Running external test suites with OpenSSL

It is possible to integrate external test suites into OpenSSL's `make test`. This capability is considered a developer option and does not work on all platforms.

Python PYCA/Cryptography test suite

This python test suite runs cryptographic tests with a local OpenSSL build as the implementation.

First checkout the `PYCA/Cryptography` module into `./pyca-cryptography` using:

```
$ git submodule update --init
```

Then configure/build OpenSSL compatible with the python module:

```
$ ./config shared enable-external-tests
$ make
```

The tests will run in a python virtual environment which requires virtualenv to be installed.

```
$ make test VERBOSE=1 TESTS=test_external_pyca
```

Test failures and suppressions

Some tests target older (<=1.0.2) versions so will not run. Other tests target other crypto implementations so are not relevant. Currently no tests fail.

krb5 test suite

Much like the PYCA/Cryptography test suite, this builds and runs the krb5 tests against the local OpenSSL build.

You will need a git checkout of krb5 at the top level:

```
$ git clone https://github.com/krb5/krb5
```

krb5's master has to pass this same CI, but a known-good version is krb5-1.15.1-final if you want to be sure.

```
$ cd krb5
$ git checkout krb5-1.15.1-final
$ cd ..
```

OpenSSL must be built with external tests enabled:

```
$ ./config enable-external-tests
$ make
```

krb5's tests will then be run as part of the rest of the suite, or can be explicitly run (with more debugging):

```
$ VERBOSE=1 make TESTS=test_external_krb5 test
```

Test-failures suppressions

krb5 will automatically adapt its test suite to account for the configuration of your system. Certain tests may require more installed packages to run. No tests are expected to fail.

GOST engine test suite

Much like the PYCA/Cryptography test suite, this builds and runs the GOST engine tests against the local OpenSSL build.

You will need a git checkout of gost-engine at the top level:

```
$ git submodule update --init
```

Then configure/build OpenSSL enabling external tests:

```
$ ./config shared enable-external-tests
$ make
```

GOST engine requires CMake for the build process.

GOST engine tests will then be run as part of the rest of the suite, or can be explicitly run (with more debugging):

```
$ make test VERBOSE=1 TESTS=test_external_gost_engine
```

Updating test suites

To update the commit for any of the above test suites:

- Make sure the submodules are cloned locally:

  $ git submodule update --init --recursive

- Enter subdirectory and pull from the repository (use a specific branch/tag if required):

  $ cd `<submodule-dir>` $ git pull origin master

- Go to root directory, there should be a new git status:

  $ cd ../ $ git status ... # modified: `<submodule-dir>` (new commits) ...

- Add/commit/push the update

  $ git add `<submodule-dir>` $ git commit -m `"Updated <submodule> to latest commit"` $ git push