VTAP

Application Note -
ECC key pairs for VTAP readers

Firmware from v1.1.12.1

VTAP50 and VTAP100

Revised December 2022 v1.1

DOT ORIGIN

If you need help to set up or use your VTAP reader, beyond what is contained in this Application Note, then please contact our support team.

Email: **vtap-support@dotorigin.com**

Download the latest documentation and firmware from **https://vtapnfc.com**

Telephone UK and Europe: +44 (0) 1428 685861

Telephone North America and Latin America: +1 (562) 262-9642

If you have any feedback on setting up or using your VTAP reader or this documentation, then please contact our support team. The product is constantly being reviewed and improved and we value feedback about your experience.

Dot Origin Ltd
Unit 7, Coopers Place Business Park, Combe Lane, Wormley
Godalming GU8 5SZ United Kingdom
+44 (0) 1428 685861

Contents

1 ECC key pairs for VTAP readers

When you set up a VTAP mobile pass NFC reader to work with your passes, you have to load a private key into your VTAP reader to unencrypt pass data. This needs to match the public key you provided for generating the passes.

This Application Note is concerned with the key pair(s) to secure your mobile passes. This is not the same key pair used in your Apple developer/NFC entitlement certificate. The Apple certificate authenticates you in their system. It is private to your organisation, and not to be shared.

So you will need to generate one or more ECC public-private key pairs to manage your passes. It is the responsibility of the pass owner to generate the key pair and keep them secure.

What is an ECC key pair?

An ECC key pair comprises two data files. One is the **public key** which you may need to share with third parties. The other is the **private key** which remains in your possession (on your hardware) at all times.

If you have a key pair, you can send someone the public key, and ask them to encrypt data they send to you with that key.

Data that is encrypted using the public key of a pair, can be decrypted by anyone with the matching private key.

A public key can be saved in a compressed format (as required by Apple) or an uncompressed format.

It is computationally simple to derive a public key from a private one, and to decrypt data if you have the private key, but it is very difficult (if not impossible) to find the private key to match a known public one.

1.1 How many ECC key pairs do I need?

To generate mobile NFC passes, you have to supply a public key to use in its encoding. You will be asked to a supply public key when working through each of the Google and Apple pass issuance processes or working with a pass provider company. The matching private key(s) need to be saved on your VTAP reader, in a `.pem` file, to decrypt the pass data that it reads.

It is entirely your choice, whether to use the same public-private key pair to set up Google SmartTap and Apple VAS passes, or different ones. They are generated in a standard way and are not signed by a certificate authority. The VTAP reader can accommodate up to six keys.

More key pairs increases your security but also increases the management complexity of your application. Consider what action you would take if one of the keys was compromised. Would you need to reissue passes and update keys on readers in the field?

1.2 Key type specification

The key type specification for both Apple and Google is as follows:

EC key length = 256 bit

Curve = prime256v1 / P-256

1.3 How to generate an ECC key pair

We recommend generating keys using OpenSSL, but you can generate your keys in other ways, as long as you save them in text files following the `.pem` format. If you have not followed the `.pem` format exactly you will find that the VTAP reader will not consume the key.

Option 1: Use OpenSSL to generate an ECC key pair

OpenSSL is a widely used open source command line tool for generating keys and certificates.

This command will generate a private key called `private1.pem`, named ready for use in the first key slot on the VTAP reader.

Example: OpenSSL command to generate a private key

```
openssl ecparam -name prime256v1 -genkey -out private1.pem
```

The text of this `private1.pem` file will look something like this:

```
-----BEGIN EC PARAMETERS-----
BggqhkjOPQMBBw==
-----END EC PARAMETERS-----
-----BEGIN EC PRIVATE KEY-----
MHcCAQEEIIEtIyvdGuRj+gRrTPn7+wpQ7XAhWfLAfmBzhtzjdrnQoAoGCCqGSM49
AwEHoUQDQgAEYzDKBwanQZs1TtuTsmrkpYjow8idfqMd0U/lwfpBdtqIjcCRowD1
lznasiT971AkZqvOZBfZTRnnjNBMu1uXzg==
-----END EC PRIVATE KEY-----
```

The following command will take the private key `private1.pem` and generate a matching `apple_public1.pem` or `google_public1.pem` public key from it.

