strongSwan - Bug #2574

PKCS#8 key file written by OpenSSL 1.1 can't be loaded because it uses SHA-2 based PRFs for PKCS#5

01.03.2018 16:35 - Harald Dunkel

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	libstrongswan		
Target version:	5.6.3		
Affected version:	5.6.2	**Resolution:**	Fixed

Description

Hopefully its allowed to post an issue about the NetworkManager applet here.

Problem: It seems that the applet (1.4.2) is more picky about the x509 key files than charon itself.

Long story:

The Network Manager applet doesn't accept p12 files (AFAIK), so I have to split it using this procedure:

```
basename=mykey
p12=${basename}.p12
passw=secret
openssl pkcs12 -in ${p12} -passin env:passw -clcerts -nokeys -out "${basename}.cert.pem"
openssl pkcs12 -in ${p12} -passin env:passw -cacerts -nokeys -out "ca.cert.pem"
openssl pkcs12 -in ${p12} -passin env:passw -passout env:passw -nocerts -aes256 -out "${basename}.
key.pem"
```

Problem is: charon-nm complains about the key

```
Mar  1 15:20:56 ppcl001 charon-nm: 02[LIB] building CRED_PRIVATE_KEY - RSA failed, tried 10 builde
rs
```

Workaround is to pass the key through openssl once more, i.e. to use this procedure to split the p12 file instead:

```
basename=mykey
p12=${basename}.p12
passw=secret
openssl pkcs12 -in ${p12} -passin env:passw -clcerts -nokeys -out "${basename}.cert.pem"
openssl pkcs12 -in ${p12} -passin env:passw -cacerts -nokeys -out "ca.cert.pem"
openssl pkcs12 -in ${p12} -passin env:passw -passout env:passw -nocerts -aes256 -out "${basename}.
key.tmp.pem"
openssl rsa -in ${basename}.key.tmp.pem -passin env:passw -passout env:passw -aes256 -out ${basena
me}.key.pem -outform pem
```

The interesting part is, charon (without "-nm") doesn't need this workaround, so I wonder WTH? Would it be possible to improve the applet?

The "bad" key looks like this:

```
Bag Attributes
    localKeyID:  AA 52 F0 9B 62 63 30 47 E9 C9 2C EA 76 39 2C 8B 84 5D 81 AD
    friendlyName: ppcl001
Key Attributes: <No Attributes>
-----BEGIN ENCRYPTED PRIVATE KEY-----
:
:
```

```
-----END ENCRYPTED PRIVATE KEY-----
```

This is how the good key looks like:

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-256-CBC,21665EB5BAA2D4EEF93E3B43723C5B44

:
:
-----END RSA PRIVATE KEY-----
```

Associated revisions

Revision f71cccec - 07.03.2018 15:24 - Tobias Brunner

Merge branch 'pbkdf2-sha2'

Adds support for common SHA-2 based PRFs in PKCS#5/PBKDF2 as used by
OpenSSL 1.1 when generating PKCS#8-encoded private keys.

Fixes #2574.

History

#1 - 01.03.2018 17:15 - Tobias Brunner

- Category set to networkmanager (charon-nm)

- Status changed from New to Feedback

> The Network Manager applet doesn't accept p12 files (AFAIK)

No, it never has.

> It seems that the applet (1.4.2) is more picky about the x509 key files than charon itself.

It's not related to the applet (which is really only the GUI), but charon-nm, which loads the key from the configured file when the connection is initiated.

> The "bad" key looks like this:

Looks like a PKCS#8-encoded file.

> This is how the good key looks like:

And this is a plain PKCS#1-encoded file.

charon-nm uses exactly the same call to load the key as pki --print --type rsa --in "${basename}.key.pem" does. And that seems to work fine here with PKCS#8 files. Could you check which plugins charon-nm has loaded on your system? (You should see that in the log after Starting charon NetworkManager backend....)

#2 - 02.03.2018 08:30 - Harald Dunkel

Here it is:

