NETWORK WORKING GROUP
Internet-Draft
Intended status: Standards Track
Expires: May 7, 2009

L. Zhu
Microsoft Corporation
J. Altman
Secure Endpoints
N. Williams
Sun
November 3, 2008

Public Key Cryptography Based User-to-User Authentication - (PKU2U)
draft-zhu-pku2u-09

Status of this Memo

Abstract

This document defines a Generic Security Services Application Program Interface (GSS-API) mechanism based on Public Key Infrastructure (PKI) - PKU2U. This mechanism is based on Kerberos V messages and the Kerberos V GSS-API mechanism, but without requiring a Kerberos Key Distribution Center (KDC).

Table of Contents

1. Introduction

The Generic Security Services Application Programming Interface (GSS-API) is a generic protocol and API for providing authentication and session protection to applications. It is generic in that it supports multiple authentication mechanisms. Today there exists only one workable, widely deployed, standards-track GSS-API mechanism: the Kerberos V GSS-API mechanism [RFC1964] [RFC4121], which is based on Kerberos V [RFC4120]. There is a need to provide a GSS-API mechanism which does not require Kerberos V Key Distribution Center (KDC) infrastructure, and which supports the use of public key cryptography, particularly Public Key Infrastructure (PKI) [RFC5280], including the use of public key certificates without a PKI.

This document specifies such a mechanism: the Public Key User to User mechanism (PKU2U).

PKU2U is based on building blocks taken from Kerberos V [RFC4120], PKINIT, [RFC4556] (which in turn uses PKI [RFC5280]) building blocks), and the Kerberos V GSS-API mechanism [RFC1964] [RFC4121]. In spite of using Kerberos V building blocks, PKU2U does not require any Kerberos V KDC infrastructure. And though PKU2U also uses PKI building blocks, PKU2U can be used without a PKI by pre-sharing certificates and/or pre-associating name/certificate bindings.

Therefore PKU2U can be used for true peer-to-peer authentication, as well as for PKI-based authentication.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

In this document, the GSS-API initiator or acceptor is referred to as the peer when the description is applicable to both the initiator and the acceptor.

3. The PKU2U Realm Name

The PKU2U realm name is defined as a reserved Kerberos realm name per [KRB-NAMING], and it has the value of "WELLKNOWN:PKU2U".

The PKU2U realm name has no meaning, but is intended to be used in the Kebreros V Protocol Data Units (PDUs) that are re-used by PKU2U wherever realm names are needed. Unless otherwise specified, the

realm name in any Kerberos message used by PKU2U is the PKU2U realm name.

4. The NULL Principal Name

The NULL Kerberos principal name is defined as a well-known Kerberos principal name based on [KRB-NAMING]. The value of the name-type field is KRB_NT_WELLKNOWN [KRB-NAMING], and the value of the name-string field is a sequence of two KerberosString components: "WELLKNOWN", "NULL".

The NULL Kerberos principal name is used in the Kerberos messages where there is no Kerberos representation of the principal name, for example, when the client name is a Distinguished Name. When the NULL principal name is used in the Kerberos messages, the principal name is either not used or provided separately (for example in the PA_PKU2U_NAME padata defined in Section 6.1).

5. PKU2U Principal Naming

The GSS-API targ_name supplied for the initiator MUST NOT be GSS_C_NO_NAME in PKU2U.

PKU2U principal names can be Kerberos principal names, and they can also be distiguished names, or subject alternative names [RFC5280] as they appear in the certificate of any PKU2U peer, as well as any names agreed to out of band that do not appear in the peer certificates.

Certificates may be associated with multiple principal names. This presents problems for the GSS-API bindings of a PKI-based mechanism because, for example, for any given, established GSS-API security mechanism there can be only one initiator name, and one acceptor name, and credential handles may be associated with only one name. We resolve these problems as follows:

- We define multiple GSS-API name types corresponding to several GeneralName choices [RFC5280], along with syntaxes, display forms, and exported name token formats for each. For most of the name-types listed below the exported name token formats consists of a GeneralName with the usual exported name token header as per-RFC2743. Two name-types are shared with the Kerberos V mechanism and use the Kerberos V mechanism's query and display syntaxes, canonicalization rules, and exported name token format.

- The cred_name of a credential handle acquired with GSS_C_NT_NO_NAME as the desired_name SHOULD be the Distinguished Name (DN) of the certificate underlying the credential. If there are multiple certificates and private keys, then either one MUST be selected by local, implementation-specific means, or credential acquisition with GSS_C_NT_NO_NAME MUST fail (implementers may choose which of these two behaviors to provide).

