Network Working Group
Internet-Draft
Intended status: Informational
Expires: 23 August 2026

A. Escobar
independent

19 February 2026

The otpauth URI Scheme
draft-andesco-otpauth-uri-00

Abstract

This document defines syntax and processing rules for the otpauth: URI scheme used to provision one-time-password (OTP) credentials (verification codes).

This document defines a common baseline for interoperability, including issuer handling rules that improve account matching while maintaining compatibility with existing deployments.

Discussion Venues

This note is to be removed before publishing as an RFC.

Source for this draft and an issue tracker can be found at https://github.com/andesco/otpauth-uri.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 23 August 2026.

Copyright Notice

Table of Contents

1. Introduction

The otpauth: URI scheme is used to transfer OTP configuration, including a shared secret and related parameters, into OTP clients. The scheme is currently registered with IANA as a provisional URI scheme [URI-SCHEMES].

Current ecosystem behavior is primarily described by vendor documentation, including Google's otpauth key URI format [GOOGLE-KEYURI] and Apple's verification code guidance [APPLE-VERIFICATION-CODES]. Those documents are largely aligned, but differ on semantics for issuer information.

This document defines interoperable behavior for URI producers and consumers, with emphasis on reliable account association in modern password managers.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This specification uses the Augmented Backus-Naur Form (ABNF) notation defined in [RFC5234].

This specification uses the OTP concepts from [RFC4226] and [RFC6238]. In this document, secret, issuer, and account identify credential attributes associated with HOTP and TOTP methods.

The term URI is imported from [RFC3986]. Mentions of "query string" in this document refer to the URI query component defined in Section 3.4 of [RFC3986].

The Base16, Base32, and Base64 data encodings referenced in this document are from [RFC4648].

The terms "producer" and "consumer" are used throughout:

* A producer creates an otpauth: URI.

* A consumer parses and imports an otpauth: URI.

3. URI Format

An otpauth: URI uses this general form:

otpauth://<type>/<label>?<parameters>

where <type> identifies the OTP algorithm family and <parameters> contains URL query parameters.

3.1. Syntax

The syntax is defined using ABNF from [RFC5234] and URI productions from [RFC3986].

```
uri = "otpauth://" otp-type "/" label "?" parameter
      *( "&" parameter )

otp-type = "totp" / "hotp"

label = issuer-label ( ":" / "%3A" ) *"%20" account
        / issuer-label
        / account

issuer-label = 1*( unreserved / pct-encoded / sub-delims / "@" )
account      = 1*( unreserved / pct-encoded / sub-delims / "@" )

parameter = secret / algorithm / digits / counter
            / period / issuer / extension

secret    = "secret=" 1*( %x41-5A / %x32-37 ) ; Base32 A-Z2-7
algorithm = "algorithm=" ( "SHA1" / "SHA256" / "SHA512" )
digits    = "digits=" ( "6" / "8" )
counter   = "counter=" 1*DIGIT
period    = "period=" 1*DIGIT
issuer    = "issuer=" *pchar ; domain name recommended
extension = 1*( ALPHA / DIGIT / "-" / "_" ) "=" *pchar
```

Per [RFC5234], quoted ABNF literals are case-insensitive. Therefore, otpauth, totp, and hotp are matched case-insensitively by this grammar. Producers SHOULD emit lowercase forms for consistency. The same case-insensitive matching applies to quoted parameter names and quoted enumerated values in this ABNF.

This ABNF names commonly used parameters explicitly and allows extension parameters. Parameter requirements are defined in the following section. Parameter values MUST percent-encode literal & characters as %26.

A consumer MUST parse label before decoding as follows:

* If label contains a separator, split the raw string at the first separator, where separator is either literal : or percent-encoded %3A (case-insensitive).

* If label does not contain a separator, treat the entire label as account.

* Percent-decode each parsed component using standard URI decoding rules from [RFC3986].

* If decoded issuer-label or decoded account contains a colon, the consumer MUST reject the URI.

3.2. otp-type

otp-type identifies the OTP method and MUST be either totp or hotp.

3.3. label

label identifies the account being provisioned:

<issuer-label>: <account>

In this structure:

* issuer-label is optional

* the separator is either a literal colon (:) or %3A

* optional spaces before account are encoded as %20

* issuer-label and account MUST NOT themselves contain a colon

Producers SHOULD percent-encode label components using URI encoding rules from [RFC3986]. Valid examples include alice@example.com, Example:alice@example.com, and Example%20Issuer%3A%20alice@example.com.

3.4. Parameters

Parameter order is not significant.

A producer MUST include exactly one secret parameter.

A consumer MUST reject the URI if secret is missing.

A producer MAY include algorithm, digits, counter, period, and issuer; each known parameter MUST appear at most once.

A consumer MUST reject the URI if any known parameter appears more than once.

3.4.1. secret

The secret parameter carries the shared OTP secret and is mandatory. The value MUST use unpadded Base32 with alphabet A-Z2-7, as specified by [RFC4648]. Producers SHOULD emit uppercase Base32 text. Consumers MAY accept lowercase Base32 text for interoperability.

3.4.2. issuer

The issuer parameter identifies the relying party that issued the OTP credential.

A producer SHOULD include issuer.

