# [MS-KILE]:
# Kerberos Protocol Extensions

## Revision Summary

| Date | Revision History | Revision Class | Comments |
|---|---|---|---|
| 03/14/2007 | 1.0 | | Version 1.0 release |
| 04/10/2007 | 1.1 | | Version 1.1 release |
| 05/18/2007 | 1.2 | | Version 1.2 release |
| 06/08/2007 | 1.2.1 | Editorial | Revised and edited the technical content. |
| 07/10/2007 | 1.3 | Minor | Updated the technical content. |
| 08/17/2007 | 1.3.1 | Editorial | Revised and edited the technical content. |
| 09/21/2007 | 1.4 | Minor | Revised content based on feedback. |
| 10/26/2007 | 2.0 | Major | Converted document to unified format. |
| 01/25/2008 | 2.0.1 | Editorial | Revised and edited the technical content. |
| 03/14/2008 | 2.1 | Minor | Updated the technical content. |
| 06/20/2008 | 3.0 | Major | Updated and revised the technical content. |
| 07/25/2008 | 3.1 | Minor | Updated the technical content. |
| 08/29/2008 | 4.0 | Major | Updated and revised the technical content. |
| 10/24/2008 | 4.1 | Minor | Updated the technical content. |
| 12/05/2008 | 5.0 | Major | Updated and revised the technical content. |
| 01/16/2009 | 5.1 | Minor | Updated the technical content. |
| 02/27/2009 | 6.0 | Major | Updated and revised the technical content. |
| 04/10/2009 | 7.0 | Major | Updated and revised the technical content. |
| 05/22/2009 | 8.0 | Major | Updated and revised the technical content. |
| 07/02/2009 | 9.0 | Major | Updated and revised the technical content. |
| 08/14/2009 | 9.1 | Minor | Updated the technical content. |
| 09/25/2009 | 10.0 | Major | Updated and revised the technical content. |
| 11/06/2009 | 11.0 | Major | Updated and revised the technical content. |
| 12/18/2009 | 12.0 | Major | Updated and revised the technical content. |
| 01/29/2010 | 13.0 | Major | Updated and revised the technical content. |
| 03/12/2010 | 13.1 | Minor | Updated the technical content. |
| 04/23/2010 | 14.0 | Major | Updated and revised the technical content. |

| Date | Revision History | Revision Class | Comments |
|---|---|---|---|
| 06/04/2010 | 14.1 | Minor | Updated the technical content. |
| 07/16/2010 | 14.2 | Minor | Clarified the meaning of the technical content. |
| 08/27/2010 | 14.3 | Minor | Clarified the meaning of the technical content. |
| 10/08/2010 | 14.4 | Minor | Clarified the meaning of the technical content. |
| 11/19/2010 | 15.0 | Major | Significantly changed the technical content. |
| 01/07/2011 | 16.0 | Major | Significantly changed the technical content. |
| 02/11/2011 | 16.1 | Minor | Clarified the meaning of the technical content. |

# Contents

# 1 Introduction

Kerberos Protocol Extensions (KILE) specifies extensions to the Kerberos Network Authentication Service (V5) protocol [RFC4120]. These extensions provide additional capability for authorization information including group memberships, interactive logon information, and integrity levels.

**Note** Throughout the remainder of this specification the Kerberos Network Authentication Service (V5) protocol will be referred to simply as Kerberos V5.

## 1.1 Glossary

The following terms are defined in [MS-GLOS]:

**Active Directory**
**AP exchange**
**AS exchange**
**Authentication Service (AS)**
**authenticator**
**authorization data**
**directory**
**directory service (DS)**
**distinguished name (DN)**
**domain**
**fully qualified domain name (FQDN)**
**Generic Security Services (GSS)**
**Internet host name**
**Kerberos principal**
**key**
**Key Distribution Center (KDC)**
**KRB_AP_REQ/KRB_AP_REP**
**KRB_AS_REQ/KRB_AS_REP**
**KRB_PRIV exchange**
**KRB_SAFE exchange**
**object identifier (OID)**
**objectGuid**
**preauthentication**
**privilege attribute certificate (PAC)**
**read-only domain controller (RODC)**
**realm**
**secret key**
**Security Support Provider Interface (SSPI)**
**service**
**service principal**
**service principal name (SPN)**
**service (SRV) resource record**
**service ticket**
**session**
**session key**
**ticket**
**ticket-granting service (TGS)**
**ticket-granting service (TGS) exchange**
**ticket-granting ticket (TGT)**

The following terms are specific to this document:

**context session key:** A variant of a cryptographic key used in the generation and processing of per-message tokens that uses the Kerberos session key directly ([RFC1964] section 1.2).

**integrity level:** The attributed trustworthiness of an entity or object.

**"RestrictedKrbHost" services:** The class of services that use SPNs with the *serviceclass* string equal to "RestrictedKrbHost", whose service tickets use the computer account's key and share a session key. For information on the *serviceclass* string, see section 3.1.5.12.

**security package:** The software implementation of a security protocol. Security packages are contained in security support provider components or security support provider/authentication package components.

**ticket session key:** The **session key** within a **ticket**.

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

## 1.2 References

### 1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[C706] The Open Group, "DCE 1.1: Remote Procedure Call", C706, August 1997, http://www.opengroup.org/public/pubs/catalog/c706.htm

[FIPS140] National Institute of Standards and Technology, "Federal Information Processing Standards Publication 140-2: Security Requirements for Cryptographic Modules", December 2002, http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

[MS-ADA1] Microsoft Corporation, "Active Directory Schema Attributes A-L", June 2007.

[MS-ADA2] Microsoft Corporation, "Active Directory Schema Attributes M", July 2006.

[MS-ADA3] Microsoft Corporation, "Active Directory Schema Attributes N-Z", July 2006.

[MS-ADTS] Microsoft Corporation, "Active Directory Technical Specification", July 2006.

[MS-ADSC] Microsoft Corporation, "Active Directory Schema Classes", July 2006.

[MS-DRSR] Microsoft Corporation, "Directory Replication Service (DRS) Remote Protocol Specification", July 2006.

[MS-DTYP] Microsoft Corporation, "Windows Data Types", January 2007.

[MS-GPSB] Microsoft Corporation, "Group Policy: Security Protocol Extension", July 2006.

[MS-LSAD] Microsoft Corporation, "Local Security Authority (Domain Policy) Remote Protocol Specification", July 2006.

[MS-PAC] Microsoft Corporation, "Privilege Attribute Certificate Data Structure", July 2006.

[MS-RPCE] Microsoft Corporation, "Remote Procedure Call Protocol Extensions", July 2006.

[MS-SAMR] Microsoft Corporation, "Security Account Manager (SAM) Remote Protocol Specification (Client-to-Server)", July 2006.

[MS-SNTP] Microsoft Corporation, "Network Time Protocol (NTP) Authentication Extensions", July 2006.

[MS-SPNG] Microsoft Corporation, "Simple and Protected GSS-API Negotiation Mechanism (SPNEGO) Extension", July 2006.

[MS-UCODEREF] Microsoft Corporation, "Windows Protocols Unicode Reference", July 2007.

[Referrals-11] Raeburn, K., and Zhu, L., "Kerberos Principal Name Canonicalization and KDC-Generated Cross-Realm Referrals", July 2008, http://tools.ietf.org/internet-drafts/draft-ietf-krb-wg-kerberos-referrals-11

[RFC1964] Linn, J., "The Kerberos Version 5 GSS-API Mechanism", RFC 1964, June 1996, http://www.ietf.org/rfc/rfc1964.txt

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[RFC2743] Linn, J., "Generic Security Service Application Program Interface Version 2, Update 1", RFC 2743, January 2000, http://www.ietf.org/rfc/rfc2743.txt

[RFC2744] Wray, J., "Generic Security Service API Version 2 : C-bindings", RFC 2744, January 2000, http://www.ietf.org/rfc/rfc2744.txt

[RFC2279] Yergeau, F., "UTF-8, A Transformation Format of ISO10646", RFC 2279, January 1998, http://www.ietf.org/rfc/rfc2279.txt

[RFC3961] Raeburn, K., "Encryption and Checksum Specifications for Kerberos 5", RFC 3961, February 2005, http://www.ietf.org/rfc/rfc3961.txt

[RFC3962] Raeburn, K., "Advanced Encryption Standard (AES) Encryption for Kerberos 5", RFC 3962, February 2005, http://www.ietf.org/rfc/rfc3962.txt

[RFC4120] Neuman, C., Yu, T., Hartman, S., and Raeburn, K., "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005, http://www.ietf.org/rfc/rfc4120.txt

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

[RFC4556] Zhu, L., and Tung, B., "Public Key Cryptography for Initial Authentication in Kerberos", RFC 4556, June 2006 http://www.ietf.org/rfc/rfc4556.txt

[RFC4757] Jaganathan, K., Zhu, L., and Brezak, J., "The RC4-HMAC Kerberos Encryption Types Used by Microsoft Windows", RFC 4757, December 2006, http://www.ietf.org/rfc/rfc4757.txt

[RFC5349] Zhu, L., Jaganathan, K., and Lauter, K., "Elliptic Curve Cryptography (ECC) Support for Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)", RFC 5349, September 2008, http://www.ietf.org/rfc/rfc5349.txt

[X680] ITU-T, "Abstract Syntax Notation One (ASN.1): Specification of Basic Notation", Recommendation X.680, July 2002, http://www.itu.int/rec/T-REC-X.680/en

**Note** There is a charge to download the specification.

[X690] ITU-T, "Information Technology - ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", Recommendation X.690, July 2002, http://www.itu.int/rec/T-REC-X.690/en

**Note** There is a charge to download the specification.

### 1.2.2 Informative References

[ADDLG] Microsoft Corporation, "Security Briefs: Credentials and Delegation", September 2005, http://msdn.microsoft.com/en-us/magazine/cc163740.aspx

[DIALOGUE] Bryant, B. and Ts'o, T., "Designing an Authentication System: A Dialogue in Four Scenes", February 1997, http://web.mit.edu/kerberos/www/dialogue.html

[KAUFMAN] Kaufman, C., Perlman, R., and M. Speciner, "Network Security: Private Communication in a Public World, Second Edition", Prentice Hall, 2002, ISBN: 0130460192.

[MS-APDS] Microsoft Corporation, "Authentication Protocol Domain Support Specification", July 2006.

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-PKCA] Microsoft Corporation, "Public Key Cryptography for Initial Authentication (PKINIT) in Kerberos Protocol Specification", July 2006.

[MS-SFU] Microsoft Corporation, "Kerberos Protocol Extensions: Service for User and Constrained Delegation Protocol Specification", July 2006.

[MSDN-WIMD] Microsoft Corporation, "Windows Integrity Mechanism Design", http://msdn.microsoft.com/en-us/library/bb625963.aspx

[RFC1510] Kohl, J., and Neuman, C., "The Kerberos Network Authentication Service (V5)", RFC 1510, September 1993, http://www.ietf.org/rfc/rfc1510.txt

[RFC2222] Myers, J., "Simple Authentication and Security Layer (SASL)", RFC 2222, October 1997, http://www.ietf.org/rfc/rfc2222.txt

[RFC2396] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998, http://www.ietf.org/rfc/rfc2396.txt

[UNICODE] The Unicode Consortium, "Unicode Home Page", 2006, http://www.unicode.org/

[UUKA-GSSAPI] Swift, M., Brezak, J., and Moore, P., "User to User Kerberos Authentication using GSS-API", October 2001, http://www.watersprings.org/pub/id/draft-swift-win2k-krb-user2user-03.txt

If you have any trouble finding [UUKA-GSSAPI], please check here.

## 1.3 Overview

KILE is a security protocol that authenticates entities on a network and provides additional **services** after the parties are authenticated with each other. KILE specifies extensions to the Kerberos V5 protocol.

### 1.3.1 Security Background

Because KILE is a security protocol, the normative references (section 1.2.1) and this specification use terms that are commonly used in the security field. In this specification, every effort was made to use terms (such as **kerberos principal**, **key**, and service) in the same way that they are used in [RFC4120] section 1.7.

A working knowledge of the Kerberos protocol is required in order to understand the variations between KILE and Kerberos V5, or among all the Kerberos implementations. Several informative references (section 1.2.2), specifically [DIALOGUE] and [KAUFMAN], provide an excellent high-level understanding of the Kerberos protocol and message flow. [KAUFMAN] also provides an excellent survey of other security protocols and concepts, and helps explain the terminology that is used in this document.

Finally, there are details in [RFC4120] and [RFC4121], and the predecessor documents [RFC1964], [RFC2743], and [RFC1510], that are not always immediately apparent. Careful study must be made, particularly of how **Generic Security Services (GSS)** [RFC2743] and the Kerberos implementation of GSS [RFC4121] tie together.

### 1.3.2 Kerberos Network Authentication Service (V5) Synopsis

The Kerberos V5 protocol provides a mechanism for mutual authentication between a client and a server before application data is transmitted between them. Kerberos V5 is composed of three exchanges described in detail in [RFC4120] sections 1.1 and 3.

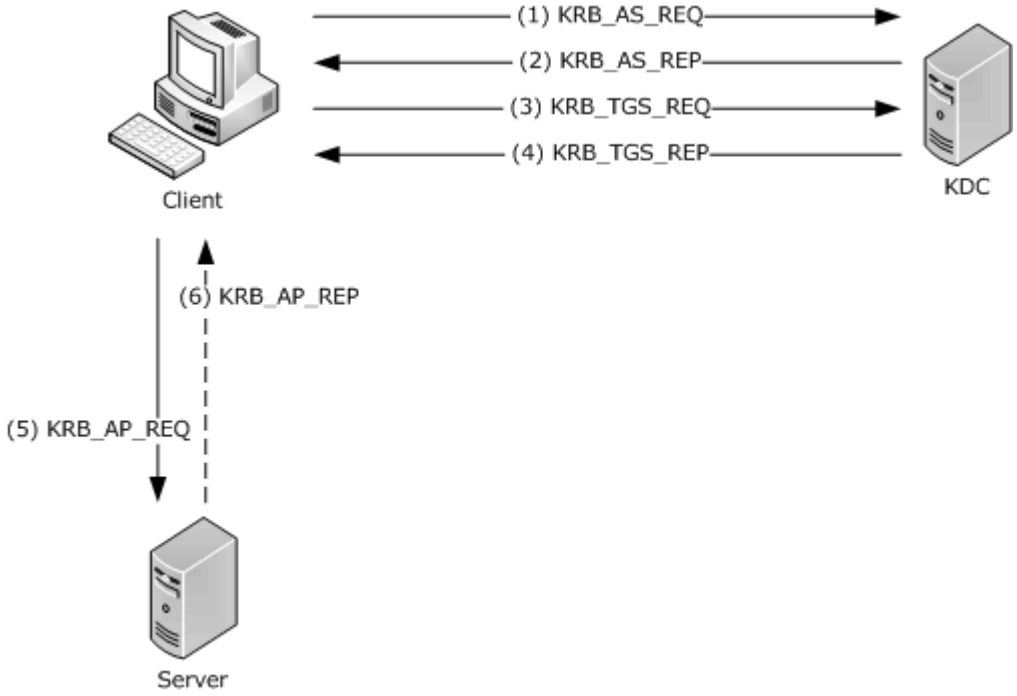

**Figure 1: Kerberos V5 Exchanges**

**Note** The terms client, server and **Key Distribution Center (KDC)**, as used in this section, refer to Kerberos V5 implementations of each entity. Unless explicitly noted, use of these terms in the remainder of this specification refers to KILE implementations of each entity.

The **Authentication Service (AS)** exchange ([RFC4120] section 3.1):

- Kerberos authentication service request (KRB_AS_REQ) ([RFC4120] section 5.4.1): The client sends a request to the KDC for a **ticket-granting ticket (TGT)** ([RFC4120] section 5.3). The client presents its principal name and can present pre-authentication information.
- Kerberos authentication service response (KRB_AS_REP) ([RFC4120] section 5.4.2): The KDC returns a TGT and a **session key** the client can use to encrypt and authenticate communication with the KDC for **ticket-granting service (TGS)** requests, without reusing the persistent key.