- When using desired_name values other than GSS_C_NT_NO_NAME for credential handle acquisition then the implementation MUST use exact matching of the given desired_name to a certificate's DN or Subject Alternative Names (SANs) for all name-types given below, except for GSS_C_NT_DN, where matching rules are fuzzier and given below. The names of a X.509 certificate will be compared to the given desired_name in this order: certificate DN first, then SANs in the order in which they appear in the certificate. When multiple certificates and private keys are available the order in which the various certificates are searched is significant; no canonical certificate collation order is defined herein.

- The cred_name of a credential object acquired with a desired_name other than GSS_C_NT_NO_NAME MUST be equal to the certificate DN or SAN matched by the given desired_name.

- We provide a method (see below) by which initiators can assert, in their context tokens, one of these names of the initiator. We also provide a method of asserting names that do not appear in a X.509 certificate, in which case the binding of X.509 certificate to the asserted name is done out-of band. The name to be asserted, of course, is the cred_name of the cred_handle passed to GSS_Init_sec_context().

- The initiator's context tokens may also indicate what is the expected name of the acceptor -- the targ_name passed in to GSS_Init_sec_context().

- No attempt is made to map Kerberos V realm names to trust anchor certificate authority (CA) names.

- We provide a method of matching host-based service principal names to acceptor certificates, so that: a) initiators need not know the particulars of an acceptor's certificates' names a priori, b) acceptors can select a credential to accept a security context with that the initiator will accept, c) existing certificates for web servers, may be used as host-based service principal names as though the service name were "HTTP".

Thus GSS-API initiator applications that use the GSS_C_NO_NAME as the

desired_name arguments of GSS_Acquire_cred() and GSS_Add_cred(), or GSS_C_NO_CREDENTIAL as the cred argument of GSS_Init_sec_context() will assert the selected X.509 certificate's subject DN, and that X.509 certificate's subject DN will be the name returned by GSS_Inquire_cred() and GSS_Inquire_cred_by_mech().

And portable GSS-API initiator applications using GSS_C_NT_HOSTBASED_SERVICE for naming acceptors (i.e., for importing a name to use as the targ_name input argument of GSS_Init_sec_context()) will have a reasonable chance of success in authenticating peers with X.509 certificates predating this specification.

5.1. GSS_C_NT_DN

The name type GSS_C_NT_DN, with Object Identifier (OID) <TBD> (see Section 10), is defined. This corresponds to the 'directoryName' choice of the 'GeneralName' Abstract Syntax Notation One (ASN.1) [CCITT.X680.2002] type defined in [RFC5280].

The query syntax and display form for names of this type SHALL be as described in [RFC4514].

As RFC4514 says, "[c]omparison of DNs for equality is to be performed in accordance with the distinguishedNameMatch matching rule [RFC4517]".

There is no reasonable way to canonicalize names of this type without providing certificates to match query forms of GSS_C_NT_DN against, such as in the form of a directory. Therefore GSS_Canonicalize_name() as applied to names imported with the GSS_C_NT_DN name-type MUST search available certificate databases, or directories, or MUST fail. No method of locating and searching directories for matching certificate DNs is specified herein. Note though that GSS_Inquire_cred_by_mech() and GSS_Inquire_sec_context() can and, indeed, MUST return "mechanism names" (MN) (see [RFC2743]).

The exported name token format for names of this type SHALL be the Distinguished Encoding Rules (DER) [CCITT.X680.2002] [CCITT.X690.2002] encoding of a GeneralName with directoryName as the choice.

Implementation support for this name type is REQUIRED.

5.2. GSS_C_NT_HOSTNAME

The name type GSS_C_NT_HOSTNAME, with OID <TBD>, is defined. This corresponds to the 'dNSName' choice of the 'GeneralName' ASN.1 type

defined in [RFC5280].

The query syntax for names of this type SHALL be a DNS name [RFC1034] in either ACE or Unicode form [RFC3490].

The display and canonical form of names of this type SHALL be a DNS domain name in ACE form, with character case folded down. Canonicalization consists merely of applying the ToASCII() function and case-folding the result.

The exported name token format for names of this type SHALL be the DER encoding of a GeneralName with dNSName as the choice and the DNS domain name in ACE form and case folded down.

Implementation support for this name type is OPTIONAL.

5.3. GSS_C_NT_EMAIL_ADDR

The name type GSS_C_NT_EMAIL_ADDR, with OID <TBD>, is defined. This corresponds to the 'rfc822Name' choice of the 'GeneralName' ASN.1 type defined in [RFC5280].

The query syntax and display form for names of this type SHALL be the text representation of an 'addr-spec' as defined in [RFC0822].

