

HOTP and TOTP

Martin Stanek

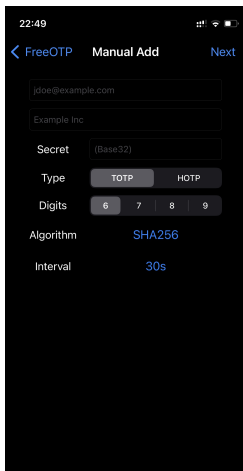
Department of Computer Science
Comenius University
`stanek@dcs.fmph.uniba.sk`

Cryptology 1 (2023/24)

Introduction

- ▶ multifactor authentication, 2-step verification, ...
 - ▶ something you know/have/are
 - ▶ often mobile phone: SMS, push notifications, authenticator app
- ▶ one-time passwords
- ▶ HOTP and TOTP
 - ▶ HOTP: HMAC-Based One-Time Password Algorithm (RFC 4226)
 - ▶ TOTP: Time-Based One-Time Password Algorithm (RFC 6238)

FreeOTP example



22:49

< FreeOTP Manual Add Next

jdoo@example.com

Example Inc

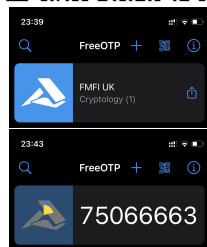
Secret (Base32)

Type TOTP HOTP

Digits 6 7 8 9

Algorithm SHA256

Interval 30s



otpauth://totp/FMFI%20UK:Cryptology%20(1)?secret=ONUG65LMM
RRGKYTFOR2GK4TUNBQW45DINFZTCMRT&algorithm=SHA256&digits=8
&period=30&lock=false

HOTP

- ▶ actors: HOTP generator (client), HOTP validator (server)
- ▶ $\text{HMAC}_K(\cdot)$, usually based on SHA-1 (default)
- ▶ parameters:
 - ▶ K – shared secret (static symmetric key, ≥ 128 bits)
 - ▶ C – counter value (8B, synchronized, starts with 0)
 - ▶ Digits – output length (≥ 6)

$$\text{HOTP}(K, C) = \text{Truncate}(\text{HMAC}_K(C))$$

- ▶ Truncate – transform HMAC output to HOTP value
 - ▶ focus on uniformity and implementation clarity
- ▶ client increments C , and then calculates the next HOTP value
- ▶ server recalculates and compares received HOTP value
 - ▶ server increments C after a successful authentication

HOTP – remarks

- ▶ authentication protocol over a secure channel, e.g. TLS, IPsec
- ▶ security of shared secret is important (obviously)
- ▶ validation failure (HOTP values do not match)
 - ▶ resynch protocol (look-ahead window)
 - ▶ look-ahead parameter s – server validates against s consecutive values
 - ▶ if unsuccessful → failed attempt
- ▶ brute-force attack prevention
 - ▶ brute-force attack is, in theory, the best attack possible
 - ▶ throttling parameter – the maximum number of failed attempts
- ▶ in some scenarios, server can request multiple HOTP values
- ▶ bidirectional authentication possible

TOTP

- ▶ extension of HOTP: counter value C replaced by time
 - ▶ short-lived OTP values (instead of “valid until next successful authentication”)
- ▶ HMAC based on SHA-1 (default), SHA-256, SHA-512
- ▶ parameters:
 - ▶ X – time step in seconds (usually $X = 30$ seconds)
 - ▶ time – current Unix time (seconds since 1.1.1970)
 - ▶ $T = \lfloor \text{time}/X \rfloor$ – number of time steps

$$\text{TOTP}(K, T) = \text{HOTP}(K, T)$$

TOTP – remarks

- ▶ time step size: security vs. usability
- ▶ “one-time only” requirement: the server must not accept the second attempt after the successful validation
- ▶ delay window – accept TOTP value from the previous time step
 - ▶ time when the value was entered vs. time when it is validated
 - ▶ recommended 1 time step
- ▶ resynchronization
 - ▶ clock drift
 - ▶ server can set limits on forward and backward time drifts
 - ▶ remember the drift and adjust for next validation