Network File System Version 4
Internet-Draft
Intended status: Standards Track
Expires: 7 August 2026

R. Macklem
FreeBSD
C. Lever, Ed.
Oracle

3 February 2026

Remote Procedure Call Identity Squashing via x.509 Certificate Fields
draft-cel-nfsv4-rpc-tls-othername-02

Abstract

This document extends RPC-with-TLS, as described in [RFC9289], so that a client's x.509 certificate may carry instructions to the RPC server to execute all RPC transactions from that client as a single user identity.

About This Document

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

The latest revision of this draft can be found at https://chucklever.github.io/i-d-rpc-tls-othername/#go.draft-cel-nfsv4-rpc-tls-othername.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-cel-nfsv4-rpc-tls-othername/.

Discussion of this document takes place on the nfsv4 Working Group mailing list (mailto:nfsv4@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/nfsv4/. Subscribe at https://www.ietf.org/mailman/listinfo/nfsv4/.

Source for this draft and an issue tracker can be found at https://github.com/chucklever/i-d-rpc-tls-othername.

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 7 August 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

1.1. Background

The Remote Procedure Call version 2 protocol (RPC, for short) has been a Proposed Standard for three decades (see [RFC5531] and its antecedents). Several important upper layer protocols, such as the family of Network File System protocols (most recently described in [RFC8881] are based on RPC.

In 2022, the IETF published [RFC9289], which specifies a mechanism by which RPC transactions can be cryptographically protected during transit. This protection includes maintaining confidentiality and integrity, and the authentication of the communicating peers.

1.2. Problem Statement

Section 4.2 of [RFC9289] states that:

> RPC user authentication is not affected by the use of transport layer security. When a client presents a TLS peer identity to an RPC server, the protocol extension described in the current document provides no way for the server to know whether that identity represents one RPC user on that client or is shared amongst many RPC users. Therefore, a server implementation cannot utilize the remote TLS peer identity to authenticate RPC users.

Mobile devices such as laptops are typically used by a single user and do not have a fixed, well known IP host address or fully qualified DNS name. The lack of a well known fixed IP host address or fully qualified DNS name weakens the verification checks that may be done on the client's X.509 certificate by the server. As such, this extension allows the client to be restricted to a single user entity on the server, limiting the scope of risk associated with allowing access to the server.

When a service is running in a dedicated VM or container, it often runs as a single assigned user identity. Handling this user identity using Kerberos is problematic, since Kerberos TGTs typically expire in a matter of hours and the service is typically a long running task. This extension allows the client to specify the single assigned user identity to the server in a manner that will not expire for a significant period of time.

When an RPC server replaces incoming RPC user identities with a single user identity, for brevity we refer to this as "identity squashing".

1.3. Summary of Proposed Solution

In the interest of enabling the independent creation of interoperating implementations of RPC identity squashing, this document proposes the use of the x.509 SubjectAltName otherName field to carry a RPC user identity. For these user squashing instructions, this document establishes a fixed object identifier carried in the "type-id" part of the otherName field, and specifies the format of the "value" part of the otherName field when "type-id" carries the new object identifier. The document also provides normative guidance on how the "value" is to be interpreted by RPC servers.

2. 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.

3. x.509 Certificate SubjectAltName Field

As specified in Section 4.2.1.6 of [RFC5280]:

> The subjectAltName MAY carry additional name types through the use of the otherName field. The format and semantics of the name are indicated through the OBJECT IDENTIFIER in the type-id field. The name itself is conveyed as value field in otherName.

A SubjectAltName extension MAY contain multiple entries of different types (e.g., dNSName, iPAddress, otherName). When processing a certificate for identity squashing purposes, the server examines only the otherName entries with type-id values defined in this document. Other SubjectAltName entries are used for their normal purposes (such as hostname verification for TLS).

This document specifies new uses of the otherName field to carry an RPC user identity. The receiving system (an RPC server) then replaces the RPC user, as carried in the RPC header credential and verifier fields in each RPC request within the TLS session, with the user identity specified in the certificate used to authenticate that session.

3.1. Server Processing of otherName Fields

When an RPC server receives a client certificate containing a SubjectAltName extension, it MUST process the otherName fields as follows:

1. The server MUST examine all otherName entries in the SubjectAltName extension.

2. If the server finds an otherName with a type-id that matches one of the identity squashing OIDs defined in this document (id-on-rpcAuthSys, id-on-gssExportedName, or id-on-nfsv4Principal), it SHOULD extract and validate the identity information from that otherName.