```
Mar  2 07:56:24 ppcl001 systemd[1]: Started strongSwan IPsec IKEv1/IKEv2 daemon using ipsec.conf.
Mar  2 07:56:24 ppcl001 ipsec[4583]: Starting strongSwan 5.6.2 IPsec [starter]...
Mar  2 07:56:24 ppcl001 charon: 00[DMN] Starting IKE charon daemon (strongSwan 5.6.2, Linux 4.9.0-6-amd64, x86
_64)
Mar  2 07:56:24 ppcl001 charon: 00[CFG] PKCS11 module '<name>' lacks library path
Mar  2 07:56:24 ppcl001 charon: 00[CFG] loading ca certificates from '/etc/ipsec.d/cacerts'
Mar  2 07:56:24 ppcl001 charon: 00[CFG] loading aa certificates from '/etc/ipsec.d/aacerts'
Mar  2 07:56:24 ppcl001 charon: 00[CFG] loading ocsp signer certificates from '/etc/ipsec.d/ocspcerts'
Mar  2 07:56:24 ppcl001 charon: 00[CFG] loading attribute certificates from '/etc/ipsec.d/acerts'
Mar  2 07:56:24 ppcl001 charon: 00[CFG] loading crls from '/etc/ipsec.d/crls'
Mar  2 07:56:24 ppcl001 charon: 00[CFG] loading secrets from '/etc/ipsec.secrets'
```

```
Mar  2 07:56:24 ppcl001 charon: 00[CFG] expanding file expression '/var/lib/strongswan/ipsec.secrets.inc' fail
ed
Mar  2 07:56:24 ppcl001 charon: 00[LIB] loaded plugins: charon test-vectors ldap pkcs11 tpm aesni aes rc2 sha2
 sha1 md5 mgf1 rdrand random nonce x509 revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp dnskey sshk
ey pem openssl gcrypt af-alg fips-prf gmp curve25519 agent xcbc cmac hmac ctr ccm gcm curl attr kernel-netlink
 resolve socket-default connmark stroke updown counters
Mar  2 07:56:24 ppcl001 charon: 00[LIB] dropped capabilities, running as uid 0, gid 0
Mar  2 07:56:24 ppcl001 charon: 00[JOB] spawning 16 worker threads
Mar  2 07:56:24 ppcl001 ipsec[4583]: charon (4637) started after 180 ms
```

On my laptop the pki command line fails as well:

```
% pki --print --type rsa --in ${basename}.key.pem.bad
Private key passphrase:
building CRED_PRIVATE_KEY - RSA failed, tried 9 builders
parsing input failed
```

It works for the "good" key (not shown here).

#3 - 02.03.2018 08:42 - Harald Dunkel

PS: Platform is Debian 9.3. libgcrypt is version 1.7.6-2+deb9u2.

Not sure if this helps, but here is a list of shared objects loaded by pki, as reported by strace:

```
14155 open("/usr/lib/ipsec/tls/x86_64/libstrongswan.so.0", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or di
rectory)
14155 open("/usr/lib/ipsec/tls/libstrongswan.so.0", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory
)
14155 open("/usr/lib/ipsec/x86_64/libstrongswan.so.0", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or direct
ory)
14155 open("/usr/lib/ipsec/libstrongswan.so.0", O_RDONLY|O_CLOEXEC) = 3
14155 open("/usr/lib/ipsec/libpthread.so.0", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
14155 open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
14155 open("/lib/x86_64-linux-gnu/libpthread.so.0", O_RDONLY|O_CLOEXEC) = 3
14155 open("/usr/lib/ipsec/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
14155 open("/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
14155 open("/lib/x86_64-linux-gnu/libdl.so.2", O_RDONLY|O_CLOEXEC) = 3
14155 open("/lib/x86_64-linux-gnu/libcap.so.2", O_RDONLY|O_CLOEXEC) = 3
14155 open("/lib/x86_64-linux-gnu/libsystemd.so.0", O_RDONLY|O_CLOEXEC) = 3
14155 open("/lib/x86_64-linux-gnu/libselinux.so.1", O_RDONLY|O_CLOEXEC) = 3
14155 open("/lib/x86_64-linux-gnu/librt.so.1", O_RDONLY|O_CLOEXEC) = 3
14155 open("/lib/x86_64-linux-gnu/liblzma.so.5", O_RDONLY|O_CLOEXEC) = 3
14155 open("/usr/lib/x86_64-linux-gnu/liblz4.so.1", O_RDONLY|O_CLOEXEC) = 3
14155 open("/lib/x86_64-linux-gnu/libgcrypt.so.20", O_RDONLY|O_CLOEXEC) = 3
14155 open("/lib/x86_64-linux-gnu/libpcre.so.3", O_RDONLY|O_CLOEXEC) = 3
14155 open("/lib/x86_64-linux-gnu/libgpg-error.so.0", O_RDONLY|O_CLOEXEC) = 3
14155 open("/usr/lib/ipsec/plugins/libstrongswan-test-vectors.so", O_RDONLY|O_CLOEXEC) = 5
14155 open("/usr/lib/ipsec/plugins/libstrongswan-pkcs11.so", O_RDONLY|O_CLOEXEC) = 5
14155 open("/usr/lib/ipsec/plugins/libstrongswan-tpm.so", O_RDONLY|O_CLOEXEC) = 5
14155 open("/usr/lib/ipsec/libtpmtss.so.0", O_RDONLY|O_CLOEXEC) = 5
14155 open("/usr/lib/ipsec/plugins/libstrongswan-aesni.so", O_RDONLY|O_CLOEXEC) = 5
14155 open("/usr/lib/ipsec/plugins/libstrongswan-aes.so", O_RDONLY|O_CLOEXEC) = 5
14155 open("/usr/lib/ipsec/plugins/libstrongswan-rc2.so", O_RDONLY|O_CLOEXEC) = 5
14155 open("/usr/lib/ipsec/plugins/libstrongswan-sha2.so", O_RDONLY|O_CLOEXEC) = 5
14155 open("/usr/lib/ipsec/plugins/libstrongswan-sha1.so", O_RDONLY|O_CLOEXEC) = 5
14155 open("/usr/lib/ipsec/plugins/libstrongswan-md5.so", O_RDONLY|O_CLOEXEC) = 5
14155 open("/usr/lib/ipsec/plugins/libstrongswan-mgf1.so", O_RDONLY|O_CLOEXEC) = 5
14155 open("/usr/lib/ipsec/plugins/libstrongswan-rdrand.so", O_RDONLY|O_CLOEXEC) = 5
14155 open("/usr/lib/ipsec/plugins/libstrongswan-random.so", O_RDONLY|O_CLOEXEC) = 5
14155 open("/usr/lib/ipsec/plugins/libstrongswan-x509.so", O_RDONLY|O_CLOEXEC) = 7
14155 open("/usr/lib/ipsec/plugins/libstrongswan-revocation.so", O_RDONLY|O_CLOEXEC) = 7
14155 open("/usr/lib/ipsec/plugins/libstrongswan-pubkey.so", O_RDONLY|O_CLOEXEC) = 7
14155 open("/usr/lib/ipsec/plugins/libstrongswan-pkcs1.so", O_RDONLY|O_CLOEXEC) = 7
14155 open("/usr/lib/ipsec/plugins/libstrongswan-pkcs7.so", O_RDONLY|O_CLOEXEC) = 7
14155 open("/usr/lib/ipsec/plugins/libstrongswan-pkcs8.so", O_RDONLY|O_CLOEXEC) = 7
14155 open("/usr/lib/ipsec/plugins/libstrongswan-pkcs12.so", O_RDONLY|O_CLOEXEC) = 7
14155 open("/usr/lib/ipsec/plugins/libstrongswan-dnskey.so", O_RDONLY|O_CLOEXEC) = 7
14155 open("/usr/lib/ipsec/plugins/libstrongswan-sshkey.so", O_RDONLY|O_CLOEXEC) = 7
14155 open("/usr/lib/ipsec/plugins/libstrongswan-pem.so", O_RDONLY|O_CLOEXEC) = 7
14155 stat("/usr/lib/ipsec/plugins/libstrongswan-openssl.so", {st_mode=S_IFREG|0644, st_size=91560, ...}) = 0
14155 open("/usr/lib/ipsec/plugins/libstrongswan-openssl.so", O_RDONLY|O_CLOEXEC) = 7
14155 open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 7
```