The Ticket-Granting Service (TGS) exchange ([RFC4120] section 3.3):

- Kerberos ticket-granting service request (KRB_TGS_REQ) ([RFC4120] section 5.4.1): The client sends a request to the KDC for a ticket ([RFC4120] section 5.3) for the server. The client presents the TGT ([RFC4120] section 5.3), an **authenticator** ([RFC4120] section 5.5.1), and the **Service Principal Name (SPN)**.
- Kerberos ticket-granting service response (KRB_TGS_REP) ([RFC4120] section 5.4.2): The KDC validates the TGT ([RFC4120] section 5.3) and the authenticator ([RFC4120] section 5.5.1). If these are valid, the KDC returns a service ticket ([RFC4120] section 5.3) and session key the client can use to encrypt communication with the server.

The Client/Server Authentication Protocol (AP) exchange ([RFC4120] section 3.2):

- Kerberos application server request (KRB_AP_REQ) ([RFC4120] section 5.5.1): The client requests access to the server. The client presents the ticket ([RFC4120] section 5.3) and a new authenticator ([RFC4120] section 5.5.1). The server will decrypt the ticket, validate the authenticator, and can use any **authorization data** ([RFC4120] section 5.2.6) contained in the ticket for access control.
- Kerberos application server response (KRB_AP_REP) ([RFC4120] section 5.5.2): Optionally, the client might request that the server verify its own identity. If mutual authentication is requested, the server returns the client's timestamp from the authenticator encrypted with the session key.

The **AS exchange** and TGS exchange are transported by Kerberos implementations. The AP exchange is passive and relies on an upper-layer application protocol to carry the **AP exchange** messages. Applications that use AP exchange messages directly are typically called "kerberized" applications. Most applications use the Generic Security Service Application Program Interface (GSS-API) and may even be wrapped by higher-level abstractions such as Simple Authentication and Security Layer (SASL) [RFC2222], which allows for "kerberized" connections to mail servers.

### 1.3.3 KILE Synopsis

By extending the authorization data ([RFC4120] section 5.2.6), KILE provides the server with additional information such as:

- Group membership
- Interactive logon information
- Integrity levels

By extending the KDC's account database, KILE provides control at the principal level for things such as delegation and Data Encryption Standard (DES) usage.

How authorization is accomplished using **Privilege Attribute Certificate (PAC)** data is described in [MS-PAC].

## 1.4 Relationship to Other Protocols

Kerberos V5 AS and TGS exchanges rely on either the User Datagram Protocol (UDP) or the Transmission Control Protocol (TCP) ([RFC4120] section 7.2.1) as a transport. KILE relies on a working Domain Name System (DNS) infrastructure.

Kerberos V5 AP exchange messages are only carried in other application protocols and never exist by themselves on the network. Almost any application can (theoretically) use Kerberos V5 authentication; applications that already adopt a GSS-style approach to security are most applicable.

Other non-RFC standard specifications relevant to the implementation of Kerberos are:

- Microsoft Active Directory, including: Active Directory Schema Attributes A-L [MS-ADA1], Active Directory Schema Attributes M [MS-ADA2], Active Directory Schema Attributes N-Z [MS-ADA3], Active Directory Schema Classes [MS-ADSC], and Active Directory Technical Specification [MS-ADTS].
- Group Policy: Security Protocol Extension [MS-GPSB]
- Local Security Authority (Domain Policy) Remote Protocol Specification [MS-LSAD]

KILE is only one part of the Microsoft Windows® implementation of Kerberos. The following are additional Kerberos extensions:

- Authentication Protocol Domain Support Specification [MS-APDS]
- Privilege Attribute Certificate Data Structure [MS-PAC]
- Public Key Cryptography for Initial Authentication (PKINIT) in Kerberos Protocol Specification [MS-PKCA]
- Kerberos Protocol Extensions: Service for User and Constrained Delegation Protocol Specification [MS-SFU]
- User to User Kerberos Authentication using GSS-API [UUKA-GSSAPI]

## 1.5 Prerequisites/Preconditions

The Kerberos V5 protocol assumes the following:

- The clocks of the participants (clients, servers, and KDCs) MUST be synchronized within a reasonable window of time. In [RFC4120], the recommended acceptable clock skew is five minutes. Time synchronization uses the Network Time Protocol and Authentication Extensions [MS-SNTP], for synchronization of the time between the three parties, but a conformant implementation can use another protocol if they choose.
- The KDC shares a secret key with the client and a separate secret key with the server. The provisioning of these secret keys is done out-of-band and is not part of KILE. Kerberos V5 implementations have a directory or database that contains at least the list of accounts and the associated secret keys.

- A source of cryptographically useful random numbers is available for generating keys and other cryptographically sensitive information.

General Kerberos V5 protocol assumptions are as specified in [RFC4120] section 1.6.

## 1.6 Applicability Statement

The Kerberos V5 protocol provides suitable authentication for clients and servers on a network that receives some level of management. The Kerberos V5 protocol is not applicable for stand-alone machines or among machines that do not have a common management infrastructure (for example, between clients and Web servers on the Internet).

KILE is applicable to any application protocol that also requires integrated authorization and group management. These extensions are also applicable to any other use for which the Kerberos V5 protocol alone is suitable.

## 1.7 Versioning and Capability Negotiation

Kerberos Protocol Extensions  does not extend the Kerberos V5 [RFC4120] protocol version number.

### 1.7.1 Pre-Authentication

The Kerberos V5 protocol supports **pre-authentication**, which takes place during the AS exchange and occurs when the client first authenticates to the KDC. A client pre-authenticates if it supplies additional information that proves it knows the key it shares with the KDC before the TGT is issued. See Pre-authentication Data (section 3.1.5.1) for a complete specification of these types supported by KILE.

### 1.7.2 Encryption Types

The Kerberos V5 protocol supports multiple encryption types, which are the actual algorithms for encrypting the **tickets** or other data. The Kerberos V5 protocol negotiates which encryption type to use for a particular connection ([RFC4120] section 3.1.3). See Encryption Types (section 3.1.5.2) for a complete specification of these types supported by KILE.

## 1.8 Vendor-Extensible Fields

The Kerberos V5 protocol includes several areas for vendor extension.

KILE does not provide vendor extensibility beyond what is specified in [RFC4120].

## 1.9 Standards Assignments

Assignment of Kerberos V5 IANA numbers is as specified in [RFC4120] section 9. UDP port 88 and TCP port 88 are used when communication between the client and the KDC occurs.

### 1.9.1 Use of Constants Assigned Elsewhere

Kerberos V5 protocol has been assigned the following **object identifier (OID)**: iso.member-body.United States.mit.infosys.gssapi.krb5<1> (1.2.840.113554.1.2.2).

# 2 Messages

## 2.1 Transport

The Kerberos V5 protocol uses UDP and TCP for transport ([RFC4120] section 7.2). KILE SHOULD use UDP by default; however, if the message size exceeds a specific configurable value (message size threshold), TCP SHOULD be used.<2> The threshold applies to **AS** and TGS messages. They do not apply to AP messages because the transport is controlled by the application protocol.

KILE MUST have a working DNS infrastructure. KILE SHOULD NOT use the Internet Protocol (IP) addresses of the KDCs. For more information about DC SRV records registration, see [MS-ADTS] section 7.3.2.3.

## 2.2 Message Syntax

KILE does not alter the syntax of any Kerberos V5 messages ([RFC4120] sections 5.4 through 5.9). KILE extensions provide platform-specific data to support encoding of authorization data ([MS-PAC] section 2) in the authorization data field ([RFC4120] sections 5.2.6 and 5.2.7) of the ticket.

The authorization data, which MUST be encoded as a PAC, MUST be marked as AD-IF-RELEVANT, which means that it SHOULD be ignored by implementations that do not understand the format.

Kerberos V5 messages are defined using Abstract Syntax Notation One (ASN.1), as specified in [X680], and encoded using Distinguished Encoding Rules (DER), as specified in [X690] section 10.

### 2.2.1 KERB-ERROR-DATA

This structure is a Windows-specific structure returned by the application server in the e-data field in the KRB-ERROR message ([RFC4120] section 5.9.1) when clock skew recovery is attempted.

```
KERB-ERROR-DATA ::= SEQUENCE {
    data-type               [1] INTEGER,
    data-value              [2] OCTET STRING OPTIONAL
}
```

**Data-type**: This value SHOULD be as follows.

| Value | Meaning |
|---|---|
| KERB_AP_ERR_TYPE_SKEW_RECOVERY | Represents the integer value 0x00000002 |

**Data-value**: This value SHOULD be NULL.

### 2.2.2 KERB-PA-PAC-REQUEST

This structure is a PA-DATA type that is defined to explicitly request to include or exclude a PAC in the ticket. Its structure is defined using ASN.1 notation and the syntax is as follows.

```
KERB-PA-PAC-REQUEST ::= SEQUENCE {
include-pac[0] BOOLEAN --If TRUE, and no pac present, include PAC.
                       --If FALSE, and PAC present, remove PAC
}
```

### 2.2.3 KERB-LOCAL

The KERB-LOCAL structure contains implementation-specific data used when the Kerberos client and application server are on the same host.<3>

```
typedef struct KERB-LOCAL {
  OCTET STRING Reserved;
} KERB-LOCAL,
 *PKERB-LOCAL;
```

**Reserved:** Implementation-specific data which MUST be ignored if Kerberos client is not local.

### 2.2.4 LSAP_TOKEN_INFO_INTEGRITY

The **LSAP_TOKEN_INFO_INTEGRITY** structure specifies the **integrity level** information for the client.<4>

```
typedef struct _LSAP_TOKEN_INFO_INTEGRITY {
  unsigned long Flags;
  unsigned long TokenIL;
  unsigned char MachineID[32];
} LSAP_TOKEN_INFO_INTEGRITY,
 *PLSAP_TOKEN_INFO_INTEGRITY;
```

**Flags:** A 32-bit unsigned integer indicating the token information type. This value MUST be one of the following.

| Value | Meaning |
|---|---|
| 0x00000000 | Full token. |
| 0x00000001 | User Account Control (UAC) restricted token. |

**TokenIL:** A 32-bit unsigned integer indicating the integrity level of the calling process. For more information about integrity levels, see [MSDN-WIMD]. This value MUST be one of the following.

| Value | Meaning |
|---|---|
| 0x00000000 | Untrusted. |
| 0x00001000 | Low. |
| 0x00002000 | Medium. |
| 0x00003000 | High. |
| 0x00004000 | System. |
| 0x00005000 | Protected process. |

**MachineID:** The machine ID (section 3.1.1.4), which is used to identify the calling machine.

### 2.2.5 KERB-AD-RESTRICTION-ENTRY

The KERB-AD-RESTRICTION-ENTRY structure specifies additional restrictions for the client.<5> Its structure is defined using ASN.1 notation and the syntax is as follows.

```
KERB-AD-RESTRICTION-ENTRY ::= SEQUENCE {
restriction-type            [0] Int32,
restriction                 [1] OCTET STRING
}
```

**Restriction-Type**: MUST be set to 0x00000000.

**Restriction**: An **LSAP_TOKEN_INFO_INTEGRITY** structure that contains the integrity information for the client.

### 2.2.6 Supported Encryption Types Bit Flags

The data in the **msDS-SupportedEncryptionTypes** attribute ([MS-ADA2] section 2.324), and in fields that specify which encryption types are supported, contains a 32-bit unsigned integer in **little-endian** format that contains a combination of the following flags, and which specifies what encryption types are supported by the server or service. An encryption type is supported if its value is equal to 1.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | E | D | C | B | A |

Where the bits are defined as:

| Value | Description |
|---|---|
| A | DES-CBC-CRC |
| B | DES-CBC-MD5 |
| C | RC4-HMAC |
| D | AES128-CTS-HMAC-SHA1-96 |
| E | AES256-CTS-HMAC-SHA1-96 |

All other bits MUST be set to zero when sent and MUST be ignored when they are received.

### 2.2.7 PA-SUPPORTED-ENCTYPES

The PA-SUPPORTED-ENCTYPES structure specifies the encryption types supported and contains a bit field of the supported encryption types bit flags (section 2.2.6).<6>

```
PA-SUPPORTED-ENCTYPES ::= Int32 -- Supported Encryption Types Bit Field --
```

### 2.2.8 OCTET STRING

An ASN.1 OCTET STRING, which is binary data whose length is a multiple of eight, as defined in [X680], section 22.

## 2.3 Directory Service Schema Elements

KILE accesses the **directory service** schema classes and attributes listed in the following table.

For the syntactic specifications of the following <Class> or <Class><Attribute> pairs, refer to Active Directory Domain Services (AD DS) ([MS-ADA2], [MS-ADA3] and [MS-ADSC]).

| Class | Attribute |
|---|---|
| trustedDomain | msDS-SupportedEncryptionTypes |
| user | logonHours<br>msDS-SupportedEncryptionTypes<br>servicePrincipalName<br>userAccountControl<br>userPrincipalName |

# 3 Protocol Details

This section specifies details of KILE, including abstract data models and message processing rules, as follows:

- Common Details (section 3.1) specifies extensions to common elements.
- Client Details (section 3.2) specifies extensions specific to the client during the AS, TGS, and AP exchanges.
- KDC Details (section 3.3) specifies extensions specific to the KDC processing of AS and TGS requests.
- Application Server Details (section 3.4) specifies extensions to the server processing of the AP requests.

## 3.1 Common Details

### 3.1.1 Abstract Data Model

Kerberos V5 specifies the abstract data model for common elements.

KILE specifies the following extensions to common elements:

- Replay Cache
- Cryptographic Material
- Ticket Cache
- Machine ID
- Kerberos OID

#### 3.1.1.1 Replay Cache

Kerberos V5 specifies that servers MUST utilize a replay cache unless the application server provides replay protection ([RFC4120] section 3.2.3).

KILE MUST implement a replay cache regardless of the application server replay functionality.

#### 3.1.1.2 Cryptographic Material

Kerberos V5 establishes a secret key that is shared by a principal and the KDC and a session key that forms the basis for privacy or integrity in the communication channel between client and server. When KILE creates an AES128 key, the password MUST be converted from a Unicode (UTF16) string to a UTF8 string ([UNICODE], chapter 3.9). KILE concatenates the following information to use as the key salt for principals:

- User accounts: < DNS of the realm, converted to upper case> | <user name>
- Computer accounts: < DNS name of the realm, converted to upper case > | "host" | < computer name, converted to lower case with trailing "$" stripped off > | "." | < DNS name of the realm, converted to lower case >

Using KILE, application clients (for example, CIFS/SMB clients) MAY use the negotiated key directly. When an application client uses the session key, the application protocol MUST document the explicit use of the key in its protocol specification. The key MAY be exported as an attribute of the completed security context in the **SSPI** API.

The subkey in the **EncAPRepPart** of the KRB_AP_REP message SHOULD be used as the session key when MutualAuthentication is requested. (The KRB_AP_REP message and its fields are defined in section 5.5.2 of [RFC4120].) When DES and RC4 are used, the implementation is as described in [RFC1964]. With DES and RC4, the subkey in the KRB_AP_REQ message can be used as the session key, as it is the same as the subkey in KRB_AP_REP message; however when AES is used (see [RFC4121]), the subkeys are different and the subkey in the KRB_AP_REP SHOULD be used. (The KRB_AP_REQ message is defined in section 5.5.1 of [RFC4120]).

#### 3.1.1.3 Ticket Cache

Kerberos V5 specifies that clients MAY cache TGTs ([RFC4120] section 3.3.1).

KILE implements a ticket cache that preserves service tickets and TGTs.<7>

#### 3.1.1.4 Machine ID

KILE implements a 32-byte binary random string machine ID. <8>

#### 3.1.1.5 SupportedEncryptionTypes

KILE implements a 32-bit unsigned integer that contains a combination of flags that specify what encryption types (section 2.2.6) are supported by Kerberos.<9> The default is 0000001C.<10>

#### 3.1.1.6 Kerberos OID

Kerberos V5 specifies the Kerberos principal name form ([RFC1964] section 2.1.1). KILE also implements a truncated Kerberos OID value: (1.2.840.48018.1.2.2)

### 3.1.2 Timers

None.