3. If multiple identity squashing otherName fields are present in the same SubjectAltName extension, the server MUST reject the certificate to avoid ambiguity. See Section 7 for details.

4. If the server encounters otherName entries with type-id values it does not recognize, it MUST ignore those entries and continue processing. This ensures forward compatibility with future extensions.

5. Other types of SubjectAltName entries (dNSName, iPAddress, etc.) are processed independently and do not affect identity squashing behavior.

The server performs identity squashing only if it successfully validates an identity squashing otherName field and authorizes its use for the authenticated TLS peer.

3.2. Server Processing

This section provides a non-normative example of how an RPC server implementation might process identity squashing otherName fields. Implementers are free to use alternative approaches.

A typical server processing flow might include these steps:

1. During TLS session establishment, extract and validate the client's X.509 certificate according to [RFC5280] and [RFC9289].

2. If the certificate contains a SubjectAltName extension, examine each otherName entry to determine if any contain identity squashing type-id values (id-on-rpcAuthSys, id-on-gssExportedName, or id-on-nfsv4Principal).

3. If exactly one identity squashing otherName is found, extract and parse the identity information according to the ASN.1 definition for that type-id. If parsing fails, reject the certificate.

4. Perform authorization checks to determine whether the authenticated TLS peer is permitted to use the specified identity. This might involve:

   * Consulting an access control list mapping certificate subjects to allowed user identities

   * Verifying that the requested UID/GID values are within acceptable ranges

* Validating that the user@domain string matches expected domain patterns

* Checking that the GSS-API mechanism is trusted and the principal is authorized

5. If authorization succeeds, associate the extracted identity with the TLS session state.

6. For each incoming RPC request on this TLS session, replace the credential information in the RPC header with the identity extracted from the certificate. The original credential information in the RPC header is ignored.

7. Process the RPC request using the squashed identity for all authorization and access control decisions.

Implementations should consider caching the parsed and validated identity information at TLS session establishment time to avoid repeated parsing for each RPC request.

3.3. Interoperability with Non-Supporting Servers

RPC servers that do not implement this specification will not recognize the otherName OIDs defined in this document. Such servers MUST ignore unrecognized otherName entries per Section 4.2.1.6 of [RFC5280]. These servers will process RPC requests using the credential information contained in the RPC header, subject to their normal authentication and authorization policies. This ensures that clients presenting certificates with identity squashing otherName fields can interoperate with servers that do not support this specification, though without identity squashing.

3.4. AUTH_SYS Identities

3.4.1. otherName OID for AUTH_SYS

The otherName OID for AUTH_SYS identities is id-on-rpcAuthSys, defined in Appendix A.

3.4.2. Format of the otherName Value

The otherName value for AUTH_SYS identities contains an RPCAuthSys structure as defined in Appendix A. This structure consists of a 32-bit unsigned integer specifying a numeric UID, and a sequence of 32-bit unsigned integers specifying numeric GIDs.

The use of these integers is further explained in [RFC5531].

3.5. GSS-API Principals

3.5.1. otherName OID for GSS-API Principals

The otherName OID for GSS-API exported names is id-on-gssExportedName, defined in Appendix A.

3.5.2. Format of the otherName Value

The otherName value contains a GSSExportedName structure as defined in Appendix A, consisting of a GSS-API mechanism OID and a mechanism-specific exported name value as described in Section 3.2 of [RFC2743].

3.6. NFSv4 User @ Domain String Identities

3.6.1. otherName OID for String Identities

The otherName OID for NFSv4 user@domain principals is id-on-nfsv4Principal, defined in Appendix A. This principal appears in the same form as an internatialized electronic mail addresses, following the normative rules specified by Section 7.5 of [RFC5280] and its updates.

3.6.2. Format of the otherName Value

The otherName value contains an NFSv4Principal structure as defined in Appendix A, consisting of a UTF-8 encoded user name, the literal "@" character, and a UTF-8 encoded domain name, as described in Section 5.9 of [RFC8881].

4. Extending This Mechanism

It is possible that in the future, RPC servers might implement other forms of RPC user identity, such as Windows Security Identifiers. This section describes how standards action can extend the mechanism specified in this document to accommodate new forms of user identity.

