

[MS-KILE-Diff]:

Kerberos Protocol Extensions

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards as well as overviews of the interaction among each of these technologies support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you maycan make copies of it in order to develop implementations of the technologies that are described in the Open Specifications-~~this documentation~~ and maycan distribute portions of it in your implementations usingthat use these technologies or in your documentation as necessary to properly document the implementation. You maycan also distribute in your implementation, with or without modification, any schema, IDL'sschemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications-documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that maymight cover your implementations of the technologies described in the Open Specifications-documentation. Neither this notice nor Microsoft's delivery of thethis documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specification maySpecifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications>this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation maymight be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mailemail addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standardstandards specifications and network programming art, and assumes, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
10/22/2006	0.01	<u>New</u>	Version 0.01 release
1/19/2007	1.0	<u>Major</u>	Version 1.0 release
3/2/2007	1.1	<u>Minor</u>	Version 1.1 release
4/3/2007	1.2	<u>Minor</u>	Version 1.2 release
5/11/2007	1.3	<u>Minor</u>	Version 1.3 release
6/1/2007	1.3.1	Editorial	Changed language and formatting in the technical content.
7/3/2007	2.0	Major	Revised technical content in several sections and created two new sections.
7/20/2007	2.0.1	Editorial	Changed language and formatting in the technical content.
8/10/2007	3.0	Major	Updated content based on feedback.
9/28/2007	3.1	Minor	Made technical and editorial changes based on feedback.
10/23/2007	3.2	Minor	Made technical and editorial changes based on feedback.
11/30/2007	3.3	Minor	Made technical and editorial changes based on feedback.
1/25/2008	3.3.1	Editorial	Changed language and formatting in the technical content.
3/14/2008	3.4	Minor	Clarified the meaning of the technical content.
5/16/2008	4.0	Major	Updated and revised the technical content.
6/20/2008	5.0	Major	Updated and revised the technical content.
7/25/2008	5.1	Minor	Clarified the meaning of the technical content.
8/29/2008	6.0	Major	Updated and revised the technical content.
10/24/2008	6.1	Minor	Clarified the meaning of the technical content.
12/5/2008	7.0	Major	Updated and revised the technical content.
1/16/2009	7.1	Minor	Clarified the meaning of the technical content.
2/27/2009	8.0	Major	Updated and revised the technical content.
4/10/2009	9.0	Major	Updated and revised the technical content.
5/22/2009	10.0	Major	Updated and revised the technical content.
7/2/2009	11.0	Major	Updated and revised the technical content.
8/14/2009	11.1	Minor	Clarified the meaning of the technical content.
9/25/2009	12.0	Major	Updated and revised the technical content.
11/6/2009	13.0	Major	Updated and revised the technical content.
12/18/2009	14.0	Major	Updated and revised the technical content.

