ID TECH Reading Demo Passes Guide

1. Overview

This guide provides instructions for reading ID TECH demo passes on supported ID TECH readers.

1.1. Before You Begin

1. Make sure your PiP reader has firmware version 1.00.017 or later.
2. Make sure you've downloaded the most recent version of the Universal SDK Demo app.
3. Add the demo pass for your desired mobile platform:
   - Apple Wallet
   - Google Pay Smart Tap 2.0

2. Google Pay Smart Tap 2.0

The section below describes steps for using the demo pass with Google Pay Smart Tap 2.0. Refer to the *NEO Interface Developer's Guide* for details on the commands listed in this document.

Note the following information for the demo pass:

- Collector ID = 87133300
- Key Version = A

2.1. Reader Configuration Sequence

Send the following commands to the target reader:

1. Send the **Set Collector ID (04-03)** command with the following data:
   ```
   FFE4018EDFEE3B0405318C74DFEE3C00DFEE3D00DFEF2500DFED0100DFED02050
   000000001DFED030101DFED040101DFED050101DFED060100DFED070100DFED270
   10DDFED3F0100DFED490100DFEF770100
   ```

2. Send the **Load LTPK (C7-65)** command with the following data:
   ```
   0000000AF5368708933920553B7B9FFB16AEED9C77D5BFD9662AF149A6B9F965B
   73F0CCA
   ```

3. Apple VAS

The section below describes steps for using the demo pass with Apple VAS. Refer to the *NEO Interface Developer's Guide* for details on the commands listed in this document.

Note the following information for the demo pass:
Passtype ID = pass.com.pronto.id-tech.demo

3.1. Reader Configuration Sequence

Send the following commands to the demo reader:

1. Send the **Set Merchant Record (04-11)** command with the following data:
```
0101AD9887C78E412F835E89D0A4F71E423320C7BB53B6FAACD8D1D1EED9E1E38D
390000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000
```

2. Send the **Load Key (C7-66)** command with the following data:
```
F536870893392055 3B7B9FFB16AEED9C77D5BFD9662AF149A6B9F965B73F0CCA
```

3. Send the **Set Configuration (04-00)** command to set the encryption switch with the following data:
```
DFED3F0102
```

4. Examples

After configuring your reader with steps above, send the **Quick Set Work Mode (01-12)** command to set the reader's work mode.

Work Mode

Mode	Poll Mode	Data Output Mode	USB Interface
00h	Auto poll	Normal mode	USBHID
01h	Auto poll	Normal mode	USBKB
02h*	**Auto poll**	**Simplified output mode**	**USBKB**
03h	Auto poll	Tags only	USBHID
04h	Auto poll	Tags only	USBKB
05h*	**Poll on demand**	**Normal mode**	**USBHID**
06h	Poll on demand	Normal mode	USBKB

* **Note: 02h** and **05h** are the most frequently-used work mode settings.

4.1. Google Pay Smart Tap 2.0 Examples

Example 1: Poll on demand mode in USBHID. Send the **Quick Set Work Mode (01-12)** command with data **05h** and include the additional tag **FFEE080ADFEF1A010ADFED280100** in the ACT command.

```
10:37:22.315 OUT:
5669564F7465636832000240000F0FFFEE080ADFEF1A010ADFED28010087C5
10:37:25.842 IN:
5669564F746563683200025700BF41FFEE0881A8DFEF76818C940337617376940106690
40105318C745402286C799403096F69640496CD987A07500516590113015 46E00357C74
555237646A5361736E475F4254775754034961737694010669040271797971540339637
57394031163696404000000000000000000000000000000000190103035463706C00656E
5403116375740403500FC422B91D5768EE5F548319F6CCDFEC1613357C74555237646A5
361736E475F425477570D9F390107FFEE0104DF300100DFEE260141CE7E
```

The string will be in Tag DFEC16: **357C74555237646A5361736E475f42547757** (5|tUR7djSasnG_BTwW in ASCII).