### 3.1.3 Initialization

The random number generator for keys and nonces is initialized by other components but complies with [FIPS140] section 4.7.1.

A machine ID (section 3.1.1.4) is created at computer startup.

### 3.1.4 Higher-Layer Triggered Events

None.

### 3.1.5 Message Processing Events and Sequencing Rules

The following sections detail variations in tickets and naming that are common to all parts of the Kerberos protocol.

### 3.1.5.1 Pre-authentication Data

Pre-authentication ([RFC4120] sections 3.1.1, 5.4.1, and 5.2.7) is an extensibility point for the Kerberos V5 protocol. Pre-authentication is performed by supplying one or more pre-authentication messages in the PA-data field of the AS-REQ and AS-REP messages.

KILE supports the following pre-authentication types described in ([RFC4120] section 7.5.2):

- PA-TGS-REQ [1]
- PA-ENC-TIMESTAMP [2]
- PA-ETYPE-INFO [11]
- PA-PK-AS-REQ_OLD [14]
- PA-PK-AS-REP_OLD [15]
- PA-PK-AS-REQ [16]
- PA-PK-AS-REP [17]
- PA-ETYPE-INFO2 [19]
- PA-PAC-REQUEST [128]

KILE supports the following pre-authentication types described in ([Referrals-11] Appendix A):

- PA-SVR-REFERRAL-INFO [20]

KILE adds the following pre-authentication type:

- PA-SUPPORTED_ENCTYPES [165] (section 2.2.7)

Unknown pre-authentication types MUST be ignored by KDCs.

When clients perform a password-based initial authentication, they MUST supply the PA-ENC-TIMESTAMP pre-authentication type when they construct the initial AS request. They SHOULD request, via the PA-PAC-REQUEST pre-authentication type, that a privilege attribute certificate (PAC) be included in issued tickets.

If the KDC does not receive the required pre-authentication message in the AS exchange, an error MUST be returned to the client. The exact error depends on what pre-authentication types were supplied.

### 3.1.5.2 Encryption Types

KILE SHOULD support the Advanced Encryption Standard (AES) encryption types:<11>

- AES256-CTS-HMAC-SHA1-96 [18] ([RFC3962] section 7)
- AES128-CTS-HMAC-SHA1-96 [17] ([RFC3962] section 7)

and MAY<12> support the other following encryption types, which are listed in order of relative strength:

- RC4-HMAC [23] [RFC4757]<13>
- RC4-HMAC-EXP [24] [RFC4757]<14>

- DES-CBC-MD5 [3] [RFC3961]<15>
- DES-CBC-CRC [1] [RFC3961]<16>

Kerberos V5 encryption type assigned numbers are specified in [RFC3961] section 8, [RFC4757] section 5, and [RFC3962] section 7.<17>

#### 3.1.5.3 Encryption Checksum Types

KILE supports the following checksum types. Each checksum type is described, and a number is specified, in the corresponding RFC.

- CRC32 [1] [RFC3961]
- rsa-md4 [2] [RFC3961]
- rsa-md4-des [3] [RFC3961]
- des-mac [4] [RFC3961]
- des-mac-k [5] [RFC3961]
- rsa-md4-des-k [6] [RFC3961]
- rsa-md5 [7] [RFC3961]
- rsa-md5-des [8] [RFC3961]
- sha1 (unkeyed) [-131] [RFC3961]
- hmac-sha1-96-aes128 [15] [RFC3962]
- hmac-sha1-96-aes256 [16] [RFC3962]
- hmac-md5-string [-138] [RFC4757]

#### 3.1.5.4 Ticket Flag Details

The Kerberos V5 protocol specifies a number of options and behaviors with regard to the flags ([RFC4120] section 2) that are encoded in a ticket.

KILE implements the following ticket flags:

- The INITIAL and PRE-AUTHENT flags ([RFC4120] section 2.1): By default, KDCs require pre-authentication when they issue tickets. Clients SHOULD pre-authenticate. KDCs MUST enforce pre-authentication. Therefore, unless the account has been explicitly set to not require Kerberos pre-authentication, the ticket will have the PRE-AUTHENT flag set.
- The HW-AUTHENT flag ([RFC4120] section 2.1): This flag was originally intended to indicate that hardware-supported authentication was used during pre-authentication. This flag is no longer recommended in the Kerberos V5 protocol. KDCs MUST NOT issue a ticket with this flag set. KDCs SHOULD NOT preserve this flag if it is set by another KDC.
- The RENEWABLE flag ([RFC4120] section 2.3): Renewable tickets SHOULD be supported in KILE.
- The POSTDATED/MAY-POSTDATE flag ([RFC4120] section 2.4): Postdated tickets SHOULD NOT be supported in KILE.

- The PROXY/PROXIABLE flag ([RFC4120] section 2.5): Proxiable tickets SHOULD NOT be supported in KILE.
- The FORWARDABLE/FORWARDED flag ([RFC4120] section 2.6): Forwarded tickets SHOULD be supported in KILE.
- The TRANSITED-POLICY-CHECKED flag ([RFC4120] section 2.7): KILE MUST NOT check for transited domains on servers or a KDC. Application servers MUST ignore the TRANSITED-POLICY-CHECKED setting.
- The OK-AS-DELEGATE flag ([RFC4120] section 2.8): The KDC MUST set the OK-AS-DELEGATE flag if the service account is trusted for delegation (section 3.3.1.1). For more information, see [ADDLG].

#### 3.1.5.5 Other Elements and Options

The Kerberos V5 protocol defines optional authorization data elements ([RFC4120] section 5.2.6).

KILE has added the following elements:

- AD-AUTH-DATA-AP-OPTIONS (section 3.2.5.6).
- KERB_AUTH_DATA_TOKEN_RESTRICTIONS (sections 3.2.5.6 and 3.4.5.3).

KILE SHOULD NOT support the following elements:

- The AD-KDC-ISSUED element ([RFC4120] section 5.2.6.2).
- The AD-AND-OR element ([RFC4120] section 5.2.6.3).
- The AD-MANDATORY-FOR-KDC element ([RFC4120] section 5.2.6.4).

KILE SHOULD NOT fail on unknown authorization data ([RFC4120] section 1.5.1). The server SHOULD NOT generate an error; instead, it SHOULD ignore the unknown data and proceed to authenticate the client.

KILE MUST support the KRB_ERR_RESPONSE_TOO_BIG error message ([RFC4120] section 7.2.1).

#### 3.1.5.6 Addressing

KILE SHOULD support IPv6 addresses ([RFC4120] section 7.1).<18>

KILE MUST NOT support directional addresses ([RFC4120] section 7.1). If the directional addresses are present, they MUST be ignored.

#### 3.1.5.7 Internationalization and Case Sensitivity

The Kerberos V5 protocol specifies rules for encoding and processing names, both for character set and case ([RFC4120] section 6).

Name comparisons, whether for users or **domains**, MUST NOT be case sensitive in KILE. KILE MUST use UTF-8 encoding of these names [RFC2279]. Normalization MUST NOT be performed and surrogates MUST NOT be supported.To match names, the GetWindowsSortKey algorithm ([MS-UCODEREF] section 3.1.5.2.4) with the following flags NORM_IGNORECASE, NORM_IGNOREKANATYPE, NORM_IGNORENONSPACE, and NORM_IGNOREWIDTH SHOULD be used then the CompareSortKey algorithm ([MS-UCODEREF] section 3.1.5.2.2) SHOULD be used to compare the names.Note that this applies only to names; passwords (and the transformation of a

password to a key) are governed by the actual key generation specification ([RFC4120], [RFC4757], and [RFC3962]).

#### 3.1.5.8 Key Version Numbers

The Kerberos V5 protocol specifies key version numbers ([RFC4120] section 5.2.9). Key version numbers are used in the Kerberos V5 protocol to distinguish between different keys in the same domain.

KILE supports key version numbers for **read-only domain controllers (RODCs)**. Each RODC will have a different key version number.<19> This allows the domain controller to distinguish between keys that are issued to different RODCs.

The key version number consists of 32 bits. The first 16 bits SHOULD identify the RODC and the remaining 16 bits SHOULD be the version number of the key.

#### 3.1.5.9 Key Usage Numbers

The Kerberos V5 protocol specifies key usage numbers ([RFC4120] section 7.5.1).

Kerberos Protocol Extensions  define the following additional Key Usage Numbers:

- KERB_NON_KERB_SALT [16]
- KERB_NON_KERB_CKSUM_SALT [17]

#### 3.1.5.10 Referrals

The Kerberos V5 protocol specifies cross-**realm** behavior and the nature of referrals ([RFC4120] section 1.2).

KILE MUST support cross-realm referrals ([RFC4120] sections 1.2 and 3.3.1) and extended referrals [Referrals-11].

#### 3.1.5.11 PAC Generation

The PAC [MS-PAC] MUST be generated by the KDC under one of the following conditions:

- During an Authentication Service (AS) request that has been validated with pre-authentication.
- During a TGS request when the TGT for the client in the request does not contain a PAC and the ticket to be returned is a cross-realm referral TGT ([RFC4120] section 1.2).
- During a TGS request when the client has domain local groups.

The KDC MUST collect the user's initial set of group information and add it to the PAC in the TGT.

The PAC MUST be subsequently updated when the client requests a **service ticket** to contain additional domain local groups that are specific to the server's domain.

By default, the KDC MUST generate a PAC. However, a client MAY explicitly request that a PAC be excluded through the use of a KERB-PA-PAC-REQUEST PA-DATA type (2.2.2).

#### 3.1.5.12 Naming

Kerberos V5 specifies a variety of name types ([RFC4120] section 7.5.8) for specifying the name of the server during a TGS request.

KILE SHOULD use service principal names (SPNs) to identify servers in TGS-REQs. An SPN is a single-string representation of a Kerberos principal name according to section 2.1.1 of [RFC1964] that identifies the server. The Directory Service attribute **servicePrincipalName**, as defined in [MS-ADA3] section 2.252, is a multi-value attribute on a user or computer object that contains a list of service principal names, with each list item corresponding to a string representation of a Kerberos name that can be used to identify the server.

An SPN is a string of the following format. For more information on the <alphanum> element, see [RFC2396] section 1.6.

```
SPN = serviceclass "/" hostname [":"port] ["/" servicename]
serviceclass = alphanum
servicename = alphanum
```

Where:

- *serviceclass* is a string that identifies the class of the service, such as "www" for a Web service or "ldap" for a directory service.
- *hostname* ([RFC2396] section 3.2.2) is a string that is the name of the system. This SHOULD be the fully qualified domain name (FQDN).
- *port* ([RFC2396] section 3.2.2) is a number that is the port number for the service.
- The *servicename* segment is a string that is the **distinguished name (DN)**, **objectGuid**, **Internet host name**, or **fully qualified domain name (FQDN)** for the service.

An application can supply a name of the form "RestrictedKrbHost/<hostname>" when its callers have provided the hostname but not the correct SPN for the service. Applications SHOULD NOT use "RestrictedKrbHost/<hostname>" due to the security considerations in section 5.1.2. Applications calling GSS-API directly MUST provide the SPN for their service applications for Kerberos authentication.<20>

### 3.1.6 Timer Events

KILE introduces no timer events.

### 3.1.7 Other Local Events

KILE introduces no local events.

### 3.1.8 Implementing Public Keys

The use of public keys in KILE is specified in [MS-PKCA].

## 3.2 Client Details

### 3.2.1 Abstract Data Model

After a connection is established through the AP exchange, Kerberos V5 does not directly influence the application protocol. The client parameters MUST be set when establishing a security context that supports the signing or encryption of messages. The higher-layer application protocol will invoke the per-message functions. The following parameters are logically available for the application to set. These logical parameters can influence various protocol-defined flags.

**Note** The following variables are logical, abstract parameters that an implementation MUST maintain and expose to provide the proper level of service. How these variables are maintained and exposed is up to the implementation.

**ChannelBinding:** A Boolean setting that indicates the caller's channel binding information ([RFC2743] section 1.1.6 and [RFC2744]). <21>

**Confidentiality:** A Boolean setting that indicates that the caller is requiring encryption of messages so that they cannot be read while in transit.

**DatagramStyle:** A Boolean setting that indicates that the caller is requiring the use of **Datagram** semantics (section 3.4.5.2).

**DCE Style:** A Boolean setting that indicates that the caller requires three-leg, DCE Style authentication ([MS-RPCE] and [C706]).

**Delegate:** A Boolean setting that indicates that the caller is requiring the use of forwardable tickets.

**ExtendedError:** A Boolean setting that indicates that the caller requires additional error handling, possibly including retries, with the context of the GSS exchange in progress.

**Identify:** A Boolean setting that indicates that the caller shares its identity with the server but does not allow the server to impersonate the caller to resources on that system.

**Integrity:** A Boolean setting that indicates that the caller has elected to sign messages so that they cannot be tampered with while in transit.

**MessageBlockSize:** An integer that indicates the minimum size of the input_message for GSS_WrapEx (section 3.4.5.4). The size of the input_message MUST be a multiple of this value. This value depends on the encryption type:

- For AES, the value equals the message block size ([RFC3962] section 6)
- For RC4, it equals 1 ([RFC4757] section 7.3)
- For DES, it equals 8 ([RFC1964] section 1.2.2.3)

**MutualAuthentication:** A Boolean setting that indicates that the client requires authentication of the server. Even with this flag, mutual authentication cannot be assured until the first message is passed by the application protocol and the message is signed or encrypted.

**ReplayDetect:** A Boolean setting that indicates that the caller requires replay detection so that the application can determine when messages are replayed.

**SequenceDetect:** A Boolean setting that indicates that the caller requires sequence detection so that messages cannot be received out of order.

**UseSessionKey:** A Boolean setting that indicates that the caller requests user-to-user authentication exchanges ([RFC4120] section 3.7).

### 3.2.2 Timers

When the client sends an AS-REQ or TGS-REQ to the KDC, it uses a timer to determine when to retry. The operation of this timer, along with its default values, is as specified in section 3.2.6.

### 3.2.3 Initialization

Before the client can send an AS or TGS message, it MUST discover the KDC to which the AS or TGS message will be sent. Clients SHOULD use **SRV record** discovery ([RFC4120] section 7.2.3.2) by default. When SRV record discovery is not supported by KDCs, clients can use a list of KDCs for a specified realm.

If the client has a ticket cache, the ticket cache MUST be initialized to an empty state.

All parameters that are specified in section 3.2.1 are reset and then set according to the higher-layer protocols request.

### 3.2.4 Higher-Layer Triggered Events

#### 3.2.4.1 Initial Logon

Initial logon is the process by which a user first authenticates to the KDC. The client engages in an AS exchange (see section 1.3.2) with the KDC, using domain password or smartcard authentication and receives a TGT and session key. The TGT and session key are then used in subsequent protocol exchanges with the KDC in requesting service tickets.

The client SHOULD request a service ticket to its own workstation during initial logon from the KDC because the service ticket contains information about the logged on user contained in the user's PAC within the service ticket. The client can use the information in that PAC for access control purposes.

Standard Kerberos requires that the user principal name (UPN) refers to a valid domain the KDC defines (for example, user@windows.example.com). KILE SHOULD allow authentication with valid AD DS UPNs ([MS-ADTS] section 5.1.1.1.1).

#### 3.2.4.2 Authentication to Services

When the initial authentication is complete and the TGT is obtained, the user typically wants to use a network resource. For a Kerberos-aware application, the Kerberos client initiates a **TGS exchange** requesting a service ticket to the named service, for example, "host/hostname.domain.name".

The Kerberos client then initiates an AP exchange which MAY be encoded in a GSS–API style wrapper, if the Kerberos-aware application requests it.

KILE provides no support for direct access to the Kerberos **KRB_SAFE** or **KRB_PRIV** messages.

