

Übungsblatt 5

Aufgabe 2: HOTP (Praxisaufgabe)

HMAC-SHA1 ist in [hmac.rs](#) implementiert.

HOTP ist in [hotp.rs](#) implementiert.

[main.rs](#) implementiert einen "YubiKey-Simulator" zum generieren von HOTP-Werten und einen kommandozeilen-basierten HOTP Validator.

Eine kompilierte Linux x86-64 Binary kann von [hier](#) heruntergeladen werden.

Zum selbst kompilieren ist es ausreichend, `cargo build --release` auszuführen.

Kommandozeilen-Parameter

Das Programm kann mit Kommandozeilen-Argumenten ausgeführt werden, um es zu konfigurieren. Valide Schreibweisen zum Konfigurieren der HOTP-Wert-Länge sind beispielsweise `--length=6` und `-l6`.

Die folgenden Parameter werden akzeptiert. Unbekannte Parameter-Namen/Kurzformen werden ignoriert, ungültige Werte führen zur Ausgabe eines Fehlers und zum Ausführungsabbruch.

Name	Kurzform	Beschreibung	Default
<code>--mode</code>	<code>-m</code>	Ausführungsmodus. Erlaubte Werte: "key", "validator"	"key"
<code>--counter</code>	<code>-c</code>	Initialer Wert des HOTP Moving Factors. Integer ≥ 0	0
<code>--key</code>	<code>-k</code>	HOTP Secret Key. Bytes als Hex-String	0000...0 (40 Zeichen)
<code>--length</code>	<code>-l</code>	HOTP-Wert-Länge. Integer ≥ 0	6
<code>--window</code>	<code>-w</code>	Fenster-Größe für den Validator	20

Ausführungsmodi

Als YubiKey-Simulator (HOTP-Generator)

Im `key`-Modus gibt das Programm für jede Eingabe eines Zeilenumbruchs einen HOTP-Wert aus. Es beginnt mit dem konfigurierten Counter-Wert und erhöht den Wert dann stets um 1. Ein mit

[illegible]

konfigurierter YubiKey ist äquivalent zu dem mit

```
./hotp_linux_x86-64 \
--length=6 --mode=key --counter=29344 \
--key=1212121212121212121212121212121212121212
```

ausgeführten Programm.

[illegible]

Als HOTP Validator

Im `validator`-Modus überprüft das Programm eingegebene HOTP-Werte auf Validität über die konfigurierte Fenstergröße. Alle Zeichen einer Eingabezeile bis auf die letzten `length` Zeichen werden dabei ignoriert, sodass das Programm direkt gegen einen "echten" YubiKey getestet werden kann.

Ein mit

[illegible]

```
-oath-imf=29344 -ofixed=cccccccc -oserial-api-visible -oappend-cr
```

konfiguriertes YubiKey erzeugt mit

```
./hotp_linux_x86-64 \  
--length=6 --mode=validator --counter=29344 --window=20 \  
--key=1212121212121212121212121212121212121212121212121212121212121212
```

validierbare HOTP-Werte.

```
tmp ykpersonalize \  
-1 -a1212121212121212121212121212121212121212121212121212121212121212 -oath-hotp \  
-oath-imf=29344 -ofixed=cccccccc -oserial-api-visible -oappend-cr  
Firmware version 3.4.9 Touch level 1029 Program sequence 3  
  
Configuration data to be written to key configuration 1:  
  
OATH id: 00000000000000  
uid: n/a  
key: h:1212121212121212121212121212121212121212121212121212121212121212  
acc_code: h:00000000000000  
OATH IMF: h:72a0  
ticket_flags: APPEND_CR|OATH_HOTP  
config_flags:  
extended_flags: SERIAL_API_VISIBLE  
  
Commit? (y/n) [n]: y  
tmp ./hotp_linux_x86-64 \  
--length=6 --mode=validator --counter=29344 --window=20 \  
--key=1212121212121212121212121212121212121212121212121212121212121212  
Executing as HOTP validator. Start providing HOTP values, for example by using your YubiKey.  
> 00000000486073  
"486073" is valid at counter value 29344  
> 00000000528312  
"528312" is valid at counter value 29345  
> 00000000976603  
"976603" is valid at counter value 29346  
> 00000000278233  
"278233" is valid at counter value 29347  
> 00000000310690  
"310690" is valid at counter value 29348  
> 00000000544536  
"544536" is valid at counter value 29349  
> 00000000300456  
"300456" is valid at counter value 29350  
> █
```