Example 2: Auto poll mode in USBKB. Send the **Quick Set Work Mode (01-12)** command with data **02h** and the **Set Configuration (04-00)** command with data **FFEE080ADFEF1A010ADFED280100**.

```
10:37:57.656 OUT:
5669564F746563683200040000EFFEE080ADFEF1A010ADFED280100244B
10:37:57.718 IN: 5669564F74656368320004000000AE16
10:38:33.607 OUT: 5669564F7465636832000112000102CF84
10:38:33.659 IN: 5669564F74656368320001000000 1253
```

When presenting the Google Pass, the data should output as **5|tUR7djSasnG_BTwW** on notepad.

4.1. Apple VAS Examples

Example1: Poll on demand mode in USBHID. Send the **Quick Set Work Mode (01-12)** command with data **05h** and include the additional tag **FFEE06189F220201009F2604000000029F2B050100000000DF010101** in the ACT command.

```
11:08:53.404 OUT:
5669564F7465636832000240001D0FFFEE06189F220201009F2604000000029F2B05010
0000000DF01010177BA
11:09:00.598 IN:
5669564F7465636832000257006141FFEE06469A032203099F21031910399F2520AD988
7C78E412F835E89D0A4F71E423320C7BB53B6FAACD8D1D1EED9E1E38D399F2A009F2712
357C74555237646A5361736E475F425477579F390107FFEE0104DF300100DFEE260141D
FED60010259CA
```

The string will be in Tag 9F27= **357C74555237646A5361736E475F42547757** (**5|tUR7djSasnG_BTwW** in ASCII).

Example 2: Auto poll mode in USBKB. Send the **Quick Set Work Mode (01-12)** command with data **02h** and the **Set Configuration (04-00)** command with data **FFEE06189F220201009F2604000000029F2B050100000000DF010101**.

When presenting the Apple Wallet pass, the data should output **5|tUR7djSasnG_BTwW** on notepad.

5. APPENDIX A: ECC Key Pair

Merchants or other administrators who wish to use SmartTap must create and amanage the Elliptical Curve Cryptography (ECC) key pair used to for securing communication between the reader and the wallet.

Public Key: administators must communicate the public key to Google. It is public and can be visible to anyone.

Private Key: the private key must be kept private and injected into the ViVOpay device, where it will be stored securely.

5.1. How to Create an ECC Key Pair Using Open-SSL

Users have several options for generating the ECC key pair (or the ECDSA digital signature key pair). The example below uses the freely available OpenSSL package to generate a prime256v1 Elliptical Curve Cipher key pair (and to sign messages).

To generate EC private key:

```
openssl> ecparam -out PRIVATE.key.pem -name prime256v1 -genkey
```

To generate EC public key from private key:

```
openssl> ec -in PRIVATE.key.pem -pubout -out PUBLIC.key.pem -conv_form
compressed
```

Sign message:

```
openssl> dgst -sha256 -sign LONG_TERM_PRIVATE.pem message.txt >
signature.bin
```

Verify message:

```
openssl> dgst -sha256 -verify LONG_TERM_PUBLIC.pem -signature
signature.bin message.txt
```

Generate ECDH shared secret:

```
openssl> pkeyutl -derive -inkey TERMINAL_EPHEMERAL_PRIVATE.pem -
peerkey HANDSET_EPHEMERAL_PUBLIC.pem -out secret.bin
```

5.2. Example LTPK for an ID TECH Pass

```
-----BEGIN EC PRIVATE KEY-----
MHcCAQEEIPU2hwiTOSBVO3uf+xau7Zx31b/ZZirxSaa5+WW3PwzKoAoGCCqGSM49
AwEHoUQDQgAEOAuPfwpDM6fk8iqWsc6ow+s4eq/YNmMtYtzApmGczCi0KMW/hjjX
DpMoxrRhOR6y796o27/+k8F9FOLmlNyOTA==
-----END EC PRIVATE KEY-----
```