The client application then takes the AP message and supplies it, in band with the application protocol, to the server. The Kerberos server processes the message as specified in [RFC4120] and completes the connection. The AP exchange is covered further in section 3.4.

### 3.2.5 Message Processing Events and Sequencing Rules

#### 3.2.5.1 Request Flags Details

Kerberos V5 specifies Kerberos ticket-issuing behavior defined by a set of options that are passed to the KDC during the AS exchange or TGS exchange.

Clients SHOULD set the canonicalize flag ([RFC4120] section 5.4.1).

If Delegate is set to TRUE, the client SHOULD set the FORWARDABLE option in the TGS request. When the client receives a forwardable ticket, it puts the ticket in a KRB_CRED structure ([RFC4120]

section 3.6). The client SHOULD NOT forward the ticket unless the TGT is marked OK-AS-DELEGATE ([RFC4120] section 2.8).

If MutualAuthentication is set to TRUE, the client SHOULD set the MUTUAL-REQUIRED flag in the KRB_AP_REQ message ([RFC4120] sections 3.2.2 and 3.2.4).

#### 3.2.5.2 Authenticator Checksum Flags

If the following variables are set to TRUE, the client SHOULD set the corresponding GSS flag ([RFC4121] section 4.1.1) to TRUE in the authenticator's checksum ([RFC4121] section 4.1.1):

- *Confidentiality*: GSS_C_CONF_FLAG ([RFC1964] section 1.1.1).
- *Delegate*: GSS_C_DELEGE_FLAG.
- *ExtendedError*: GSS_C_EXTENDED_ERROR_FLAG ([RFC4757] section 7.1).
- *Identify*: GSS_C_IDENTIFY_FLAG ([RFC4757] section 7.1) and set in the GSS_Init_sec_context call ([RFC4757] section 7.1).
- *Integrity*: GSS_C_INTEG_FLAG ([RFC1964] section 1.1.1).
- *MutualAuthentication*: GSS_C_MUTUAL_FLAG ([RFC1964] section 1.1.1).
- *ReplayDetect*: GSS_C_REPLAY_FLAG ([RFC1964] section 1.1.1).
- *SequenceDetect*: GSS_C_SEQUENCE_FLAG ([RFC1964] section 1.1.1).

#### 3.2.5.3 AS Exchange

The Kerberos V5 protocol specifies the AS exchange ([RFC4120] section 3.1). KILE also supports extensions to the AS exchange as specified in [Referrals-11], [RFC5349], [RFC4556], and [MS-PKCA].

The client will always include a PAC request PA-data type when generating an AS-REQ message. The PAC is specified in [MS-PAC].

#### 3.2.5.4 Forwardable TGT Request

When the client requests a forwardable TGT ([RFC4120] Section 2.6) for the application server, the client SHOULD<22>:

- Set the **etype** field of the TGS-REQ to the contents of the **keytype** field in the previous TGS-REP to specify the common encryption type.
- Provide a PA-SUPPORTED-ENCTYPES value for padata, based on the encryption types mutually supported by the KDC and the application server for the session key with the delegated TGT. The client uses the KDC encryption types provided in the AS-REP from the KDC and the application server encryption types provided in the previous TGS-REP for the application server.

#### 3.2.5.5 TGS Exchange

When the server name is not Krbtgt, the client SHOULD send an authorization data field ([RFC4120] section 5.2.6) with ad-type KERB-LOCAL (142) and ad-data containing KERB-LOCAL structure (section 2.2.3) in an AD-IF-RELEVANT element ([RFC4120] section 5.2.6.1) in the enc-authorization-data field ([RFC4120] section 5.2.6).<23>

#### 3.2.5.6 AP Exchange

If *UseSessionKey* is set to TRUE, the client SHOULD set the USE-SESSION-KEY flag to TRUE in the ap-options field of the AP-REQ ([RFC4120] section 5.5.1).

When the server name is not Krbtgt, the client SHOULD send an AP request as an authorization data field ([RFC4120] section 5.2.6), initialized as follows:

- ad-type KERB-LOCAL (142) and ad-data containing KERB-LOCAL structure (section 2.2.3).<24>
- KERB_AUTH_DATA_TOKEN_RESTRICTIONS (141), containing the KERB-AD-RESTRICTION-ENTRY structure (section 2.2.5).<25>

If ChannelBinding is set to TRUE, the client SHOULD send AD-AUTH-DATA-AP-OPTIONS data in an AD-IF-RELEVANT element ([RFC4120] section 5.2.6.1). The Authorization Data Type AD-AUTH-DATA-AP-OPTIONS has an ad-type of 143 and ad-data of KERB_AP_OPTIONS_CBT (0x4000). The presence of this element indicates that the client expects the applications running on it to include channel binding information ([RFC2743] section 1.1.6 and [RFC2744]) in AP requests whenever Kerberos authentication takes place over an "outer channel" such as TLS. Channel binding is provided using the *ChannelBinding* parameter specified in section 3.2.1.

When the client receives a KRB_AP_ERR_SKEW error ([RFC4120] section 3.2.3) with a KERB-ERROR-DATA structure (section 2.2.1) in the e-data field of the KRB-ERROR message ([RFC4120] section 5.9.1), the client SHOULD retry the AP-REQ using the time in the KRB-ERROR message ([RFC4120] section 5.9.1) to create the authenticator ([RFC4120] section 5.5.1).

### 3.2.6 Timer Events

The Kerberos V5 protocol requires the client to contact the KDC and recognizes that a specific KDC could be offline or unavailable to service the request. The actual behavior is not specified in [RFC4120]; these behavior details are determined by the implementation. Detection of a KDC's failure to reply requires a timer. Clients can use the initial time-out and increase the time-out by some interval to retry multiple times before failing the AS-REQ or TGS-REQ message.<26>

### 3.2.7 Other Local Events

KILE introduces no local events.

## 3.3 KDC Details

### 3.3.1 Abstract Data Model

KILE uses the abstract data model and default values specified in Kerberos V5, except for the following default configuration values. KILE implementations, which use the LSAD for the configuration database, SHOULD set:

- **Minimum lifetime** ([RFC4120] section 8.2): 0 minutes.
- **Maximum renewable lifetime** ([RFC4120] section 8.2): A 64-bit signed integer shared from the **MaxRenewAge** field in the Kerberos Policy Information ([MS-LSAD] section 3.1.1.1)
- **Acceptable clock skew** ([RFC4120] section 8.2): A 64-bit signed integer shared from the **MaxClockSkew** field in the Kerberos Policy Information ([MS-LSAD] section 3.1.1.1).

KILE also adds the following new KDC configuration settings:

- **MaximumServiceTicketLifetime:** A 64-bit signed integer shared from the **MaxServiceTicketAge** field in the Kerberos Policy Information ([MS-LSAD] section 3.1.1.1). The default is 10 hours.
- **MaximumTGTLifetime:** A 64-bit signed integer shared from the **MaxTicketAge** field in the Kerberos Policy Information ([MS-LSAD] section 3.1.1.1). The default is 10 hours.
- **AuthenticationOptions:** A 32-bit unsigned integer shared from the **AuthenticationOptions** field in the Kerberos Policy Information ([MS-LSAD] section 3.1.1.1). Only the POLICY_KERBEROS_VALIDATE_CLIENT flag is supported and SHOULD be set by default.

KILE implementations that use an Active Directory for the account database SHOULD support the following variables:

- **NetbiosServerName:** The NetBIOS name for the server. This Abstract Data Model element is shared with **ComputerName.NetBIOS** ([MS-DISO]).
- **NetbiosDomainName:** The NetBIOS domain name for the domain to which the server belongs. This Abstract Data Model element is shared with **DomainName.NetBIOS** ([MS-DISO]).
- **DomainSid:** A security identifier for the domain. This Abstract Data Model element is shared with **DomainSid** ([MS-DISO]).

#### 3.3.1.1 Account Database Extensions

The Kerberos V5 protocol specifies which KDCs MUST maintain a database of principals with their secret keys and corresponding supported encryption types:

- Secret keys: KILE implementations that use an **Active Directory** for the accountdatabase SHOULD use the **supplementalCredentials** attribute ([MS-ADA3] section 2.286).
- KerbSupportedEncryptionTypes: A 32-bit unsigned integer that contains a combination of flags that specify what encryption types (section 2.2.6) are supported by the application server.<27> KILE implementations that use an Active Directory for the accountdatabase SHOULD use the **msDS-SupportedEncryptionTypes** attribute ([MS-ADA2] section 2.324).

To support all functionality of KILE, the account database MUST be extended to support the following additional information for each principal:

- AuthorizationDataNotRequired: A Boolean setting to control when to include a PAC in the service ticket. KILE implementations that use an Active Directory for the account database SHOULD use the userAccountControl attribute ([MS-ADTS] section 2.2.15) NA flag. The default is FALSE.
- DelegationNotAllowed: A Boolean setting to prevent PROXIABLE or FORWARDABLE ticket flags ([RFC4120] sections 2.5 and 2.6) in tickets for the principal. KILE implementations that use an Active Directory for the account database SHOULD use the userAccountControl attribute ([MS-ADTS] section 2.2.15) ND flag. The default is FALSE.
- Disabled: A Boolean setting to control when the account is disabled. KILE implementations that use an Active Directory for the account database SHOULD use the userAccountControl attribute ([MS-ADTS] section 2.2.15) D flag. The default is FALSE.
- Expired: A Boolean setting to control when the password has expired. KILE implementations that use an Active Directory for the account database SHOULD use the userAccountControl attribute ([MS-ADTS] section 2.2.15) PE flag. The default is FALSE.

- GroupMembership: A list of GROUP_MEMBERSHIP ([MS-PAC] section 2.2.2) structures that contain the groups to which the account belongs in the realm.
- Locked: A Boolean setting to control when the account is locked out. KILE implementations that use an Active Directory for the account database SHOULD use the userAccountControl attribute ([MS-ADTS] section 2.2.15) L flag. The default is FALSE.
- LogonHours: A binary value with the structure SAMPR_LOGON_HOURS ([MS-SAMR] section 2.2.7), indicating a logon policy describing the time periods during which the user can authenticate. KILE implementations that use an Active Directory for the account database SHOULD use the **logonHours** attribute ([MS-ADA1] section 2.376).
- PasswordMustChange: A FILETIME value indicating when the password must change. Setting to 0x7FFFFFFF FFFFFFFF never requires password change. KILE implementations that use an Active Directory for the account database SHOULD generate the value with the same method as the SAM ([MS-SAMR] section 3.1.5.14.4). The default is 0.
- Pre-AuthenticationNotRequired: A Boolean setting to control when pre-authentication data is required. KILE implementations that use an Active Directory for the account database SHOULD use the userAccountControl attribute ([MS-ADTS] section 2.2.15) DR flag. The default is 0.
- TrustedForDelegation: A Boolean setting to control when to set the OK-AS-DELEGATE ticket flag ([RFC4120] section 2.8) in tickets for the principal. KILE implementations that use an Active Directory for the account database SHOULD use the userAccountControl attribute ([MS-ADTS] section 2.2.15) TD flag. The default is FALSE.
- UseDESOnly: A Boolean setting to control when only the des-cbc-md5 and/or des-cbc-crc keys [RFC3961] are used in the Kerberos exchanges for this account. KILE implementations that use an Active Directory for the account database SHOULD use the userAccountControl attribute ([MS-ADTS] section 2.2.15) DK flag. The default is FALSE.

For KILE implementations that use an Active Directory for the account database, the previous Boolean settings are accessible in the **userAccountControl** attribute ([MS-ADTS] section 2.2.15):

- D flag: Disabled
- DK flag: UseDESOnly
- DR flag: Pre-AuthenticationNotRequired
- L flag: Locked
- NA flag: AuthorizationDataNotRequired
- ND flag: DelegationNotAllowed
- PE flag: Expired
- TA flag: TrustedToAuthenticationForDelegation
- TD flag: TrustedForDelegation

### 3.3.2 Timers

There are no KDC timers.

### 3.3.3 Initialization

Kerberos V5 specifies that all KDCs in a domain MUST have the same key, and the name of the service for the TGS is "krbtgt/domain-name" SPN ([RFC4120] section 6.2).

KILE implementations that use the LSAD for the configuration database load the KDC configuration from the Kerberos Policy Information ([MS-LSAD] section 3.1.1.1). The KDC SHOULD call the LsarQueryDomainInformation Policy method ([MS-LSAD] section 3.1.4.4.7) and the *InformationClass* parameter SHOULD be PolicyDomainKerberosTicketInformation to retrieve the current values. The KDC SHOULD set its configuration settings as follows:

- Maximum renewable lifetime ([RFC4120] section 8.2) to the value of the **MaxRenewAge** field.
- Acceptable clock skew ([RFC4120] section 8.2) to the value of the **MaxClockSkew** field.
- MaximumServiceTicketLifetime to the value of the **MaxServiceTicketAge** field.
- MaximumTGTLifetime to the value of the **MaxTicketAge** field.

Implementations of KILE KDCs which use an AD for the account database MUST use the krbtgt account in the AD.

If the KDC has a ticket replay cache, it MUST be reset when the KDC starts up.

If the KDC has a ticket cache, the ticket cache MUST be initialized to an empty state.

### 3.3.4 Higher-Layer Triggered Events

For KILE implementations which use the LSAD for the configuration database, a KDC ConfigurationChange event is triggered whenever the KDC configuration policy is changed in the LSAD database.

#### 3.3.4.1 KDC Configuration Changes

If an implementation supports multiple KDCs for a realm, then it SHOULD have a mechanism for keeping the KDC configuration database consistent across all the KDCs. KDC configuration change details are determined by the implementation.

When KILE implementations that use the LSAD for the configuration database receive a KDC ConfigurationChange event, the KDC SHOULD call the **LsarQueryDomainInformation Policy** method (as specified in **[MS-LSAD] (section 3.1.4.4.7)**, and the *InformationClass* parameter SHOULD be PolicyDomainKerberosTicketInformation to retrieve the current values. The KDC SHOULD set its configuration settings as follows:

- Maximum renewable lifetime ([RFC4120] section 8.2) to the value of the **MaxRenewAge** field.
- Acceptable clock skew ([RFC4120] section 8.2) to the value of the **MaxClockSkew** field.
- MaximumServiceTicketLifetime to the value of the **MaxServiceTicketAge** field.
- MaximumTGTLifetime to the value of the **MaxTicketAge** field.

### 3.3.5 Message Processing Events and Sequencing Rules

#### 3.3.5.1 Request Flag Ticket-issuing Behavior

Kerberos V5 specifies Kerberos ticket-issuing behavior defined by the kdc-options ([RFC4120] section 5.4.1) that are passed to the KDC during the AS or TGS exchange.

KILE KDCs SHOULD<28> ignore the canonicalize flag except for referrals [Referrals-11].

Canonicalization was designed to allow aliasing for principals. This allowed the client to request a ticket to "cifs/hostname" and the KDC to issue a ticket to "host/hostname" which allowed for exposing the "true" name of the principal. This behavior resulted in inefficiencies and confusion for several reasons:

- The client ticket cache became unusable because all the tickets were named "host/hostname" and a cache lookup for "cifs/hostname" never succeeded.
- Third-party implementations of the Kerberos-aware applications that used the Kerberos protocol expected the name in the ticket to match the requested name and ran into problems when they did not. This confusion was mitigated by disabling strict name checking in the third-party implementations when they interoperate with older versions of KILE.

Kerberos V5 specifies Kerberos TicketFlags ([RFC4120] Section 5.3) that can be set by the KDC on tickets.

KILE KDCs use the following account variables to enforce TicketFlags:

- If DelegationNotAllowed is set to TRUE on the principal, the KILE KDC MUST NOT set the PROXIABLE or FORWARDABLE ticket flags ([RFC4120] sections 2.5 and 2.6).
- If TrustedForDelegation is set to TRUE on the principal, the KILE KDC MUST set the OK-AS-DELEGATE ticket flag ([RFC4120] section 2.8).