Here, we'll provide the base level of general requirements that must be met, as instructions to future authors. These are to include:

* New identity types must define an ASN.1 module
* Must request IANA OID allocation
* Should provide security considerations specific to that identity type

* Should provide examples and test vectors

5. Client Certificate Generation

This section provides non-normative guidance for Certificate Authorities and administrators who generate client certificates containing identity squashing otherName fields.

5.1. Choosing an Identity Format

The choice of which identity format to use depends on the deployment environment:

RPCAuthSys Appropriate for environments where numeric UIDs and GIDs are the primary form of user identity, such as traditional UNIX/Linux systems. This format is compact but requires that UID/GID mappings be consistent between the certificate and the server's user database.

GSSExportedName Suitable for environments using GSS-API mechanisms like Kerberos. This format provides the strongest integration with existing enterprise authentication infrastructure but requires that servers support the specific GSS-API mechanism indicated by the nameType OID.

NFSv4Principal Recommended for heterogeneous environments or when human-readable identities are preferred. The user@domain format is familiar to administrators and supports internationalization, but requires that servers perform name-to-UID mapping similar to NFSv4 identity mapping.

5.2. Populating Identity Fields

When generating certificates, consider these guidelines:

UID/GID values Ensure that the numeric values in RPCAuthSys correspond to valid entries in the server's user database. Avoid using privileged UIDs (such as 0 for root) unless there is a specific operational requirement and strong authorization controls are in place.

GSS-API exported names The nameValue field should contain a properly formatted exported name token as defined by the specific GSS-API mechanism. For Kerberos, this follows the format specified in [RFC4121]. Consult the mechanism specification for proper encoding.

User@domain strings Both the user and domain components should be

UTF-8 encoded. Domain names should typically match the DNS domain under which the server operates. International domain names should be encoded in UTF-8, not in Punycode (ACE) form.

5.3. Certificate Validity Period

Certificates containing identity squashing otherName fields grant access to server resources under a specific user identity. Administrators should consider appropriate validity periods based on their security requirements. Shorter validity periods reduce the window of exposure if a certificate is compromised, but may increase operational overhead for certificate renewal.

The choice of validity period might also consider whether certificate revocation checking (CRL or OCSP) is deployed and how quickly revocation information propagates in the environment.

6. Implementation Status

| RFC Editor: This section is to be removed before publishing
| this document as an RFC.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs.

Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

6.1. FreeBSD NFS Server and Client

Organization: FreeBSD

URL: https://www.freebsd.org (https://www.freebsd.org)

Maturity: Complete.

Coverage: The mechanism to represent user@domain strings has been implemented using an OID from the FreeBSD arc.

Licensing: BSD 3-clause

Implementation experience: None to report

7. Security Considerations

7.1. General Security Considerations

The security considerations for RPC-with-TLS described in Section 8 of [RFC9289] apply to this specification. In particular, the discussion about certificate validation, trust anchors, and the establishment of secure TLS sessions remains relevant.

7.2. Identity Squashing and Authorization

This specification enables a client to request that all RPC operations within a TLS session be executed under a single user identity specified in the client's X.509 certificate. This "identity squashing" mechanism has several security implications:

7.2.1. Trust in the Certificate Authority

The server MUST carefully consider which Certificate Authorities (CAs) it trusts to issue certificates containing the otherName extensions defined in this document. A compromised or malicious CA could issue certificates that allow unauthorized access to server resources under arbitrary user identities.

Servers SHOULD maintain separate trust anchors for certificates containing identity squashing otherName fields versus certificates used solely for TLS peer authentication. This allows administrators to tightly control which CAs are authorized to assert user identities.

7.2.2. Authorization Decisions

The presence of an otherName field specifying a user identity does not by itself grant any authorization. Servers MUST perform their normal authorization checks to determine whether the requested identity is permitted for the authenticated TLS peer.

For example, a server might maintain an access control list mapping certificate subjects or distinguished names to the set of user identities they are permitted to assume. Only if such authorization succeeds should the server execute RPC operations under the specified identity.

