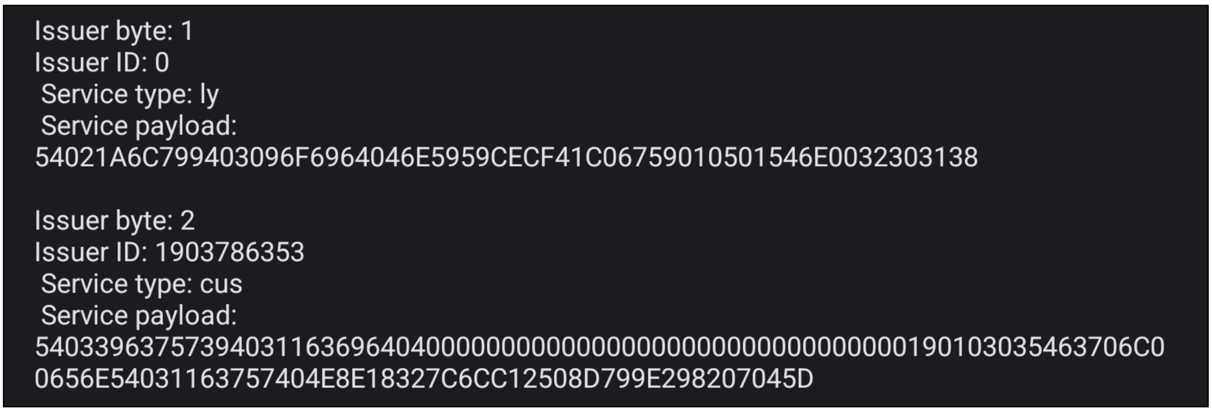
Google VAS

Collector ID: 20180608
Location ID: 1
Terminal ID: 1
Merchant Name: ACS
Merchant Category: 1
☐ Repeat

☑ All services	☐ All except PPSE	☐ PPSE
☐ Loyalty	☐ Offer	☐ Gift card
☐ Private label card	☐ Event ticket	☐ Flight
☐ Transit	☐ Cloud-based wallet	☐ Mobile marketing
☐ Generic	☐ Generic private pass	☐ Wallet customer

2. Click *Get Data*
3. If there is a pop message, click *Allow*
4. Unlock the screen, turn on the NFC on your android device, and tap the android mobile phone on top of WalletMate Mobile Wallet NFC Reader, the Google Demo Test Pass shall pop up immediately.

5. Payload message and timestamp is shown in message box.

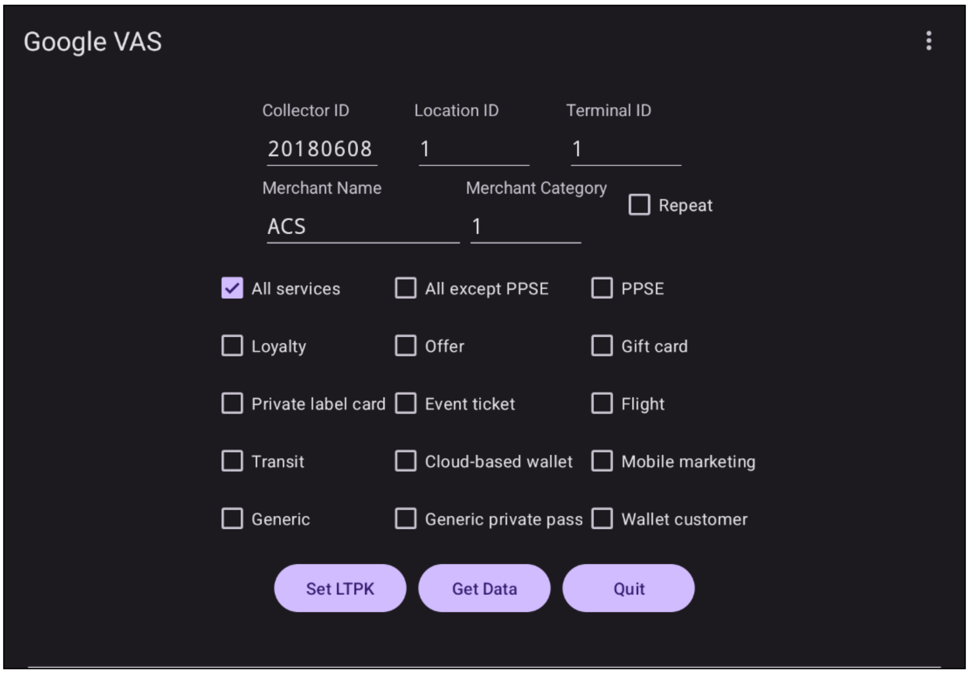


```
Issuer byte: 1
Issuer ID: 0
 Service type: ly
 Service payload:
54021A6C799403096F6964046E5959CECF41C06759010501546E0032303138

Issuer byte: 2
Issuer ID: 1903786353
 Service type: cus
 Service payload:
54033963757394031163696404000000000000000000000000000000019010303546370 6C0
0656E54031163757404E8E18327C6CC12508D799E298207045D
```

5.3.6. Test Custom NFC-Enabled Google Pass

If you want to develop your NFC-Enabled Pass with Google and generate the encrypted LTPK and Public Key, and had uploaded the Public Key into your Google Wallet API Issuer Account, you can follow the below guidelines and test your pass with ACS VAS Test Tool.

1. Input the Collector ID.

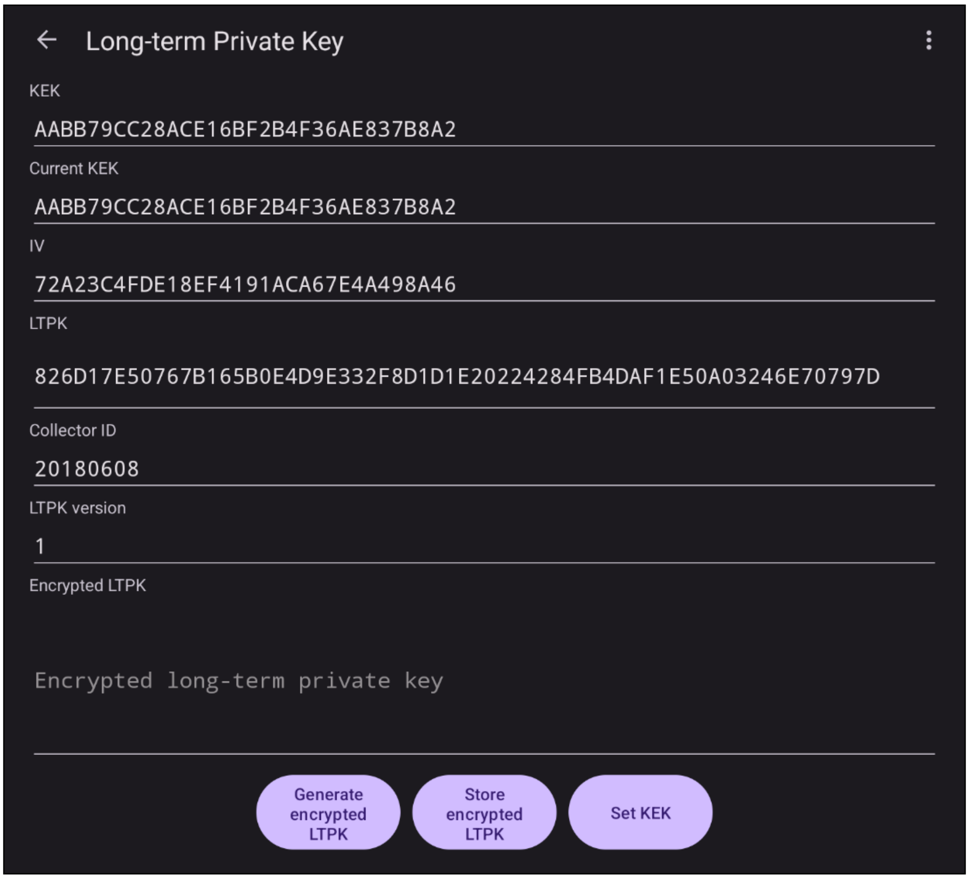


Google VAS

Collector ID: 20180608
Location ID: 1
Terminal ID: 1
Merchant Name: ACS
Merchant Category: 1
□ Repeat

☑ All services □ All except PPSE □ PPSE
□ Loyalty □ Offer □ Gift card
□ Private label card □ Event ticket □ Flight
□ Transit □ Cloud-based wallet □ Mobile marketing
□ Generic □ Generic private pass □ Wallet customer

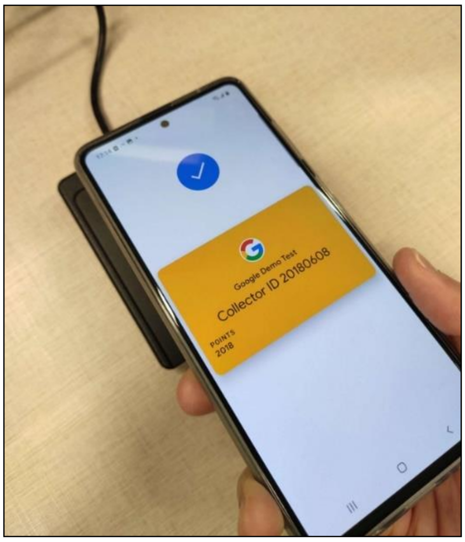
2. Click *Set LTPK*, fill in the Encrypted LTPK (Refer to Get Your Key Pair For Smart Tap), together with your Collector ID and LTPK Key Version, and then press *Store encrypted LTPK*.



Long-term Private Key

KEK: AABB79CC28ACE16BF2B4F36AE837B8A2
Current KEK: AABB79CC28ACE16BF2B4F36AE837B8A2
IV: 72A23C4FDE18EF4191ACA67E4A498A46
LTPK: 826D17E50767B165B0E4D9E332F8D1D1E20224284FB4DAF1E50A03246E70797D
Collector ID: 20180608
LTPK version: 1
Encrypted LTPK: Encrypted long-term private key

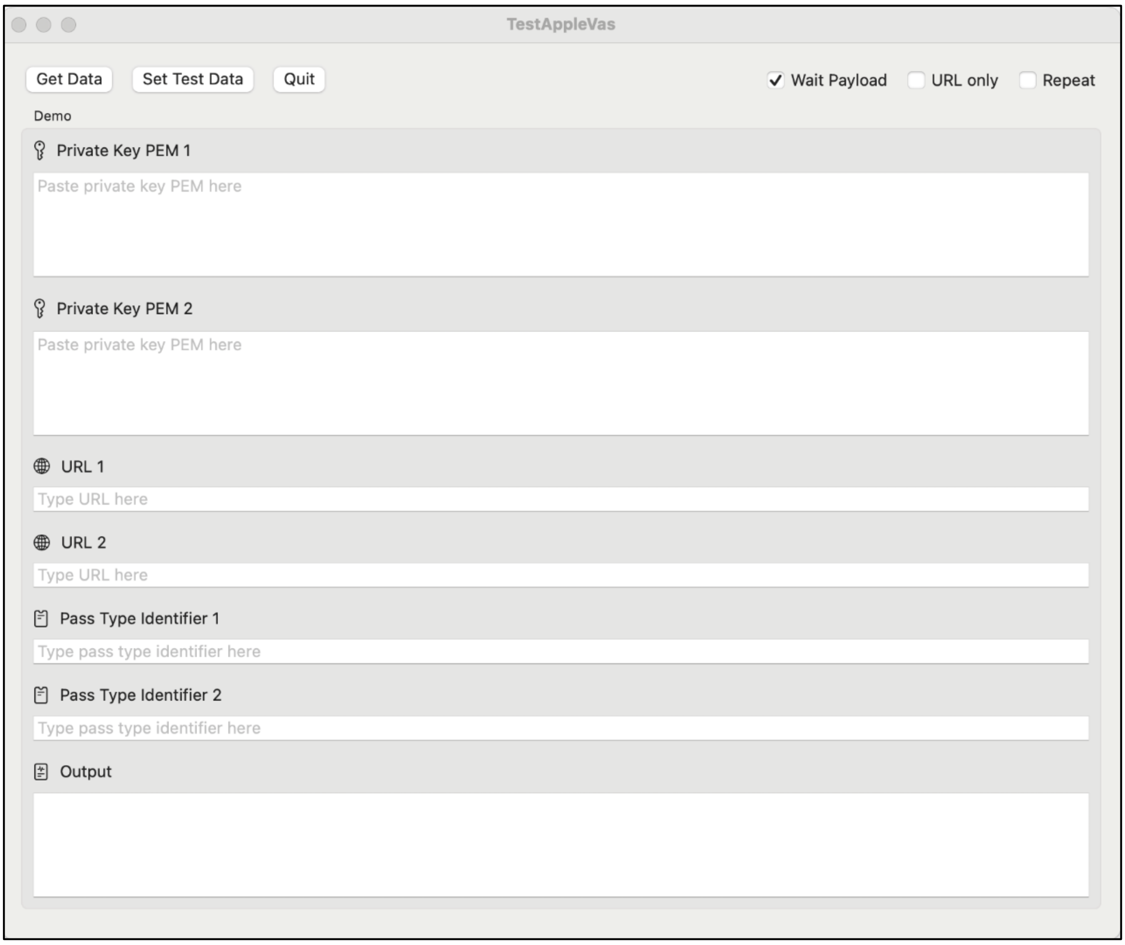
3. Click *return* at the top-left corner
4. Click *Get Data*
5. If there is a pop message, click *Allow*
6. Then follow the rest procedures on Using ACS VAS Test Tool with Google Test Pass, you should be able to retrieve the information stored on your own Custom NFC-Enabled Pass.