Date	Revision History	Revision Class	Comments
1/29/2010	15.0	Major	Updated and revised the technical content.
3/12/2010	15.1	Minor	Clarified the meaning of the technical content.
4/23/2010	16.0	Major	Updated and revised the technical content.
6/4/2010	16.1	Minor	Clarified the meaning of the technical content.
7/16/2010	16.2	Minor	Clarified the meaning of the technical content.
8/27/2010	16.3	Minor	Clarified the meaning of the technical content.
10/8/2010	16.4	Minor	Clarified the meaning of the technical content.
11/19/2010	17.0	Major	Updated and revised the technical content.
1/7/2011	18.0	Major	Updated and revised the technical content.
2/11/2011	18.1	Minor	Clarified the meaning of the technical content.
3/25/2011	19.0	Major	Updated and revised the technical content.
5/6/2011	20.0	Major	Updated and revised the technical content.
6/17/2011	21.0	Major	Updated and revised the technical content.
9/23/2011	21.0	None	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	22.0	Major	Updated and revised the technical content.
3/30/2012	23.0	Major	Updated and revised the technical content.
7/12/2012	24.0	Major	Updated and revised the technical content.
10/25/2012	25.0	Major	Updated and revised the technical content.
1/31/2013	26.0	Major	Updated and revised the technical content.
8/8/2013	27.0	Major	Updated and revised the technical content.
11/14/2013	28.0	Major	Updated and revised the technical content.
2/13/2014	29.0	Major	Updated and revised the technical content.
5/15/2014	29.0	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	30.0	Major	Significantly changed the technical content.
10/16/2015	31.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	7
1.1	Glossary	7
1.2	References	11
1.2.1	Normative References	11
1.2.2	Informative References	13
1.3	Overview	14
1.3.1	Security Background	14
1.3.2	Kerberos Network Authentication Service (V5) Synopsis	14
1.3.3	FAST	16
1.3.4	Compound Identity	16
1.3.5	KILE Synopsis	16
1.4	Relationship to Other Protocols	17
1.5	Prerequisites/Preconditions	17
1.6	Applicability Statement	18
1.7	Versioning and Capability Negotiation	18
1.7.1	Pre-Authentication	18
1.7.2	Encryption Types	18
1.8	Vendor-Extensible Fields	18
1.9	Standards Assignments.....	18
1.9.1	Use of Constants Assigned Elsewhere	18
2	Messages.....	19
2.1	Transport.....	19
2.2	Message Syntax.....	19
2.2.1	KERB-EXT-ERROR.....	19
2.2.2	KERB-ERROR-DATA.....	19
2.2.3	KERB-PA-PAC-REQUEST	20
2.2.4	KERB-LOCAL	20
2.2.5	LSAP_TOKEN_INFO_INTEGRITY.....	20
2.2.6	KERB-AD-RESTRICTION-ENTRY	21
2.2.7	Supported Encryption Types Bit Flags.....	21
2.2.8	PA-SUPPORTED-ENCTYPES	22
2.2.9	OCTET STRING.....	22
2.2.10	PA-PAC-OPTIONS	22
2.3	Directory Service Schema Elements	22
3	Protocol Details.....	24
3.1	Common Details	24
3.1.1	Abstract Data Model.....	24
3.1.1.1	Replay Cache	24
3.1.1.2	Cryptographic Material.....	24
3.1.1.3	Ticket Cache.....	25
3.1.1.4	Machine ID.....	25
3.1.1.5	SupportedEncryptionTypes	25
3.1.1.6	Kerberos OID	25
3.1.2	Timers	25
3.1.3	Initialization	25
3.1.4	Higher-Layer Triggered Events	25
3.1.5	Message Processing Events and Sequencing Rules	25
3.1.5.1	Pre-authentication Data	26
3.1.5.2	Encryption Types	26
3.1.5.3	Encryption Checksum Types	27
3.1.5.4	Ticket Flag Details.....	27
3.1.5.5	Other Elements and Options.....	28
3.1.5.6	Addressing	28