The canonical form of names of this type SHALL be the query form with case folded down. Note that the domain name part of an addr-spec is a "domain name slot" and so the canonicalization rules for GSS_C_NT_HOSTNAME given above apply here as well.

The exported name token form for this name type SHALL be the DER-encoding of a GeneralName with the rfc822Name choice.

Implementation support for this name type is OPTIONAL.

5.4. GSS_KRB5_NT_PRINCIPAL_NAME

PKU2U supports the use of GSS_KRB5_NT_PRINCIPAL_NAME names [RFC1964].

The query, display, canonical and exported name token forms of names of this type SHALL be as specified in RFC4121. The realm name part of GSS_KRB5_NT_PRINCIPAL_NAME names is optional for the query syntax; when canonicalized, names of this type lacking a realm name will have the well-known PKU2U realm name affixed.

When the realm name of a GSS_KRB5_NT_PRINCIPAL_NAME NAME is defaulted or otherwise is the well-known PKU2U realm name, then the "cname"' or sname fields of the Kerberos V PDUs used to construct PKU2U security

context tokens MUST be set to the principal name part of the given NAME. Otherwise the PA_PKU2U_NAME pre-authentication data MUST be used to indicate a name of id-pkinit-san type [RFC4556] corresponding to the given NAME. See Section 5.4.

No attempt is made to map Kerberos V realm names to trust anchor certificate authority (CA) names.

Note that having more than one mechanism share name-types has implications for multi-mechanism, pluggable GSS-API implementations (commonly referred to as "mechglue"). Specifically:

- It must be the responsibility of the mechanism, not of the mechglue, to ensure that the standard exported name token header (which includes a mechanism OID), is included in exported name tokens. The exported name token for a GSS_KRB5_NT_PRINCIPAL_NAME MN produced by PKU2U would have PKU2U's mechanism OID in the header.

- A pluggable mechglue must be able to find a mechanism that can import an exported name token if an available mechanism can produce that exported name token. For example, a pluggable mechglue where PKU2U is available but where the Kerberos V mechanism [RFC1964] is not should still be able to import exported Kerberos V name tokens since PKU2U can create such tokens. One way to do this would be for the mechglue to try the mechanism named in the exported name token header, if it is available, else try all other available mechanisms until one succeeds or all fail. It would be reasonable for a mechglue implementer to require that the Kerberos V mechanism be available if PKU2U is too.

- It must be possible for GSS_Acquire_cred(), GSS_Add_cred() to use a Kerberos V "mechanism name" (MN; see [RFC2743]) as desired_name argument value to acquire a PKU2U credential. Similarly, it must be possible to use a Kerberos V MN as the target_name in a call to GSS_Init_sec_context with PKU2U as the mech OID. A multi-mechanism mechglue implementer would likely have a mechglue-layer NAME object that internally keeps a reference to a NAME object produced by the underlying mechanism, but a pluggable mechglue could not expect two different mechanisms to be able to share their internal NAME objects. A clever implementer can work around this by exporting the one mechanism's MN and then re-importing using the target mechanism's GSS_Import_name() service function.

- It must be possible for the credential inquiry functions (e.g., GSS_Inquire_cred() and GSS_Inquire_cred_by_mech()) to return a cred_name that is an MN of a different mechanism than the credential element being inquired.

Implementation support for this name type, with defaulted realm name or with the PKU2U realm name is REQUIRED, but it is OPTIONAL for use with any other realm names.

5.5. GSS_C_NT_ANONYMOUS

This is a generic GSS-API name-type. Implementation support for this name type is OPTIONAL. See Section 6.1 for more information.

See [RFC2743] and [RFC2744] for more information about this name type.

The PKU2U mechanism only supports anonymous initiators, not acceptors.

Implementation support for this name type is RECOMMENDED.

5.6. GSS_C_NT_HOSTBASED_SERVICE - Matching Host-based Principal Names to Acceptor Certificates

Support for GSS_C_NT_HOSTBASED_SERVICE names is REQUIRED as described herein.

The query form of this name type is as per-RFC2743. The canonical and exported name token forms are as per-RFC1964. The display form of this name type is left unspecified, but should either be as per-RFC2743 or the same as the display form for GSS_KRB5_NT_PRINCIPAL_NAME [RFC1964].

Initiators using names of type GSS_C_NT_HOSTBASED_SERVICE to identify target acceptors represent these names as Kerberos V principal names as per [RFC1964] but with a well-known realm name of "WELLKNOWN: PKU2U" (see Section 5.4).

Acceptors match such names to acceptor certificates as follows. Initiators then match the certificate chosen by the acceptor in the same manner.

