cpackget v0.8.0 demo - signed packs

Luís Tonicha
06/09/2022

Current status

Local only

These operations (create/verify) can only be performed on local files (specification is required).

No “secure store”

Keys are either generated on-the-fly and written to the FS, or passed through a path. Ideally, they should be read from the OS keychain/keyring (as suggested).

Verifying with private key

For ease of usage, the signature is verified against the private key, which will have to change to the public key (that will be the published one)

Main goals

Verifying on pack installation

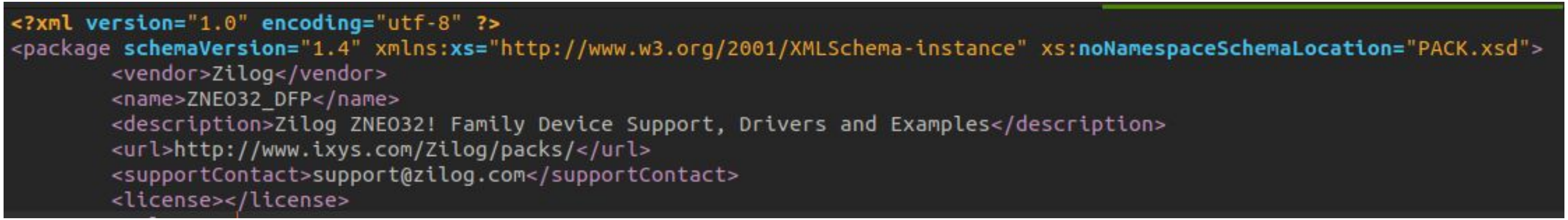
“checksum-verify” and “signature-verify” should be “hidden” from the user, and ran when a pack from a public index is going to be installed (if a *.checksum*/*.signature* is published)

Key handling

The generated/provided keys should be read from the keychain/keystore - careful implementation as this varies a lot by OS

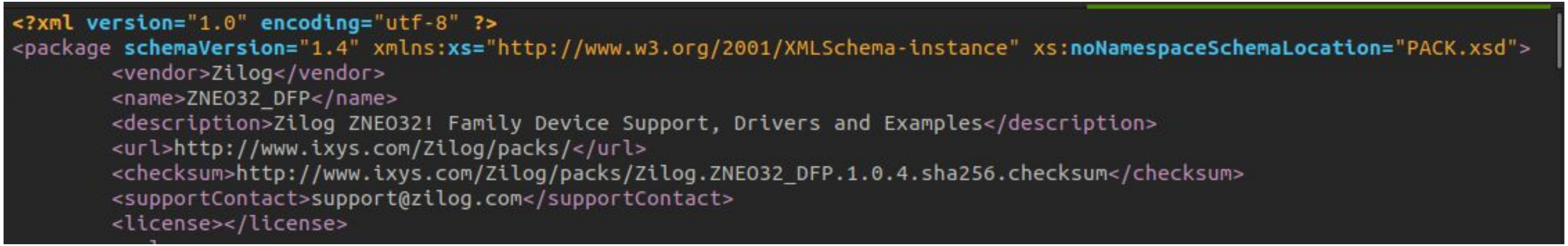
Proposal for checksum publishing

Before - typical .pdsc



```
<?xml version="1.0" encoding="utf-8" ?>
<package schemaVersion="1.4" xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xs:noNamespaceSchemaLocation="PACK.xsd">
        <vendor>Zilog</vendor>
        <name>ZNEO32_DFP</name>
        <description>Zilog ZNEO32! Family Device Support, Drivers and Examples</description>
        <url>http://www.ixys.com/Zilog/packs/</url>
        <supportContact>support@zilog.com</supportContact>
        <license></license>
```

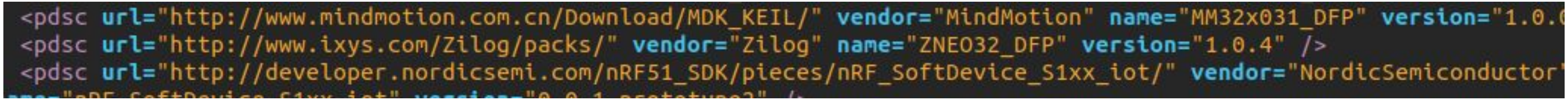
After - new tag points to remote .*checksum* of the latest version



```
<?xml version="1.0" encoding="utf-8" ?>
<package schemaVersion="1.4" xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xs:noNamespaceSchemaLocation="PACK.xsd">
        <vendor>Zilog</vendor>
        <name>ZNEO32_DFP</name>
        <description>Zilog ZNEO32! Family Device Support, Drivers and Examples</description>
        <url>http://www.ixys.com/Zilog/packs/</url>
        <checksum>http://www.ixys.com/Zilog/packs/Zilog.ZNEO32_DFP.1.0.4.sha256.checksum</checksum>
        <supportContact>support@zilog.com</supportContact>
        <license></license>
```