```
14155 open("/usr/lib/x86_64-linux-gnu/libcrypto.so.1.1", O_RDONLY|O_CLOEXEC) = 7
14155 open("/usr/lib/ipsec/plugins/libstrongswan-gcrypt.so", O_RDONLY|O_CLOEXEC) = 7
14155 open("/usr/lib/ipsec/plugins/libstrongswan-af-alg.so", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/ipsec/plugins/libstrongswan-gmp.so", O_RDONLY|O_CLOEXEC) = 8
14155 open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libgmp.so.10", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/ipsec/plugins/libstrongswan-curve25519.so", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/ipsec/plugins/libstrongswan-hmac.so", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/ipsec/plugins/libstrongswan-curl.so", O_RDONLY|O_CLOEXEC) = 8
14155 open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libcurl.so.4", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libnghttp2.so.14", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libidn2.so.0", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/librtmp.so.1", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libssh2.so.1", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libpsl.so.5", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libssl.so.1.0.2", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libcrypto.so.1.0.2", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libgssapi_krb5.so.2", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libkrb5.so.3", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libk5crypto.so.3", O_RDONLY|O_CLOEXEC) = 8
14155 open("/lib/x86_64-linux-gnu/libcom_err.so.2", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/liblber-2.4.so.2", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libldap_r-2.4.so.2", O_RDONLY|O_CLOEXEC) = 8
14155 open("/lib/x86_64-linux-gnu/libz.so.1", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/tls/x86_64/libunistring.so.0", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
14155 open("/usr/lib/x86_64-linux-gnu/tls/libunistring.so.0", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or
 directory)
14155 open("/usr/lib/x86_64-linux-gnu/x86_64/libunistring.so.0", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file
 or directory)
14155 open("/usr/lib/x86_64-linux-gnu/libunistring.so.0", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libgnutls.so.30", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libhogweed.so.4", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libnettle.so.6", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libkrb5support.so.0", O_RDONLY|O_CLOEXEC) = 8
14155 open("/lib/x86_64-linux-gnu/libkeyutils.so.1", O_RDONLY|O_CLOEXEC) = 8
14155 open("/lib/x86_64-linux-gnu/libresolv.so.2", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libsasl2.so.2", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libp11-kit.so.0", O_RDONLY|O_CLOEXEC) = 8
14155 open("/lib/x86_64-linux-gnu/libidn.so.11", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libtasn1.so.6", O_RDONLY|O_CLOEXEC) = 8
14155 open("/usr/lib/x86_64-linux-gnu/libffi.so.6", O_RDONLY|O_CLOEXEC) = 8
```

#4 - 02.03.2018 08:46 - Tobias Brunner

> On my laptop the pki command line fails as well:

Hm, strange. Could you send me such a "bad" key? Debian 9 apparently comes with OpenSSL 1.1.0f, I wonder if they changed something in the format. Because this works fine here with 1.0.2g using the exact same commands you used (the encoded key looks the same as the example you gave in your original report).

#5 - 02.03.2018 10:07 - Tobias Brunner

This looks fun:

```
...
14155 open("/usr/lib/ipsec/plugins/libstrongswan-openssl.so", O_RDONLY|O_CLOEXEC) = 7
...
14155 open("/usr/lib/x86_64-linux-gnu/libcrypto.so.1.1", O_RDONLY|O_CLOEXEC) = 7
...
14155 open("/usr/lib/x86_64-linux-gnu/libcurl.so.4", O_RDONLY|O_CLOEXEC) = 8
...
14155 open("/usr/lib/x86_64-linux-gnu/libcrypto.so.1.0.2", O_RDONLY|O_CLOEXEC) = 8
```

No idea, if mixing two versions of libcrypto in the same process causes any problems.

#6 - 02.03.2018 10:16 - Harald Dunkel

- File secret.pem added

I could reproduce the problem using