3.1.5.7	Internationalization and Case Sensitivity.....	28
3.1.5.8	Key Version Numbers	29
3.1.5.9	Key Usage Numbers	29
3.1.5.10	Referrals.....	29
3.1.5.11	Naming.....	29
3.1.6	Timer Events.....	30
3.1.7	Other Local Events.....	30
3.1.8	Implementing Public Keys.....	30
3.2	Client Details.....	30
3.2.1	Abstract Data Model.....	30
3.2.2	Timers	32
3.2.3	Initialization.....	32
3.2.4	Higher-Layer Triggered Events	32
3.2.4.1	Initial Logon	32
3.2.4.2	Authentication to Services.....	32
3.2.5	Message Processing Events and Sequencing Rules	33
3.2.5.1	Request Flags Details	33
3.2.5.2	Authenticator Checksum Flags	33
3.2.5.3	Locate a DS_BEHAVIOR_WIN2012 DC	34
3.2.5.4	Using FAST When the Realm Supports FAST	34
3.2.5.5	AS Exchange	35
3.2.5.6	Forwardable TGT Request	35
3.2.5.7	TGS Exchange	35
3.2.5.8	AP Exchange	36
3.2.6	Timer Events.....	36
3.2.7	Other Local Events.....	36
3.3	KDC Details.....	36
3.3.1	Abstract Data Model.....	36
3.3.1.1	Account Database Extensions	38
3.3.2	Timers	39
3.3.3	Initialization.....	39
3.3.4	Higher-Layer Triggered Events	40
3.3.4.1	KDC Configuration Changes.....	40
3.3.5	Message Processing Events and Sequencing Rules	40
3.3.5.1	Request Flag Ticket-issuing Behavior	40
3.3.5.1.1	Server Principal Lookup.....	41
3.3.5.1.2	Canonicalization of Server Principals	42
3.3.5.2	User Account Objects Without UPN.....	42
3.3.5.3	PAC Generation	43
3.3.5.4	Determining Authentication Policy Silo Membership	43
3.3.5.5	Determining Authentication Policy Settings.....	43
3.3.5.6	AS Exchange	45
3.3.5.6.1	Client Principal Lookup	45
3.3.5.6.2	Referrals	47
3.3.5.6.3	Check Account Policy for Every TGT Request.....	47
3.3.5.6.4	Initial Population of the PAC.....	48
3.3.5.6.4.1	KERB_VALIDATION_INFO Structure	48
3.3.5.6.4.2	PAC_CLIENT_INFO Structure.....	50
3.3.5.6.4.3	Server Signature	50
3.3.5.6.4.4	KDC Signatures	51
3.3.5.6.4.5	UPN_DNS_INFO Structure.....	51
3.3.5.6.4.6	PAC_CLIENT_CLAIMS_INFO Structure	51
3.3.5.7	TGS Exchange	52
3.3.5.7.1	Check Account Policy for Every Session Ticket Request	53
3.3.5.7.2	TGT without a PAC.....	53
3.3.5.7.3	Domain Local Group Membership	54
3.3.5.7.4	Compound Identity	55
3.3.5.7.5	Cross-Domain Trust and Referrals	56

3.3.5.7.6	FORWARDED TGT etype	56
3.3.5.7.7	Read-only Domain Controller (RODC).....	57
3.3.6	Timer Events.....	57
3.3.7	Other Local Events.....	57
3.4	Application Server Details	57
3.4.1	Abstract Data Model.....	57
3.4.2	Timers	57
3.4.3	Initialization.....	57
3.4.3.1	msDS-SupportedEncryptionTypes attribute	58
3.4.4	Higher-Layer Triggered Events	58
3.4.5	Message Processing Events and Sequencing Rules	58
3.4.5.1	Three-Leg DCE-Style Mutual Authentication	59
3.4.5.2	Datagram-Style Authentication	59
3.4.5.3	Processing Authorization Data	60
3.4.5.4	GSS_WrapEx() Call	61
3.4.5.4.1	Kerberos Binding of GSS_WrapEx()	62
3.4.5.5	GSS_UnwrapEx() Call.....	63
3.4.5.6	GSS_GetMICEx() Call	63
3.4.5.7	GSS_VerifyMICEx() Call	64
3.4.6	Timer Events.....	64
3.4.7	Other Local Events.....	64
4	Protocol Examples	65
4.1	Interactive Logon Using Passwords.....	65
4.2	Network Logon	67
4.3	GSS_WrapEx with AES128-CTS-HMAC-SHA1-96	67
4.4	AES 128 Key Creation.....	69
4.5	RC4 GSS_WrapEx	70
5	Security	72
5.1	Security Considerations for Implementers	72
5.1.1	RODC Key Version Numbers.....	72
5.1.2	SPNs with Serviceclass Equal to "RestrictedKrbHost"	72
5.1.3	Account Revocation Checking	72
5.1.4	FORWARDED TGT etype	72
5.1.5	DES Downgrade Protection	72
5.2	Index of Security Parameters	72
6	Appendix A: Product Behavior	73
7	Change Tracking.....	79
8	Index.....	81

1 Introduction

Kerberos Network Authentication Service V5 Extensions apply 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.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative ~~and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain these terms.~~ All other sections and examples in this specification are informative.