5.4. ACS VAS Test Tool (Mac)

5.4.1. Launch the TestAppleVas

1. Download and install the latest WalletMate Driver from the ACS Official website https://www.acs.com.hk/
2. Unzip the file *ACSVas-vX.X.X.zip* which provided by our salesman.
3. Connect your WalletMate Mobile Wallet NFC Reader to your MacBook via USB cable.
4. If there is a pop message, click *Allow*
5. Navigate to ACSVas-vX.X.X/ACSVas-vX.X.X/binaries/TestAppleVas/, double-click *TestAppleVas* to launch the application. The Swift Source code is located inside ACSVas-vX.X.X/ACSVas-vX.X.X/sources/TestAppleVas/.
6. The following screen shall appear.



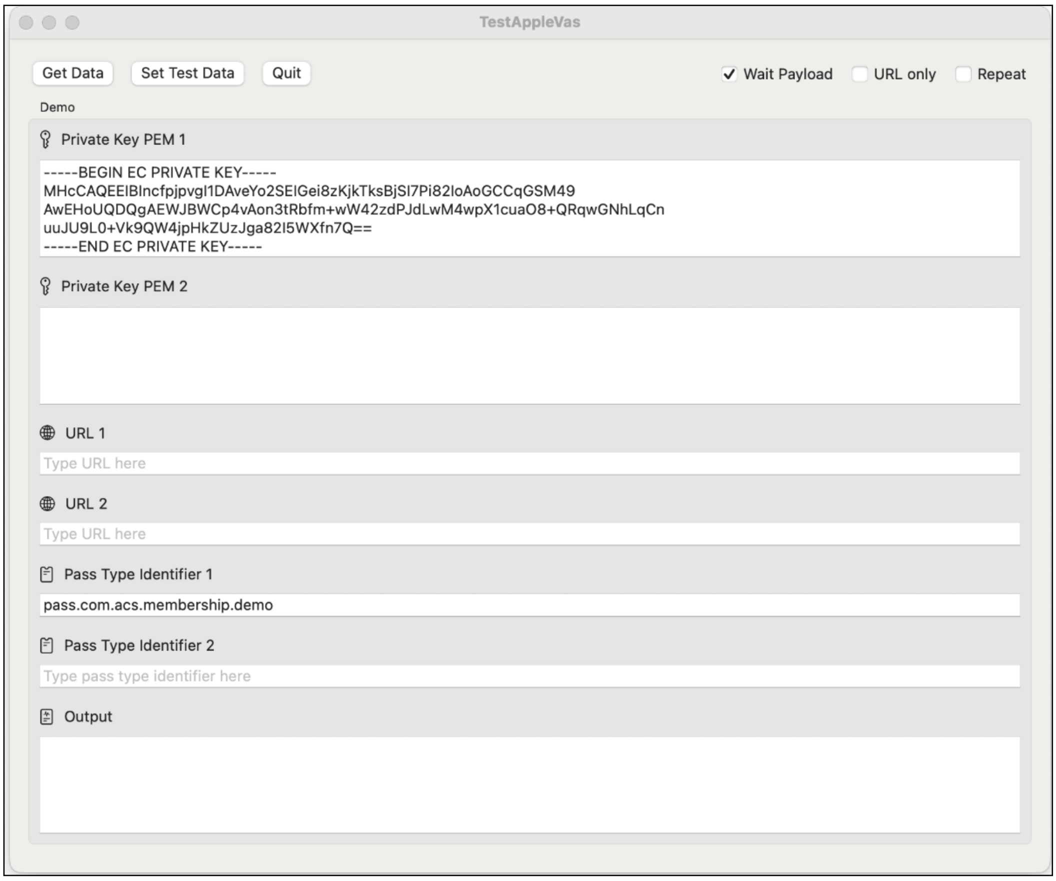
☑ Wait Payload □ URL only □ Repeat

Figure 6: TestAppleVas (Mac)

5.4.2. Using ACS VAS Test Tool with ACS Test Apple Pass

This section provides simple step-by-step procedures on how to test ACS Test Pass stored in Apple Wallet with ACS VAS Test Tool.

1. Click *SET TEST DATA*
2. Adjust the *URL Only*, Wait Payload, Repeat for the polling, and then press *Set Test Data*.



✓ Wait Payload □ URL only □ Repeat

Demo

Private Key PEM 1

```
-----BEGIN EC PRIVATE KEY-----
MHcCAQEEIBIncfpjpvgl1DAveYo2SEIGei8zKjkTksBjSI7Pi82IoAoGCCqGSM49
AwEHoUQDQgAEWJBWCp4vAon3tRbfm+wW42zdPJdLwM4wpX1cuaO8+QRqwGNhLqCn
uuJU9L0+Vk9QW4jpHkZUzJga82I5WXfn7Q==
-----END EC PRIVATE KEY-----
```

Private Key PEM 2

URL 1

Type URL here

URL 2

Type URL here

Pass Type Identifier 1

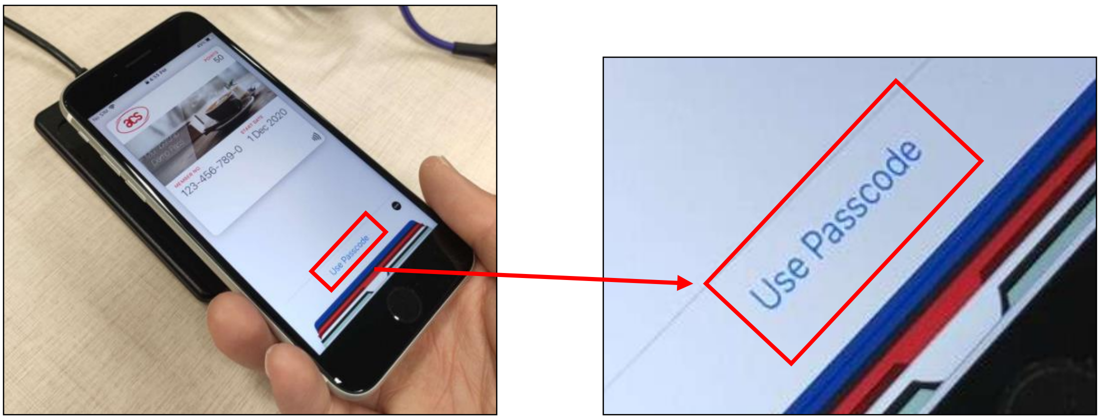
pass.com.acs.membership.demo

Pass Type Identifier 2

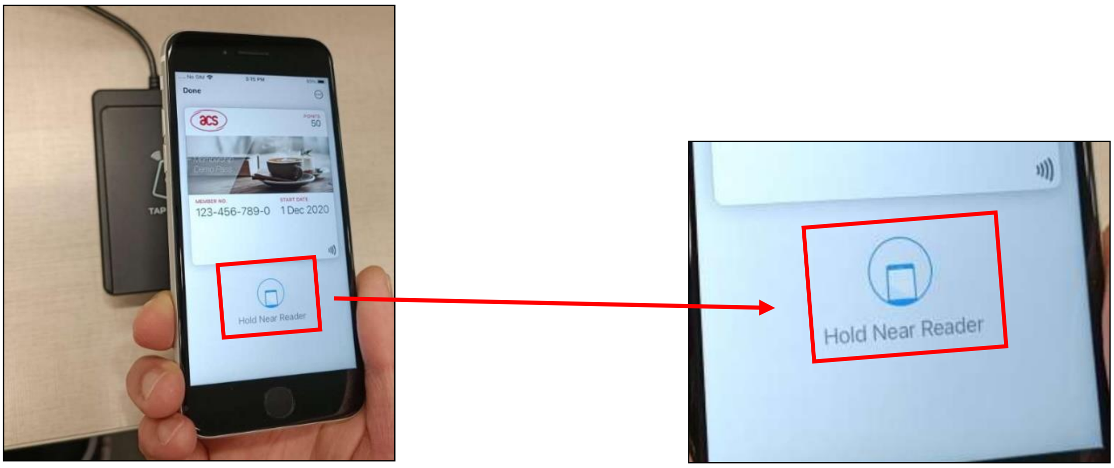
Type pass type identifier here

Output

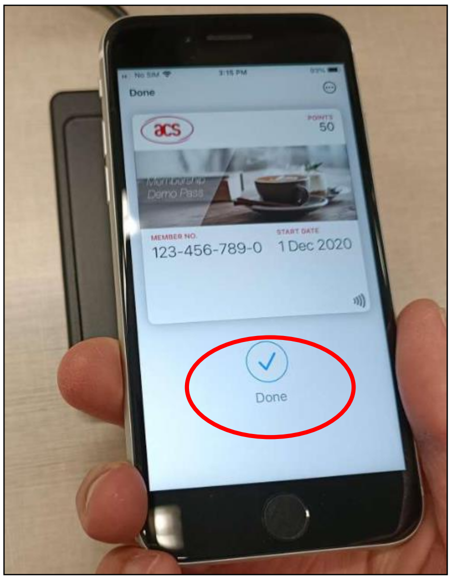
3. Tap the iPhone / Apple Watch on top of WalletMate Mobile Wallet NFC Reader.
4. ACS Test Pass will pop up. If your iPhone is password/Touch ID/Face ID protected, there shall be a prompt asking you to unlock the Pass.



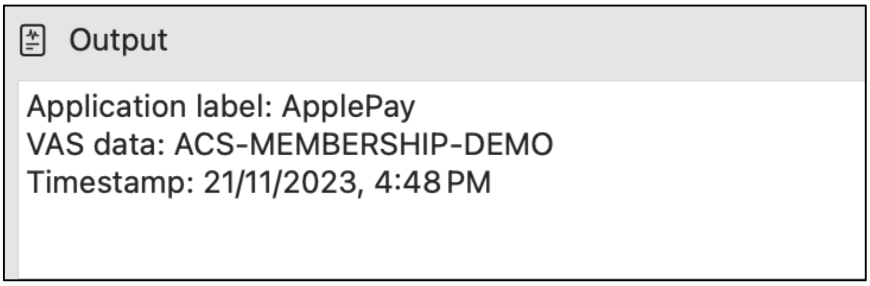
5. Tap the iPhone / Apple Watch on top of WalletMate Mobile Wallet NFC Reader again.



6. A tick mark should be shown on the screen, and the pass shall disappear.



7. Payload message and timestamp is shown in message box.



Output

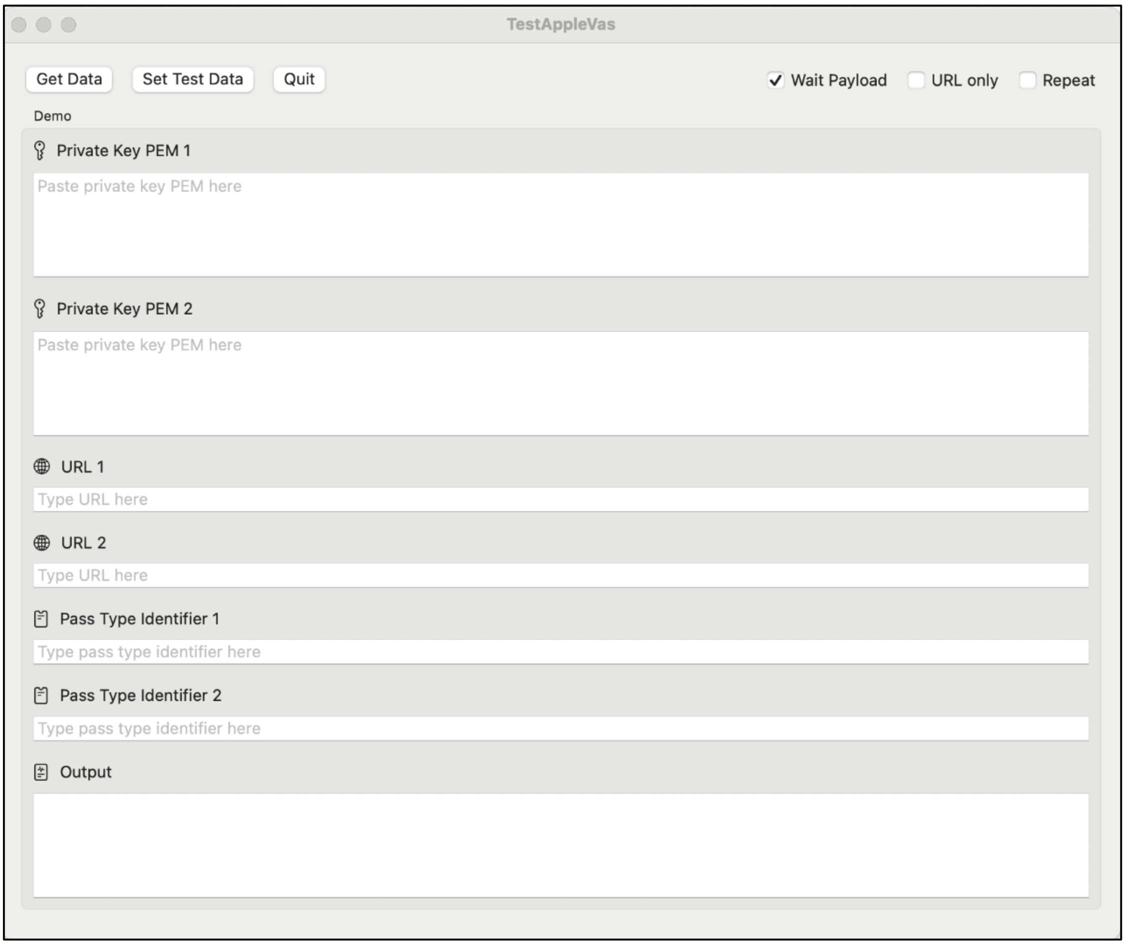
```
Application label: ApplePay
VAS data: ACS-MEMBERSHIP-DEMO
Timestamp: 21/11/2023, 4:48 PM
```

5.4.3. Test Custom NFC-Enabled Apple Pass

This section provides a simple step-by-step procedure on how to use ACS VAS Test Tool with your custom NFC-Enabled Pass. If you had created your custom test pass with Apple, you should follow the guidance of this section.