Initiators can also assert host-based service names as the initiator name. In this case acceptors MUST also apply the matching rules below, in order, to validate the initiator's assertion.

1. If there is an out-of-band binding of the peer's host-based service name to its certificate, then the certificate matches.

2. If the peer has a certificate with an id-pkinit-san subject alternative name matching the initiator-provided acceptor name, then the X.509 certificate matches.

3. If a X.509 certificate has a dNSName SAN that matches the hostname part of the host-based service principal name, and either the anyExtendedKeyUsage extended key usage (EKU), or no EKU is present, or an EKU is present which corresponds to the service part of the host-based service principal name, then the X.509 certificate matches. The id-kp-serverAuth EKU SHALL be considered to match the 'HTTP' service name. (See Section 10, IANA considerations, where the GSS-API service name registry is extended to include an EKU for each service name.)

4. Implementations SHOULD, subject to local configuration, allow matches where the single-component cn of the DN of a X.509 certificate matches the hostname part of the host-based service name, for some or all service names. This feature is needed to allow the use of existing X.509 web certificates.

Implementation support for this name type as an acceptor name is REQUIRED. Implementation support for this name type as an initiator name is OPTIONAL.

6. The Protocol Description and the Context Establishment Tokens

The PKU2U mechanism is a GSS-API mechanism based on [RFC4120], [RFC4556] and [RFC4121].

The per-message tokens of the PKU2U mechanism are the same as those of the Kerberos V GSS-API mechanism [RFC4121].

The GSS_Pseudo_random() function [RFC4401] of the PKU2U is the same as that of the Kerberos V GSS-API mechanism [RFC4402].

The PKU2U security context token exchange consists of KRB_AS_REQ and KRB_AS_REP (and KRB_ERROR) Kerberos KDC PDUs (with no changes to their ASN.1 description, but with other minor changes/requirements described below) as context tokens, with the acceptor as the KDC, followed by context tokens from [RFC4121] using the Kerberos V Ticket PDU issued by the acceptor-as-KDC. PKINIT [RFC4556] is the only acceptable pre-authentication method in this document. Caching the ticket issued by the acceptor allows subsequent security context exchanges between the same to peers to use a single context token round-trip -- a "fast reconnect" feature.

PKU2U differs from Kerberos V with PKINIT in several minor ways as follows (this is not a complete list):

- KDC PDUs are not exchanged as usual in Kerberos, but wrapped as [the first two] GSS-API context tokens.

- PKU2U does not use KDC certificates.

- PKU2U adds pa-data types for carrying the initiator's assertion of its name and the targ_name passed to GSS_Init_sec_context().

PKU2U differs from the Kerberos V GSS-API mechanism in several ways:

- KDC PDUs are not exchanged as described in [RFC4120], but wrapped as GSS-API context tokens.

- PKU2U allows the use of principal names matching PKI naming (see Section 5). PKU2U does support the use of Kerberos V naming, but requires only support of Kerberos V naming to a limited extent (full support is optional).

- PKU2U adds an extension [GSS-EXTS] to the RFC4121 initial context token for binding the AP-REQ to the AS exchange that precedes is (that is, when the initiator has to request a ticket from the acceptor).

- The number of round-trips can vary. If the initiator already has a ticket for the acceptor then the context token exchange will be half a round-trip or one round-trip, as per RFC4121. Otherwise one or two round-trips are added for the AS exchanges needed to acquire a ticket. Note that two AS exchanges may be required when the initiator's initial choice of X.509 certificate does not match the acceptor's trust anchors, in which case the acceptor SHOULD reply with a KRB-ERROR with TD-TRUSTED-CERTIFIERS indicating what the acceptor's trust anchors are, and then the initiator can engage in a second AS exchange within the same GSS-API context.

To recapitulate, the acceptor and the initiator communicate by tunneling the authentication service exchange messages through the use of the GSS-API tokens and application traffic. In the event of security context token loss, message duplication, or out of order message delivery, the security context MUST fail to establish.

All security context establishment tokens MUST follow the InitialContextToken syntax defined in Section 3.1 of [RFC2743]. PKU2U is identified by the Objection Identifier (OID) id-kerberos-pku2u.

The PKU2U OID is:

id-kerberos-pku2u ::=
{ iso(1) org(3) dod(6) internet(1) security(5) kerberosV5(2)
pku2u(7) }

All context establishment tokens consist of some Kerberos V PDU or another, prefixed with a two-octet token type ID, and the InitialContextToken header (see above).