Example: OpenSSL command to generate a public key

```
openssl ec -in private1.pem -pubout -conv_form compressed -out apple_
public1.pem
```

The text of this `apple_public1.pem` file will look something like this:

```
-----BEGIN PUBLIC KEY-----
MDkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDIgACWpF1zC3h+dCh+eWyqV8unVddh2LQ
aUoV8LQrgb3BKkM=
-----END PUBLIC KEY-----
```

Apple requires this compressed format key.

Google prefer the uncompressed format, which can be generated using:

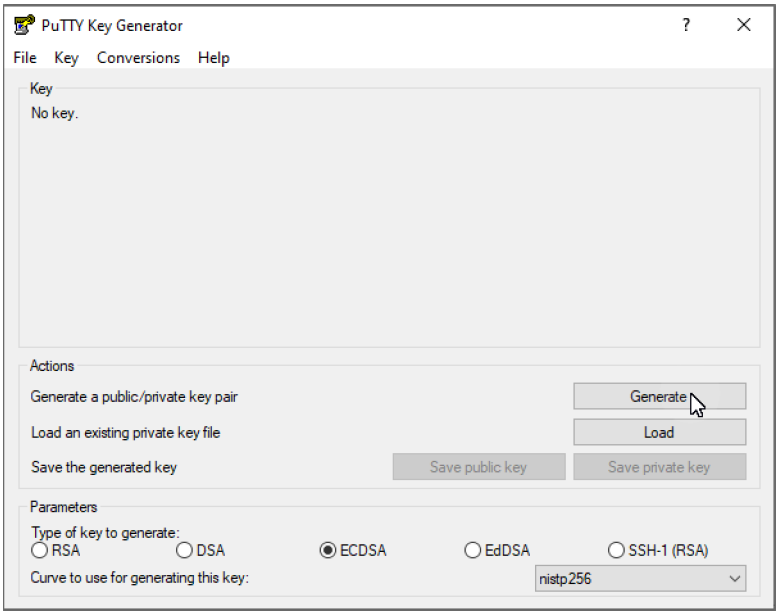
```
openssl ec -in private1.pem -pubout -out google_public1.pem
```

The text of this `google_public1.pem` file will look something like this:

```
-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEWpF1zC3h+dCh+eWyqV8unVddh2LQ
aUoV8LQrgb3BKkMA/aUIQ4EnhI+l9LeBEmO5Fc0xpQGjGuUL9G4ZQbAMNA==
-----END PUBLIC KEY-----
```

Option 2: Use PuTTYgen to generate an ECC key pair

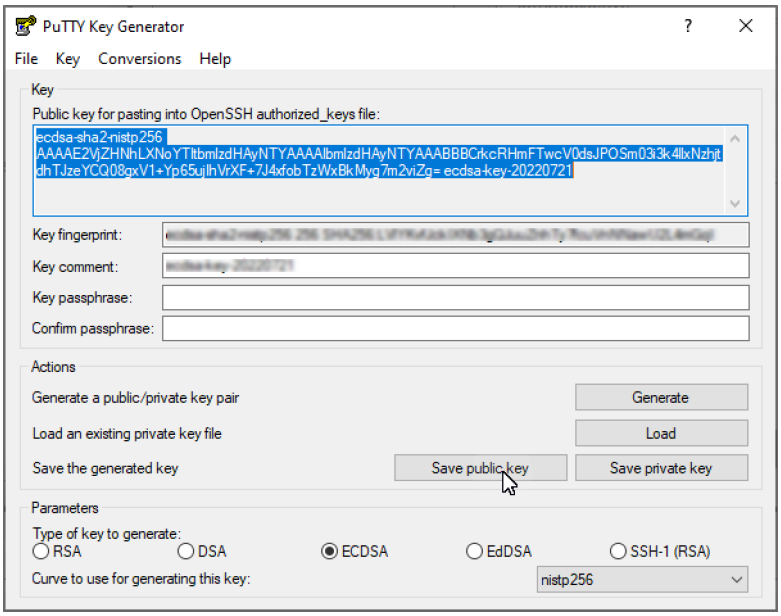
You can download and install the free PuTTY.msi, which includes PuTTYgen. When you run PuTTYgen you will see this:



Choose ECDSA as the type of key to generate, at the bottom of the window.