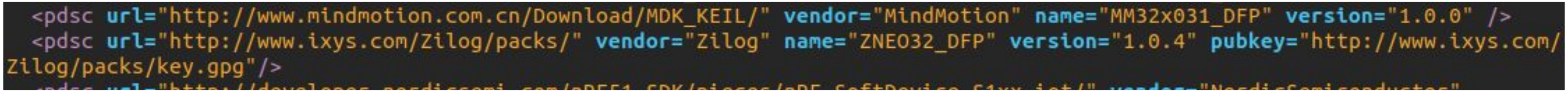
Proposal for signature publishing

Before - typical index.pidx



```
 <pdsc url="http://www.mindmotion.com.cn/Download/MDK_KEIL/" vendor="MindMotion" name="MM32x031_DFP" version="1.0.0
 <pdsc url="http://www.ixys.com/Zilog/packs/" vendor="Zilog" name="ZNEO32_DFP" version="1.0.4" />
 <pdsc url="http://developer.nordicsemi.com/nRF51_SDK/pieces/nRF_SoftDevice_S1xx_iot/" vendor="NordicSemiconductor"
ame="nRF_SoftDevice_S1xx_iot" version="0.0.1-prototype2" />
```

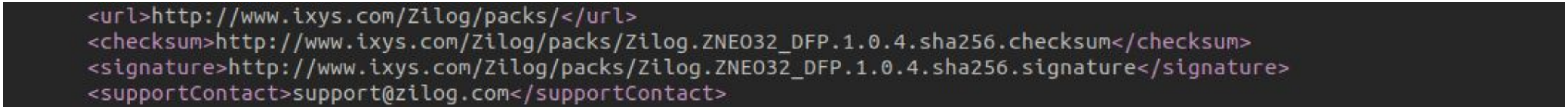
After - new tag on the index.pidx points to vendor’s public GPG key



```
  <pdsc url="http://www.mindmotion.com.cn/Download/MDK_KEIL/" vendor="MindMotion" name="MM32x031_DFP" version="1.0.0" />
  <pdsc url="http://www.ixys.com/Zilog/packs/" vendor="Zilog" name="ZNEO32_DFP" version="1.0.4" pubkey="http://www.ixys.com/
Zilog/packs/key.gpg"/>
  <pdsc url="http://developer.nordicsemi.com/nRF51_SDK/pieces/nRF_SoftDevice_S1xx_iot/" vendor="NordicSemiconductor"
```

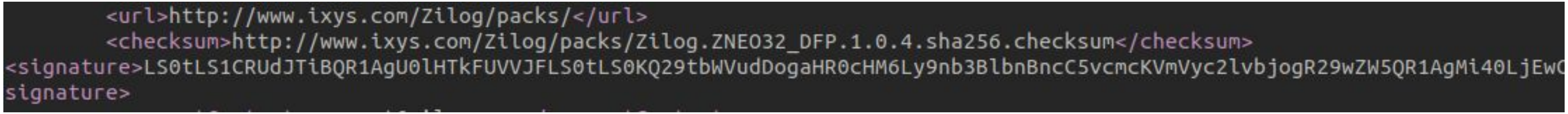
Proposal for signature publishing

Signature is either published as a separate file, specified by an URL



```
        <url>http://www.ixys.com/Zilog/packs/</url>
        <checksum>http://www.ixys.com/Zilog/packs/Zilog.ZNE032_DFP.1.0.4.sha256.checksum</checksum>
        <signature>http://www.ixys.com/Zilog/packs/Zilog.ZNE032_DFP.1.0.4.sha256.signature</signature>
        <supportContact>support@zilog.com</supportContact>
```

Or inline, encoded as base64 via “signature-create –output-base64”