A producer SHOULD set issuer to a stable service identifier that is useful for account matching and credential suggestion. A domain name controlled by the relying party (for example, example.com) is RECOMMENDED where available, but non-domain identifiers are allowed.

A consumer SHOULD use issuer as the primary identifier for account matching and credential suggestion.

3.4.3. algorithm

algorithm is OPTIONAL. If absent, the default is SHA1.

Valid algorithm values are SHA1, SHA256, and SHA512. Producers SHOULD use one of these values.

* Consumers MUST support SHA1.

* Consumers SHOULD support SHA256 and SHA512.

For hotp URIs, [RFC4226] defines HOTP with HMAC-SHA-1, so SHA1 is RECOMMENDED for maximum compatibility.

If a consumer receives an unsupported algorithm value, it SHOULD reject the URI.

3.4.4. digits

digits is OPTIONAL. If absent, the default is 6. Valid digits values are 6 and 8.

* Producers SHOULD use 6 or 8.

* Consumers MUST support 6.

* Consumers SHOULD support 8.

If a consumer receives an unsupported digits value, it SHOULD reject the URI.

3.4.5. period

period is OPTIONAL for totp. If absent, the default is 30. It MUST be ignored for hotp.

3.4.6. counter

counter is REQUIRED for hotp and MUST be ignored for totp.

These parameters map to HOTP and TOTP behavior defined in [RFC4226] and [RFC6238], while allowing extension values where explicitly noted.

3.4.7. Additional Parameters

Producers MAY include additional parameters that are not defined by this document.

Consumers SHOULD ignore unrecognized parameters unless local policy requires rejecting them.

4. Issuer Label Prefix and Issuer Parameter

In this document:

* The issuer label prefix (issuer-label) is presentation-oriented text for display.

* The issuer parameter (issuer) is the canonical identifier for account matching.

A producer MAY include an issuer label prefix for backward compatibility.

A producer that includes both values SHOULD use a human-readable service name for the issuer label prefix and a stable matching identifier for the issuer parameter.

If both issuer label prefix and issuer parameter are present:

* A consumer MUST NOT reject the URI only because the two values differ.

* A consumer SHOULD treat the issuer parameter as authoritative for account matching.

* A consumer MAY use issuer label prefix for display.

A consumer MUST continue to accept URIs where issuer label prefix and issuer parameter are equal, as this remains common in deployed systems [GOOGLE-KEYURI].

5. Examples

The following are valid otpauth: URIs (where PB4XU is the unpadded Base32 encoding of xyz):

otpauth://totp/Example?secret=PB4XU&issuer=example.com

otpauth://totp/Example%3Aalice?secret=PB4XU&issuer=example.com

otpauth://hotp/Example?secret=PB4XU&counter=42&issuer=example.com

6. Security Considerations

This URI format does not in itself pose a security threat. However, the secret parameter carries a bearer credential. Disclosure of secret enables OTP generation and can lead to account compromise.

Producers and consumers:

* SHOULD treat otpauth: URIs as sensitive secrets.

* SHOULD avoid writing full URIs to logs, analytics, and crash reports.

* SHOULD transport provisioning data only over authenticated and encrypted channels.

Consumers SHOULD display issuer and account information clearly before import so users can detect phishing or provisioning mistakes.

7. Privacy Considerations

otpauth: URIs may reveal account identifiers and service associations through label and issuer fields.

Applications handling these URIs SHOULD minimize retention and SHOULD avoid sharing raw URI values across process boundaries unless necessary for explicit user action.

8. IANA Considerations

The otpauth: URI scheme is registered as provisional in the IANA URI Schemes registry [URI-SCHEMES].

Upon publication of this document as an RFC, IANA is requested to update the reference for the existing otpauth provisional registration to point to this RFC.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/rfc/rfc3986>.

[RFC4226] M'Raihi, D., Bellare, M., Hoornaert, F., Naccache, D., and O. Ranen, "HOTP: An HMAC-Based One-Time Password Algorithm", RFC 4226, DOI 10.17487/RFC4226, December 2005, <https://www.rfc-editor.org/rfc/rfc4226>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/rfc/rfc4648>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/rfc/rfc5234>.

[RFC6238] M'Raihi, D., Machani, S., Pei, M., and J. Rydell, "TOTP: Time-Based One-Time Password Algorithm", RFC 6238, DOI 10.17487/RFC6238, May 2011, <https://www.rfc-editor.org/rfc/rfc6238>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

9.2. Informative References

[APPLE-VERIFICATION-CODES] Apple, "Securing Logins with iCloud Keychain Verification Codes", <https://developer.apple.com/documentation/authenticationservices/securing-logins-with-icloud-keychain-verification-codes>.

[GOOGLE-KEYURI]
Google, "Key Uri Format", <https://github.com/google/google-authenticator/wiki/Key-Uri-Format>.

[URI-SCHEMES]
Internet Assigned Numbers Authority, "Uniform Resource Identifier (URI) Schemes", <https://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml>.

Acknowledgments

The author thanks the maintainers and implementers of OTP ecosystems, including the teams at Apple and Google whose documentation helped shape existing interoperable behavior.

Author’s Address

Andrew Escobar
independent
Canada
Email: ietf@andrewe.dev