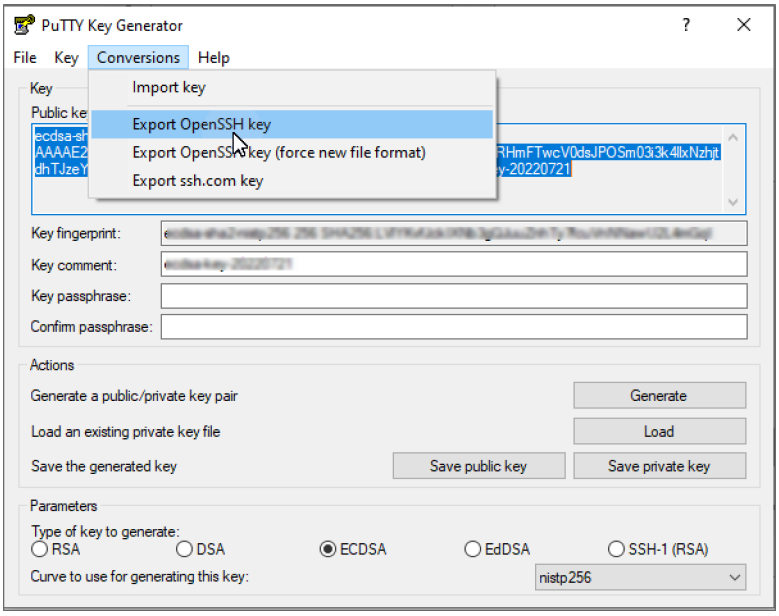
Select the <Generate> button. You will then be asked to move your mouse around for a short time, to add random activity to the generation process.

When the process completes your screen will look like this:



Select <Save public key>, to save the public key in a text file named `apple_public1`. This is in a compressed format. To derive an uncompressed format for Google you will have to use code to uncompress it.

Go to the Conversions tab and choose <Export OpenSSH key>.



Choose <Yes> to ignore the warning about not setting a passphrase. Save the private key as `private1.pem`.

Option 3: Leave it to your pass provider

If a pass provider offers to generate a key-pair for you, do ensure that you take control of both the generated public and private keys. This will ensure that you are free to change pass provider in the future, without needing to update all of the readers with new keys.

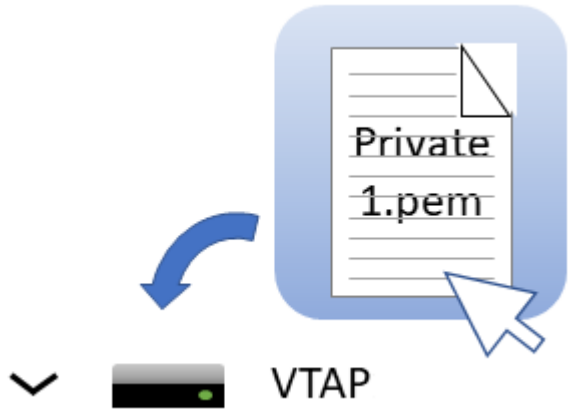
1.4 Use your ECC key pair

Supply the public key `apple_public1` or `google_public1` (just the long string) to your pass issuer, Apple or Google when they request your public key, so that the passes you issue can be unencrypted using your private key.

Apple use the public key in the generated pass. Google use the public key in your issuer configuration. It goes in the Google Pay and Wallet Console and is used in the API calls to generate a pass. (You will be asked to supply Google with a key version number alongside the public key, which must later match the one you set in the VTAP configuration.)

Note: String formatting can be quite sensitive, for instance, do not leave a trailing new line character at the end of the string.

Load the private key `private1.pem` onto your VTAP reader, by copying the file across to it, where it shows up in the file system of your PC as a mass storage device. (Consult the VTAP Configuration Guide for alternative ways to upload a key and how to link passes for a particular merchant/collector ID to this particular key in `config.txt`.)



Note: When you reboot the VTAP reader your key will have been stored in hardware, and will no longer be listed as a file on the device. You can confirm key file(s) have been loaded by checking in `Boot.txt`

2 About Application Notes

Application Notes address topics of interest to small groups of customers, or topics around the use of a VTAP reader with third-party systems.

The main documents available to support your use of the VTAP50 and VTAP100 are the Installation Guide for your VTAP reader model and the VTAP Configuration Guide. You will find the latest versions of these, and other useful information at **https://vtapnfc.com**.

If you need further help do contact us by email to **vtap-support@dotorigin.com**, or by phone +44 (0) 1428 685861 from Europe and Asia, or +1 (562) 262-9642 from Northern and Latin America.