# Update on Kerberos Extensibility

`draft-ietf-krb-wg-rfc1510ter-00.txt`

Tom Yu

IETF 62

# Typed Hole Assignments

- Relative OID vs absolute OID
- Self-assignment of absolute OID
- Number assignment policy?
- Which holes should allow private use numbers?
- What things not IANA?
- AuthorizationData special (only allow private uses if enclosed in AD-IF-RELEVANT?)
- Transited only by standards action?

# String Types

- TicketExt should only contain UTF-8?
- Ext types should only contain UTF-8?
- Exception for KDC-REQ-EXT to use IA5, request Ticket1510?

# ASN.1 Readability Changes?

- String type as parameter?
- Derive 1510/ext types from common types at all?
- How many constraints expressed in ASN.1 vs text?
  - KDC-REQ most complex
  - Stop using WITH COMPONENTS to “reach into” structures?
- Drop advisory parameters? (EncryptedData, Checksum, etc.)
- “Signed” still useful

# Existing Ticket, TicketCommon

```
Ticket          ::= CHOICE {
    rfc1510     [APPLICATION 1] Ticket1510,
    ext         [APPLICATION 4] Signed {
        TicketExt, { key-server }, { ku-Ticket-cksum }
    }
}
TicketCommon { EncPart } ::= SEQUENCE {
    tkt-vno     [0] INTEGER (5),
    realm       [1] Realm,
    sname       [2] PrincipalName,
    enc-part    [3] EncryptedData {
        EncPart, { key-server }, { ku-Ticket }
    },
    ...,
    extensions          [4] TicketExtensions OPTIONAL,
    ...
}
```

# Existing Ticket1510, TicketExt

```
Ticket1510 ::= SEQUENCE {
    COMPONENTS OF TicketCommon { EncTicketPart1510 }
} (WITH COMPONENTS {
    ...,
    -- explicitly force IA5 in strings
    realm (RealmIA5),
    sname (PrincipalNameIA5)
})
TicketExt ::= [APPLICATION 4] TicketCommon {
    EncTicketPartExt
} (WITH COMPONENTS {
    ...,
    -- explicitly force UTF-8 in strings
    realm (RealmExt),
    sname (PrincipalNameExt)
})
```

# Parameterized Realm, PrincipalName

```
Realm { StrType }       ::= KerberosString (StrType)
RealmIA5                ::= Realm { KerberosStringIA5 }
RealmExt                ::= Realm { KerberosStringExt }

PrincipalName { StrType } ::= SEQUENCE {
    name-type   [0] NameType,
    name-string [1] SEQUENCE OF KerberosString (StrType)
}
PrincipalNameIA5 ::= PrincipalName { KerberosStringIA5 }
PrincipalNameExt ::= PrincipalName { KerberosStringExt }
```

# TicketCommon with string parameter

```
Ticket           ::= CHOICE {
    rfc1510      Ticket1510,
    ext          TicketExt
}
TicketCommon { StrType, EncPart } ::= SEQUENCE {
    tkt-vno      [0] INTEGER (5),
    realm        [1] Realm { StrType },
    sname        [2] PrincipalName { StrType },
    enc-part     [3] EncryptedData {
        EncPart, { key-server }, { ku-Ticket }
    },
    ...,
    extensions   [4] TicketExtensions OPTIONAL,
    ...
}
```

# Ticket1510, TicketExt using string parameter

```
Ticket1510 ::= [APPLICATION 1] SEQUENCE {
    COMPONENTS OF TicketCommon {
        KerberosStringIA5, EncTicketPart1510
    }
}
TicketExt ::= [APPLICATION 4] Signed {
    [APPLICATION 4] TicketCommon {
        KerberosStringExt, EncTicketPartExt
    },
    { key-server }, { ku-Ticket-cksum }
}
```

# Separated Ticket1510, TicketExt

```
Ticket1510 ::= [APPLICATION 1] SEQUENCE {
    tkt-vno     [0] INTEGER (5),
    realm       [1] RealmIA5,
    sname       [2] PrincipalNameIA5,
    enc-part    [3] EncryptedData {
        EncTicketPart1510, { key-server }, { ku-Ticket }
    }
}

TicketExt ::= [APPLICATION 4] Signed {
    [APPLICATION 4] SEQUENCE {
        tkt-vno     [0] INTEGER (5),
        realm       [1] RealmExt,
        sname       [2] PrincipalNameExt,
        enc-part    [3] EncryptedData {
            EncTicketPartExt, { key-server }, { ku-Ticket }
        },
        ...,
        extensions  [4] TicketExtensions OPTIONAL,
        ...
    }, { key-server }, { ku-Ticket-cksum }
}
```

# EncryptedData pulled out

```
Ticket1510 ::= [APPLICATION 1] SEQUENCE {
    tkt-vno     [0] INTEGER (5),
    realm       [1] RealmIA5,
    sname       [2] PrincipalNameIA5,
    enc-part    [3] EncTicketPart1510Cipher
}
EncTicketPart1510Cipher ::= EncryptedData {
    EncTicketPart1510, { key-server }, { ku-Ticket }
}

TicketExt ::= [APPLICATION 4] Signed {
    [APPLICATION 4] SEQUENCE {
        tkt-vno     [0] INTEGER (5),
        realm       [1] RealmExt,
        sname       [2] PrincipalNameExt,
        enc-part    [3] ,
        ...,
        extensions  [4] TicketExtensions OPTIONAL,
        ...
    }, { key-server }, { ku-Ticket-cksum }
}
EncTicketPartExtCipher ::= EncryptedData {
    EncTicketPartExt, { key-server }, { ku-Ticket }
}
```

# Future Plans

- Clarify capability negotiation
- Resolve IANA assignment policies
- ASN.1 module reworking