7.2.3. Name Canonicalization

7.2.3.1. NFSv4 Principals

When processing NFSv4Principal otherName values, servers MUST apply the same name canonicalization and domain validation procedures described in Section 5.9 of [RFC8881]. In particular:

* Domain names SHOULD be validated against expected domain suffixes
* Internationalized domain names MUST be properly normalized
* Case-sensitivity rules for usernames and domains MUST be consistently applied

7.2.3.2. GSS-API Exported Names

When processing GSSExportedName otherName values, servers MUST verify that:

* The mechanism OID in the nameType field corresponds to a GSS-API mechanism the server supports and trusts
* The nameValue field conforms to the exported name format defined by that specific GSS-API mechanism
* The mechanism-specific name validation and canonicalization procedures are followed

Servers SHOULD NOT accept exported names from GSS-API mechanisms they do not fully support, as improper name handling could lead to authorization bypass vulnerabilities.

7.2.3.3. AUTH_SYS Credentials

When processing RPCAuthSys otherName values, servers MUST:

* Validate that the UID and GIDs fall within acceptable ranges for the local system’s user database
* Verify that the UID corresponds to a valid user account
* Confirm that the GIDs represent valid groups and that the user is authorized to be a member of those groups

Servers SHOULD reject certificates containing UID 0 (root) or other privileged UIDs unless there is an explicit and well-justified operational requirement, and additional strong authorization controls are in place.

7.3. Session Binding

All RPC operations within a TLS session containing an identity squashing otherName execute under the same user identity. Servers MUST ensure that session state cannot be hijacked or transferred between different TLS sessions, as this could allow an attacker to gain the privileges associated with the squashed identity.

7.4. Revocation

Servers SHOULD support certificate revocation checking (via CRL, OCSP, or similar mechanisms) for certificates containing identity squashing otherName fields. Since these certificates grant user-level access to server resources, timely revocation is critical when a certificate is compromised or a user’s access should be terminated.

7.5. Privacy Considerations

The otherName fields defined in this specification reveal user identity information in the client’s X.509 certificate. This information is transmitted during the TLS handshake and may be visible to network observers if the handshake is not properly protected.

While TLS 1.3 encrypts most of the handshake including certificates, earlier TLS versions may expose this information. Deployments concerned about privacy SHOULD use TLS 1.3 or later.

7.6. Multiple Identity Formats

Implementations MUST NOT allow multiple identity squashing otherName fields to be present simultaneously in the same SubjectAltName extension. If multiple such fields are present (e.g., both RPCAuthSys and NFSv4Principal), the server MUST reject the certificate to avoid ambiguity about which identity should be used.

8. IANA Considerations

8.1. SMI Security for PKIX Module Identifier

IANA is requested to assign an object identifier for the ASN.1 module specified in this document in the "SMI Security for PKIX Module Identifier" registry (1.3.6.1.5.5.7.0):

Decimal	Description	References
TBD1	id-mod-rpc-tls-identity-squashing	RFC-TBD

Table 1

8.2. SMI Security for PKIX Other Name Forms

IANA is requested to assign three object identifiers for the otherName types specified in this document in the "SMI Security for PKIX Other Name Forms" registry (1.3.6.1.5.5.7.8):

Decimal	Description	References
TBD4	id-on-rpcAuthSys	RFC-TBD
TBD5	id-on-gssExportedName	RFC-TBD
TBD6	id-on-nfsv4Principal	RFC-TBD

Table 2

These otherName identifiers are used in the SubjectAltName extension of X.509 certificates to carry RPC user identity information for the purpose of identity squashing as described in this document.

"RFC-TBD" is to be replaced with the actual RFC number when this document is published.

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>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/rfc/rfc5280>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/rfc/rfc7942>.

[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>.

[RFC9289] Myklebust, T. and C. Lever, Ed., "Towards Remote Procedure Call Encryption by Default", RFC 9289, DOI 10.17487/RFC9289, September 2022, <https://www.rfc-editor.org/rfc/rfc9289>.

9.2. Informative References

[RFC2743] Linn, J., "Generic Security Service Application Program Interface Version 2, Update 1", RFC 2743, DOI 10.17487/RFC2743, January 2000, <https://www.rfc-editor.org/rfc/rfc2743>.

[RFC4121] Zhu, L., Jaganathan, K., and S. Hartman, "The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2", RFC 4121, DOI 10.17487/RFC4121, July 2005, <https://www.rfc-editor.org/rfc/rfc4121>.

[RFC5531] Thurlow, R., "RPC: Remote Procedure Call Protocol Specification Version 2", RFC 5531, DOI 10.17487/RFC5531, May 2009, <https://www.rfc-editor.org/rfc/rfc5531>.