1.1 Glossary

~~This document uses the~~ following terms ~~are specific to this document:~~

Active Directory: A general-purpose network **directory service**. **Active Directory** also refers to the Windows implementation of a **directory service**. **Active Directory** stores information about a variety of objects in the network. Importantly, user accounts, computer accounts, groups, and all related credential information used by the Windows implementation of Kerberos are stored in **Active Directory**. **Active Directory** is either deployed as Active Directory Domain Services (AD DS) or Active Directory Lightweight Directory Services (AD LDS). [MS-ADTS] describes both forms. For more information, see [MS-AUTHSOD] section 1.1.1.5.2, Lightweight Directory Access Protocol (LDAP) versions 2 and 3, Kerberos, and **DNS**.

~~**AP exchange:** See **Authentication Protocol (AP) exchange**.~~

Authentication Protocol (AP) exchange: The Kerberos subprotocol called the "authentication protocol", sometimes referred to as the "Client/Server Authentication Exchange", in which the client presents a **service ticket** and an **authenticator** to a service to establish an authenticated communication session with the service. ~~The protocol is specified in (see [RFC4120] section 3.2-).~~

Authentication Service (AS): A service that issues ticket granting tickets (TGTs), which are used for authenticating principals within the **realm** or **domain** served by the **Authentication Service**.

Authentication Service (AS) exchange: The Kerberos subprotocol in which the **Authentication Service (AS)** component of the **key distribution center (KDC)** accepts an initial logon or authentication request from a client and provides the client with a **ticket-granting ticket (TGT)** and necessary cryptographic keys to make use of the **ticket**. This is specified in [RFC4120] section 3.1. The **AS exchange** is always initiated by the client, usually in response to the initial logon of a principal such as a user.

authenticator: When used in reference to Kerberos, see **Kerberos authenticator**.

authorization data: An extensible field within a Kerberos **ticket**, used to pass authorization data about the principal on whose behalf the **ticket** was issued to the application service.

claim: An assertion about a security principal expressed as ~~an~~ ~~the~~ n-tuple ~~containing an~~ {Identifier, ValueType ~~and~~ m ~~Values Value(s)~~ of type ValueType} where m ~~>~~ ~~is greater than or equal to~~ 1. A claim with only ~~1 value one Value~~ in the n-tuple is called a single-valued claim ~~and~~; a claim with more than ~~1 value one Value~~ is called a multi-valued claim.

Compound identity TGS-REQ: A **FAST** TGS-REQ that uses explicit **FAST** armoring using the computer's **ticket-granting ticket (TGT)**.

datagram: A style of communication offered by a network transport protocol where each message is contained within a single network packet. In this style, there is no requirement for establishing a session prior to communication, as opposed to a connection-oriented style.

directory: The database that stores information about objects such as users, groups, computers, printers, and the **directory service** that makes this information available to users and applications.

directory service (DS): A service that stores and organizes information about a computer network's users and network shares, and that allows network administrators to manage users' access to the shares. See also **Active Directory**.

distinguished name (DN): ~~In Active Directory, a name that uniquely identifies an object by using the Active Directory directory service, relative distinguished name (RDN) for the unique identifier object, and the names of an object container objects and domains that contain the object. The distinguished name (DN) identifies the object and its location in Active Directory, as described in [MS-ADTS] and [RFC2251]-a tree.~~

domain: A set of users and computers sharing a common namespace and management infrastructure. At least one computer member of the set must act as a **domain controller (DC)** and host a member list that identifies all members of the domain, as well as optionally hosting the **Active Directory** service. The domain controller provides authentication (2) of members, creating a unit of trust for its members. Each domain has an identifier that is shared among its members. For more information, see [MS-AUTHSOD] section 1.1.1.5 and [MS-ADTS].