The innerToken described in section 3.1 of [RFC2743] and subsequent GSS-API mechanism tokens have the following formats: it starts with a two-octet token-identifier (TOK_ID), followed by a Kerberos message. The TOK_ID values for the AS-REQ message and the AS-REP message are defined in the table blow:

Token	TOK_ID Value in Hex
KRB_AS_REQ	05 00
KRB_AS_REP	06 00

The TOK_ID values for all other Kerberos messages are the same as defined in [RFC4121].

It should be noted that by using anonymous PKINIT [KRB-ANON], PKU2U can authenticate the acceptor without revealing the initiator's identity

6.1. Context Token Derived from KRB_AS_REQ

When the initiator does not have a service ticket to the acceptor, it requests a ticket from the acceptor instead of from the KDC by constructing a KRB_AS_REQ PDU [RFC4120] and using it as the context token, with a token type ID prefixed. This will be the initiator's initial context token, therefore it MUST also have the standard header bearing the OID of the mechanism being used (in this case, PKU2U's OID).

The initiator MUST NOT set any KDC options in the 'kdc-options' field of the AS-REQ.

The 'realm' field of the AS-REQ MUST be set to the PKU2U well-known PKU2U realm name ("WELLKNOWN:PKU2U" [KRB-NAMING].

If the initiator wishes to assert a name of type GSS_KRB5_NT_PRINCIPAL_NAME or GSS_C_NT_HOSTBASED_SERVICE, then it MUST set the 'cname' field of the AS-REQ accordingly if and only if the realm name part of the given name object is defaulted or the

well-known PKU2U realm name. Otherwise the initiator MUST add a pa-data element (see below) stating the name that the initiator wishes to assert, it MUST set the cname field to the NULL principal name as defined in Section 4.

If the targ_name passed to GSS_Init_sec_context() is of type GSS_C_NT_HOSTBASED_NAME then the initiator sets the 'sname' field of the AS-REQ to match the parsed name as per [RFC4121]. If the target name does not have a representation as a Kerberos principal name per [RFC1964], then the initiator MUST add a pa-data element (see below) stating the given targ_name and the initiator MUST set the 'sname' field of the AS-REQ to the NULL principal name as defined in Section 4.

The padata used to convey initiator and target names is of type PA_PKU2U_NAME <136> and it's value consists of the DER [CCITT.X680.2002] [CCITT.X690.2002] encoding of the ASN.1 type InitiatorNameAssertion (with explicit tagging).

```
InitiatorName ::= CHOICE {
    -- -1 -> certificate DN
    -- 0..16384 -> subjectAltName named by
    --             this index
    sanIndex INTEGER (-1..16384), -- DN or SAN
    nameNotInCert GeneralName,    -- name not present in cert
                                  -- (see RFC5280 for definition
                                      of GeneralName)
    ...
}

TargetName ::= CHOICE {
    exportedTargName OTCET STRING, -- exported krb5 name
    generalName [0] GeneralName,   -- all other PKI names
                                   -- (tagged to distinguish
                                   -- from nameNotInCert
                                   -- choice of InitiatorName)
    ...
}

InitiatorNameAssertion ::= SEQUENCE {
    initiatorName InitiatorName OPTIONAL,
    targetName TargetName OPTIONAL,
    ...
}
```

The initiatorName, if present, contains the initiator's name. The initiator can fill out either the sanIndex field or the nameNotInCert field to indicate the name of the initiator.

The sanIndex field, if present, is used to refer to either the Distinguished Name or the SubjectAltName in the initiator's X.509 certificate. A sanIndex value of -1 value refers to the initiator's certificate's DN. All other legal values of sanIndex refer to the corresponding element of the SubjectAltName sequence. A value of 0 means the first instance of GeneralName in the SubjectAltName sequence, and 1 means the second, and so on. If the sanIndex value is equal or biger than the number of GeneralName elements in the SubjectAltName, the security context establishment attempt MUST fail.

The nameNotInCert field, if present, contains the initiator's GeneralName.

If an initiator name assertion is included, the acceptor MUST verify that this asserted name is either present in the initiator's certificate or otherwise bound to the initiator's certificate by out-of-band provisioning (e.g., by a table lookup). Failure to validate the asserted initiator's name MUST cause GSS_Accept_sec_context() to return an error and, optionally, to output a KRB_ERROR context token as per-RFC4121.

The initiatorName field MUST NOT be present if the initiator is anonymous or if the 'cname' field of the AS-REQ is not the NULL name (see Section 4).