You may refer to Apple VAS to learn more about the pre-requires of designing your own Apple Pass.

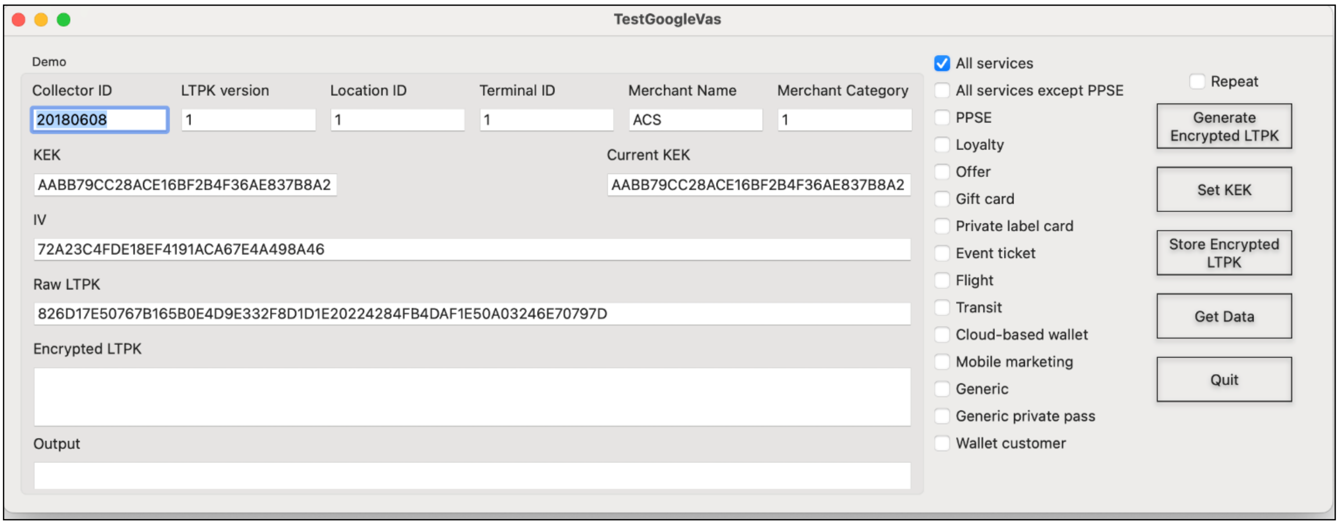
1. Fill in the *Pass Type Identifier 1*, *Private Key PEM 1* and URL 1(Be ware of the syntax) fields.
2. If you have two custom test passes wanted to test, please paste the 2nd passTypeIdentifier, Private Key and URL in *Pass Type Identifier 2*, *Private Key PEM 2* and *URL 2* respectively.
3. Press *Set Test Data*



4. Then follow the rest procedures on Using ACS VAS Test Tool with ACS Test Apple Pass, you should be able to retrieve the information stored on your own Custom NFC-Enabled Pass.

5.4.4. Launch the TestGoogleVas

1. Download and install the latest WalletMate Driver from the ACS Official website https://www.acs.com.hk/
2. Unzip the file *ACSVas-vX.X.X.zip* which provided by our salesman.
3. Connect your WalletMate Mobile Wallet NFC Reader to your MacBook via USB cable.
4. If there is a pop message, click *Allow*
5. Navigate to ACSVas-vX.X.X/ACSVas-vX.X.X/binaries/TestGoogleVas/, double-click *TestGoogleVas* to launch the application. The Swift Source code is located inside ACSVas-vX.X.X/ACSVas-vX.X.X/sources/TestGoogleVas/.
6. The following screen shall appear.



TestGoogleVas

Demo

Collector ID	LTPK version	Location ID	Terminal ID	Merchant Name	Merchant Category
20180608	1	1	1	ACS	1

KEK
AABB79CC28ACE16BF2B4F36AE837B8A2

Current KEK
AABB79CC28ACE16BF2B4F36AE837B8A2

IV
72A23C4FDE18EF4191ACA67E4A498A46

Raw LTPK
826D17E50767B165B0E4D9E332F8D1D1E20224284FB4DAF1E50A03246E70797D

Encrypted LTPK

Output

☑ All services
□ All services except PPSE
□ PPSE
□ Loyalty
□ Offer
□ Gift card
□ Private label card
□ Event ticket
□ Flight
□ Transit
□ Cloud-based wallet
□ Mobile marketing
□ Generic
□ Generic private pass
□ Wallet customer

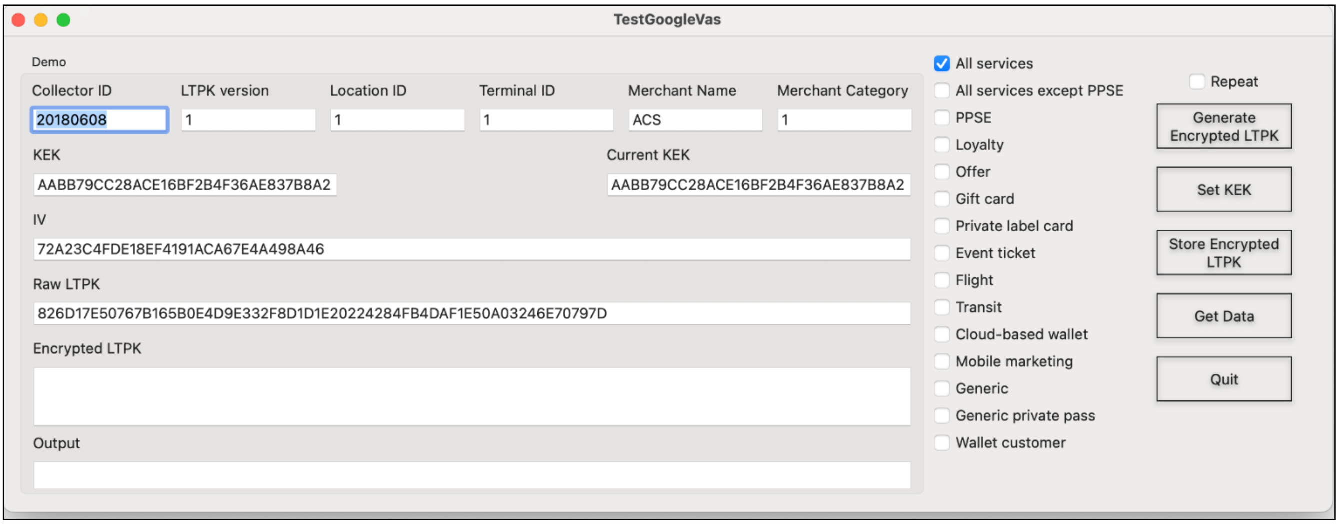
□ Repeat

Figure 7: TestGoogleVas (Mac)

5.4.5. Using ACS VAS Test Tool with Google Test Pass

This section provides simple step-by-step procedures on how to test ACS Test Pass stored in Google Wallet with ACS VAS Test Tool.

1. You may adjust *Repeat* for the polling, press *Get Data*. Leave the default setting if you wish to test with Google Demo Test Pass



Collector ID: 20180608
LTPK version: 1
Location ID: 1
Terminal ID: 1
Merchant Name: ACS
Merchant Category: 1

KEK: AABB79CC28ACE16BF2B4F36AE837B8A2
Current KEK: AABB79CC28ACE16BF2B4F36AE837B8A2

IV: 72A23C4FDE18EF4191ACA67E4A498A46

Raw LTPK: 826D17E50767B165B0E4D9E332F8D1D1E20224284FB4DAF1E50A03246E70797D

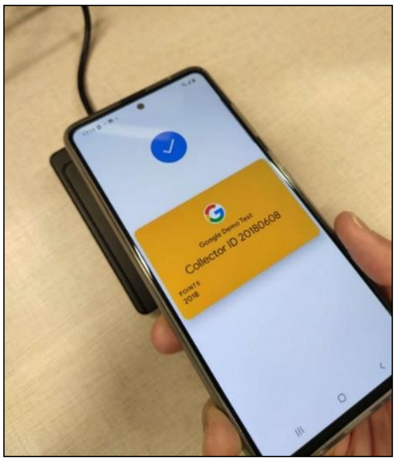
Encrypted LTPK

Output

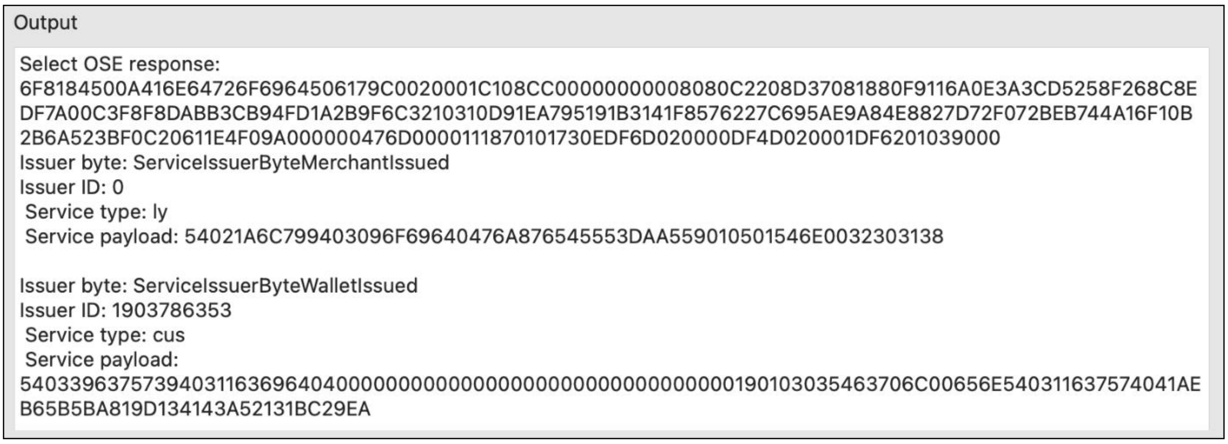
☑ All services
☐ All services except PPSE
☐ PPSE
☐ Loyalty
☐ Offer
☐ Gift card
☐ Private label card
☐ Event ticket
☐ Flight
☐ Transit
☐ Cloud-based wallet
☐ Mobile marketing
☐ Generic
☐ Generic private pass
☐ Wallet customer

☐ Repeat

2. Click *Get Data*
3. Unlock the screen, turn on the NFC on your android device, and tap the android mobile phone on top of WalletMate Mobile Wallet NFC Reader, the Google Demo Test Pass shall pop up immediately.



4. Payload message and timestamp is shown in message box.

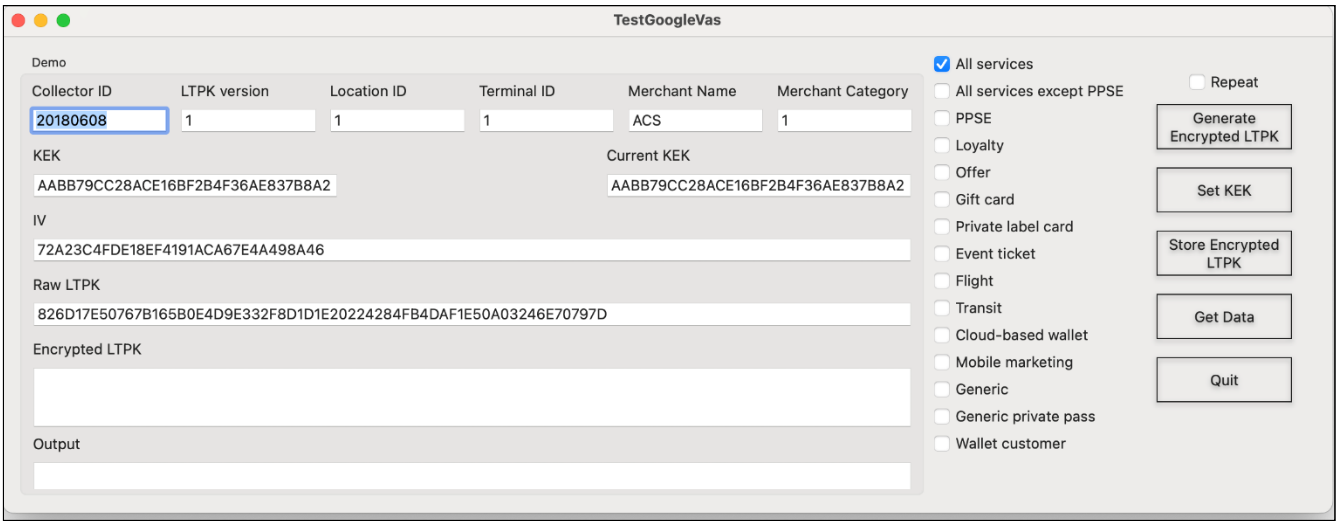


```
Select OSE response:
6F8184500A416E64726F6964506179C0020001C108CC00000000008080C2208D37081880F9116A0E3A3CD5258F268C8E
DF7A00C3F8F8DABB3CB94FD1A2B9F6C3210310D91EA795191B3141F8576227C695AE9A84E8827D72F072BEB744A16F10B
2B6A523BF0C20611E4F09A000000476D0000111870101730EDF6D020000DF4D020001DF6201039000
Issuer byte: ServiceIssuerByteMerchantIssued
Issuer ID: 0
 Service type: ly
 Service payload: 54021A6C799403096F69640476A876545553DAA559010501546E0032303138

Issuer byte: ServiceIssuerByteWalletIssued
Issuer ID: 1903786353
 Service type: cus
 Service payload:
5403396375739403116369640400000000000000000000000000000000190103035463706C00656E540311637574041AE
B65B5BA819D134143A52131BC29EA
```

5.4.6. Test Custom NFC-Enabled Google Pass

If you want to develop your NFC-Enabled Pass with Google and generate the encrypted LTPK and Public Key, and had uploaded the Public Key into your Google Wallet API Issuer Account, you can follow the below guidelines and test your pass with ACS VAS Test Tool.