```
% openssl genrsa | openssl pkcs8 -topk8 -inform PEM -outform PEM | pki --print --type rsa
Generating RSA private key, 2048 bit long modulus
.....................................................................................................+++
.......+++
e is 65537 (0x010001)
Enter Encryption Password:
Verifying - Enter Encryption Password:
Private key passphrase:
building CRED_PRIVATE_KEY - RSA failed, tried 9 builders
parsing input failed
```

Attached is a sample key with passphrase "secret".

PS: I get the same problem on Sid using openssl 1.1.0g-2 and strongswan-pki 5.6.2.

#7 - 02.03.2018 11:15 - Harald Dunkel

I think I found something:

```
# openssl genrsa | openssl pkcs8 -topk8 -v2 aes256 -v2prf hmacWithSHA1 -inform PEM -outform PEM | pki --print
--type rsa
Generating RSA private key, 2048 bit long modulus
....+++
.............................+++
e is 65537 (0x010001)
Enter Encryption Password:
Verifying - Enter Encryption Password:
Private key passphrase:
  privkey:   RSA 2048 bits
  keyid:     a6:1a:88:05:d1:03:55:f8:1d:b2:f4:1b:94:ca:b9:db:7c:f6:0d:f6
  subjkey:   ae:12:42:5b:6f:89:85:df:a6:af:db:e0:28:96:40:b2:e4:69:67:1d

# openssl genrsa | openssl pkcs8 -topk8 -v2 aes256 -v2prf hmacWithSHA256 -inform PEM -outform PEM | pki --prin
t --type rsa
Generating RSA private key, 2048 bit long modulus
.............................................................+++
...+++
e is 65537 (0x010001)
Enter Encryption Password:
Verifying - Enter Encryption Password:
Private key passphrase:
building CRED_PRIVATE_KEY - RSA failed, tried 9 builders
parsing input failed
```

Not much, but I hope this helps.

#8 - 05.03.2018 09:53 - Tobias Brunner

- Tracker changed from Issue to Bug

- Subject changed from Network Manager applet: x509 key file written by pkcs12 appears to be bad to PKCS#8 key file written by OpenSSL 1.1 can't be loaded because it uses SHA-2 based PRFs for PKCS#5

- Category changed from networkmanager (charon-nm) to libstrongswan

- Target version set to 5.6.3

> I think I found something:
> [...]
> Not much, but I hope this helps.

Yep, that's exactly it. OpenSSL 1.1 started using more modern algorithms for PKCS#5, by default (more specifically for the PRF used to derive the encryption key from the password with PBKDF2), which were only defined in RFC 8018 last year (I guess RSA's PKCS#5 standard might have been updated earlier besides that, as OpenSSL refers to v2.0 not v2.1). I pushed a fix to the *2574-pkcs5* branch.

#9 - 05.03.2018 12:56 - Harald Dunkel

I applied the patch to 5.6.2 for testing. There were a few failures:

```
% openssl genrsa -passout env:passw | openssl pkcs8 -passin env:passw -passout env:passw -topk8 -v2 aes256 -v2
prf hmacWithSHA1 -inform PEM -outform PEM | pki --print --type rsa
Generating RSA private key, 2048 bit long modulus
.................+++
.....................................+++
e is 65537 (0x010001)
Private key passphrase:
  privkey:   RSA 2048 bits
  keyid:     23:75:a8:c2:b0:f9:79:40:1d:91:70:6a:c8:de:2c:26:26:79:49:73
  subjkey:   b3:58:b6:e8:42:d5:bf:12:e0:94:b6:33:a8:74:e1:62:96:b1:7a:30

% openssl genrsa -passout env:passw | openssl pkcs8 -passin env:passw -passout env:passw -topk8 -v2 aes256 -v2
prf hmacWithSHA256 -inform PEM -outform PEM | pki --print --type rsa
Generating RSA private key, 2048 bit long modulus
..............+++
....+++
e is 65537 (0x010001)
Private key passphrase:
  privkey:   RSA 2048 bits
  keyid:     81:01:f8:ec:b4:bc:ee:ac:54:c1:b7:5f:a4:38:57:60:61:62:1b:f4
  subjkey:   c9:2b:a2:d5:fb:97:0e:13:00:05:ce:00:4c:a3:e2:40:40:50:2f:5d

