Bumblebee Quick Start Guide
Last updated: Nov 26, 2023

This document will provide a set of steps that will guide you through installing Bumblebee, sharing keys with another user, and using a few methods for sharing secrets with them.

This document is focused on simple use cases that will show you some basic functionality in Bumblebee. While there are a number of features and options in Bumblebee, this document will not go into those details.

Step 1. Installing Bumblebee

Option A: Download runtime from Github repository

Bumblebee is a single runtime. You can get the latest, pre-built version for your platform in the "Releases" section at https://github.com/thoughtrealm/bumblebee. Simply download and place the runtime in a common path in your OS. You can place it in a directory and just execute it directly from there, but that can result in command lines that are longer than necessary. It is recommended to place the runtime in a common path.

Option B: Build and install using the Go compiler

If you have the Go compiler installed, you can clone the repo, then simply run "make install" in the root path of the repo.

If you are on Windows and do not have the *make* utility installed, you can run "go install" instead. This build should work fine, with the one exception that the output of the "bee version" command will not be fully populated with build times.

Once installed, you can verify it is running correctly by simply typing…

```
bee
```

That will output the root help info. You can also run this to see the version info…

```
bee version
```

2. Initialize the Bumblebee environment

The first step is to initialize the Bumblebee environment. This will create the default profile, populate the initial random key sets and some other artifacts that are required that are required for sharing secrets.

To do so, just run…

```
bee init
```

You will be asked about several options.

When asked, "*Enter a default sender key name or leave empty for none*", provide a name you wish to use for the default sender account in this profile. It could be a name, a handle, an email address, whatever you wish to use for identifying yourself. The other user will be able to use whatever name

they wish to use in their user store for your identity. Bumblebee will always validate the sending identity, regardless of any name assignment.

However, in a formal environment, like in a corporate environment, it is recommended to use something unique like your email address or an LDAP account name, etc.

Otherwise, for the other questions, just press enter for each to accept the default options for now.

Once the initialization is complete you can view the default profile identities by running…

```
bee list keypairs
```

That will show you the public keys only for the default and system key pairs.

You can use the "--**`show-all`**" flag to see the seed and private keys as well…

```
bee list keypairs --show-all
```

Of course, be aware that the you must never share your private keys with anyone. By default, they are not printed out when listing the key pairs unless you provide the "--**`show-all`**" flag.

> ***Note**: Bumblebee makes use of X25519 key pair functionality. Specifically, it uses the **NKEYS** repo/packages (https://github.com/nats-io/nkeys). **NKEYS** is provided by the **NATS** messaging server (https://nats.io/).*
>
> *Each identity is configured with two key pairs: a **Cipher** and a **Signing** key pair. The **Cipher** key pair is a curve25519 key pair construction and is used for the encrypting and decrypting processes. The **Signing** key pair is an ed25519 key pair and is used for signing secrets sent by that identity, so that the receiving user can validate the sender's identity. The curve25519 support is found in the **XKEYS** package of the **NKEYs** repo.*

You can also see the users that you have setup by running…

```
bee list users
```

Of course, at this point, you will find that your user list is empty. You must add or import users to your local profile(s). We will do this in this Quick Start Guide.

Step 3. Export your keys in order to share them with another user

To share secrets with another user, you must provide them with your public keys. This can be done easily by exporting your keys. There are several ways to do this, but will just focus on exporting them to a file.

To do this, run…

```
bee export user <username> --from-keypairs
```