1. Input the Collector ID.
2. Click *Set LTPK*, fill in the Encrypted LTPK (Refer to Get Your Key Pair For Smart Tap), together with your Collector ID and LTPK Key Version, and then press *Store Encrypted LTPK*.



TestGoogleVas

Demo

Collector ID	LTPK version	Location ID	Terminal ID	Merchant Name	Merchant Category
20180608	1	1	1	ACS	1

KEK: AABB79CC28ACE16BF2B4F36AE837B8A2

Current KEK: AABB79CC28ACE16BF2B4F36AE837B8A2

IV: 72A23C4FDE18EF4191ACA67E4A498A46

Raw LTPK: 826D17E50767B165B0E4D9E332F8D1D1E20224284FB4DAF1E50A03246E70797D

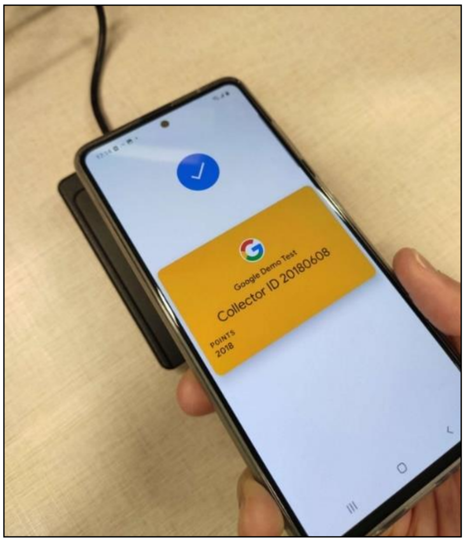
Encrypted LTPK:

Output:

☑ All services
☐ All services except PPSE
☐ PPSE
☐ Loyalty
☐ Offer
☐ Gift card
☐ Private label card
☐ Event ticket
☐ Flight
☐ Transit
☐ Cloud-based wallet
☐ Mobile marketing
☐ Generic
☐ Generic private pass
☐ Wallet customer

☐ Repeat

3. Click *Get Data*
4. Then follow the rest procedures on Using ACS VAS Test Tool with Google Test Pass, you should be able to retrieve the information stored on your own Custom NFC-Enabled Pass.



6.0. Switch Polling

WalletMate is a versatile device that can handle different VAS types. If the VAS type is not specified, WalletMate can automatically switch the polling mode to read various cards. The following C# code shows how to switch the polling mode in the order of Apple Vas, Google Smart Tap, and other smartcards.

C# Sample Code From the Windows demo

```
//Declare Task
private async Task GetVASDataTask(CancellationToken ct2)
{
    await Task.Run(() =>
    {
        if (ct2.IsCancellationRequested == false)
        {

            //Declare the get ose response variable
            string outputOSEResponseHex = "";
            string outputATRmessage = "";

            Console.WriteLine("Calling get Apple VAS Data");
            //Function Call for getting the Apple VAS Data
            int    retValue1    =    GetAppleVASData(ref    outputOSEResponseHex,    ref
outputATRmessage, _currentPassIndex);

            //SetScriptLogOut("Ret value = "+retValue1.ToString());
            if (retValue1 == 0)
            {
                // 0 - indicate the Apple Vas Get the Data successfully
                return;
            }
            else if (retValue1 == 3)
            {
                //Display no reader found message in the UI as well as pop up as Message
Box
                SetScriptLogOut("Device for the VAS Demo Program is not found");
                MessageBox.Show("Device for the VAS Demo Program is not found");
            }
            Console.WriteLine("Calling get Google Smart Tap Data");
            //MessageBox.Show("Call get google smart tap");
            GetGoogleSmartTapData(outputOSEResponseHex);
            //Print the get Google error output to the log
            SetScriptLogOut(_errorCode.ToString());
            //Check if the User Intervention is needed for the Google Smart Tap
            if (_errorCode == GoogleVas.ErrorCode.VasUserIntervention)
            {
                //Reset the OSE Hex data for another Polling
                outputOSEResponseHex = "";
                //Polling again after the VAS User Intervention is returned
                GetGoogleSmartTapData(outputOSEResponseHex);
            }
            //Check error code returned from GetData to see if other card is present
            if (_errorCode == GoogleVas.ErrorCode.VasOtherCardFound)
            {
                //Polling operation for other card
                GetOtherCardData(outputATRmessage);
            }
        }
        else
        {
            Console.WriteLine("=== IsCancellationRequested is true ===");
        }
    });
}
```

Appendix A. Key Pair Settings

Appendix A.1. Apple VAS

Appendix A.1.1. Enroll In Apple Developer’s Program For Apple Pass Generation

You will need to enroll in the Apple developer’s program before you can generate any custom Apple Passes, you can enroll in the developer’s program at the following website:

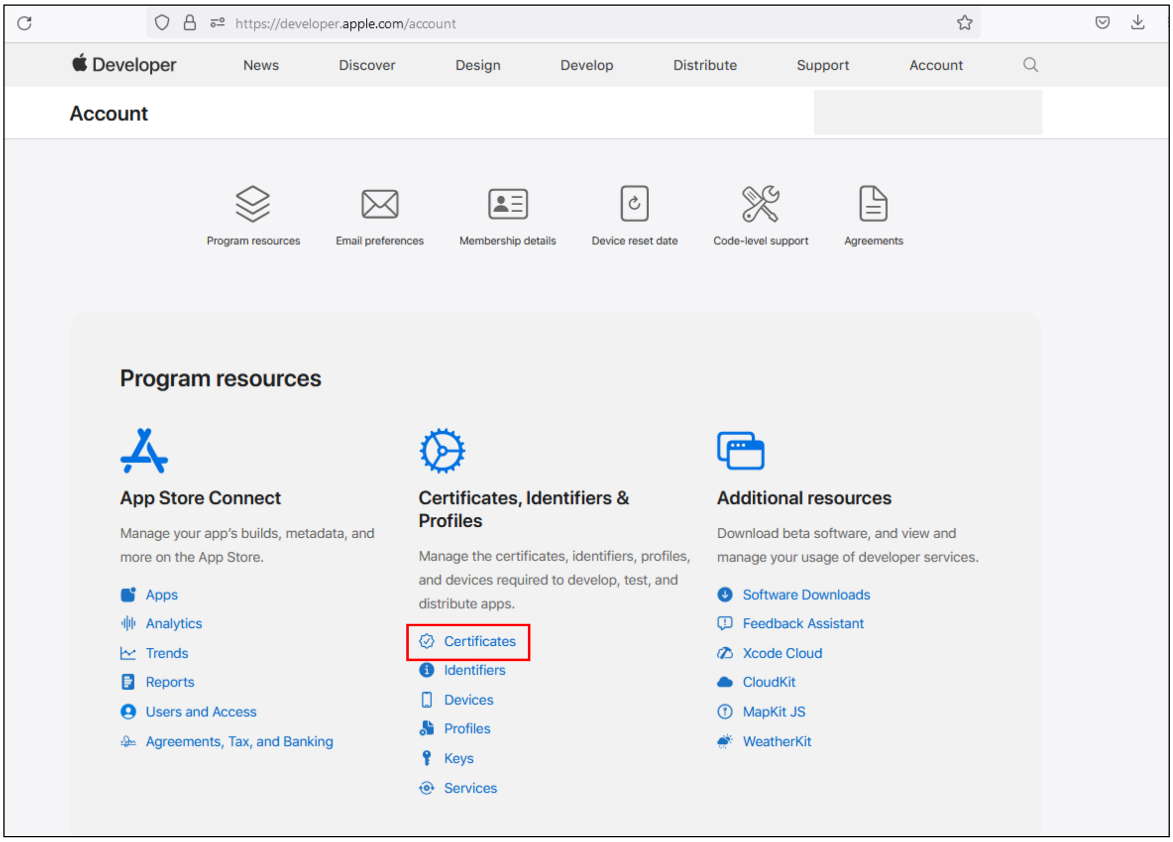
https://developer.apple.com/programs/enroll/

After you successfully enroll in the Apple developer’s program, please visit *https://developer.apple.com/account/*.

The following screens will guide how to generate a signing certificate with Apple, documentation can be found here: *https://developer.apple.com/documentation/walletpasses/building_a_pass*

You can refer to Generate Key Pair For Apple Wallet Pass for steps to generate key pair.

1. In the main page of the Apple developer’s website, select *Certificates*



Account

Program resources

App Store Connect

Manage your app’s builds, metadata, and more on the App Store.

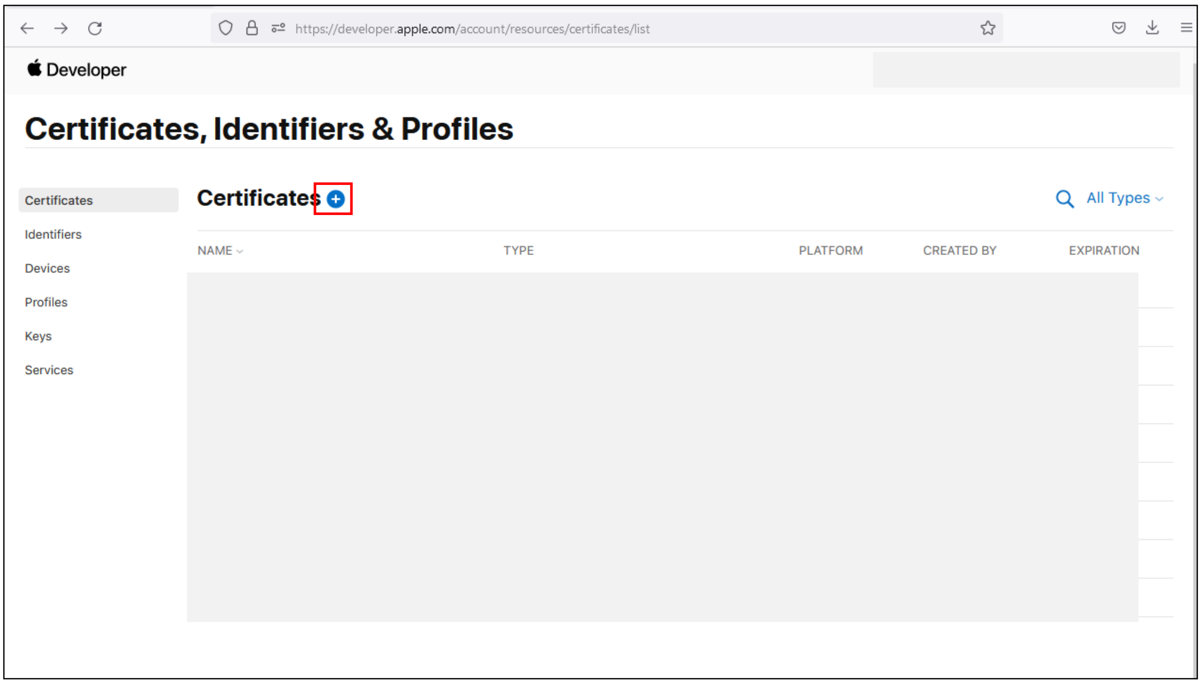
Certificates, Identifiers & Profiles

Manage the certificates, identifiers, profiles, and devices required to develop, test, and distribute apps.

Additional resources

Download beta software, and view and manage your usage of developer services.