% openssl genrsa -passout env:passw | openssl pkcs8 -passin env:passw -passout env:passw -topk8 -v2 aes256 -v2
prf hmacWithSHA224 -inform PEM -outform PEM | pki --print --type rsa
Generating RSA private key, 2048 bit long modulus
.............................+++
...............................................+++
e is 65537 (0x010001)
building CRED_PRIVATE_KEY - RSA failed, tried 9 builders
parsing input failed

% openssl genrsa -passout env:passw | openssl pkcs8 -passin env:passw -passout env:passw -topk8 -v2 aes256 -v2
prf hmacWithSHA384 -inform PEM -outform PEM | pki --print --type rsa
Generating RSA private key, 2048 bit long modulus
.........................+++
...+++
e is 65537 (0x010001)
Private key passphrase:
  privkey:   RSA 2048 bits
  keyid:     48:76:7a:19:e4:ac:be:1e:86:b0:ce:ac:45:ef:10:2f:bc:7d:6b:28
  subjkey:   53:4e:92:ac:b6:5f:76:88:55:8b:45:b0:07:e6:79:50:26:84:ac:fd

% openssl genrsa -passout env:passw | openssl pkcs8 -passin env:passw -passout env:passw -topk8 -v2 aes256 -v2
prf hmacWithSHA512 -inform PEM -outform PEM | pki --print --type rsa
Generating RSA private key, 2048 bit long modulus
............+++
..............................+++
e is 65537 (0x010001)
Private key passphrase:
  privkey:   RSA 2048 bits
  keyid:     37:bf:b7:06:b1:0e:f0:84:a5:e6:78:66:b9:01:5c:08:43:6e:44:88
  subjkey:   52:7d:15:72:16:96:3f:4f:93:1d:09:f2:35:69:2b:3b:27:2a:b1:92

% openssl genrsa -passout env:passw | openssl pkcs8 -passin env:passw -passout env:passw -topk8 -v2 aes256 -v2
prf hmacWithSHA512-224 -inform PEM -outform PEM | pki --print --type rsa
pkcs8: Unknown PRF algorithm hmacWithSHA512-224
pkcs8: Use -help for summary.
Generating RSA private key, 2048 bit long modulus
.+++
....building CRED_PRIVATE_KEY - RSA failed, tried 9 builders
parsing input failed
.................................................................................+++
e is 65537 (0x010001)

% openssl genrsa -passout env:passw | openssl pkcs8 -passin env:passw -passout env:passw -topk8 -v2 aes256 -v2
prf hmacWithSHA512-256 -inform PEM -outform PEM | pki --print --type rsa
pkcs8: Unknown PRF algorithm hmacWithSHA512-256
pkcs8: Use -help for summary.
Generating RSA private key, 2048 bit long modulus
......building CRED_PRIVATE_KEY - RSA failed, tried 9 builders
parsing input failed
.............................................................+++
...........+++
e is 65537 (0x010001)
```

I am not sure how to verify the list of supported PRF algorithms, but AFAICT the hmacWithSHA224 should have worked.

#10 - 05.03.2018 18:37 - Tobias Brunner

Yeah, strongSwan currently does not support the PRFs that failed (see source:src/libstrongswan/crypto/prfs/prf.h#L36).

#11 - 06.03.2018 09:05 - Harald Dunkel

Do you think this fix could be included into the next strongswan release?

#12 - 06.03.2018 09:23 - Tobias Brunner

> Do you think this fix could be included into the next strongswan release?

You did see the "Target version" field? :)

#13 - 07.03.2018 15:03 - Tobias Brunner

> I am not sure how to verify the list of supported PRF algorithms, but AFAICT the hmacWithSHA224 should have worked.

By the way, I just noticed this:

```
pkcs8: Unknown PRF algorithm hmacWithSHA512-256
```

Same for the 224-bit truncated version. OpenSSL apparently does not support these two algorithms either.

#14 - 07.03.2018 15:38 - Tobias Brunner

- Status changed from Feedback to Closed

- Assignee set to Tobias Brunner

- Resolution set to Fixed

#15 - 09.03.2018 08:09 - Harald Dunkel

Sorry, I missed the "Target Version" at the top.

Thanx very much for your support.

Files

secret.pem	1.83 KB	02.03.2018	Harald Dunkel