Registry Extensions Working Group
Internet-Draft
Intended status: Standards Track
Expires: 5 June 2026

Z. Zhou
Namefi

2 December 2025

Domain Transfer Authorization Using Cryptographic Signatures
draft-zzn-authcodesec-02

Abstract

This document specifies a mechanism to enhance domain transfer security by transitioning from a shared-secret authorization model to a asymmetric cryptographic signature-based validation system. Using asymmetric cryptographic signatures enables many benefits over a shared secret, such as non-repudiation, improved auditability through clear identification of the authorizing entity, elimination of the need for registries to store and secure shared secrets, replay protection through timestamp validation, and reduced risk of interception since only the private key needs to remain secret.

This document establishes the following AuthCodeSEC extension of EPP with the following protocol elements:

1. Where to place the signature and hash data in the EPP command and

2. How is data hashed and signed, and how to verify the signature

3. How to verify the signature;

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 5 June 2026.

Table of Contents

1. Introduction

The current domain transfer process relies on a shared secret (authInfo) which is susceptible to interception, misuse, and lack of auditability. This document proposes a simplified, secure replacement using cryptographic signatures, building upon the Extensible Provisioning Protocol (EPP) [RFC5730].

Key features of this specification:

* Mandatory use of ECDSA Curve P-256 with SHA-256 for strong security.

* A flexible "Signer Payload" to clearly identify who authorized the transfer (e.g., Registrar, Regulator, Registry, or Registrant).

* Removal of complex transition models in favor of a clean, signature-based approach.

1.1. Motivation

The evolution of Internet protocols has consistently moved from trust-based, shared-secret, or unauthenticated models toward cryptographic verification to address modern security threats.

Two notable precedents within the IETF serve as guiding examples for this specification:

* HTTP to HTTPS (RFC 2818): The transition from plain HTTP to HTTP Over TLS addressed the inherent risks of eavesdropping, tampering, and man-in-the-middle attacks. By layering HTTP over TLS, the protocol ensured confidentiality and server authentication, eliminating the reliance on clear-text communication.

* DNS to DNSSEC (RFC 4033, 4034, 4035): The original Domain Name System lacked data origin authentication and integrity. DNSSEC introduced cryptographic signatures to the DNS hierarchy, allowing resolvers to verify that data originated from the authoritative source and was not modified in transit, thus mitigating cache poisoning and spoofing attacks.

AuthCodeSEC applies these same principles to the domain transfer process. The current shared-secret (AuthCode) model resembles the early, unauthenticated era of other protocols. It is susceptible to interception, unauthorized reuse, and lacks non-repudiation.

By adopting asymmetric cryptographic signatures, this specification achieves:

1. Strong Authentication: Replaces a static password with a digital signature, proving possession of a private key without revealing it.

2. Elimination of Shared Secret Risks: Significantly reduces the attack surface by removing the shared secret entirely.

    * Transit: The private key is never transmitted, making credential eavesdropping impossible.

    * Storage: Registries no longer need to store and secure password databases, eliminating a major source of potential data leakage.

3. Integrity: Ensures the transfer request data (domain, gaining registrar, timestamp) has not been tampered with.

4. Non-repudiation: Provides cryptographic proof of the authorizing entity’s intent, which is critical for audit trails and dispute resolution.

5. Replay Protection: Mitigates the risk of intercepted authorization codes being reused maliciously or replayed.

This transition modernizes domain transfers to meet the security standards established by other critical Internet infrastructure protocols.

1.2. Scope of this Specification

This specification defines the following elements of the AuthCodeSEC protocol:

1. EPP Extension Data Structure: The definition of the XML schema and element placement within the EPP domain:transfer command to transport the signature, signer identity, and metadata.

2. Data Canonicalization and Digest Generation: The specific rules for formatting the transfer data (including domain name, gaining registrar ID, and timestamp) into a canonical string and hashing it with a specific hash algorithm.

3. Asymmetric Cryptographic Authentication: The specific rules for signing the hash of the transfer data with a specific asymmetric cryptographic algorithm.

4. Distribution of Public Keys: The specific rules for distributing the public key to the gaining registrar or other validating parties such as the registry or regulator.

1.3. Current AuthCode Processes

The current [RFC5731] defines the following transfer process:

‘‘‘mermaid sequenceDiagram autonumber

```
participant GC as Gaining Client<br/>(GC)
participant S as EPP Server<br/>(Registry)
participant LC as Losing Client<br/>(LC)

Note over LC,S: Object is sponsored by LC

%% 1. LC maintains AuthInfo
LC->>S: Update object (set / refresh AuthInfo)
S-->>LC: Update OK

Note over LC,GC: Registrant obtains AuthInfo from LC (out-of-band)\nand provides it to GC

%% 2. GC initiates transfer using AuthInfo
GC->>S: Transfer Request (with AuthInfo)
S-->>GC: Transfer Pending\n(AuthInfo validated, transfer created)

%% 3. LC makes decision
alt LC approves
    LC->>S: Transfer Approve
    S-->>LC: Transfer Approved
    Note over S: Sponsorship moves to GC\nTransfer completes immediately
else LC rejects
    LC->>S: Transfer Reject
    S-->>LC: Transfer Rejected
    Note over S: Sponsorship remains with LC
else LC takes no action
    Note over S: Pending period lapses → Auto-approve\nSponsorship moves to GC
end

%% 4. GC may query final status
GC->>S: Transfer Query (optional)
S-->>GC: Final transfer status '''
```

Domain Transfer Request (AuthCode) XML Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <transfer op="request">
      <domain:transfer
       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.com</domain:name>
        <domain:period unit="y">1</domain:period>
        <domain:authInfo>
          <domain:pw roid="JD1234-REP">2fooBAR</domain:pw>
        </domain:authInfo>
      </domain:transfer>
    </transfer>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Key Fields:

* op="request": Indicates we are initiating a transfer.