[RFC5912] Hoffman, P. and J. Schaad, "New ASN.1 Modules for the Public Key Infrastructure Using X.509 (PKIX)", RFC 5912, DOI 10.17487/RFC5912, June 2010, <https://www.rfc-editor.org/rfc/rfc5912>.

[RFC8881] Noveck, D., Ed. and C. Lever, "Network File System (NFS) Version 4 Minor Version 1 Protocol", RFC 8881, DOI 10.17487/RFC8881, August 2020, <https://www.rfc-editor.org/rfc/rfc8881>.

Appendix A. ASN.1 Module

The following ASN.1 module normatively specifies the structure of the new otherName values described in this document. This specification uses the ASN.1 definitions from [RFC5912] with the 2002 ASN.1 notation used in that document. [RFC5912] updates normative documents using older ASN.1 notation.

A.1. RPC TLS Identity Squashing Module

```
RPCTLSIdentitySquashing
    { iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0)
      id-mod-rpc-tls-identity-squashing(TBD) }

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

IMPORTS
    OTHER-NAME
    FROM PKIX1Implicit-2009
        { iso(1) identified-organization(3) dod(6) internet(1)
          security(5) mechanisms(5) pkix(7) id-mod(0)
          id-mod-pkix1-implicit-02(59) } ;

-- Object Identifier Arc
id-pkix OBJECT IDENTIFIER ::=
    { iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) }

id-on OBJECT IDENTIFIER ::= { id-pkix 8 }  -- other names

-- ================================================================
-- RPC AUTH_SYS Identity Squashing
-- ================================================================

-- OID for RPC AUTH_SYS credentials in otherName
id-on-rpcAuthSys OBJECT IDENTIFIER ::= { id-on TBD }

-- RPC AUTH_SYS Credentials Structure
-- UID and GID list as used in RPC AUTH_SYS authentication flavor
-- See RFC 5531 (ONC RPC) and related specifications
RPCAuthSys ::= SEQUENCE {
    uid        INTEGER (0..4294967295),  -- 32-bit UID
    gids       SEQUENCE OF INTEGER (0..4294967295)  -- List of 32-bit GIDs
}

-- For use in SubjectAltName otherName
rpcAuthSys OTHER-NAME ::= {
    RPCAuthSys IDENTIFIED BY id-on-rpcAuthSys
}

-- ================================================================
-- GSS-API Exported Name Identity Squashing
-- ================================================================
```

```
   -- OID for GSS-API Exported Name in otherName
   id-on-gssExportedName OBJECT IDENTIFIER ::= { id-on TBD }

   -- GSS-API Exported Name Structure
   -- As defined in RFC 2743 Section 3.2
   GSSExportedName ::= SEQUENCE {
       nameType   OBJECT IDENTIFIER,  -- GSS-API mechanism OID
       nameValue  OCTET STRING        -- Mechanism-specific exported name
   }

   -- For use in SubjectAltName otherName
   gssExportedName OTHER-NAME ::= {
       GSSExportedName IDENTIFIED BY id-on-gssExportedName
   }

   -- ================================================================
   -- NFSv4 User@Domain Principal Identity Squashing
   -- ================================================================

   -- OID for NFSv4 user@domain principal in otherName
   id-on-nfsv4Principal OBJECT IDENTIFIER ::= { id-on TBD }

   -- NFSv4 User@Domain Principal Structure
   -- As defined in RFC 8881 Section 5.9
   NFSv4Principal ::= SEQUENCE {
       principal  UTF8String          -- user@domain string
   }

   -- For use in SubjectAltName otherName
   nfsv4Principal OTHER-NAME ::= {
       NFSv4Principal IDENTIFIED BY id-on-nfsv4Principal
   }

   END
```

Appendix B. Certificate Examples

This appendix provides examples of X.509 certificates containing the otherName extensions defined in this document. These examples are provided in both human-readable notation and hexadecimal DER encoding to assist implementers in verifying their implementations.

B.1. NFSv4 Principal Example

This example shows a certificate for user "alice" at domain "nfs.example.com":

```
SubjectAltName ::= SEQUENCE {
    otherName [0] IMPLICIT SEQUENCE {
        type-id OBJECT IDENTIFIER ::= id-on-nfsv4Principal,
        value [0] EXPLICIT NFSv4Principal ::= {
            user "alice",
            atSign "@",
            domain "nfs.example.com"
        }
    }
}
```

DER encoding (hexadecimal):

```
30 2B A0 29 06 08 2B 06 01 05 05 07 08 XX A0 1D
0C 05 61 6C 69 63 65 13 01 40 0C 0F 6E 66 73 2E
65 78 61 6D 70 6C 65 2E 63 6F 6D
```

Note: XX represents the TBD value for id-on-nfsv4Principal.