2. Press + *button*



https://developer.apple.com/account/resources/certificates/list

Developer

Certificates, Identifiers & Profiles

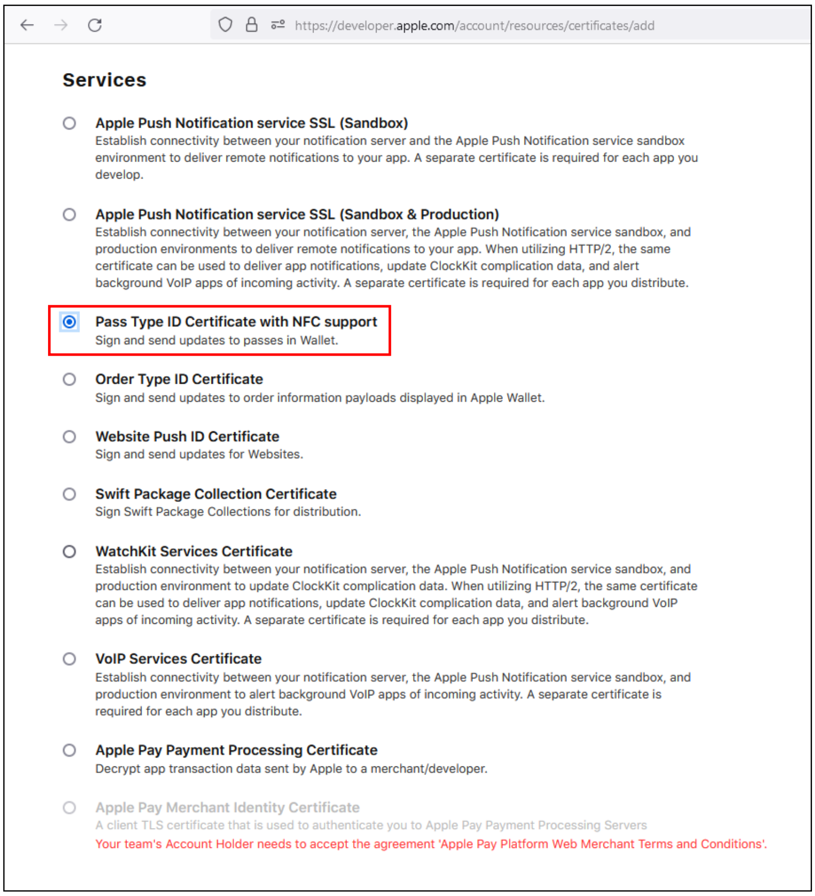
- Certificates
- Identifiers
- Devices
- Profiles
- Keys
- Services

Certificates ⊕

All Types

NAME	TYPE	PLATFORM	CREATED BY	EXPIRATION

3. Select *Pass Type ID Certificate with NFC support* and fill-in in the necessary information.



https://developer.apple.com/account/resources/certificates/add

Services

○ **Apple Push Notification service SSL (Sandbox)**
Establish connectivity between your notification server and the Apple Push Notification service sandbox environment to deliver remote notifications to your app. A separate certificate is required for each app you develop.

○ **Apple Push Notification service SSL (Sandbox & Production)**
Establish connectivity between your notification server, the Apple Push Notification service sandbox, and production environments to deliver remote notifications to your app. When utilizing HTTP/2, the same certificate can be used to deliver app notifications, update ClockKit complication data, and alert background VoIP apps of incoming activity. A separate certificate is required for each app you distribute.

● **Pass Type ID Certificate with NFC support**
Sign and send updates to passes in Wallet.

○ **Order Type ID Certificate**
Sign and send updates to order information payloads displayed in Apple Wallet.

○ **Website Push ID Certificate**
Sign and send updates for Websites.

○ **Swift Package Collection Certificate**
Sign Swift Package Collections for distribution.

○ **WatchKit Services Certificate**
Establish connectivity between your notification server, the Apple Push Notification service sandbox, and production environment to update ClockKit complication data. When utilizing HTTP/2, the same certificate can be used to deliver app notifications, update ClockKit complication data, and alert background VoIP apps of incoming activity. A separate certificate is required for each app you distribute.

○ **VoIP Services Certificate**
Establish connectivity between your notification server, the Apple Push Notification service sandbox, and production environment to alert background VoIP apps of incoming activity. A separate certificate is required for each app you distribute.

○ **Apple Pay Payment Processing Certificate**
Decrypt app transaction data sent by Apple to a merchant/developer.

○ **Apple Pay Merchant Identity Certificate**
A client TLS certificate that is used to authenticate you to Apple Pay Payment Processing Servers
Your team's Account Holder needs to accept the agreement 'Apple Pay Platform Web Merchant Terms and Conditions'.

Appendix A.1.2. Generate Key Pair For Apple Wallet Pass

You can use **OpenSSL** to generate the key pair for issuing Apple Pass.

The terminal uses the private key when reading the pass and decoding the payload.

The compressed public key is what goes into your pass.json when issuing a pass.

Inputs:

openssl ecparam -name prime256v1 -genkey -noout -out nfcKey.pem

openssl ec -in nfcKey.pem -pubout -out nfcPubkey.pem -conv_form compressed

cat nfcPubkey.pem

You need to convert the public key **(Hex)** into **Base64** format before inputting it in pass.json.

https://base64.guru/converter/encode/hex

For the private key, there is no need for you to convert it to Base64 format, the WalletMate API will do the job for you.

After you had successfully generated a key pair for the Apple pass, you can continue to read *Apple VAS documentation* and follow the guidance to create and issue Apple Pass., you can also use ACS VAS Test Tool to test your Apple pass, you can refer to Test Custom NFC-Enabled Apple Pass for more details.

Appendix A.2. Google Smart Tap

Appendix A.2.1. Communication Flow With ACS WalletMate

It is important for you to understand the Google Smart Tap workflow.

We summarized the *Smart Tap communication flow* and made three communication flow diagrams, it should be easier for you to understand who will be responsible, and how to manage ACS WalletMate with Google Smart Tap.

Example 1 (One Redemption Issuer)

Example 2 (Multiple Redemption Issuers)

Example 3 (No Aggregator)

In order to make Google Smart Tap works with our pass, you will need to generate and inject Encryption LTPK, Collector ID, and Key Version to WalletMate. Please refer to Get Your Key Pair For Smart Tap for key generation and Test Custom NFC-Enabled Google Pass for steps to inject Encrypted LTPK by using ACS VAS Test Tool into WalletMate.

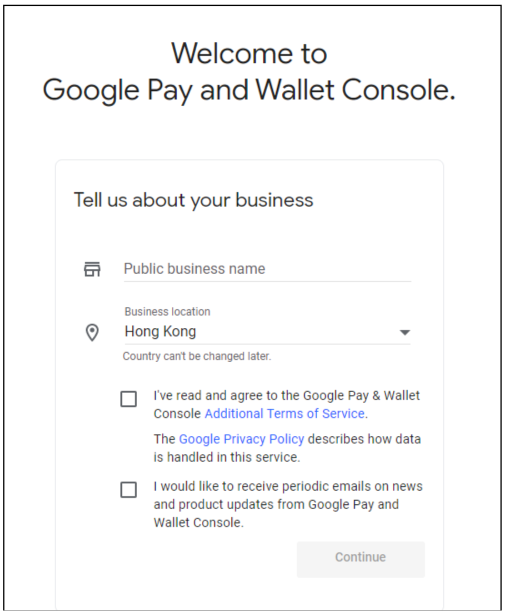
Remarks:

LTPK needed to be rotated if it was compromised or after a reasonable amount of time. To aid in key rotation, a Key Version must be defined with the LTPK.

For the Public Key, you should upload it to Google Wallet Issuer Account. There are two ways to create Google Wallet Issuer Account, either manually (Via *Google Pay and Wallet Console*) or programmatically, further information can refer to *here*.

Appendix A.2.2. Get Started With Google Pay And Wallet Console Manually

1. Sign in to Google Pay and Wallet Console at https://pay.google.com/business/console/
2. Enter your public business name, agree to the additional Terms of Service, and click *Continue*.



Welcome to
Google Pay and Wallet Console.

Tell us about your business

Public business name

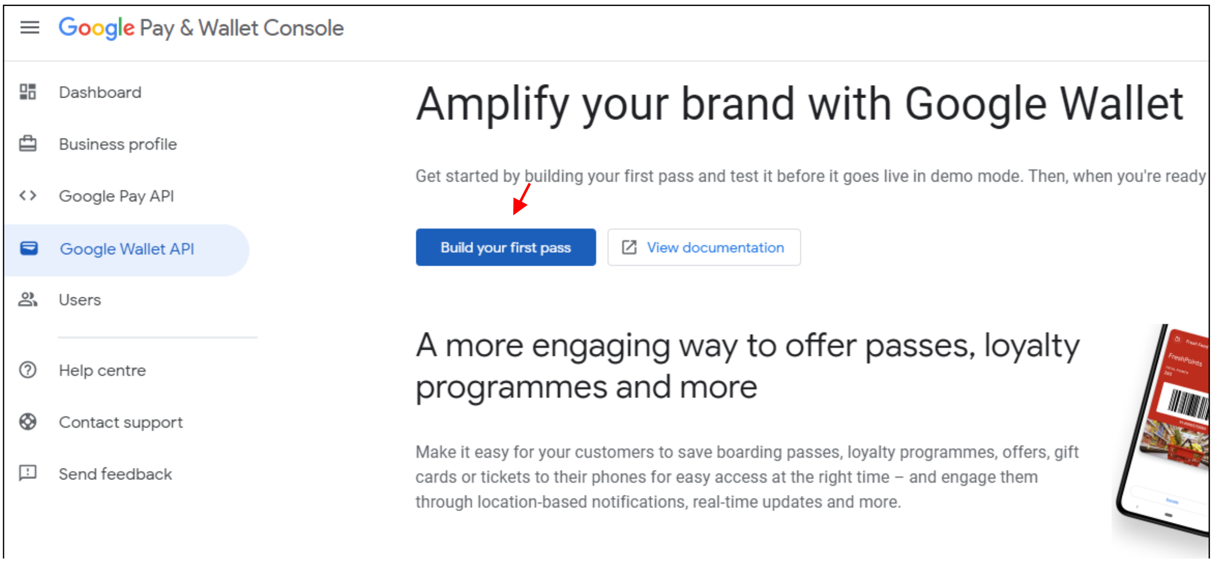
Business location
Hong Kong
Country can't be changed later.

□ I've read and agree to the Google Pay & Wallet Console Additional Terms of Service.

The Google Privacy Policy describes how data is handled in this service.

□ I would like to receive periodic emails on news and product updates from Google Pay and Wallet Console.

3. Select *Google Wallet API* on the left-hand side, then click *Build your first pass*.



Amplify your brand with Google Wallet

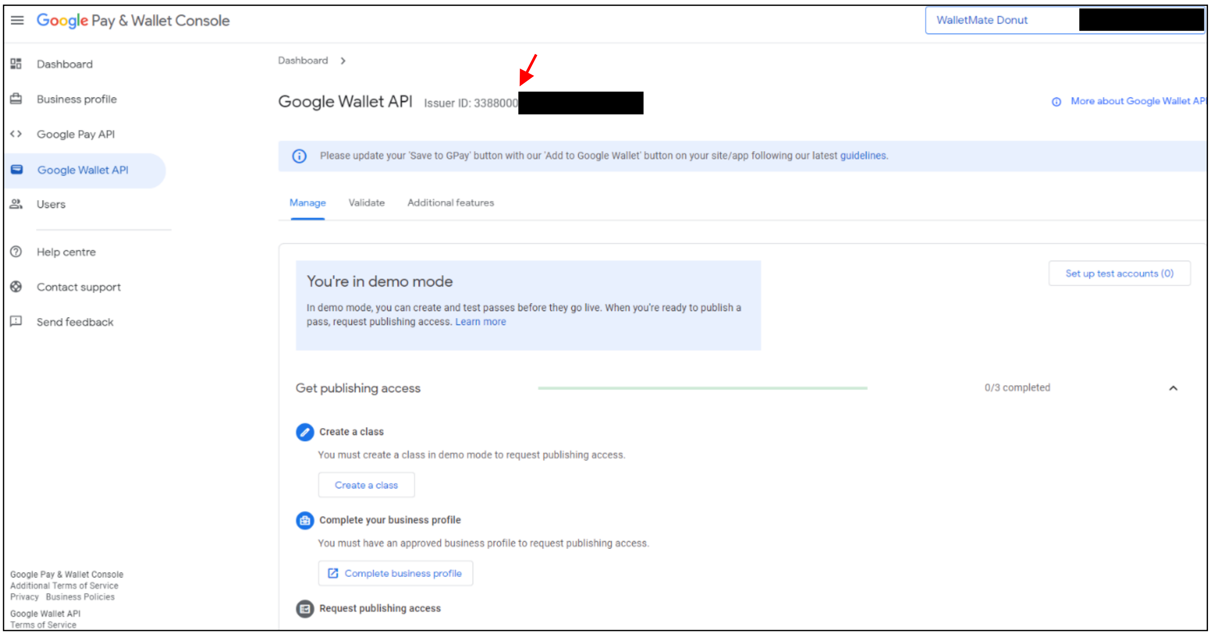
Get started by building your first pass and test it before it goes live in demo mode. Then, when you're ready

A more engaging way to offer passes, loyalty programmes and more

Make it easy for your customers to save boarding passes, loyalty programmes, offers, gift cards or tickets to their phones for easy access at the right time – and engage them through location-based notifications, real-time updates and more.

Appendix A.2.3. Obtain The Issuer ID

Once you agree to the terms and conditions, an Issuer ID will be auto-generated. You will need to Create a pass class and Complete business profit before you can request publishing access to Google Pass.