domain controller (DC): The service, running on a server, that implements **Active Directory**, or the server hosting this service. The service hosts the data store for objects and interoperates with other **DCs** to ensure that a local change to an object replicates correctly across all **DCs**. When **Active Directory** is operating as Active Directory Domain Services (AD DS), the **DC** contains full NC replicas of the configuration naming context (config NC), schema naming context (schema NC), and one of the domain NCs in its forest. If the AD DS **DC** is a global catalog server (GC server), it contains partial NC replicas of the remaining domain NCs in its forest. For more information, see [MS-AUTHSOD] section 1.1.1.5.2 and [MS-ADTS]. When **Active Directory** is operating as Active Directory Lightweight Directory Services (AD LDS), several AD LDS **DCs** can run on one server. When **Active Directory** is operating as AD DS, only one AD DS **DC** can run on one server. However, several AD LDS **DCs** can coexist with one AD DS **DC** on one server. The AD LDS **DC** contains full NC replicas of the config NC and the schema NC in its forest. The domain controller is the server side of Authentication Protocol Domain Support [MS-APDS].

Domain Name System (DNS): A hierarchical, distributed database that contains mappings of domain names (1) to various types of data, such as IP addresses. DNS enables the location of computers and services by user-friendly names, and it also enables the discovery of other information stored in the database.

FAST armor: Using a **ticket-granting ticket (TGT)** for the principal to protect Kerberos messages, as described in [RFC6113].

Flexible Authentication Secure Tunneling (FAST): FAST provides a protected channel between the client and the **Key Distribution Center (KDC)**.

fully qualified domain name (FQDN): ~~In Active Directory, a fully qualified domain name (FQDN) that identifies a domain~~ An unambiguous domain name (2) that gives an absolute location in the Domain Name System's (DNS) hierarchy tree, as defined in [RFC1035] section 3.1 and [RFC2181] section 11.

Generic Security Services (GSS): An Internet standard, as described in [RFC2743], for providing security services to applications. It consists of an application programming interface (GSS-API) set, as well as standards that describe the structure of the security data.

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

Internet host name: The name of a host as defined in [RFC1123] section 2.1, with the extensions described in [MS-HNDS].

Kerberos authenticator: A record sent with a **ticket** to a server to certify the client's knowledge of the **session key** in the **ticket**; to help the server detect replay attacks by proving that the authenticator is recently constructed; and to help the two parties select additional **session keys** for a particular connection authenticated by Kerberos. The use of authenticators, including how authenticators are validated, is specified in [RFC4120] section 5.5.1. For more information, see [KAUFMAN].

Kerberos principal: A unique individual account known to the **Key Distribution Center (KDC)**. Often a user, but it can be a service offering a resource on the network.

key: In cryptography, a generic term used to refer to cryptographic data that is used to initialize a cryptographic algorithm. **Keys** are also sometimes referred to as keying material.

Key Distribution Center (KDC): The Kerberos service that implements the authentication (2) and **ticket** granting services specified in the Kerberos protocol. The service runs on computers selected by the administrator of the **realm** or domain; it is not present on every machine on the network. It must have access to an account database for the **realm** that it serves. Windows **KDCs** are integrated into the **domain controller** role of a Windows Server operating system acting as a Domain Controller. It is a network service that supplies **tickets** to clients for use in authenticating to services.