B.2. GSS-API Exported Name Example

This example shows a certificate containing a Kerberos V5 principal for "bob@EXAMPLE.COM":

```
SubjectAltName ::= SEQUENCE {
    otherName [0] IMPLICIT SEQUENCE {
        type-id OBJECT IDENTIFIER ::= id-on-gssExportedName,
        value [0] EXPLICIT GSSExportedName ::= {
            nameType 1.2.840.113554.1.2.2,  -- Kerberos V5
            nameValue '04 01 00 0B 06 09 2A 86 48 86 F7 12 01 02 02
                       00 00 00 11 62 6F 62 40 45 58 41 4D 50 4C 45
                       2E 43 4F 4D'H
        }
    }
}
```

DER encoding (hexadecimal):

```
30 47 A0 45 06 08 2B 06 01 05 05 07 08 YY A0 39
30 37 06 09 2A 86 48 86 F7 12 01 02 02 04 2A 04
01 00 0B 06 09 2A 86 48 86 F7 12 01 02 02 00 00
00 11 62 6F 62 40 45 58 41 4D 50 4C 45 2E 43 4F
4D
```

Note: YY represents the TBD value for id-on-gssExportedName.

The nameValue field contains the GSS-API exported name token format as defined by the Kerberos V5 mechanism. The first four bytes (04 01 00 0B) are the token ID and length fields defined in Section 3.2 of [RFC2743].

B.3. RPC AUTH_SYS Example

This example shows a certificate containing UID 1000 and GIDs 1000, 10, and 100:

```
SubjectAltName ::= SEQUENCE {
    otherName [0] IMPLICIT SEQUENCE {
        type-id OBJECT IDENTIFIER ::= id-on-rpcAuthSys,
        value [0] EXPLICIT RPCAuthSys ::= {
            uid 1000,
            gids { 1000, 10, 100 }
        }
    }
}
```

DER encoding (hexadecimal):

```
30 20 A0 1E 06 08 2B 06 01 05 05 07 08 ZZ A0 12
30 10 02 02 03 E8 30 0A 02 02 03 E8 02 01 0A 02
01 64
```

Note: ZZ represents the TBD value for id-on-rpcAuthSys.

Breaking down the encoding: - 02 02 03 E8: INTEGER 1000 (UID) - 30 0A: SEQUENCE OF (GIDs) - 02 02 03 E8: INTEGER 1000 - 02 01 0A: INTEGER 10 - 02 01 64: INTEGER 100

B.4. Complete Certificate Example

This example shows a minimal self-signed certificate containing an NFSv4Principal otherName. Line breaks and whitespace have been added for readability:

```
-----BEGIN CERTIFICATE-----
MIICXzCCAcigAwIBAgIUAbCdEfG7KH0FjLbI8N9cJQqQoLwwDQYJKoZIhvcNAQEL
BQAwRDELMAkGA1UEBhMCVVMxEzARBgNVBAgMCkNhbGlmb3JuaWExDzANBgNVBAcM
BklydmluZTEPMA0GA1UECgwGT3JhY2xlMB4XDTI1MDEwMTAwMDAwMFoXDTI2MDEw
MTAwMDAwMFowRDELMAkGA1UEBhMCVVMxEzARBgNVBAgMCkNhbGlmb3JuaWExDzAN
BgNVBAcMBklydmluZTEPMA0GA1UECgwGT3JhY2xlMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQC7VJTUt9Us8cKjMzEfYyjiWA4R4ypbHqGC0H0+tG3hGbN3MYHa
... [additional base64-encoded certificate data] ...
oxUwEwYDVR0lBAwwCgYIKwYBBQUHAwEwKwYDVR0RBCQwIqAfBggrBgEFBQcIAKAT
DBVhbGljZUBuZnMuZXhhbXBsZS5jb20wDQYJKoZIhvcNAQELBQADgYEAk3+...
-----END CERTIFICATE-----
```

The SubjectAltName extension in this certificate is encoded at the position indicated by the bytes following the Extended Key Usage extension.