Google Wallet API Issuer ID: 3388000[redacted]

Please update your 'Save to GPay' button with our 'Add to Google Wallet' button on your site/app following our latest guidelines.

You're in demo mode

In demo mode, you can create and test passes before they go live. When you're ready to publish a pass, request publishing access. Learn more

Get publishing access 0/3 completed

Create a class

You must create a class in demo mode to request publishing access.

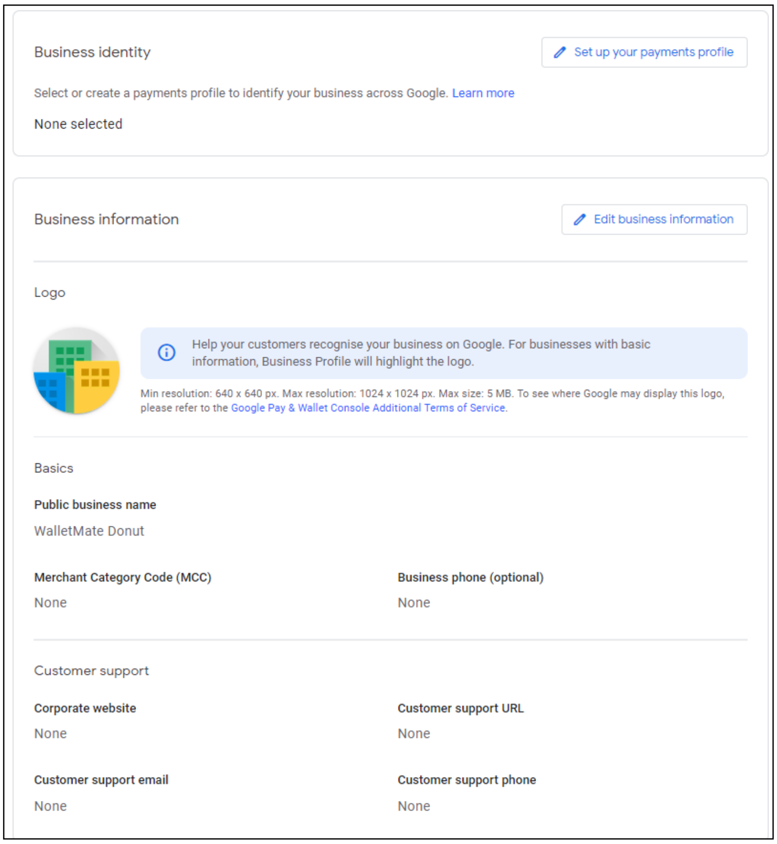
Complete your business profile

You must have an approved business profile to request publishing access.

Request publishing access

Appendix A.2.4. Complete The Business Profile

There is nothing special on **complete the business profile**, simply fill in all the required information and you are good to go.



Business identity

Select or create a payments profile to identify your business across Google. Learn more

None selected

Business information

Logo

Help your customers recognise your business on Google. For businesses with basic information, Business Profile will highlight the logo.

Min resolution: 640 x 640 px. Max resolution: 1024 x 1024 px. Max size: 5 MB. To see where Google may display this logo, please refer to the Google Pay & Wallet Console Additional Terms of Service.

Basics

Public business name
WalletMate Donut

Merchant Category Code (MCC)
None

Business phone (optional)
None

Customer support

Corporate website
None

Customer support URL
None

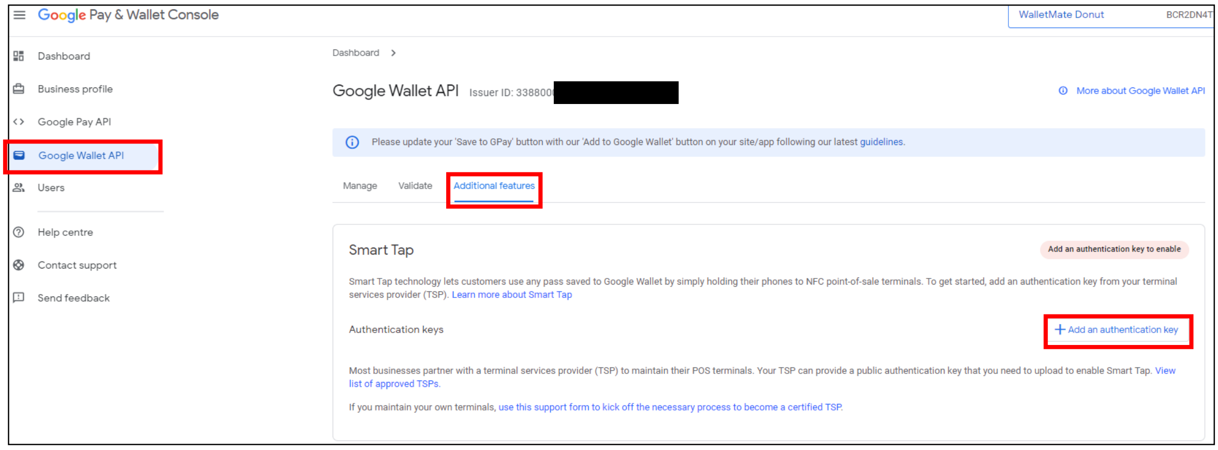
Customer support email
None

Customer support phone
None

Appendix A.2.5. Get Your Key Pair For Smart Tap

Appendix A.2.5.1. Preparation

1. Generate the ECC key pair using OpenSSL package
2. Navigates to *Google Pay & Wallet Console >> Google Wallet API >> additional features*, click *Add an authentication key*.



Google Wallet API Issuer ID: 338800[redacted]

More about Google Wallet API

Please update your 'Save to GPay' button with our 'Add to Google Wallet' button on your site/app following our latest guidelines.

Smart Tap

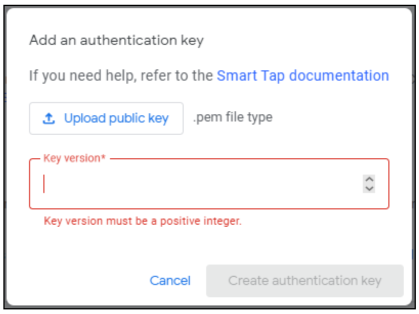
Smart Tap technology lets customers use any pass saved to Google Wallet by simply holding their phones to NFC point-of-sale terminals. To get started, add an authentication key from your terminal services provider (TSP). Learn more about Smart Tap

Authentication keys

Most businesses partner with a terminal services provider (TSP) to maintain their POS terminals. Your TSP can provide a public authentication key that you need to upload to enable Smart Tap. View list of approved TSPs.

If you maintain your own terminals, use this support form to kick off the necessary process to become a certified TSP.

3. Upload the public key .pem file, and type in your desired key version, it can be any positive integer value.



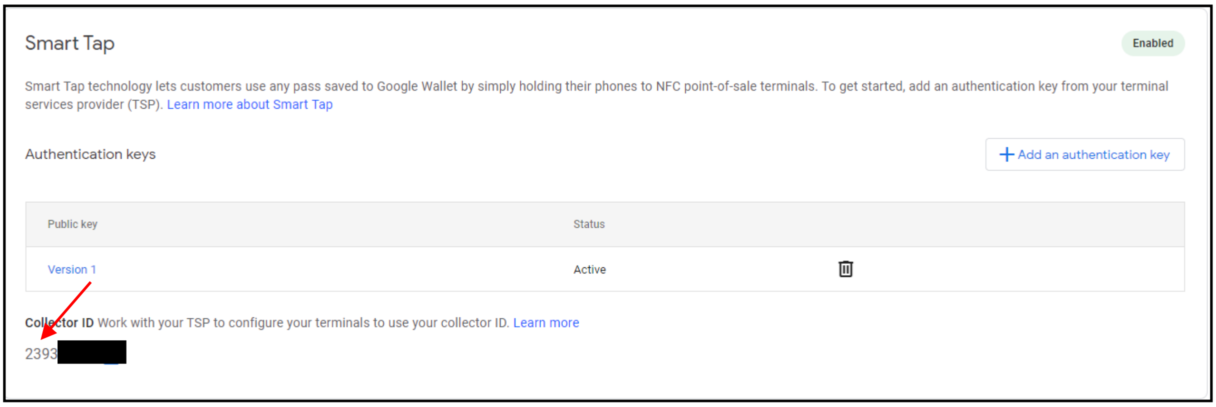
Add an authentication key

If you need help, refer to the Smart Tap documentation

Key version*

Key version must be a positive integer.

4. A Collector ID will be auto-generated after you upload the public key to Google Pay & Wallet Console.



Smart Tap

Smart Tap technology lets customers use any pass saved to Google Wallet by simply holding their phones to NFC point-of-sale terminals. To get started, add an authentication key from your terminal services provider (TSP). Learn more about Smart Tap

Authentication keys

Public key	Status
Version 1	Active

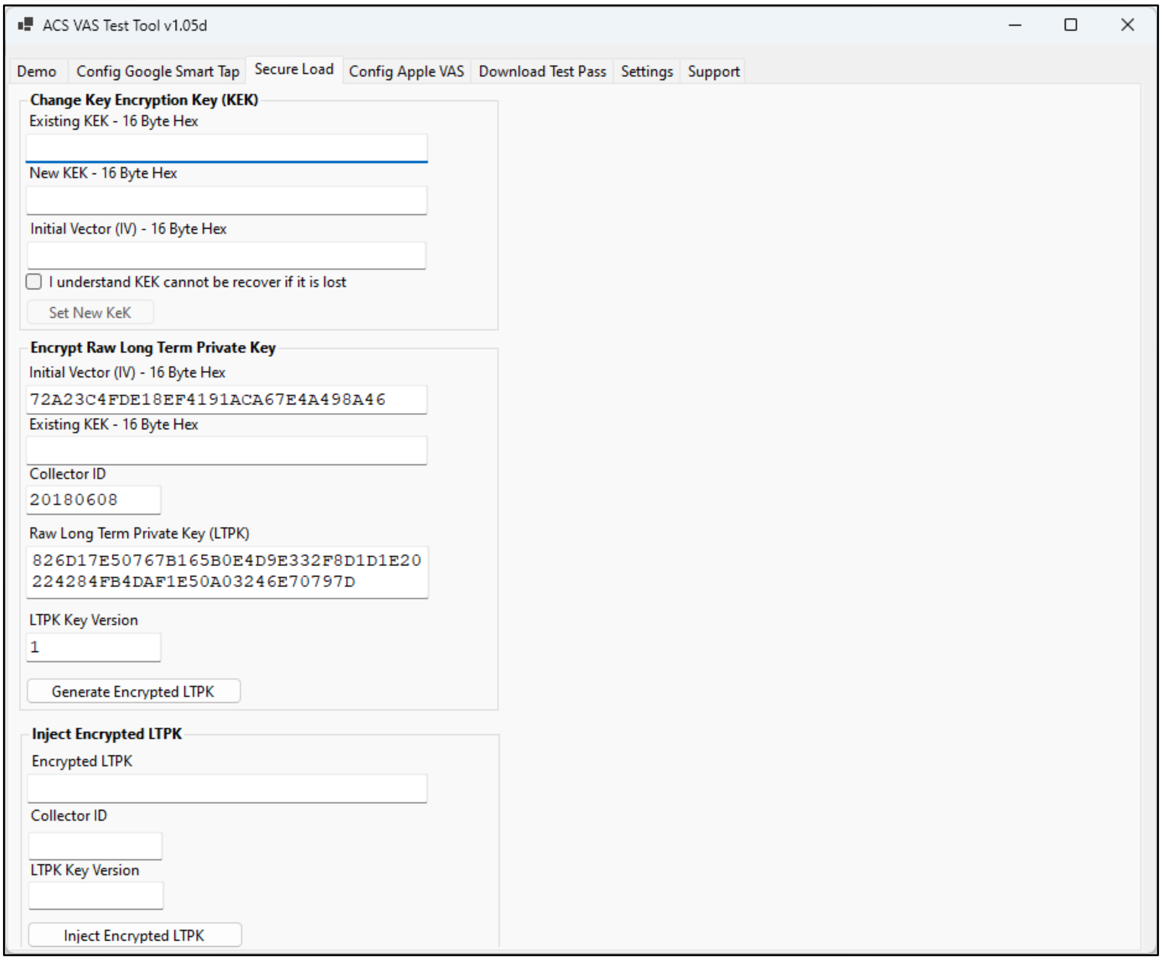
Collector ID Work with your TSP to configure your terminals to use your collector ID. Learn more

2393[redacted]

Appendix A.2.5.2. Generate LTPK (Windows)

1. Launch *ACS VAS Test Tool* and go to the *Secure Load* Tab
2. Press *F8* and Input the value inside the following box.

   Default ACS KEK: AABB79CC28ACE16BF2B4F36AE837B8A2



Initial Vector (IV) - 16 Byte Hex: 72A23C4FDE18EF4191ACA67E4A498A46

Collector ID: 20180608

Raw Long Term Private Key (LTPK): 826D17E50767B165B0E4D9E332F8D1D1E20224284FB4DAF1E50A03246E70797D

LTPK Key Version: 1

3. Click *Generate Encrypted LTPK* and it will be generated.
4. After you generated the Public Key and Encrypted LTPK, please delete the demo public key at your Google Pay and Wallet Console, and upload the Public Key and its corresponding key version, for the Encrypted LTPK, please refer to the section of Inject The Encrypted LTPK to inject into WalletMate.