#### 3.3.5.2 User Account Objects Without UPN

If the user account object does not have the **userPrincipalName** attribute ([MS-ADA3] section 2.348) set, the KDC SHOULD send a UPN_DNS_INFO structure ([MS-PAC] section 2.10) containing a **user principal name (UPN)**, constructed by concatenating the user name, the "@" symbol, and the **DNS** name of the domain. <29>

#### 3.3.5.3 AS Exchange

Kerberos V5 specifies the AS exchange ([RFC4120] section 3.1). KILE also supports extensions to the AS exchange specified in [Referrals-11], [RFC5349], [RFC4556], and [MS-PKCA].

If Pre-AuthenticationNotRequired is set to TRUE on the principal, the KDC MUST issue a TGT without validating pre-authentication data ([RFC4120] section 7.5.2) provided.

The KDC SHOULD<30> return in the encrypted part of the AS-REP message PA-DATA with padata-type set to PA-SUPPORTED-ENCTYPES (165), to indicate what encryption types are supported by the KDC.

The KDC SHOULD check whether the krbtgt account has the UseDESOnly flag:

- If the UseDESOnly flag is set: the KDC SHOULD, in the encrypted pre-auth data part ([Referrals-11], Appendix A) of the AS-REP message, include PA-DATA with the padata-type set to PA-SUPPORTED-ENCTYPES (165), and the padata-value set to 0x3 (section 2.2.6).

- Otherwise:
  - If domainControllerFunctionality returns a value < 3 ([MS-ADTS] section 3.1.1.3.2.25): the KDC SHOULD, in the encrypted pre-auth data part ([Referrals-11], Appendix A) of the AS-REP message, include PA-DATA with the padata-type set to PA-SUPPORTED-ENCTYPES (165), and the padata-value set to 0x7 (section 2.2.6).
  - If domainControllerFunctionality returns a value >= 3: the KDC SHOULD, in the encrypted pre-auth data part ([Referrals-11], Appendix A) of the AS-REP message, include PA-DATA with the padata-type set to PA-SUPPORTED-ENCTYPES (165), and the padata-value set to 0x1F (section 2.2.6).

#### 3.3.5.3.1 Referrals

The KDC supports referral processing [Referrals-11], sending a KDC and domain to use to answer a client's request.

KILE concatenates the following information to use as the key salt for realm trusts:

- Inbound trusts: <all upper case name of the remote realm> | "krbtgt" | <all upper case name of the local realm>
- Outbound trusts: <all upper case name of the local realm> | "krbtgt" | <all upper case name of the remote realm>

#### 3.3.5.3.2 Initial Population of the PAC

For KILE implementations that use an Active Directory for the account database, the KDC will create a PAC. During processing of the AS request, the KDC searches Active Directory for the user or computer account that matches the cname that was sent in the AS-REQ message. The KDC then creates the PAC structure [MS-PAC] and encodes that into the TGT using the AD-IF-RELEVANT element ([RFC4120] section 5.2.6.1).

##### 3.3.5.3.2.1 KERB_VALIDATION_INFO Structure

For KILE implementations that use an Active Directory for the account database, KDCs SHOULD retrieve the following attributes from local directory service instance with the same processing rules as defined in **SamrQueryInformationUser2()** ([MS-SAMR] section 3.1.5.5.5) message processing. The KDC populates the returned KERB_VALIDATION_INFO structure ([MS-PAC] section 2.5) fields as follows:

- The **LogonTime** field SHOULD be set to the Buffer.SAMPR_USER_ALL_INFORMATION.LastLogon field ([MS-SAMR] section 2.2.7.1) of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.
- The **LogoffTime** field SHOULD be set to the earlier of the Buffer.SAMPR_USER_ALL_INFORMATION.LogonHours field ([MS-SAMR] section 2.2.7.1) or the Buffer.SAMPR_USER_ALL_INFORMATION.AccountExpires field ([MS-SAMR] section 2.2.7.1) of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.
- The **KickOffTime** field SHOULD be set to the **LogoffTime** + the Buffer.SAMPR_USER_ALL_INFORMATION.ForceLogoff field ([MS-SAMR] section 2.2.7.1) of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.

- The **PasswordLastSet** field SHOULD be set to the Buffer.SAMPR_USER_ALL_INFORMATION.PasswordLastSet field ([MS-SAMR] section 2.2.7.1) of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.
- The **PasswordCanChange** field SHOULD be set to the Buffer.SAMPR_USER_ALL_INFORMATION.PasswordCanChange field ([MS-SAMR] section 2.2.7.1) of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.
- The **PasswordMustChange** field SHOULD be set to the Buffer.SAMPR_USER_ALL_INFORMATION.PasswordMustChange field ([MS-SAMR] section 2.2.7.1) of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.
- The **EffectiveName** field SHOULD be set to the Buffer.SAMPR_USER_ALL_INFORMATION.UserName field ([MS-SAMR] section 2.2.7.1) of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.
- The **FullName** field SHOULD be set to the Buffer.SAMPR_USER_ALL_INFORMATION.FullName field ([MS-SAMR] section 2.2.7.1) of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.
- The **LogonScript** field SHOULD be set to the Buffer.SAMPR_USER_ALL_INFORMATION.ScriptPath field ([MS-SAMR] section 2.2.7.1) of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.
- The **ProfilePath** field SHOULD be set to the Buffer.SAMPR_USER_ALL_INFORMATION.ProfilePath field ([MS-SAMR] section 2.2.7.1) of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.
- The **HomeDirectory** field SHOULD be set to the Buffer.SAMPR_USER_ALL_INFORMATION.HomeDirectory field ([MS-SAMR] section 2.2.7.1) of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.
- The **HomeDirectoryDrive** field SHOULD be set to the Buffer.SAMPR_USER_ALL_INFORMATION.HomeDirectoryDrive ([MS-SAMR] section 2.2.7.1) of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.
- The **LogonCount** field SHOULD be set to the Buffer.SAMPR_USER_ALL_INFORMATION.LogonCount ([MS-SAMR] section 2.2.7.1) of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.
- The **BadPasswordCount** field SHOULD be set to the Buffer.SAMPR_USER_ALL_INFORMATION.BadPasswordCount field ([MS-SAMR] section 2.2.7.1) of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.
- The **UserID** field SHOULD be set to the Buffer.SAMPR_USER_ALL_INFORMATION.UserId field ([MS-SAMR] section 2.2.7.1) of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.
- The **PrimaryGroupId** field SHOULD be set to the Buffer.SAMPR_USER_ALL_INFORMATION.PrimaryGroupId field ([MS-SAMR] section 2.2.7.1) of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.
- The **UserAccountControl** field SHOULD be set to the Buffer.SAMPR_USER_ALL_INFORMATION.UserAccountControl field ([MS-SAMR] section 2.2.7.1) of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.

For KILE implementations that use an Active Directory for the account database, KDCs SHOULD retrieve the following attributes from local directory service instance with the same processing rules as defined in **SamrGetGroupsForUser()** ([MS-SAMR] section 3.1.5.9.1) message processing. The KDC populates the returned KERB_VALIDATION_INFO structure ([MS-PAC] section 2.5) as follows:

- The **GroupCount** field SHOULD be set to the Groups.MembershipCount field of the **SamrGetGroupsForUser** ([MS-SAMR] section 3.1.5.9.1) response message.
- The **GroupIds** field SHOULD be set to the Groups.Group field of the **SamrGetGroupsForUser** ([MS-SAMR] section 3.1.5.9.1) response message.

The KDC populates the returned KERB_VALIDATION_INFO structure ([MS-PAC] section 2.5) fields as follows:

- The **UserFlags** field MUST set D if the ExtraSids field is populated and contains additional SIDs and all other bits MUST be set to zero.
- The **UserSessionKey** field MUST be set to zero.
- The **LogonServer** SHOULD be set to **NetbiosServerName**.
- The **LogonDomainName** SHOULD be set to **NetbiosDomainName**.
- The **LogonDomainId** SHOULD be set to **DomainSid**.
- The **Reserved1** field MUST be set to a two-element array of unsigned 32-bit integers and each element of the array MUST be zero.
- The **Reserved3** field MUST be set to a seven-element array of unsigned 32-bit integers and each element of the array MUST be zero.
- The **SidCount** field MUST be set to zero.
- The **ExtraSids** field MUST be NULL.
- The **ResourceGroupDomainSid** field MUST be set to zero.
- The **ResourceGroupCount** field MUST be set to zero.
- The **ResourceGroupIds** field MUST be set to zero.

#### 3.3.5.3.2.2 PAC_CLIENT_INFO Structure

The KDC populates the returned PAC_CLIENT_INFO structure ([MS-PAC] section 2.7) fields as follows:

- The **ClientId** field SHOULD be the Kerberos initial ticket-granting ticket TGT authentication time ([RFC4120] section 5.3).
- The **NameLength** field SHOULD be the length of the **Name** field, in bytes.
- The **Name** field SHOULD be set to cname.

#### 3.3.5.3.2.3 Server Signature

The KDC creates a keyed hash ([RFC4757]) of the entire PAC message with the Signature fields of both PAC_SIGNATURE_DATA structures set to zero using the server account key with the strongest

cryptography that the domain supports<31> and populates the returned PAC_SIGNATURE_DATA structure ([MS-PAC] section 2.8) fields as follows:

- The **SignatureType** SHOULD be the value ([MS-PAC] section 2.8) corresponding to the cryptographic system used to calculate the checksum.
- The **Signature** field SHOULD be the keyed hash ([RFC4757]) of the entire PAC message with the Signature fields of both PAC_SIGNATURE_DATA structures set to zero.

#### 3.3.5.3.2.4 KDC Signatures

The KDC creates a keyed hash ([RFC4757]) of the Server Signature field using the strongest "krbtgt" account key and populates the returned PAC_SIGNATURE_DATA structure field ([MS-PAC] section 2.8) as follows:

- The **SignatureType** SHOULD be the value ([MS-PAC] section 2.8) corresponding to the cryptographic system used to calculate the checksum.
- The **Signature** field SHOULD be the keyed hash ([RFC4757]) of the Server Signature field in the PAC message.

#### 3.3.5.3.2.5 UPN_DNS_INFO Structure

The KDC populates the returned UPN_DNS_INFO structure ([MS-PAC] section 2.10) fields<32> as follows:

- The **UpnLength** field SHOULD be the length of the UPN field, in bytes.
- The **UpnOffset** field SHOULD be the offset of the UPN field to the beginning of the buffer, in bytes, from the beginning of the UPN_DNS_INFO structure.
- The **DnsDomainNameLength** field SHOULD be the length of the **DnsDomainName** field, in bytes.
- The **DnsDomainNameOffset** field SHOULD be the offset of the **DnsDomainName** field to the beginning of the buffer, in bytes, from the beginning of the UPN_DNS_INFO structure.
- The **Flags** field SHOULD set the U bit if the user account object does not have the **userPrincipalName** attribute ([MS-ADA3] section 2.348) set.

The KDC inserts the DNS and UPN information after the UPN_DNS_INFO structure following the header and starting with the corresponding offset in a consecutive buffer. The UPN and FQDN are encoded using a two-byte UTF16 scheme, in little-endian order.

### 3.3.5.4 TGS Exchange

Kerberos V5 specifies the TGS exchange ([RFC4120] section 3.3).

KILE supports the following extensions to the TGS exchange:

- Check Account Policy for Every Session Ticket Request
- TGT without a PAC
- Domain Local Group Membership
- Cross-Domain Trust and Referrals

If the server or service has a KerbSupportedEncryptionTypes populated with supported encryption types, then the KDC SHOULD<33> return in the encrypted part ([Referrals-11] Appendix A) of TGS-REP message PA-DATA with padata-type set to PA-SUPPORTED-ENCTYPES (165), to indicate what encryption types are supported by the server or service. If not, the KDC SHOULD<34> check the server or service account's UseDESOnly flag:

- If UseDESOnly is set: the KDC SHOULD, in the encrypted pre-auth data part ([Referrals-11], Appendix A) of the TGS-REP message, include PA-DATA with the padata-type set to PA-SUPPORTED-ENCTYPES (165), and the padata-value set to 0x3 (section 2.2.6).
- Otherwise:
  - If the account is krbtgt, and domainControllerFunctionality returns a value < 3 ([MS-ADTS] section 3.1.1.3.2.25): the KDC SHOULD, in the encrypted pre-auth data part ([Referrals-11], Appendix A) of the TGS-REP message, include PA-DATA with padata-type set to PA-SUPPORTED-ENCTYPES (165), and the padata-value set to 0x7 (Section 2.2.6).
  - If the account is krbtgt, and domainControllerFunctionality returns >= 3 ([MS-ADTS] Section 3.1.1.3.2.25): the KDC SHOULD, in the encrypted pre-auth data part ([Referrals-11], Appendix A) of the TGS-REP message, include PA-DATA with the padata-type set to PA-SUPPORTED-ENCTYPES (165), and the padata-value set to 0x1F (section 2.2.6).

If the Application Server's service account AuthorizationDataNotRequired is set to TRUE, the KDC MUST NOT include a PAC in the service ticket.

If the PAC contains the SID S-1-5-1000 (Other Organization) ([MS-DTYP] section 2.4.2.4), the PAC MUST be used to perform an access check for the Allowed-To-Authenticate right ([MS-ADTS] section 7.1.1.2.7.42) against the Active Directory object of the account for which the service ticket request is being made. If the access check succeeds, the service ticket MUST be issued; otherwise, the KDC MUST return KDC_ERR_POLICY.

When KERB-LOCAL data is present, the KDC SHOULD copy the authorization data field ([RFC4120] section 5.2.6) with ad-type KERB-LOCAL (142) and ad-data containing KERB-LOCAL structure (section 2.2.3) as an AD-IF-RELEVANT to the end of authorization data in the service ticket.<35>

The KILE KDC MUST copy the populated fields from the PAC in the TGT to the newly created PAC and, after processing all fields it supports, the KILE KDC MUST generate a new Server Signature (section 3.3.5.3.2.3) and KDC Signature (section 3.3.5.3.2.4) which replace the existing signature fields in the PAC.

#### 3.3.5.4.1 Check Account Policy for Every Session Ticket Request

Kerberos V5 does not enforce revocation of accounts prior to the expiration of issued tickets.

If the POLICY_KERBEROS_VALIDATE_CLIENT bit is set in the **AuthenticationOptions** setting on the KDC then KILE will enforce revocation <36> on the KDCs. When this property is set on the KDC for the client's domain, and the TGT is older than an implementation specific time <37>, the KDC MUST verify that the account is still in good standing. Good standing means the account has not expired, been locked out, been disabled or otherwise is not allowed to log on.

- If Disabled is TRUE, then the KDC MUST return KDC_ERR_CLIENT_REVOKED.
- If Expired is TRUE, then the KDC MUST return KDC_ERR_CLIENT_REVOKED.
- If Locked is TRUE, then the KDC MUST return KDC_ERR_CLIENT_REVOKED.

- If current time is not within the LogonHours, then the KDC MUST return KDC_ERR_CLIENT_REVOKED.
- If the PasswordMustChange is in the past, then the KDC MUST return KDC_ERR_KEY_EXPIRED.
- If the PasswordMustChange is zero, then the KDC MUST return KDC_ERR_KEY_EXPIRED.

#### 3.3.5.4.2 TGT without a PAC

If a TGS request includes a TGT without a PAC, the KDC SHOULD add a PAC before issuing the service ticket. This occurs when the TGT was issued by a pure realm [RFC4120] that is trusted by the domain. The PAC MUST be inserted when there is a mapping to a domain user. There are two ways to discover the mapped user:

- If the KDC is configured locally to map principals in the realm to accounts based on name [RFC4120]. In this case, the KDC MUST search the mapping for a principal with the same name.
- If there is no default mapping rule established, the KDC MUST search Active Directory for an account which is associated with the name in the TGT.

If a matching account is found and the Application Server's service account AuthorizationDataNotRequired is set to FALSE, the KDC MUST use that account to construct a PAC and insert it into the resulting service ticket. Otherwise, the service ticket MUST be issued without a PAC.