* domain:period: (Optional) Extension to the registration period upon successful transfer (usually 1 year).

* domain:authInfo: The authorization code provided by the registrant.

* domain:pw: The actual text password (AuthCode).

1.4. Requirements Language

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.

2. Protocol Description

2.1. EPP Extension Data Structure

The EPP extension data structure is defined as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <transfer op="request">
      <domain:transfer
       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.com</domain:name>
        <domain:period unit="y">1</domain:period>
      </domain:transfer>
    </transfer>
  </command>
</epp>
```

We are replacing the <domain:pw> with <domain:ext> to support custom authentication mechanisms (e.g., digital signatures or tokens).

TODO: Define detailed schema for the <domain:ext> element.

2.2. Data Canonicalization and Digest Generation

The following data will be canonicalized and hashed to generate the digest for the signature:

* domain name
* current expiration date
* period
* receiver info
* endorsers info (optional)
* further extensions (optional)

To ensure consistent signature generation and verification, the data fields MUST be canonicalized by concatenating their UTF-8 string representations in the order listed above, separated by a pipe character ("|"). If a field is empty (like the receiver info), it contributes an empty string between delimiters.

The canonicalized string is then hashed using the SHA-256 algorithm to produce the digest that will be signed.

Receiver Info: The identifier of receiver, can be the gaining registrar (e.g., IANA ID or other identifiers e.g. domain name of the registrar). If the authorizing party wishes to restrict the transfer to a specific receiver, this field MUST be populated. If this field

is left empty, the authorization is valid for ANY receiver. This fits the use case like a gaining registrar is not yet determined or disclosed to the losing registrar.

2.3. Asymmetric Cryptographic Authentication

To ensure interoperability and security, this specification mandates the use of specific algorithms while allowing for future extensibility.

* Signing Algorithm: Implementations MUST support ECDSA Curve P-256 with SHA-256. This corresponds to:
  - DNSSEC: IANA DNS Security Algorithm Number 13 (ECDSAP256SHA256) [RFC6605] [IANA-DNS-SEC-ALG].
  - SSL/TLS: IANA TLS 1.3 TLS SignatureScheme 0x0403 (ecdsa_secp256r1_sha256) [RFC8446] [IANA-TLS-PARAMS].
  - JWT: IANA JSON Web Signature and Encryption Algorithms "ES256" (ECDSA using P-256 and SHA-256) [IANA-JOSE].
  - FIDO: FIDO Registry of Predefined Values "ALG_SIGN_SECP256R1_ECDSA_SHA256_DER" (0x0002) [FIDO-REGISTRY].
* Extensibility: The protocol allows specifying other algorithms in the "alg" field for future extensibility. As described in [RFC7518], additional algorithms may be supported.

2.4. Public Key Distribution

The public key is distributed to all verifying parties, such as the gaining registrar, the registry, and the regulator, via the following mechanisms:

* Registrar uses their DNS to publish the public key to the registry and other verifying parties.

3. IANA Considerations

The following IANA considerations are required:

* New hash algorithm registry.
* New signing algorithm registry.

TODO: Define the IANA considerations for how to register the new hash algorithms and signing algorithms.

4. Security Considerations

TODO: Define the security considerations for the protocol.

5. References

5.1. Normative References

[IANA-DNS-SEC-ALG]
IANA, "Domain Name System Security (DNSSEC) Algorithm Numbers", n.d., <https://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml>.

[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>.

[RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <https://www.rfc-editor.org/rfc/rfc5730>.

[RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, DOI 10.17487/RFC5731, August 2009, <https://www.rfc-editor.org/rfc/rfc5731>.

[RFC6605] Hoffman, P. and W.C.A. Wijngaards, "Elliptic Curve Digital Signature Algorithm (DSA) for DNSSEC", RFC 6605, DOI 10.17487/RFC6605, April 2012, <https://www.rfc-editor.org/rfc/rfc6605>.

[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>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/rfc/rfc8446>.

5.2. Informative References

[FIDO-REGISTRY]
FIDO Alliance, "FIDO Registry of Predefined Values", 2 July 2018, <https://fidoalliance.org/specs/fido-v2.0-rd-20180702/fido-registry-v2.0-rd-20180702.html>.

[IANA-JOSE]
IANA, "JSON Web Signature and Encryption Algorithms", n.d., <https://www.iana.org/assignments/jose/jose.xhtml>.

[IANA-TLS-PARAMS]
IANA, "Transport Layer Security (TLS) Parameters", n.d., <https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml>.

[RFC7518] Jones, M., "JSON Web Algorithms (JWA)", RFC 7518, DOI 10.17487/RFC7518, May 2015, <https://www.rfc-editor.org/rfc/rfc7518>.

Author's Address

Zainan (Victor) Zhou
Namefi
Email: zzn@namefi.io