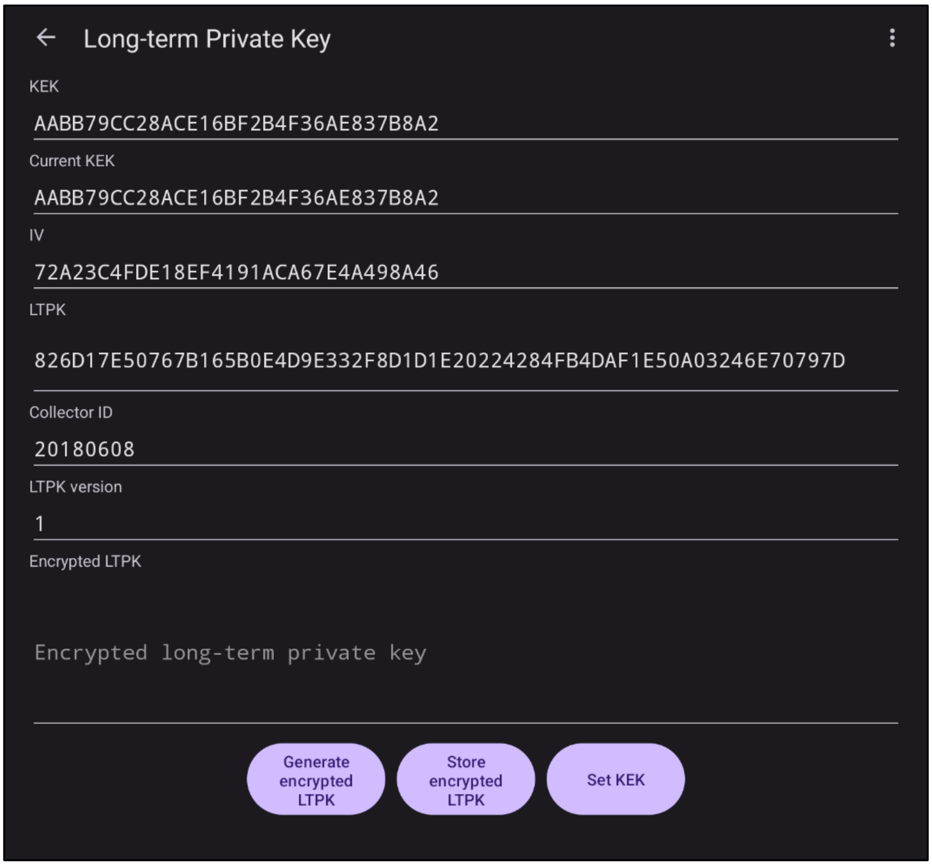
Appendix A.2.5.3. Generate LTPK (Linux)

Currently automatically generate and set to WalletMate when click the *Set* button at *Secure Load* Tap.

Appendix A.2.5.4. Generate LTPK (Android)

1. Launch *TestGoogleVas* and go to the *Set LTPK*
2. Input the value inside the following box.

   Default ACS KEK: AABB79CC28ACE16BF2B4F36AE837B8A2



Long-term Private Key

KEK: AABB79CC28ACE16BF2B4F36AE837B8A2

Current KEK: AABB79CC28ACE16BF2B4F36AE837B8A2

IV: 72A23C4FDE18EF4191ACA67E4A498A46

LTPK: 826D17E50767B165B0E4D9E332F8D1D1E20224284FB4DAF1E50A03246E70797D

Collector ID: 20180608

LTPK version: 1

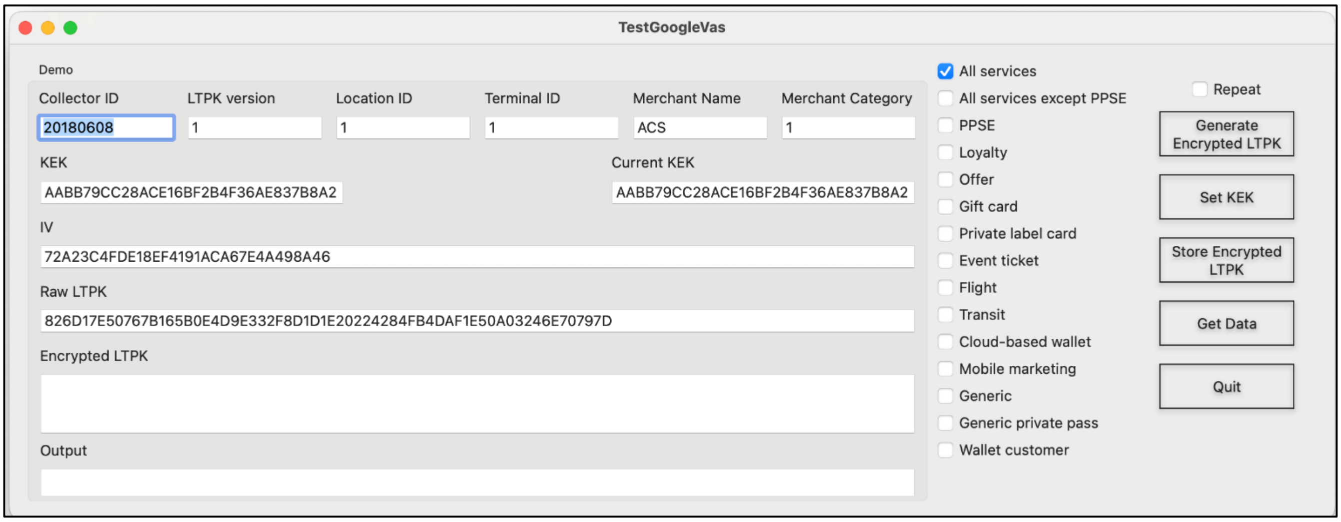
Encrypted LTPK: Encrypted long-term private key

3. Click *Generate encrypted LTPK* and it will be generated.
4. After you generated the Public Key and Encrypted LTPK, please delete the demo public key at your Google Pay and Wallet Console, and upload the Public Key and its corresponding key version, for the Encrypted LTPK, please refer to the section of Inject The Encrypted LTPK to inject into WalletMate.

Appendix A.2.5.5. Generate LTPK (Mac)

1. Launch *TestGoogleVas*
2. Input the value inside the following box.

   Default ACS KEK: AABB79CC28ACE16BF2B4F36AE837B8A2



Demo

Collector ID	LTPK version	Location ID	Terminal ID	Merchant Name	Merchant Category
20180608	1	1	1	ACS	1

KEK: AABB79CC28ACE16BF2B4F36AE837B8A2

Current KEK: AABB79CC28ACE16BF2B4F36AE837B8A2

IV: 72A23C4FDE18EF4191ACA67E4A498A46

Raw LTPK: 826D17E50767B165B0E4D9E332F8D1D1E20224284FB4DAF1E50A03246E70797D

Encrypted LTPK

Output

☑ All services
☐ All services except PPSE
☐ PPSE
☐ Loyalty
☐ Offer
☐ Gift card
☐ Private label card
☐ Event ticket
☐ Flight
☐ Transit
☐ Cloud-based wallet
☐ Mobile marketing
☐ Generic
☐ Generic private pass
☐ Wallet customer

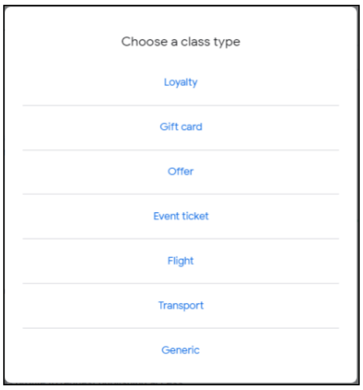
☐ Repeat

3. Click *Generate Encrypted LTPK* and it will be generated.
4. After you generated the Public Key and Encrypted LTPK, please delete the demo public key at your Google Pay and Wallet Console, and upload the Public Key and its corresponding key version, for the Encrypted LTPK, please refer to the section of Inject The Encrypted LTPK to inject into WalletMate.

Appendix A.2.6. Create A Pass Class

There are 7 types of pass classes. The below documentation describes the relationship between classes and objects, which is fundamental to how the Google Wallet API works. This page also covers how objects link to users

https://developers.google.com/wallet/generic/resources/how-classes-objects-work

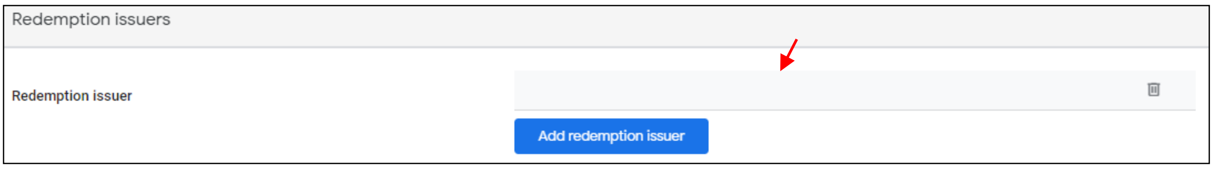


As described *here*, because merchants may use multiple businesses or contractors to develop their passes, Google uses three identifiers to facilitate sending information between Google Wallet and terminals.

There will have three case scenarios:

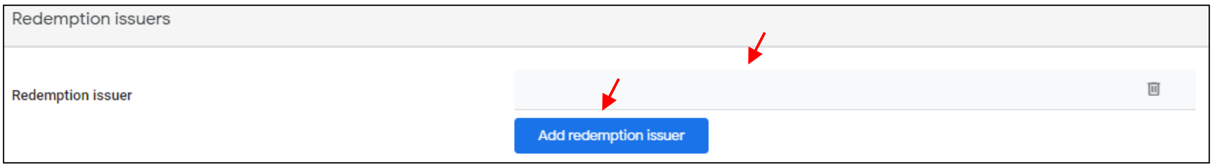
Scenario 1: *One Redemption Issuer* (One pass developer manages passes for one merchant)

If you are a pass developer (also called an Aggregator) when you create a pass class for a merchant (also called a Redemption Issuer), please be reminded to fill in the **Issuer ID** of the merchant in the **Redemption Issuer field**.



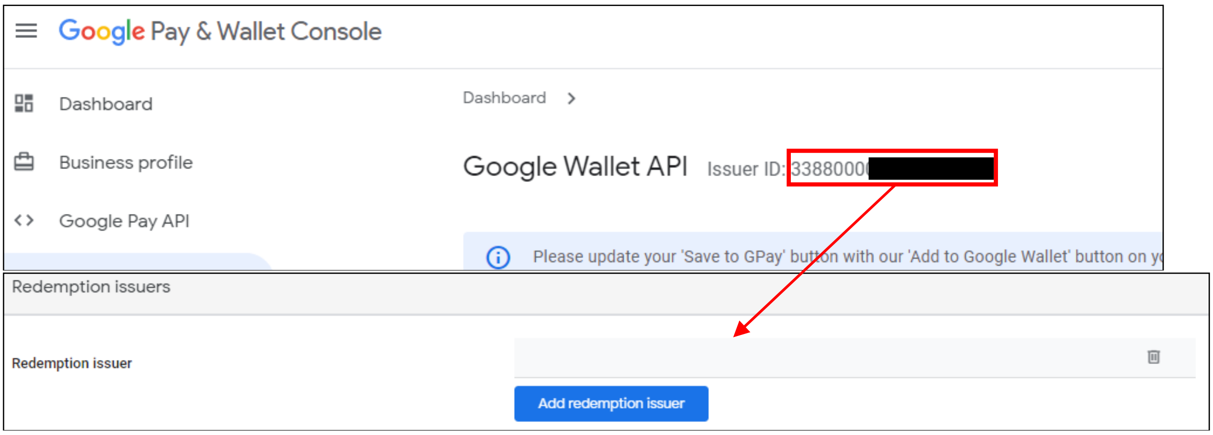
Scenario 2: *Multiple Redemption Issuers* (One pass developer manages passes for multiple merchants)

If you are a pass developer (also called an Aggregator) when you create a pass class for multiple merchants (also called a Redemption Issuer), and the pass class is redeemable by both two merchants, please be reminded to fill in both Issuer IDs of two merchants in the Redemption Issuer field.



Scenario 3: *No Aggregator* (Merchants will be responsible for managing their passes.)

In this case, the pass developer must include their Issuer ID in the Redemption Issuer field.