Target names passed to GSS_Init_sec_context() that can be represented as Kerberos V principal names, namely, names of GSS_KRB5_NT_PRINCIPAL_NAME and GSS_C_NT_HOSTBASED_SERVICE, MUST be represented as the 'sname' field of the AS-REQ or as the exportedTargName choice of TargetName (if the realm part is not the PKU2U realm name). The contents of the exportedTargName octet string MUST be an exported name token for the Kerberos V mechanism containing a Kerberos V principal name.

Other target names are represented as a generalName choice of TargetName. These may be present in an acceptor certificate, or agreed out of band.

The acceptor MUST select an appropriate acceptor credential that matches the AS-REQ's 'sname' (if not NULL) or the targetName provided in the InitiatorNameAssertion, when present.

The targetName field MUST NOT be present if the 'sname' field of the AS-REQ is not the NULL name. The targetName field MUST be present if the 'sname' field of the AS-REQ is the NULL name.

The PA_PKU2U_NAME padata SHOULD NOT be present when the initiatorName and targetName both shouldn't be present.

Implementation note: the encrypted part of a PKU2U Ticket can be anything at all since the only entity that will consumer a given PKU2U Ticket is the same entity that issued it. Implementers may choose to use the traditional EncTicketPart ASN.1 type [RFC4120] and DER encoding.

6.2. Context Token Derived from KRB_AS_REP

When the initiator's initial context token is a AS-REQ then the acceptor MUST reply with either a KRB-ERROR token as per [RFC4121] or a token derived from a KRB_AS_REP PDU [RFC4120] constructed to respond to the initiator's KRB_AS_REQ.

The initiator MUST use PKINIT pre-authentication and the acceptor MUST require it. If the initiator does not use PKINIT pre-authentication then the acceptor MUST respond with a KRB-ERROR and indicate that PKINIT is required.

If the initiator's KRB_AS_REQ token is valid, and the asserted initiator's name, if present, is bound with the initiator's certificate, and the acceptor can select a certificate based on the initiator's asserted targ_name, the acceptor then constructs a KRB_AS_REP using PKINIT as described in [RFC4556], except that the acceptor's certificate is used in the place of the KDC certificate. If and only if the initiator's X.509 certficate is validated using PKI, the acceptor SHOULD include an authorization element AD_INITIAL_VERIFIED_CAS [RFC4556] in the returned ticket. If an InitiatorName is included in the PA_PKU2U_NAME padata in the request, an authorization element of the type ad-pku2u-client-name <143> MUST be included in the returned ticet and this authorization element contains the DER encoded InitiatorName in the request.

The initiator then validates the KRB-AS-REP reply context token according to Section 3.1.5 of [RFC4120] and Section 3.2.4 of [RFC4556]. The inclusion of the EKU KeyPurposeId [RFC5280] id-pkinit-KPKdc in the X.509 certificate in the response is not applicable when PKU2U is used because there is no KDC involved in this protocol. The initiator MUST verify that the acceptor's certificate is bound with the targ_name passed in to GSS_Init_sec_context(), by verifying either the targ_name matches with either the subject DN or one instance of the SubjectAltName name in the acceptor's certificate, or otherwise the targ_name is bound with the acceptor's certificate by out-of-band provisioning (e.g., by a table lookup). Failure to validate this name binding MUST cause the authentication to be rejected.

The 'flags' field of the AS-REP MUST have only the 'initial' and 'pre-authent' flags set.

The 'authtime' field of the AS-REP MUST be set to the acceptor's current time as it is when it formats the AS-REP.

Otherwise all other aspects of the AS-REP are as described in [RFC4120].

The values of the tkt-vno, realm and 'sname' fields of the Ticket issued by the acceptor are unspecified. The initiator MUST NOT examine them for correctness. Cut-n-paste attacks are prevented by the fact that PKU2U provides integrity protection for all cleartext in Kerberos V PDUs used by PKU2U (and for the mechanism OID).

6.3. Context Tokens Imported from RFC4121

Once the initiator has a Kerberos V Ticket for the acceptor the security context token exchange will continue with those of the Kerberos V GSS-API mechanism [RFC4121] with the following modifications:

- The mechanism OID of PKU2U SHALL be used instead of that of the Kerberos V GSS-API mechanism;

- The 'crealm' field of the initiator's Authenticator MUST be set to the PKU2U realm name and if the 'cname" field is the NULL principal name, an authorization element of the type ad-pku2u-client-name <143> MUST be included in the authenticator and this authorization element contains the DER encoded InitiatorName in the AS-REQ based on which the ticket was obtained;

- The sub-session key MUST be used in the initiator's Authenticator;

- The contents of the encrypted part of the Ticket can be implementation specific since the only entity consuming it will be the same entity that issues it;

- If the initiator's initial context token is a KRB_AS_REQ token (i.e., not KRB_AP_REQ token), then the Exts field in the Authenticator of the KRB_AP_REQ-derived token MUST contain an extension [GSS-EXTS] of the type GSS_EXTS_FINISHED <2> as defined next in this section.