#### 3.3.5.4.3 Domain Local Group Membership

Groups can be created so that they are only visible to servers in the same domain. For every service ticket that is issued during a TGS request, except for cross-realm TGTs, the KDC MUST populate the PAC with domain local group membership for the user.

For KILE implementations that use an Active Directory for the account database, KDCs MUST call **IDL_DRSGetMemberships** ([MS-DRSR] section 4.1.8) where:

- **dwInVersion** is 1.
- **msgIn.cDsNames** is the count of items in the ppDsNames array.
- **msgIn.ppDsNames** is the DSNAME ([MS-DRSR] section 5.49) of the user and groups of which the user is a member contained in GroupIds ([MS-PAC] section 2.5) with **Sid** set to the SID, **SidLen** set to the length of the SID, and other fields set to NULL.
- **msgIn.dwFlags** is 0.
- **msgIn.OperationType** is set to RevMembGetResourceGroups.
- **msgIn.pLimitingDomain** is NULL.

Then the KDC MUST copy the populated fields from the PAC in the TGT to the newly created PAC and add to the KERB_VALIDATION_INFO structure ([MS-PAC] section 2.5) of the new PAC the domain local groups returned by **IDL_DRSGetMemberships** ([MS-DRSR] section 4.1.8) to the existing fields as follows:

- The **SidCount** field contains the number of groups in the **ExtraSids** field.
- The **ExtraSids** field contains the pointer to a list which is the list copied from the PAC in the TGT plus a list constructed from the domain local groups where:

- **Sid** ([MS-PAC] section 2.2.1) contains the value **pmsgOut.ppDsNames.SID** ([MS-DRSR] section 5.49).
- **Attributes** ([MS-PAC] section 2.2.1) has the A, B, C and E bits set to 1, and all other bits set to zero.

#### 3.3.5.4.4 Cross-Domain Trust and Referrals

The KDC derives its knowledge of cross-domain trusts from trusted domain objects (TDOs) in Active Directory. For more information, see [MS-ADTS].

If a cross-domain referral is determined to be necessary ([RFC4120] section 1.2 and [Referrals-11]), the appropriate inter-realm key MUST be retrieved from the TDO and used as specified in [RFC4120].

If the TRUST_ATTRIBUTE_CROSS_ORGANIZATION flag is set in the TrustAttributes field ([MS-ADTS] section 7.1.6.7.9), the OTHER_ORGANIZATION SID ([MS-DTYP] section 2.4.2.4) MUST be added to the user's PAC. The KDC MUST perform an ACL check while processing the TGS request as follows.

- The security descriptor MUST be that of the server AD account object,
- the client principal MUST be that of the client user,
- and the requested access MUST be ACTRL_DS_CONTROL_ACCESS.

If there is a failure in the check, the KDC MUST reject the authentication request with KDC_ERROR_POLICY.

#### 3.3.5.4.5 FORWARDED TGT etype

When the KDC receives a TGS-REQ, it will create the random session key as described in [RFC4120], section 3.1.3. If a TGS-REQ message requesting a FORWARDED ([RFC4120] section 2.6) TGT provides an **etype** value that is not supported by the KDC, and the client provides a PA-SUPPORTED-ENCTYPES with encryption types the KDC supports, then the KDC MAY select the strongest encryption type that is both included in the PA-SUPPORTED-ENCTYPES and supported by the KDC to generate the random session key. <38>See section 3.1.5.2 for the relative strengths of KILE-supported encryption types.

### 3.3.6 Timer Events

KILE introduces no timer events.

### 3.3.7 Other Local Events

KILE introduces no local events.

## 3.4 Application Server Details

Kerberos V5 defines a protocol subordinate to some other application protocol, via GSS-API [RFC4121]. KILE extends GSS-API (see GSS_WrapEx (section 3.4.5.4) and GSS_UnwrapEx (section 3.4.5.5)).

The AP exchange is controlled by several logical parameters that are passed in by the higher-layer application protocol that is invoking KILE.

### 3.4.1 Abstract Data Model

The abstract data model for the Application Server is identical to that specified in section 3.2.1.

Additionally, the server maintains the following parameter:

- ApplicationRequiresCBT: A Boolean setting from the application requiring channel binding.<39>

For KILE implementations that use a **security identifier (SID)**-based authorization model, the server maintains the following parameter:

- *ImpersonationAccessToken* (Public): An impersonation token.

### 3.4.2 Timers

The AP exchange does not require specific timers.

### 3.4.3 Initialization

All parameters that are specified in section 3.4.1 are reset and then set according to the higher-layer protocols request.

The replay cache MUST be initialized with no entries.

#### 3.4.3.1 msDS-SupportedEncryptionTypes attribute

If the realm is a KILE implementation that uses an Active Directory for the account database, the server SHOULD ensure that the **msDS-SupportedEncryptionTypes** attribute ([MS-ADA2] section 2.324) of its account object is set to the value of SupportedEncryptionTypes (section 3.1.1.5).

When an application server is running under the machine account and NRPC is supported on the machine, the server SHOULD call NetrLogonGetDomainInfo ([MS-NRPC] section 3.4.5.2.8) with the *Level* parameter set to 1 and **WkstaBuffer.WorkstationInfo.KerberosSupportedEncryptionTypes** set to zero.<40>If the **WkstaBuffer.WorkstationInfo.KerberosSupportedEncryptionTypes** returned is not equal to SupportedEncryptionTypes (section 3.1.1.5), then LDAP is used to update the setting:<41>

1. Establish an LDAP connection with server information set to NULL .
2. Perform an LDAP modify operation to set the msDS-SupportedEncryptionTypes attribute ([MS-ADA2] section 2.324) of the computer account object to the value of SupportedEncryptionTypes (section 3.1.1.5).

### 3.4.4 Higher-Layer Triggered Events

The AP exchange is triggered by a higher-layer application protocol that requests security services for a connection or message exchange. The higher-layer application protocol MUST specify the name of the server to which it is attempting authentication and also MUST specify any of the parameters from section 3.4.1 that are required for Kerberos V5 [RFC4120] to perform the authentication.

Calling applications use the SSPI API family to establish the connection and specify the target. Optionally, certain higher-layer protocols, such as Simple and Protected Generic Security Service Application Program Interface Negotiation Mechanism (SPNEGO) [MS-SPNG], will also specify the parameters.

### 3.4.5 Message Processing Events and Sequencing Rules

Kerberos V5 specifies several additional messages ([RFC4120] sections 3.4 through 3.6) that are associated with the session after the AP exchange has completed.

KILE does not implement KRB_SAFE messages ([RFC4120] section 3.4).

KILE does not implement KRB_PRIV messages with a time stamp ([RFC4120] section 3.5). KILE implements KRB_PRIV messages with a sequence number ([RFC4120] section 3.5).

KILE implements KRB_CRED messages ([RFC4120] section 3.6).

KILE will return a zero-length message whenever it receives a message that is either not well-formed or not supported.

If the decryption of the ticket fails and the KILE server has older versions of the server key, the server SHOULD retry decrypting the ticket with the older keys.

If the decryption routines detect a modification of the ticket, the KRB_AP_ERR_MODIFIED error message is returned.

If decryption shows that the authenticator has been modified, the KRB_AP_ERR_MODIFIED error message is returned.

When clock skew errors occur during AP exchanges, the application server SHOULD attempt a clock skew recovery by returning a KRB_AP_ERR_SKEW error ([RFC4120] section 3.2.3) containing a KERB-ERROR-DATA structure (section 2.2.1) in the e-data field of the KRB-ERROR message ([RFC4120] section 5.9.1).

When the server receives AP requests for SPNs with the serviceclass equal to **"RestrictedKrbHost"**, it will decrypt the ticket with the computer account's key and either create or use the session key for the "RestrictedKrbHost", regardless of the account the target service is running as.<42>

If the ApplicationRequiresCBT parameter (section 3.4.1) is set to TRUE, the server, if so configured, MAY return GSS_S_BAD_BINDINGS whenever the AP request message contains an all-zero channel binding value and does not contain the AD-IF-RELEVANT element ([RFC4120] section 5.2.6.1) KERB_AP_OPTIONS_CBT.<43>

#### 3.4.5.1 Three-Leg DCE-Style Mutual Authentication

An application protocol using the Kerberos protocol must exchange application protocol messages with Kerberos signing or encryption applied in order to verify mutual authentication. DCE, in the authn_dce_secret authentication service (as specified in [C706]) mandated that mutual authentication be verified before any RPC messages were exchanged. To accommodate that requirement, the DCE Kerberos implementation issued an additional AP_REPLY message from the client to the server as part of the AP exchange subprotocol.

Kerberos V5 is not interoperable with the DCE authn_dce_secret security protocol. KILE MUST have compatible extensions for third-party extensions. KILE emulates this behavior as follows:

- The AP-REQ message MUST NOT have GSS-API wrapping. It is sent as is without encapsulating it in a header ([RFC2743] section 3.1).
- The signature message and the encryption message MUST NOT include the length of the application data; they are no longer RFC 1964–compliant [RFC1964].

- The client MUST generate an additional AP reply message exactly as the server would ([RFC4120] section 3.2.4) as the final message to send to the server. The client SHOULD set the GSS_C_DCE_STYLE flag ([RFC4757] section 7.1) to TRUE in the authenticator's checksum field ([RFC4121] section 4.1.1). In GSS terms, the client must return success and a message to the server. It is up to the application to deliver the message to the server.
- The server MUST receive the additional AP reply message and verify that the message is constructed correctly ([RFC4120] section 3.2.5).

The GSS_Wrap() and GSS_WrapEx() methods are not supported with DCE Style authentication.

#### 3.4.5.2 Datagram-Style Authentication

Datagram-style authentication is another DCE RPC-inspired variation. In summary, datagram style initializes the security context but does not transmit the authentication message. Instead, the first application data packet is signed or encrypted as decided by the higher-level application protocol and sent to the server. The server, presented with a packet for which it has no security context, sends a demand for authentication back to the client. At that point, the client sends the authentication token previously obtained from the authentication mechanism. Authentication proceeds as normal.

When authentication is complete, the server verifies or decrypts the application packet. An application protocol that uses this datagram capability MUST have the means within the application protocol to indicate the nature of the security mechanism that is used (if mechanisms other than the Kerberos V5 protocol are possible), and the nature of the protection (signature or encryption) that is applied to the application protocol message. For DCE RPC the application packet is not retransmitted. Therefore, the session key that will be used MUST be decided by the client before any communication with the server. This precludes the sub-session key option of the Kerberos V5 protocol.

#### 3.4.5.3 Processing Authorization Data

Kerberos V5 specifies rules for processing the authorization data field in [RFC4120] section 5.2.6.

KILE MUST unpack the authorization data field ([RFC4120] section 5.2.6) and look for an AD-WIN2K-PAC structure ([RFC4120] section 7.5.4). If the structure is valid according to the PAC specification [MS-PAC], the server MUST verify the server signature. To verify the server signature, the **Signature** field values are removed from the PAC buffer and replaced with zeros. Then the hash is generated [RFC4757] and the resulting hash is compared with the server signature ([MS-PAC] section 2.8.1) **Signature** field value. If the PAC is valid, it SHOULD be used as the authorization information.

The server MUST check if KERB-AD-RESTRICTION-ENTRY.Restriction.MachineID (section 2.2.5) is equal to Machine ID (section 3.1.1.4):

- If equal, the server SHOULD process the authentication as a local one, because the client and server are on the same machine, and MAY use the KERB-LOCAL AuthorizationData for any local implementation purposes.<44>
- Otherwise, the server MUST ignore the KERB_AUTH_DATA_TOKEN_RESTRICTIONS [141] Authorization Data Type, the KERB-AD-RESTRICTION-ENTRY structure (section 2.2.5), the KERB-LOCAL (142), and the containing KERB-LOCAL structure (section 2.2.3).<45>

For KILE implementations that use a security identifier (SID)-based authorization model, the server SHOULD populate the User SID and Security Group SIDs in the **ImpersonationAccessToken** parameter (section 3.4.1) as follows:

- Concatenate **LogonDomainId** ([MS-PAC] section 2.5) and **UserId** [MS-PAC] section 2.5), add to the **ImpersonationAccessToken.Sids** array, and set the **ImpersonationAccessToken.UserIndex** field to this index.
- Concatenate **LogonDomainId** ([MS-NRPC] sections 2.2.1.4.11, 2.2.1.4.12, and 2.2.1.4.13) and **PrimaryGroupId** ([MS-NRPC] sections 2.2.1.4.11, 2.2.1.4.12, and 2.2.1.4.13), add the result to the **ImpersonationAccessToken.Sids** array, and set the **ImpersonationAccessToken.PrimaryGroup** field to this index.
- For each **GroupIds** ([MS-PAC] section 2.2.2), concatenate **LogonDomainId** ([MS-PAC] section 2.5) and **GroupIds.RelativeID** ([MS-PAC] section 2.2.2) and add to the **ImpersonationAccessToken.Sids** array.
- For each **ExtraSids** ([MS-PAC] section 2.2.2), add the **ExtraSids.Sid** ([MS-PAC] section 2.2.2) to the **ImpersonationAccessToken.Sids** array.

The server SHOULD call **GatherGroupMembershipForSystem**where **InitialMembership** contains the **ImpersonationAccessToken.Sids** array and set **ImpersonationAccessToken.Sids** array to **FinalMembership**.

The server SHOULD call AddPrivilegesToToken where **Token** contains **ImpersonationAccessToken**.

#### 3.4.5.4 GSS_WrapEx() Call

This call is an extension to GSS_Wrap ([RFC2743] section 2.3.3) that passes multiple buffers.

Inputs:

- context_handle CONTEXT HANDLE
- qop_req INTEGER -- 0 specifies default Quality of Protection (QOP)
- input_message ORDERED LIST of:
  - conf_req_flag BOOLEAN
  - sign BOOLEAN
  - data OCTET STRING

Outputs:

- major_status INTEGER
- minor_status INTEGER
- output_message ORDERED LIST (in same order as input_message) of:
  - conf_state BOOLEAN
  - signed BOOLEAN
  - data OCTET STRING
- signature OCTET STRING

This call is identical to GSS_Wrap, except that it supports multiple input buffers. Input data buffers for which conf_req_flag==TRUE are encrypted in output_message. Input data buffers for which sign==TRUE are included in the message, as specified in section 3.4.5.4.1.

#### 3.4.5.4.1 Kerberos Binding of GSS_WrapEx()

Kerberos GSS_WrapEx() depends on the encryption type of the session key for the context. The algorithms depend on which Kerberos encryption ciphers are negotiated by the Kerberos protocol.

If the session key encryption type is AES128-CTS-HMAC-SHA1-96 or AES256-CTS-HMAC-SHA1-96 (as specified in [RFC3961]):

- The base line is [RFC4121].
- The encrypted data is per [RFC3961] (on which [RFC4121] is based), as follows.

```
C1 | H1[1..h]
```

where

```
(C1, newIV) = E(Ke, conf | plaintext | pad, oldstate.ivec)
H1 = HMAC(Ki, conf | plaintext+encrypted-data | pad
```

where the "plaintext+encrypted-data" is all the input data buffers supply to GSS_WrapEx() concatenated in the order provided in the ordered list, input_message.

The RRC field ([RFC4121] section 4.2.5) is 12 if no encryption is requested or 28 if encryption is requested. The RRC field is chosen such that all the data can be encrypted in place. The trailing meta-data H1 is rotated by RRC+EC bytes, which is different from RRC alone ([RFC4121] section 4.2.5). Thus the token buffer contains the header ([RFC4121] section 4.2.6.2) with the rotated H1 that is placed before the encrypted confounder and after the header.

If the session key encryption type is DES-CBC-MD5 or DES-CBC-CRC per [RFC3961]:

- The base line is [RFC1964].
- The ordered list contains the header ([RFC1964] 1.2.2 ) and errata, then DER(Kerberos OID | Token | Encrypted Data | Padding).
- The data is encrypted in place.