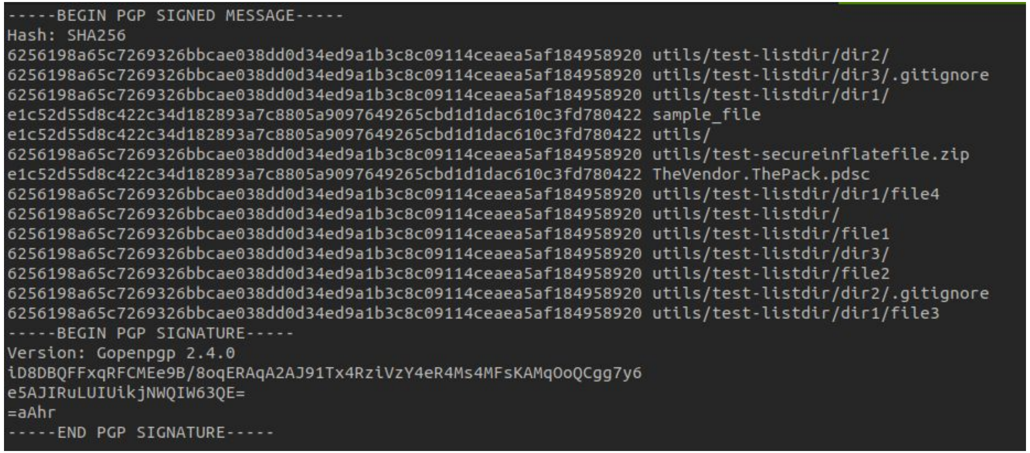
```
          <url>http://www.ixys.com/Zilog/packs/</url>
          <checksum>http://www.ixys.com/Zilog/packs/Zilog.ZNE032_DFP.1.0.4.sha256.checksum</checksum>
<signature>LS0tLS1CRUdJTiBQR1AgU0lHTkFUVVJFLS0tLS0KQ29tbWVudDogaHR0cHM6Ly9nb3BlbnBncC5vcmcKVmVyc2lvbjogR29wZW5QR1AgMi40LjEwC
signature>
```

Important notes

- The **public index maintainer** should be the one hosting/publishing the vendor’s public GPG key. This means that even if the vendor’s private key, the *.pack*, *.signature* and *.checksum* get compromised, a pack will be detected as malicious since the keys don’t match.

Important notes

- The *.signature* file could be merged with *.checksum*, as a PGP signed message instead of having it split:



```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256
6256198a65c7269326bbcae038dd0d34ed9a1b3c8c09114ceaea5af184958920 utils/test-listdir/dir2/
6256198a65c7269326bbcae038dd0d34ed9a1b3c8c09114ceaea5af184958920 utils/test-listdir/dir3/.gitignore
6256198a65c7269326bbcae038dd0d34ed9a1b3c8c09114ceaea5af184958920 utils/test-listdir/dir1/
e1c52d55d8c422c34d182893a7c8805a9097649265cbd1d1dac610c3fd780422 sample_file
e1c52d55d8c422c34d182893a7c8805a9097649265cbd1d1dac610c3fd780422 utils/
6256198a65c7269326bbcae038dd0d34ed9a1b3c8c09114ceaea5af184958920 utils/test-secureinflatefile.zip
e1c52d55d8c422c34d182893a7c8805a9097649265cbd1d1dac610c3fd780422 TheVendor.ThePack.pdsc
6256198a65c7269326bbcae038dd0d34ed9a1b3c8c09114ceaea5af184958920 utils/test-listdir/dir1/file4
6256198a65c7269326bbcae038dd0d34ed9a1b3c8c09114ceaea5af184958920 utils/test-listdir/
6256198a65c7269326bbcae038dd0d34ed9a1b3c8c09114ceaea5af184958920 utils/test-listdir/file1
6256198a65c7269326bbcae038dd0d34ed9a1b3c8c09114ceaea5af184958920 utils/test-listdir/dir3/
6256198a65c7269326bbcae038dd0d34ed9a1b3c8c09114ceaea5af184958920 utils/test-listdir/file2
6256198a65c7269326bbcae038dd0d34ed9a1b3c8c09114ceaea5af184958920 utils/test-listdir/dir2/.gitignore
6256198a65c7269326bbcae038dd0d34ed9a1b3c8c09114ceaea5af184958920 utils/test-listdir/dir1/file3
-----BEGIN PGP SIGNATURE-----
Version: Gopenpgp 2.4.0
iD8DBQFFxqRFCMEe9B/8oqERAqA2AJ91Tx4RziVzY4eR4Ms4MFsKAMqOoQCgg7y6
e5AJIRuLUIUikjNWQIW63QE=
=aAhr
-----END PGP SIGNATURE-----
```

Thank you. Questions?