The 'cusec', 'ctime', 'seq-number' and 'authorization-data' fields of the Authenticator are set as per [RFC4121] and [RFC4120].

The 'cksum' field of the Authenticator MUST be set as per [RFC4121]. The extension data of the GSS_EXTS_FINISHED extension type [GSS-EXTS] contains the DER encoding of the ASN.1 structure KRB-FINISHED.

```
GSS_EXTS_FINISHED            2
    --- The type for the checksum extension.

KRB-FINISHED ::= SEQUENCE {
    gss-mic [1] Checksum,
        -- Contains the checksum (RFC3961) of the GSS-API tokens
        -- that have been exchanged between the initiator and the
        -- acceptor and prior to the containing AP-REQ GSS-API token.
        -- The checksum is performed over the GSS-API
        -- context tokens in the order that the tokens were sent.
     ...
}
```

The gss-mic field contains a Kerberos checksum [RFC3961] that is computed over all the preceding context tokens of this GSS-API context (including the InitialContextToken header), concatenated in chronological order (note that GSS-API context token exchanges are synchronous). The checksum type is the required checksum type of the enctype of the subkey in the authenticator, the protocol key for the checksum operation is the authenticator subkey, and the key usage number is KEY_USAGE_FINISHED <41>.

The acceptor MUST process the KRB_AP_REQ token as usual for RFC4121, except that if the context token exchange included an AS exchange, then the acceptor MUST also validate the GSS_EXTS_FINISHED and return an error if it is not valid or not present. But if a KRB_AP_REQ context token is the initial context token then the acceptor MUST return an error if GSS_EXTS_FINISHED is present.

The GSS_EXTS_FINISHED (along with the ticket) binds the second part of the context token exchange to the first, and it binds the pa-data used in the request as well (this needs to be done because PKINIT does not bind pa-data other than PKINIT pa-data from the request). GSS_EXTS_FINISHED also protects all otherwise unauthenticated plaintext in Kerberos V PDUs. Note that GSS_EXTS_FINISHED also protects the mechanism OID in the InitialContextToken header.

The acceptor MUST verify that the ad-pku2u-client-name authorization element is present in the authenticator if and only there is an authorization element of the same type in the ticket and the values of these two elements MUST match exactly based on bit-wise comparison.

7. Guidelines for Credentials Selection

If a peer, either the initiator or the acceptor, has multiple pairs of public-key private keys and certificates, a choice is to be made

in choosing the best fit. The trustedCertifiers field in the PA-PK-AS-REQ structure [RFC4556] SHOULD be filled by the initiator, to provide hints for guiding the selection of an appropriate certificate chain by the acceptor.

If the initiator's X.509 certificate cannot be validated according to [RFC5280], the acceptor SHOULD send back the TD-TRUSTED-CERTIFIERS structure [RFC4556] that provides hints for guiding the selection of an appropriate certificate by the initiator. In this case GSS_Accept_sec_context() returns GSS_S_CONTINUE_NEEDED, and the initiator gets to try again in its subsequent AS-REQ token.

The GSS-API does not provide a programming interface to make this credential selection interactive, though implementers may provide methods for user interaction related to credential selection and acquisition (e.g., name and password/PIN prompts). Whenever the execution context allows for direct interaction of the mechanism with the user then it is RECOMMENDED that implementations interact with the user to select a credential whenever multiple credentials are equally usable and no other mechanism is available to inform the credential selection.

If the certificates cannot be selected interactively, multiple certificates are equally usable, and there is no other mechanism available for credential selection, then it is RECOMMENDED that initiators fail the context. Users should be able to retry using a specific credential (this requires that distinct credentials have distinct names that can be used to acquire each credential separately).

8. Security Considerations

The security considerations in [RFC4120], [RFC4121], [RFC4556] and [RFC5280] apply here. This mechanism relaxes some requirements of PKINIT and adds a device for protecting otherwise unauthenticated plaintext in the protocol (see Section 6.3) -- it is crucial that this device be faithfully implemented. It is also crucial that both the initiator and the acceptor MUST be able to verify the binding between the signing key and the asserted identity.

Note that PKU2U is just as susceptible to replays of AP-REQs as the traditional Kerberos V GSS-API mechanism [RFC4121], though only when using an AP-REQ as the initial security context token. It is important, therefore, to use a replay cache to detect replays.

9. Acknowledgements

The authors would like to thank Jeffrey Hutzelman for his insightful comments on the earlier revisions of this document.

