

Time-based One-Time Passwords (TOTP)

What is TOTP?

- TOTP = Time-based One-Time Password
 - Used for Two-Factor Authentication (2FA)
 - Generates short-lived, single-use codes
 - Based on a **shared secret** and **current time**
-

TOTP Flow Overview

1. Shared secret exchange (during setup)
 2. Device generates code using secret + time
 3. Server does the same independently
 4. User enters code
 5. Server validates code
-

TOTP in Detail

Press ↓ to step through the process

Setup Phase

- Server generates a **base32-encoded secret**
 - Shows it as a **QR code**
 - User scans it into their app (e.g., Authy, Google Authenticator)
-

Time Step Calculation

- Epoch time = seconds since Jan 1, 1970

- `T = floor(current_time / 30)`
 - This value changes every 30 seconds
-

HMAC Computation

- `HMAC = HMAC-SHA1(secret, T)`
 - SHA256 or SHA512 also allowed (RFC 6238)
 - Result: 20-byte hash
-

Dynamic Truncation

- Use the last nibble of the HMAC to get offset
 - Extract 4 bytes from HMAC starting at offset
 - Convert to 31-bit integer
-

Final Code Generation

- Modulo: `code = binary % 10^digits`
 - Usually 6 digits
 - Code is shown to user in authenticator app
-



Visual Summary

 TOTP Diagram



Key Specs

- **RFC:** 6238 (TOTP), 4226 (HOTP)
- **Hash:** HMAC-SHA1 (optionally SHA256/SHA512)
- **Time Step:** 30 seconds

- **Code Length:** 6 or 8 digits



TOTP Python Example

```
import pyotp

totp = pyotp.TOTP('JBSWY3DPEHPK3PXP') # Example base32 secret
print(totp.now()) # Show current 6-digit code
```