The "to-be-signed data" in section 1.2.2.1 of [RFC1964] is a concatenation of all the input_message data for which sign==TRUE. Only the input data with encrypt set to TRUE is encrypted in output_message. The InitialContextToken header as specified in section 1.1 of [RFC1964] is included at the beginning of the ordered list.

For [MS-RPCE], the length field in the above pseudo ASN.1 header does not include the length of the concatenated data if [RFC1964] is used.

If the session key encryption type is RC4-HMAC or RC4-HMAC-EXP per [RFC3961]:

- The base line is [RFC4757].
- The ordered list contains the header ([RFC4757] section 7.3).

- The data (excluding the conf_req_flag set to FALSE) is encrypted in place.

The "to-be-signed data" in section 7.3 of [RFC4757] is a concatenation of all the input buffers for which sign==TRUE. The InitialContextToken pseudo ASN.1 header is included at the beginning of the token header.

#### 3.4.5.5 GSS_UnwrapEx() Call

This call is an extension to GSS_Unwrap ([RFC2743] section 2.3.4) that passes multiple buffers.

Inputs:

- context_handle CONTEXT HANDLE
- input_message ORDERED LIST of:
  - conf_state BOOLEAN
  - signed BOOLEAN
  - data OCTET STRING
- signature OCTET STRING

Outputs:

- qop_req INTEGER, -- 0 specifies default QOP
- major_status INTEGER
- minor_status INTEGER
- output_message ORDERED LIST (in same order as input_message) of:
  - conf_state BOOLEAN
  - data OCTET STRING

This call is identical to GSS_Unwrap, except that it supports multiple input buffers. Input data buffers for which conf_state==TRUE are decrypted in output_message. The signature is verified for the input data buffers where signed==TRUE, that are concatenated as specified in section 3.4.5.4.1.

#### 3.4.5.6 GSS_GetMICEx() Call

Inputs:

- context_handle CONTEXT HANDLE
- qop_req INTEGER, -- 0 specifies default QOP
- message ORDERED LIST of:
  - sign BOOLEAN
  - data OCTET STRING

Outputs:

- major_status INTEGER

- minor_status INTEGER
- message ORDERED LIST of:
  - signed BOOLEAN
  - data OCTET STRING
- per_msg_token OCTET STRING

This call is identical to GSS_GetMIC, except that it supports multiple input buffers. Input data buffers where sign==TRUE are concatenated together and the resulting OCTET STRING is signed as specified by the following RFCs, depending on the session key encryption type:

- DES-CBC-MD5 or DES-CBC-CRC [RFC1964] [RFC3961]
- RC4-HMAC or RC4-HMAC-EXP per [RFC3961] [RFC4757]
- AES128-CTS-HMAC-SHA1-96 or AES256-CTS-HMAC-SHA1-96 [RFC3961] [RFC4121]

#### 3.4.5.7 GSS_VerifyMICEx() Call

Inputs:

- context_handle CONTEXT HANDLE
- message ORDERED LIST of:
  - signed BOOLEAN
  - data OCTET STRING
- per_msg_token OCTET STRING

Outputs:

- qop_state INTEGER
- major_status INTEGER
- minor_status INTEGER

This call is identical to GSS_VerifyMIC, except that it supports multiple input buffers. Input data buffers where signed==TRUE are concatenated together and the signature is verified against the resulting concatenated buffer.

### 3.4.6 Timer Events

KILE introduces no timer events.

### 3.4.7 Other Local Events

There are no other local events except what is driven by the application layer protocol.

# 4 Protocol Examples

The following sections describe four common scenarios to illustrate the function of the KILE.

## 4.1 Interactive Logon Using Passwords

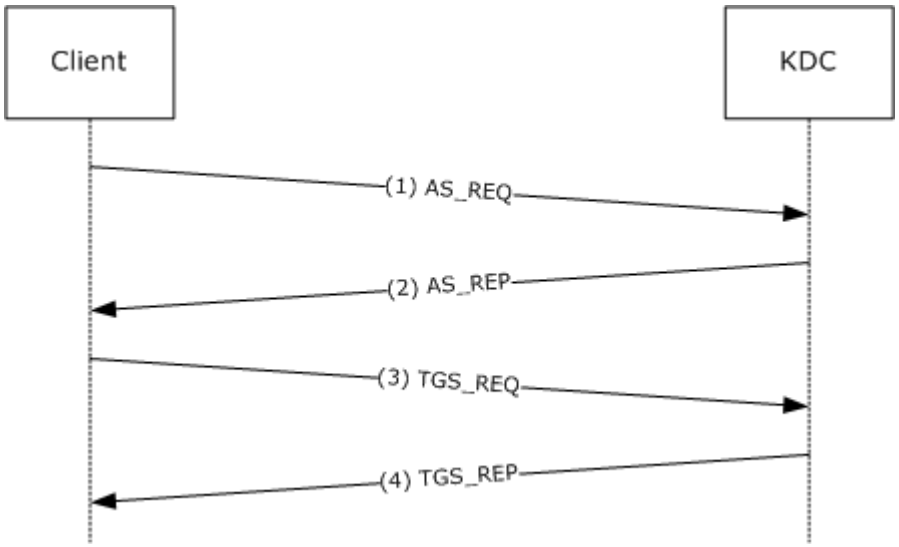


**Figure 2: Interactive logon that uses passwords**

Step 1: A user attempts to log on to a client and types a password at the logon screen, and an AS-REQ for a ticket-granting ticket (TGT) with pre-authentication data is generated. The AS-REQ, which uses the user name and password, is sent to the Key Distribution Center (KDC).

Step 2: In response to receiving the AS-REQ for a TGT, the KDC authenticates the user by checking that the credentials that are used in the AS-REQ are the same as that of the user's ([RFC4120] section 3.1). The KDC builds an AS-REP from the TGT and other requisite data, and sends it back to the client.

The KDC builds a PAC (section 3.3.5.3). Data in the PAC includes account data for the user that is used for logging onto the client. The account data is expected to be supplied by the KDC that queries an account service for the account data. The KDC inserts the PAC that contains the account data that is received from the account service into the authorization_data field of the TGT.

Step 3: The client then sends a TGS-REQ based on the TGT that is obtained in step 2 to obtain a service ticket that is formatted according to the Kerberos protocol for completing a logon process at the local workstation. The client runtime issues a request to host/hostname.domain, where hostname is the actual name of the client machine, and domain is the domain or realm of the client machine.

Step 4: The KDC responds to the TGS-REQ with a TGS-REP that contains the service ticket for the local workstation. The authorization data from step 2 is carried forward to the service ticket, with additional group processing (section 3.3.5.4). The service ticket is then interpreted by the Kerberos runtime within the local workstation.

The following fields from the PAC ([MS-PAC] is the authoritative reference for formatting and encoding these fields) are required by the Kerberos interactive logon to authorize the user for local logon, and to establish the necessary management profile for the user:

- **LogonTime**: The time when the user last logged on. This field is an absolute-format Microsoft Windows® standard time value.
- **LogoffTime**: The time when the user should log off. This field is an absolute-format Windows standard time value.
- **KickOffTime**: The time when the system forces the user to log off. This field is an absolute-format Windows standard time value. Note that Windows users are not forced to log off interactively; however, their network connections may be closed.
- **PasswordLastSet**: The time and date that the password was last changed. This field is an absolute format Windows standard time value.
- **PasswordCanChange**: The time and date when the user is reminded to change passwords. This field is an absolute-format Windows standard time value.
- **EffectiveName**: The text field that contains the effective name of the account that is validated by Active Directory.
- **FullName**: The text field that contains the user's full name.
- **LogonScript**: The text field that contains the relative path to the account's logon script.
- **ProfilePath**: The text field that contains the path to a user's roaming profile. This field is only used if the user has a roaming profile.
- **HomeDirectory**: The text field that contains the user's home directory.
- **HomeDirectoryDrive**: The text field that contains the drive that contains the user's home directory.
- **LogonCount**: The number of times the user is currently logged on.
- **BadPasswordCount**: The number of times a bad password was applied to the account since the last successful logon.
- **LogonServer**: The text field that contains the name of the server that processed the logon request.
- **LogonDomainName**: The text field that contains the name of the computer that is making the account logon request.
- **UserAccountControl**: Flags that control the behavior of the user account.

## 4.2 Network Logon

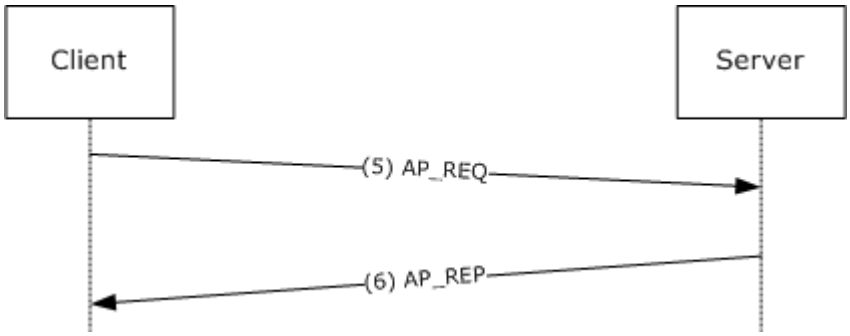

**Figure 3: Network Logon**

When an application wants to use Kerberos-based authentication, it uses either the higher-level SSPI API to invoke Kerberos directly; or it uses SPNEGO [MS-SPNG], which in turn invokes Kerberos.

This may cause steps 1 to 4 (section 4.1) to be repeated if there are new credentials supplied. It may also cause steps 3 and 4 (section 4.1) to be repeated if the server has not previously cached a ticket for the client.

Step 5: When the service ticket to the application server is obtained, the client authenticates itself to the server by sending an AP-REQ wrapped in Generic Security Services (GSS) formatting (section 3.4 and [RFC1964]).

Step 6: The Kerberos runtime on the server validates the ticket by decrypting it, and it validates the authenticator by decrypting and checking for replay and other attacks ([RFC4120] section 3.2).

Invoking the Kerberos runtime to authenticate a **session** is typically done through the SSPI API. Higher-level constructs, for example, remote file access, can also trigger the connection. After the server-side Kerberos runtime validates the ticket and authenticator, it makes the authorization data from the ticket available to the service, typically through a Microsoft Windows®-specific object that is known as an access token, which is used with the Windows system-provided authorization functions.

## 4.3 GSS_WrapEx with AES128-CTS-HMAC-SHA1-96

This is an example of using the encryption type AES128-CTS-HMAC-SHA1-96 with GSS_WrapEx() called with an input_message with four buffers:

- sign1 which has Conf_req_flag == FALSE, sign == TRUE
- enc1 which has Conf_req_flag == TRUE, sign == FALSE
- enc2 which has Conf_req_flag == TRUE, sign == FALSE
- sign2 which has Conf_req_flag == FALSE, sign == TRUE

Processing will proceed as illustrated in the following diagram.

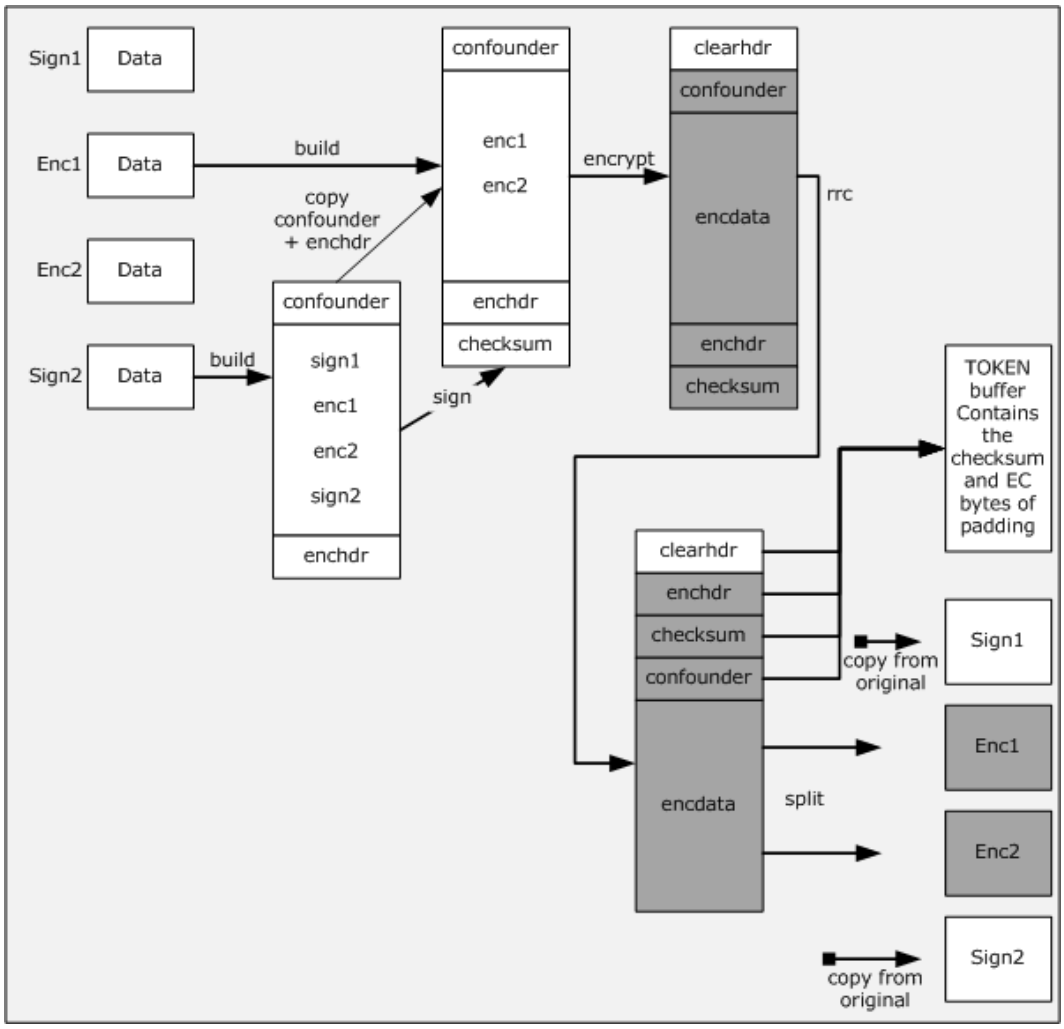


**Figure 4: Example of RRC with output message with 4 buffers**

The **enchdr** is the header ([RFC4121] section 4.2.4) for encrypted buffers. The **clearhdr** is the descriptive header ([RFC4121] section 4.2.6.2). **GSS_WrapEx()** will return an **output_message** with four buffers:

- buffer 1 contains the cleartext sign1 which has Conf_state == FALSE, signed == TRUE
- buffer 2 contains the encrypted enc1 which has Conf_state == TRUE, signed == FALSE
- buffer 3 contains the encrypted enc2 which has Conf_state == TRUE, signed == FALSE
- buffer 4 contains the cleartext sign2 which has Conf_state == FALSE, signed == TRUE and signature which contains the clearhdr + enchdr + checksum + confounder (for details, please see [RFC3961]).

The order of operations is as follows:

- build

- sign
- encrypt
- right rotation by (EC+RRC) count
- split

EC is generated during the encryption process so that there is no padding; see section 4.2.4 of [RFC4121].

## 4.4 AES 128 Key Creation

The following values are used during AES 128 key creation:

User or computer password:

```
0000000: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0000010: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0000020: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0000030: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0000040: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0000050: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0000060: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0000070: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0000080: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0000090: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
00000a0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
00000b0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
00000c0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
00000d0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
00000e0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
```

Salt:

```
0000000: 44 00 4f 00 4d 00 41 00 49 00 4e 00 2e 00 43 00   D•O•M•A•I•N•.•C•
0000010: 4f 00 4d 00 68 00 6f 00 73 00 74 00 63 00 6c 00   O•M•h•o•s•t•c•l•
0000020: 69 00 65 00 6e 00 74 00 2e 00 64 00 6f 00 6d 00   i•e•n•t•.•d•o•m•
0000030: 61 00 69 00 6e 00 2e 00 63 00 6f 00 6d 00         a•i•n•.•c•o•m•
```

IterationCount:

```
0000000: 00 00 00 00 00 00 03 e8                           ........
```

The AES 128 key is created by first converting the password from a Unicode (UTF16) string to a UTF8 string ([UNICODE], chapter 3.9).