In addition, the following individuals have provided review comments for this document: Sam Hartman, Leif Johansson, Olga Kornievskaia, Martin Rex, and Sunil Gottumukkala.

Ari Medvinsky provided help in editing the initial revisions of this document.

The text for the DN mapping is compiled from the email discussions among the following individuals: Howard Chu, Martin Rex, Jeffrey Hutzelman, Kevin Coffman, Henry B. Hotz, Leif Johansson, and Olga Kornievskaia. Howard and Jeffery clearly illustrated the challenges in creating a unique mapping, while Nicolas and Martin demonstrated the relevance and interactions to GSS-API and Kerberos.

10. IANA Considerations

This document defines the PKU2U realm and the place-holder well-known principal name. The IANA registry for the reserved names should be updated to reference this document. Two entries are added: one entry for the well-known realm "WELLKNOWN:PKU2U", and another for the well-known principal name "WELLKNOWN/NULL".

This document defines GSS_EXTS_FINISHED extension type. The corresponding IANA registry [GSS-EXTS] need to be updated to reference this document. The following single registration should be added in the registry for "Kerberos V GSS-API mechanism extension types": 2, "GSS-API token checksum", "Extension to provide a checksum for GSS-API tokens", the RFC # of this document.

This document also instructs the IANA to extend the "SMI Security for Name System Designators Codes (nametypes)" registry to include an OID for each registration, and to allocate OIDs for the following GSS-API name-types in that registry:

- gss-distinguished-name (GSS_C_NT_DN)
- gss-hostname (GSS_C_NT_HOSTNAME)
- gss-IP-address (GSS_C_NT_IP_ADDR)
- gss-e-mail-address (GSS_C_NT_EMAIL_ADDR)

11. Normative References

[CCITT.X680.2002]

International International Telephone and Telegraph Consultative Committee, "Abstract Syntax Notation One (ASN.1): Specification of basic notation", CCITT Recommendation X.680, July 2002.

[CCITT.X690.2002]
International International Telephone and Telegraph Consultative Committee, "ASN.1 encoding rules: Specification of basic encoding Rules (BER), Canonical encoding rules (CER) and Distinguished encoding rules (DER)", CCITT Recommendation X.690, July 2002.

[GSS-EXTS]
Emery, S., "Kerberos Version 5 GSS-API Channel Binding Hash Agility", draft-ietf-krb-wg-gss-cb-hash-agility (work in progress), 2007.

[KRB-ANON]
Zhu, L. and P. Leach, "Kerberos Anonymity Support", draft-ietf-krb-wg-anon (work in progress), 2007.

[KRB-NAMING]
Zhu, L., "Additional Kerberos Naming Constraints", draft-ietf-krb-wg-naming (work in progress), 2007.

[RFC0822] Crocker, D., "Standard for the format of ARPA Internet text messages", STD 11, RFC 822, August 1982.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1964] Linn, J., "The Kerberos Version 5 GSS-API Mechanism", RFC 1964, June 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2743] Linn, J., "Generic Security Service Application Program Interface Version 2, Update 1", RFC 2743, January 2000.

[RFC2744] Wray, J., "Generic Security Service API Version 2 : C-bindings", RFC 2744, January 2000.

[RFC3490] Faltstrom, P., Hoffman, P., and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 3490, March 2003.

[RFC3961] Raeburn, K., "Encryption and Checksum Specifications for

Kerberos 5", RFC 3961, February 2005.

[RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.

[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, July 2005.

[RFC4401] Williams, N., "A Pseudo-Random Function (PRF) API Extension for the Generic Security Service Application Program Interface (GSS-API)", RFC 4401, February 2006.

[RFC4402] Williams, N., "A Pseudo-Random Function (PRF) for the Kerberos V Generic Security Service Application Program Interface (GSS-API) Mechanism", RFC 4402, February 2006.

[RFC4514] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names", RFC 4514, June 2006.

[RFC4517] Legg, S., "Lightweight Directory Access Protocol (LDAP): Syntaxes and Matching Rules", RFC 4517, June 2006.

[RFC4556] Zhu, L. and B. Tung, "Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)", RFC 4556, June 2006.

[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, May 2008.

Authors' Addresses

Larry Zhu
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
US

Email: lzhu@microsoft.com

Jeffery Altman
Secure Endpoints
255 W 94th St
New York, NY 10025
US

Email: jaltman@secure-endpoints.com

Nicolas Williams
Sun Microsystems
5300 Riata Trace Ct
Austin, TX 78727
US

Email: Nicolas.Williams@sun.com

Full Copyright Statement

Intellectual Property