B.5. Internationalized Domain Name Example

This example shows an NFSv4Principal with internationalized characters:

```
SubjectAltName ::= SEQUENCE {
    otherName [0] IMPLICIT SEQUENCE {
        type-id OBJECT IDENTIFIER ::= id-on-nfsv4Principal,
        value [0] EXPLICIT NFSv4Principal ::= {
            user "逕ィ謌キ",        -- Chinese characters for "user"
            atSign "@",
            domain "萓九∴.jp"    -- Japanese IDN
        }
    }
}
```

DER encoding (hexadecimal):

```
30 2D A0 2B 06 08 2B 06 01 05 05 07 08 XX A0 1F
0C 06 E7 94 A8 E6 88 B7 13 01 40 0C 0C E4 BE 8B
E3 81 88 2E 6A 70
```

Note: The UTF-8 encoding of the Chinese characters "逕ィ謌キ" is E7 94 A8 E6 88 B7, and the Japanese text "萓九∴" is E4 BE 8B E3 81 88.

B.6. Test Vectors

This section provides test vectors for validating implementations. Each test case includes the input values, expected ASN.1 structure, and expected DER encoding.

B.6.1. Valid NFSv4Principal Test Cases

Test Case 1: Simple ASCII user and domain

Input:

* user: "bob"

* domain: "example.org"

Expected DER encoding:

```
30 22 A0 20 06 08 2B 06 01 05 05 07 08 XX A0 14
0C 03 62 6F 62 13 01 40 0C 0B 65 78 61 6D 70 6C
65 2E 6F 72 67
```

Test Case 2: User with numbers and domain with subdomain

Input:

* user: "user123"

* domain: "nfs.lab.example.com"

Expected DER encoding:

```
30 2F A0 2D 06 08 2B 06 01 05 05 07 08 XX A0 21
0C 07 75 73 65 72 31 32 33 13 01 40 0C 14 6E 66
73 2E 6C 61 62 2E 65 78 61 6D 70 6C 65 2E 63 6F
6D
```

B.6.2. Valid RPCAuthSys Test Cases

Test Case 1: Single user, single group

Input:

* uid: 1000

* gids: { 1000 }

Expected DER encoding:

```
30 13 A0 11 06 08 2B 06 01 05 05 07 08 ZZ A0 05
30 08 02 02 03 E8 30 04 02 02 03 E8
```

Test Case 2: User with empty group list

Input:

* uid: 500

* gids: (empty)

Expected DER encoding:

```
30 0F A0 0D 06 08 2B 06 01 05 05 07 08 ZZ A0 01
30 06 02 02 01 F4 30 00
```

Test Case 3: User with maximum 32-bit UID and multiple groups

Input:

* uid: 4294967295

* gids: { 1, 10, 100, 1000 }

Expected DER encoding:

```
30 24 A0 22 06 08 2B 06 01 05 05 07 08 ZZ A0 16
30 14 02 05 00 FF FF FF FF 30 0B 02 01 01 02 01
0A 02 01 64 02 02 03 E8
```

B.6.3. Invalid Test Cases

These test cases should be rejected by conforming implementations:

Test Case 1: NFSv4Principal with missing atSign field

Input (malformed):

* user: "alice"

* atSign: "" (empty)

* domain: "example.com"

Expected result: Parsing failure

Test Case 2: RPCAuthSys with UID exceeding 32-bit range

Input (malformed):

* uid: 4294967296 (2^32)

* gids: { 1000 }

Expected result: Encoding failure or rejection

Test Case 3: Certificate with multiple identity squashing otherNames

Input (malformed): SubjectAltName containing both: - id-on-nfsv4Principal with user "alice@example.com" - id-on-rpcAuthSys with uid 1000

Expected result: Certificate rejection per Security Considerations

Acknowledgments

The authors are grateful to Jeff Layton, Greg Marsden, and Martin Thomson for their input and support.

Special thanks to Area Director Gorry Fairhurst, NFSV4 Working Group Chairs Brian Pawlowski and Christopher Inacio, and NFSV4 Working Group Secretary Thomas Haynes for their guidance and oversight.

Authors' Addresses

Rick Macklem
FreeBSD Project
Canada
Email: rmacklem@uoguelph.ca

Chuck Lever (editor)
Oracle Corporation
United States of America
Email: chuck.lever@oracle.com