Manual Password Manager

Encode, decode “Indirect Passphrases”

timte.org 23.2.2023

Version

0.2

30.3.2023

General description

Storing passphrases and passwords in an indirect and encrypted form on paper.
For having a physical backup and implementing low or zero trust for passphrase storing.

Get a very compact database of passphrases which is encrypted with a master passphrase and offers the possibility to never reveal the actual passphrase to a keyboard at any time. This is done by usage of Indirect Passphrases (IPHR).

An IPHR points to handwritten notes, randomly sorted charlists, or word lists, to recover the actual passphrase (or password) using one of the specified methods.

To create the database and add passphrases, you have one of these options:

- using a one time pad for each IPHR (manual, zero trust approach)
- using a master passphrase and a generated keystream from an strong encryption algorythm (e.g. AES encryption with openssl, LibreSSL 3.6.2)

Usable with computational assistance or all manual by hand in a zero trust scenario.

Definitions

Version dependent definitions

Version	KeyID range	Blocksize	KeyID	Method	Length	IndirectKey	Subsqt-KID	Pad-ding
1.0	1-60	16 Byte	variable	1 Byte	1 Byte	Length Bytes (≤14 = Block_s - 2)	0 or 1 Byte	vari-able

Terms an concepts

Indirect Passphrase (IPHR) encoded data that points to handwritten notes to retrieve the passphrase / password.

The structure is:

KeyID, Method, Length, Indirect Key Data, Subsequent KeyID, Padding, Salt

where Method, Length, Indirect Key Data, Subsequent KeyID and Padding being encoded as described below and encrypted with a Master Passphrase and changing Salt.

```
  KeyID,      Method,Length,IndirectKey,SubsqtKID,Padding,    salt
  '--.-'      '-------------------.----------------------'    '-.-'
unencrypted                   encrypted                   unencrypted
```

KeyID Range Valid KeyIDs for specified MPM version

To encode MPM version number with the IPHR The range is permanently associated with a MPM version number.

If the KeyID range is used up, a new range is defined by a new MPM version.

On versions updates, not used KeyIDs from the old versions KeyID range become invalid.

For example MPM v. 1.0 assign KeyID range 0x01-0x60. Then MPM v. 1.01 will define KeyID range for example 0x61-0x7f in case all KeyIDs are used, or if a new MPM version is needed. In the latter case unused KeyIDs up to 0x60 will never be used.

No redefinition. If one would redefine KeyID range, an old version / copy of the decoding script could produce trouble.

KeyID (variable length since unencrypted)

ID of Indirect Passphrase matching an ID in the assignment list with the purpose of the passphrase.

KeyID range starting at 0x01. KeyID '0x00' is reserved for SubsqtKID NULL pointer.

Range borders defined with MPM version

Method (1 Byte) ID of method (“Manual Password Manager Method” - MPMM)

Method, how the original passphrase can get retrieved from the notebooks.

Length (1 Byte) length of the IndirectKey data.

For most methods in Bytes or count of IndirectKey elements.

IndirectKey (Length Bytes) data of indirect key. How this is to interpretated is defined by the MPMM. Length is up to block length -2 Bytes (Method,Length). If IndirectKey needs more space, SubsqtKID is used for following elements.

SubsqtKID (1 Byte) KeyID of next element of this key, if block length of IPHR is exceeded. 0x00 if no subsequent KeyID element.

Maybe omitted, if it is NULL and Blocksize is exceeded otherwise.

Padding (left Bytes until block length) padding of random data until block length

Salt (8 Bytes) The salt generated during encryption. Only needed for physically noted IPHRs. With digitally saved encrypted IPHRs the salt is per defaul pard of encrypted file.

Encryption method

Encrypted by openssl v. LibreSSL 3.6.2

command

```
openssl enc -e -aes-128-ctr -pbkdf2
```

in this version equivalent to

```
openssl enc -e -aes-128-ctr -iter 10000
```

Manual Password Manager Methods (MPMM)

There are different methods (“Manual Password Manager Methods” MPMMs) to retrieve the actual passphrase.

MPMMid Title

0x0 Date-method Recover passphrase from words in notebook.

The passphrase is split into parts (wordpart_n) (maybe not, then wordpart_n is only “wordpart_1”). The different parts can be noted manually e.g. in a notebook at the specified date and word position, while “word position” is the count of the word in the text on this date. This makes up the wordpart_n. But a wordpart_n could also consist of a single character not derived from a date but from a specified char_list (like in method 0x02 “char list”).

: For the example password

```
bikedrawingreally:)
```

the decoded IndirectKey data would be the 5 wordpart_n (three dates, two chars):

```
11.5.2022,11;13.2.2021,12;13.2.2021,6;14;42
```

with according notes on that dates, having the words ‘bike’ beeing word 11 on 11.5.2022, ‘drawing’ beeing word 12 on 13.2.2021 and ‘really’ beeing word 6 on 13.2.2021. And having chars ‘14’ and ‘42’ matching the chars ‘:’ and ‘)’ of a specified char_list.

Encoded Bytes of this example (grouped by fields of date_1, word_1, .., date_3, word_3, char_1, char_2):

```
627b 0b 6028 0c 6028 06 0e 2a
```

With

The wordpart_n data is 3 Bytes. Byte 1 and 2 beeing the upper two bytes of UTC unix time of this date. Byte 3 is the wordcount of this date. The lower two bytes of unix time are supposed to be `0x0000` for the reverse operation:

```
% date -R -j -f "%s" "$(( 0x627b0000 ))"
Wed, 11 May 2022 00:14:56 +0000
% date -R -j -f "%s" "$(( 0x60280000 ))"
Sat, 13 Feb 2021 16:36:16 +0000
```

With this there could be dates with two valid values. For 01.01.2023 `0x63b1` and `0x63b2` are valid

```
% export TZ=UTC
% date -R -j -f "%s" "$(( 0x63b10000 ))"
Sun, 01 Jan 2023 03:37:36 +0000
% date -R -j -f "%s" "$(( 0x63b20000 ))"
Sun, 01 Jan 2023 21:49:52 +0000
```

0x1 Charlist (randomASCII)

list of characters, randomly sorted, which passphrases are made of.

0x2 Direct Encryption (non indirect passphrase, less secure)

[description]

0x3 Dice Ware (like effs dice ware)

[description]

0x4 Custom Dice Ware (merged, selfmade word list)

[description]

0x5 Legacy Date (old notes passphrase, nonsplitted)

[description]

0x5 method (short description)

[description]