UTF8String:

```
0000000: ef bf bf ef bf bf ef bf bf ef bf bf ef bf bf ef   ................
0000010: bf bf ef bf bf ef bf bf ef bf bf ef bf bf ef bf   ................
0000020: bf ef bf bf ef bf bf ef bf bf ef bf bf ef bf bf   ................
```

```
0000030: ef bf bf ef bf bf ef bf bf ef bf bf ef bf bf ef   ................
0000040: bf bf ef bf bf ef bf bf ef bf bf ef bf bf ef bf   ................
0000050: bf ef bf bf ef bf bf ef bf bf ef bf bf ef bf bf   ................
0000060: ef bf bf ef bf bf ef bf bf ef bf bf ef bf bf ef   ................
0000070: bf bf ef bf bf ef bf bf ef bf bf ef bf bf ef bf   ................
0000080: bf ef bf bf ef bf bf ef bf bf ef bf bf ef bf bf   ................
0000090: ef bf bf ef bf bf ef bf bf ef bf bf ef bf bf ef   ................
00000a0: bf bf ef bf bf ef bf bf ef bf bf ef bf bf ef bf   ................
00000b0: bf ef bf bf ef bf bf ef bf bf ef bf bf ef bf bf   ................
00000c0: ef bf bf ef bf bf ef bf bf ef bf bf ef bf bf ef   ................
00000d0: bf bf ef bf bf ef bf bf ef bf bf ef bf bf ef bf   ................
00000e0: bf ef bf bf ef bf bf ef bf bf ef bf bf ef bf bf   ................
00000f0: ef bf bf ef bf bf ef bf bf ef bf bf ef bf bf ef   ................
0000100: bf bf ef bf bf ef bf bf ef bf bf ef bf bf ef bf   ................
0000110: bf ef bf bf ef bf bf ef bf bf ef bf bf ef bf bf   ................
0000120: ef bf bf ef bf bf ef bf bf ef bf bf ef bf bf ef   ................
0000130: bf bf ef bf bf ef bf bf ef bf bf ef bf bf ef bf   ................
0000140: bf ef bf bf ef bf bf ef bf bf ef bf bf ef bf bf   ................
0000150: ef bf bf ef bf bf ef bf bf ef bf bf ef bf bf ef   ................
0000160: bf bf ef bf bf ef bf bf                          ........
```

The salt is converted from a Unicode (UTF16) string to a UTF8 string ([UNICODE], section 3.9).

UTF8Salt:

```
0000000: 44 4f 4d 41 49 4e 2e 43 4f 4D 68 6f 73 74 63 6c   DOMAIN.COMhostcl
0000010: 69 65 6e 74 2e 64 6f 6d 61 69 6e 2e 63 6f 6d      ient.domain.com
```

Next, the UTF8 string is converted to the key ([RFC3962], section 4). When calculating the AES base 128 key, using the values above, then random2key(PBKDF2(UTF8String, UTF8Salt, IterationCount, 128) is:

```
0000000: c7 73 0d aa 23 52 1b c1 6a b8 3c be e3 b3 7f 41   .s..#R..j.<....A
```

The Kerberos key is then created using the AES 128 key above in DK(AES 128 key, "kerberos") ([RFC3962], section 4).

This results in a 128-bit key:

```
0000000: b8 2e e1 22 53 1c 2d 94 82 1a c7 55 bc cb 58 79   ..."S.-....U..Xy
```

## 4.5 RC4 GSS_WrapEx

The **GSS_WrapEx()** is specified in section 3.4.5.4.1. The RC4-HMAC usage is specified in [RFC4757] and corresponding errata. The following data is part of the security context state for the Kerberos session when the client is the initiator.

```
Confidentiality == TRUE
DCE-Style == FALSE
```

Session Key:

```
0000000: 81 a2 cb 90 af 7f c2 d1 95 54 a1 50 d8 18 53 59   üó╦É»⌂┬╤òTíP╪·SY
qop_req == 0
```

Plaintext data where `conf_req_flag == TRUE` and `sign == TRUE`:

```
0000000: 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff      ·"3DUfwêÖ¬╕╠ε
```

The signature is created as specified in [RFC4757] section 7.3 with the following inputs:

Kss:

```
0000000: 81 a2 cb 90 af 7f c2 d1 95 54 a1 50 d8 18 53 59   üó╦É»⌂┬╤òTíP╪·SY
Encrypt == TRUE
Direction == sender_is_initiator
Export == FALSE
```

Seq_num (in big-endian order as specified in [RFC4757] section 7.1):

```
0000000: 60 cb ac d3                                       `╦¼╙
```

Data:

```
0000000: 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff      ·"3DUfwêÖ¬╕╠ε
```

Confounder:

```
0000000: 52 56 f3 fb 63 0c f1 2a                           RV≤√c·±*
Padding == 01
```

The output message data and signature is created using **SEAL()** specified in section 3.4.4. **Output_message** will contain `conf_state == TRUE`, `signed == TRUE` and the following:

Data:

```
0000000: 8e d6 3f 0a c8 38 15 33 5b 72 e2 93 ba e1 f6 60   Ä╓?·╚8·3[rΓô╟ß÷`
```

Signature:

```
0000000: 60 3b 06 09 2a 86 48 86 f7 12 01 02 02 02 01 11   `;··*åHå≈·······
0000010: 00 10 00 ff ff e2 9e 8b bc 63 48 e7 40 eb aa 61   ···  Γ₧ï╝cHτ@δ¬a
0000020: 92 44 a1 56 a1 3b 5c f6 5e 3c 21 b9 aa            ÆDíVí;\÷^<!╣¬
```

# 5 Security

Older versions of MIT Kerberos do not support RC4, and therefore, the only common option for interoperability is DES. To obtain the security benefits of a stronger 128-bit key, upgrade to the latest version of MIT Kerberos.

Other general Kerberos security considerations are specified in [RFC4120] section 10.

## 5.1 Security Considerations for Implementers

KILE has the same security considerations as Kerberos V5 ([RFC4120], [RFC3961], [RFC3962], and [RFC4757]) and GSS-API ([RFC2743], [RFC1964], and [RFC4121]).

### 5.1.1 RODC Key Version Numbers

Because read-only domain controllers (RODCs) can be deployed in less secure locations, RODCs have different key version numbers (section 3.1.5.8) to ensure they are using a different key than the domain's DCs. This protects the domain if an RODC is compromised.

### 5.1.2 SPNs with Serviceclass Equal to "RestrictedKrbHost"

Supporting the "RestrictedKrbHost" service class allows client applications to use Kerberos authentication when they do not have the identity of the service but have the server name. This does not provide client-to-service mutual authentication, but rather client-to-server computer authentication. Services of different privilege levels have the same session key and could decrypt each other's data if the underlying service does not ensure that data cannot be accessed by higher services.

### 5.1.3 Account Revocation Checking

Kerberos V5 does not provide account revocation checking for TGS requests, which allows TGT renewals and service tickets to be issued as long as the TGT is valid even if the account has been revoked. KILE provides a check account policy (section 3.3.5.4.1) that limits the exposure to a shorter time. KILE KDCs in the account domain are required to check accounts when the TGT is older than 20 minutes. This limits the period that a client can get a ticket with a revoked account while limiting the performance cost for AD queries.

### 5.1.4 FORWARDED TGT etype

When the KDC can determine the etype in accordance with [RFC4120] section 3.1.3, PA-SUPPORTED-ENCTYPES should not be used because the field is not protected.

## 5.2 Index of Security Parameters

There are no security parameters for this protocol extension.

# 6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft Windows® 2000 operating system
- Windows® XP operating system
- Windows Server® 2003 operating system
- Windows Vista® operating system
- Windows Server® 2008 operating system
- Windows® 7 operating system
- Windows Server® 2008 R2 operating system

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 1.9.1: Windows 2000 does not support the RFC Kerberos OID.

<2> Section 2.1: The default values for the message size threshold are shown in the following table for different versions of Windows.

| Windows version | Message size |
|---|---|
| Windows 2000 (initial release)– Windows 2000 SP3 | 2000 bytes |
| Windows 2000 SP4 | 1465 bytes |
| Windows XP (initial release), Windows XP SP1 | 2000 bytes |
| Windows XP SP2 | 1500 bytes |
| Windows Server 2003 (initial release), Windows XP 64-Bit Edition, Windows Server 2003 with SP1, Windows Server 2003 R2, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2. | 1465 bytes |

**Note** Windows NT does not include a Kerberos implementation.

<3> Section 2.2.3: Windows 7 and Windows Server 2008 R2 support transmitting KERB-LOCAL.

<4> Section 2.2.4: The **LSAP_TOKEN_INFO_INTEGRITY** structure is supported in Windows Vista SP1, Windows Server 2008, Windows 7, and Windows Server 2008 R2.

<5> Section 2.2.5: The KERB-AD-RESTRICTION-ENTRY structure is supported in Windows Vista SP1, Windows Server 2008, Windows 7, and Windows Server 2008 R2.

<6> Section 2.2.7: PA-SUPPORTED-ENCTYPES are not supported by Windows 2000, Windows XP, or Windows Server 2003.

<7> Section 3.1.1.3: Windows has a ticket cache and makes the ticket cache available to client applications at their request. Programmatic methods for querying the contents, purging the contents, or purging individual tickets are also available.

In Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2, TGTs are automatically renewed. Renewal attempts begin at 10 minutes prior to expiration for Windows Server 2003 and at 15 minutes for Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2, unless the renew-till time (see [RFC4120], section 2.3) of the TGT is within five minutes.

<8> Section 3.1.1.4: In Windows Vista SP1, Windows Server 2008, Windows 7, and Windows Server 2008 R2, a 32-byte binary random string machine ID is sent on the wire. This machine ID is not used by KILE.

<9> Section 3.1.1.5: SupportedEncryptionTypes are not supported in Windows 2000, Windows XP, and Windows Server 2003.

<10> Section 3.1.1.5: The default for SupportedEncryptionTypes in Windows Vista and Windows Server 2008 is 0000001F. The default for Windows Server 2008 R2 DCs is 0000001F.

<11> Section 3.1.5.2: Not supported in Windows 2000, Windows XP, or Windows Server 2003.

<12> Section 3.1.5.2: In Windows 2000 and Windows Server 2003, KDCs select the encryption type based on the preference order in the client request. In Windows Server 2008, and Windows Server 2008 R2, KDCs select the encryption type used for pre-authentication, or, when pre-authentication is not used, the encryption type based on the preference order in the client request.

<13> Section 3.1.5.2: Supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2.

<14> Section 3.1.5.2: Supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2.

<15> Section 3.1.5.2: Supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008. Windows 7 and Windows Server 2008 R2 systems do not support DES by default.

<16> Section 3.1.5.2: Supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008. Windows 7 and Windows Server 2008 R2 systems do not support DES by default.

<17> Section 3.1.5.2: In addition to the encryption type values specified in section 3.1.5.2, Windows 2000 and Windows XP send the values -135, -133, and -128. Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 send the value -135. These are invalid encryption types and are ignored when received; if all encryption type values are so ignored, then the result will be as if no values were sent.

<18> Section 3.1.5.6: IPv6 addresses are not supported in Windows 2000, Windows XP and Windows Server 2003.

<19> Section 3.1.5.8: Supported in Windows Server 2008 and Windows Server 2008 R2.

<20> Section 3.1.5.12: Windows 7 and Windows Server 2008 R2 support "RestrictedKrbHost/<hostname>" to allow developer frameworks to enable Kerberos authentication for code written prior to SPN support.

<21> Section 3.2.1: The ChannelBinding parameter is supported in Windows 7 and Windows Server 2008 R2.

<22> Section 3.2.5.4: Supported in Windows Vista SP1, Windows Server 2008, Windows 7, and Windows Server 2008 R2

<23> Section 3.2.5.5: Windows 7 and Windows Server 2008 R2 support KERB-LOCAL.

<24> Section 3.2.5.6: Windows 7 and Windows Server 2008 R2 support KERB-LOCAL.

<25> Section 3.2.5.6: No version of Windows uses this field. Windows Vista SP1, Windows 7, Windows Server 2008, and Windows Server 2008 R2 send this field on the wire in anticipation of possible future use.

<26> Section 3.2.6: Windows client implementations include configured values for the initial time-out of 5 seconds, and an increase factor of 5 seconds and 10 seconds to retry 3 times.

<27> Section 3.3.1.1: KerbSupportedEncryptionTypes are not supported in Windows 2000 and Windows Server 2003.

<28> Section 3.3.5.1: Windows 2000 KDCs will canonicalize the name in the resulting ticket, based on the name of the account that is ultimately used in AD.

Windows Server 2003, Windows Server 2008, and Windows Server 2008 R2 KDCs do not honor the canonicalize flag except for referrals [Referrals-11], and they do not perform any canonicalization.

<29> Section 3.3.5.2: Windows Server 2008 and Windows Server 2008 R2 KDCs support the provisioning of UPNs.

<30> Section 3.3.5.3: Supported in Windows Server 2008 and Windows Server 2008 R2.

<31> Section 3.3.5.3.2.3: Active Directories with the **msDS-Behavior-Version** attribute on a domain NC root object equal to DS_BEHAVIOR_WIN2000, DS_BEHAVIOR_WIN20003_WITH_MIXED_DOMAINS, or DS_BEHAVIOR_WIN2003, cannot support AES.

<32> Section 3.3.5.3.2.5: Windows Server 2008 and Windows Server 2008 R2 support UPN and DNS information.

<33> Section 3.3.5.4: Supported in Windows Server 2008 and Windows Server 2008 R2.

<34> Section 3.3.5.4: Supported in Windows Server 2008 and Windows Server 2008 R2.

<35> Section 3.3.5.4: Windows 7 and Windows Server 2008 R2 support KERB-LOCAL.

<36> Section 3.3.5.4.1: In Windows 2000, Windows Server 2003, Windows Server 2008, and Windows Server 2008 R2 implementations, the default domain policy setting is Enforce user logon restrictions.

<37> Section 3.3.5.4.1: Windows uses 20 minutes as the time value at which a TGT is verified to be in good standing.

<38> Section 3.3.5.4.5: Supported in Windows Server 2008 and Windows Server 2008 R2.

<39> Section 3.4.1: Channel binding is supported in Windows 7 and Windows Server 2008 R2.

<40> Section 3.4.3.1: Not supported in Windows 2000, Windows XP and Windows Server 2003.

<41> Section 3.4.3.1: Not supported in Windows 2000, Windows XP and Windows Server 2003.

<42> Section 3.4.5: SPNs with the serviceclass equal to "RestrictedKrbHost" are supported in Windows 7 and in Windows Server 2008 R2.

<43> Section 3.4.5: The ApplicationRequiresCBT parameter is supported in Windows 7 and Windows Server 2008 R2.

<44> Section 3.4.5.3: Windows 7 and Windows Server 2008 R2 support KERB-LOCAL.

<45> Section 3.4.5.3: Supported in Windows Vista SP1, Windows Server 2008, Windows 7, and Windows Server 2008 R2.

# 7 Change Tracking

This section identifies changes that were made to the [MS-KILE] protocol document between the January 2011 and February 2011 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated.**

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact protocol@microsoft.com.

| Section | Tracking number (if applicable) and description | Major change (Y or N) | Change type |
|---|---|---|---|
| 3.3.3 Initialization | 60465<br>Removed initialization of AuthenticationOptions from the list of configuration settings. | N | Content updated. |
| 3.3.4.1 KDC Configuration Changes | 60465<br>Removed initialization of AuthenticationOptions from the list of configuration settings. | N | Content updated. |
| 3.3.5.4 TGS Exchange | 45445<br>Clarified how fields are populated in a newly created PAC. | N | Content updated. |

# 8 Index

**M**



**N**



**O**



**P**



**R**



**S**



**T**



**V**