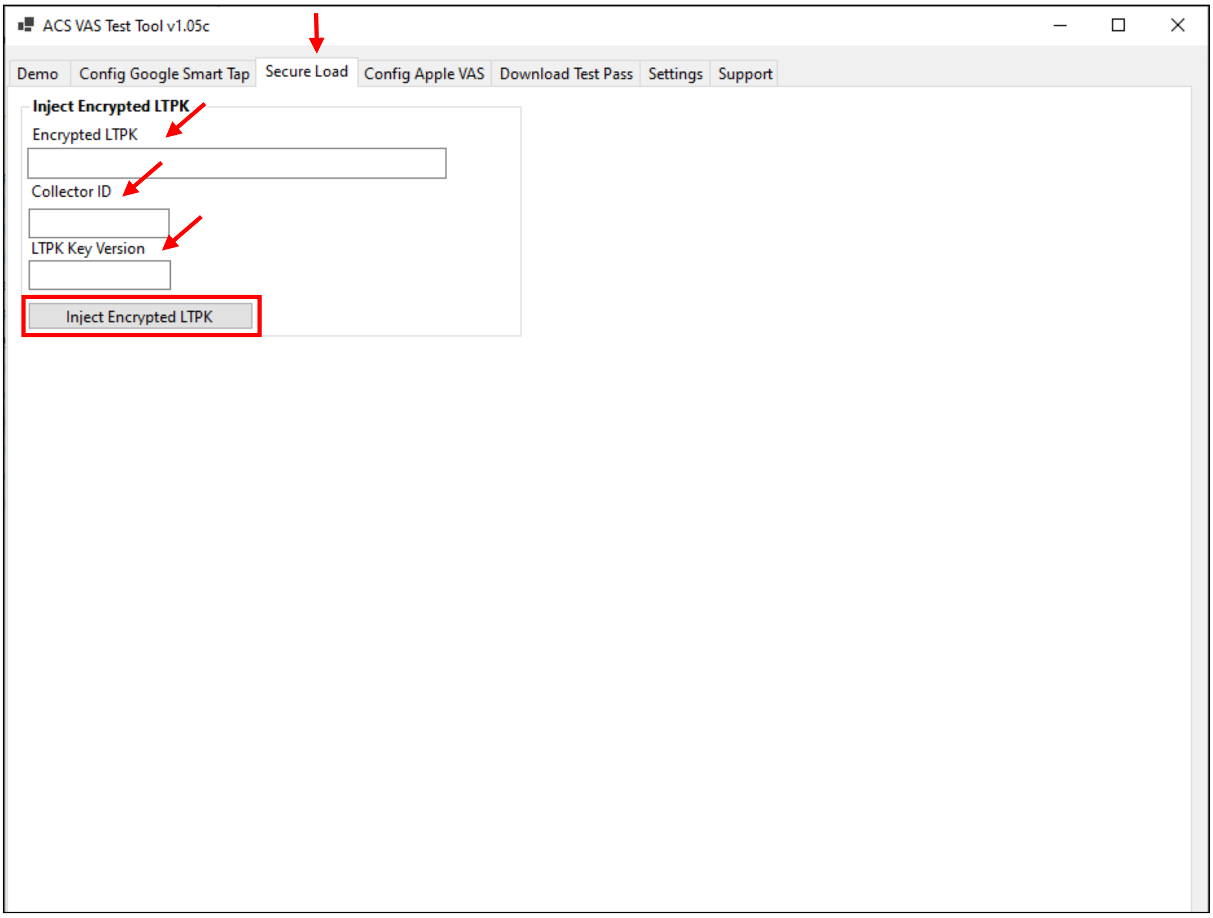
Please update your 'Save to GPay' button with our 'Add to Google Wallet' button on y

Appendix A.2.7. Inject The Encrypted LTPK

You can inject the encrypted LTPK by using the Google Smart Tap API, or ACS VAS Test Tool.

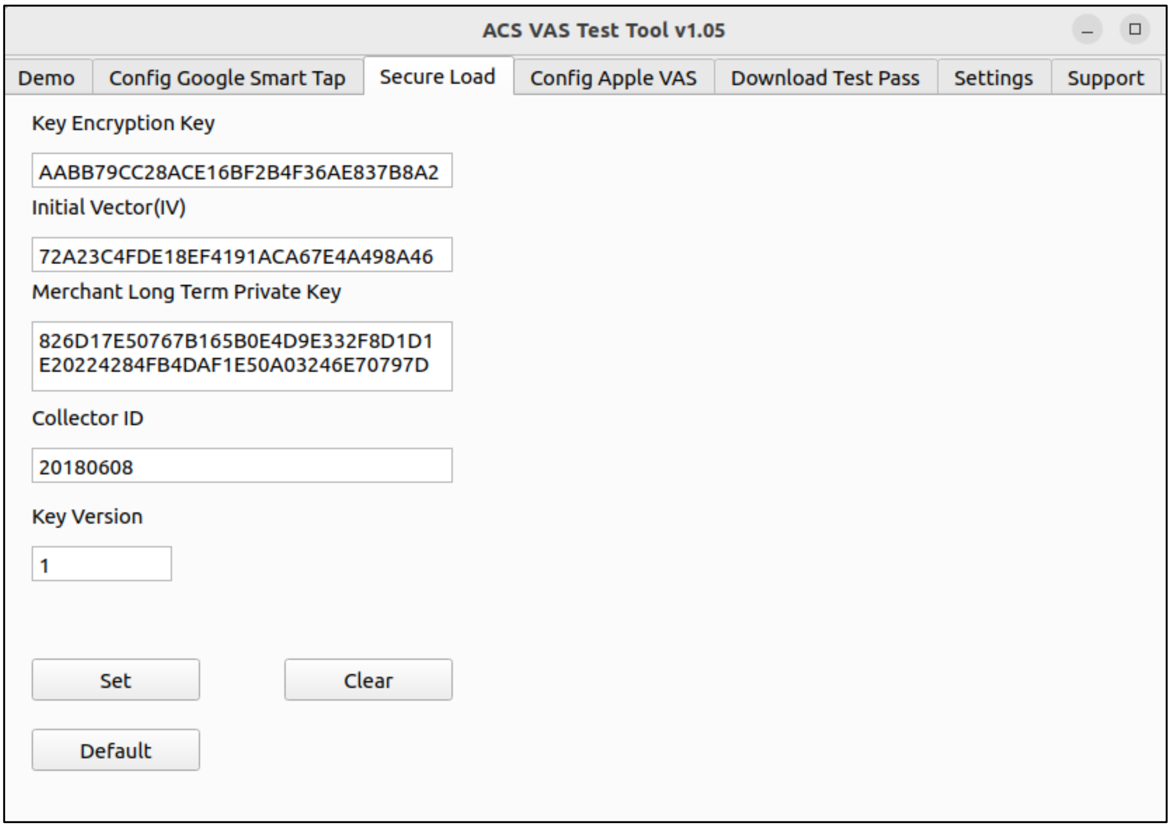
Appendix A.2.7.1. Windows

1. Connect WalletMate to PC
2. Launch ACS VAS Test Tool
3. Go to *Secure Load* Tab, fill in the Encrypted LTPK, and press *Inject Encrypted LTPK*.



Appendix A.2.7.2. Linux

1. Connect WalletMate to PC
2. Launch ACS VAS Test Tool
3. Go to *Secure Load* Tab, fill in the Key Encryption Key and Merchant Long Term Private Key, and press *Set*.



Key Encryption Key

AABB79CC28ACE16BF2B4F36AE837B8A2

Initial Vector(IV)

72A23C4FDE18EF4191ACA67E4A498A46

Merchant Long Term Private Key

826D17E50767B165B0E4D9E332F8D1D1
E20224284FB4DAF1E50A03246E70797D

Collector ID

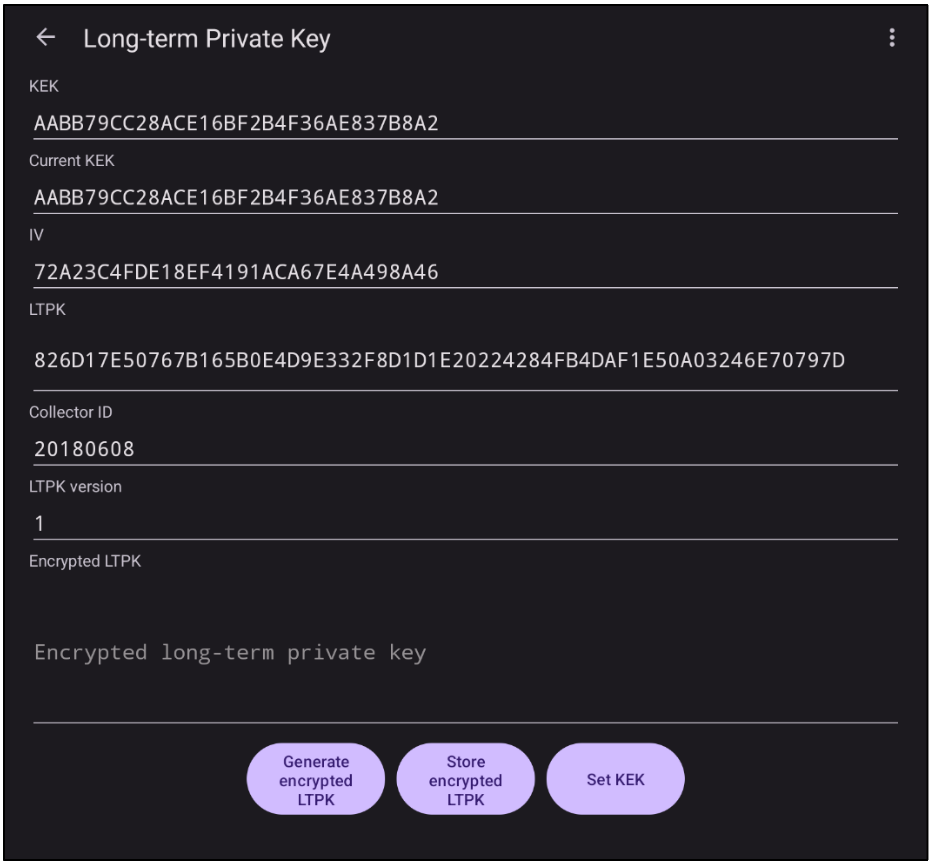
20180608

Key Version

1

Appendix A.2.7.3. Android

1. Connect WalletMate to Android Mobile or Tablet
2. Launch *TestGoogleVas*
3. Go to *Set LTPK*, fill in the Encrypted LTPK, and press *Store encrypted LTPK*.



Long-term Private Key

KEK: AABB79CC28ACE16BF2B4F36AE837B8A2

Current KEK: AABB79CC28ACE16BF2B4F36AE837B8A2

IV: 72A23C4FDE18EF4191ACA67E4A498A46

LTPK: 826D17E50767B165B0E4D9E332F8D1D1E20224284FB4DAF1E50A03246E70797D

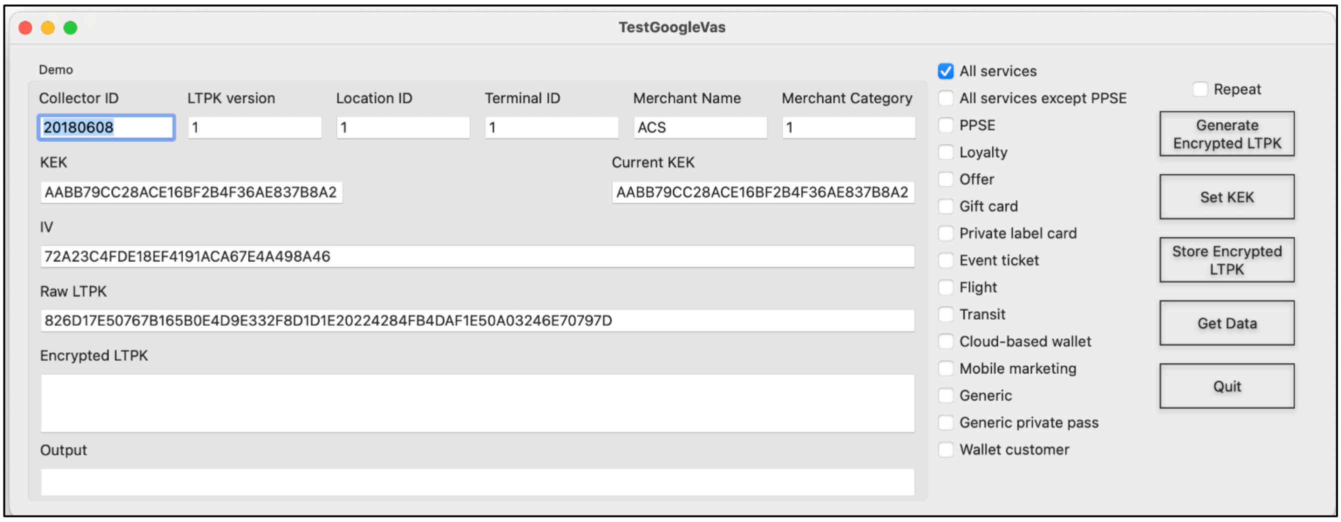
Collector ID: 20180608

LTPK version: 1

Encrypted LTPK: Encrypted long-term private key

Appendix A.2.7.4. Mac

1. Connect WalletMate to Android Mobile or Tablet
2. Launch *TestGoogleVas*
3. Fill in the Encrypted LTPK, and press *Store Encrypted LTPK*.



TestGoogleVas

Demo

Collector ID	LTPK version	Location ID	Terminal ID	Merchant Name	Merchant Category
20180608	1	1	1	ACS	1

KEK: AABB79CC28ACE16BF2B4F36AE837B8A2

Current KEK: AABB79CC28ACE16BF2B4F36AE837B8A2

IV: 72A23C4FDE18EF4191ACA67E4A498A46

Raw LTPK: 826D17E50767B165B0E4D9E332F8D1D1E20224284FB4DAF1E50A03246E70797D

Encrypted LTPK

Output

☑ All services
☐ All services except PPSE
☐ PPSE
☐ Loyalty
☐ Offer
☐ Gift card
☐ Private label card
☐ Event ticket
☐ Flight
☐ Transit
☐ Cloud-based wallet
☐ Mobile marketing
☐ Generic
☐ Generic private pass
☐ Wallet customer

☐ Repeat