~~**KRB_AP_REQ/KRB_AP_REP:** The request and response messages used in the **Authentication Protocol (AP) exchange**.~~

~~**KRB_AS_REQ/KRB_AS_REP:** The request and response messages used in the **Authentication Service (AS) Exchange**.~~

~~**KRB_PRIV:** A Kerberos message ([RFC4120] section 3.5) that is part of a **KRB_PRIV** exchange.~~

~~**KRB_SAFE exchange:** The Kerberos subprotocol used by clients to detect modifications of messages they exchange with a server in a **session** that is already established through the **AP exchange**. This exchange is initiated when a client sends a **KRB_SAFE** message, as specified in [RFC4120] section 3.4.~~

little-endian: Multiple-byte values that are byte-ordered with the least significant byte stored in the memory location with the lowest address.

object identifier (OID): In the context of an object server, a 64-bit number that uniquely identifies an object.

objectGUID: The attribute on an **Active Directory** object whose value is a GUID that uniquely identifies the object. The GUID value of an object's objectGUID is assigned when the object was created and is immutable thereafter. The integrity of object references between NCs and of replication depends on the integrity of the objectGUID attribute. For a description of the general concept of an "object", see [MS-ADTS] section 1. For more detailed information see [MS-ADTS] section 3.1.1.1.3.

pre-authentication: In Kerberos, a state in which a **key distribution center (KDC)** demands that the requestor in the **Authentication Service (AS) exchange** demonstrate knowledge of the key associated with the account. If the requestor cannot demonstrate this knowledge, the **KDC** will not issue a **ticket-granting ticket (TGT)** ([RFC4120] sections 5.2.7 and 7.5.2).

privilege attribute certificate (PAC): A Microsoft-specific authorization data present in the authorization data field of a ticket. The **PAC** contains several logical components, including

group membership data for authorization, alternate credentials for non-Kerberos authentication protocols, and policy control information for supporting interactive logon.

read-only domain controller (RODC): A **domain controller (DC)** that does not accept originating updates. Additionally, an **RODC** does not perform outbound replication. An RODC cannot be the primary domain controller (PDC) for its domain.

realm: A collection of key distribution centers (KDCs) with a common set of principals, as described in [RFC4120] section 1.2.

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 3.1.5.11.

secret key: A symmetric encryption key shared by two entities, such as between a user and the **domain controller (DC)**, with a long lifetime. A password is a common example of a secret key. When used in a context that implies Kerberos only, a principal's secret key.

security identifier (SID): An identifier for security principals in Windows that is used to identify an account or a group. Conceptually, the **SID** is composed of an account authority portion (typically a **domain**) and a smaller integer representing an identity relative to the account authority, termed the relative identifier (RID). The **SID** format is specified in [MS-DTYP] section 2.4.2; a string representation of **SIDs** is specified in [MS-DTYP] section 2.4.2 and [MS-AZOD] section 1.1.1.2.

Security Support Provider Interface (SSPI): A Windows-specific API implementation that provides the means for connected applications to call one of several security providers to establish authenticated connections and to exchange data securely over those connections. This is the Windows equivalent of Generic Security Services (GSS)-API, and the two families of APIs are on-the-wire compatible.

service: A process or agent that is available on the network, offering resources or services for clients. Examples of services include file servers, web servers, and so on.

service principal name (SPN): The name a client uses to identify a service for mutual authentication. (For more information, see [RFC1964] section 2.1.1.) An **SPN** consists of either two parts or three parts, each separated by a forward slash ('/'). The first part is the service class, the second part is the instance name, and the third part (if present) is the service name. For example, "ldap/dc-01.fabrikam.com/fabrikam.com" is a three-part **SPN** where "ldap" is the service class name, "dc-01.fabrikam.com" is the instance name, and "fabrikam.com" is the service name. See [SPNNAMES] for more information about **SPN** format and composing a unique **SPN**.

service ticket: A **ticket** for any service other than the **ticket-granting service (TGS)**. A **service ticket** serves only to classify a **ticket** as not a **ticket-granting ticket (TGT)** or cross-realm TGT, as specified in [RFC4120].

session: In Kerberos, an active communication channel established through Kerberos that also has an associated cryptographic **key**, message counters, and other state.

session key: A relatively short-lived symmetric key (a cryptographic key negotiated by the client and the server based on a shared secret). A **session key's** lifespan is bounded by the **session** to which it is associated. A **session key** ~~should~~has to be strong enough to withstand cryptanalysis for the lifespan of the **session**.

SRV record: A type of information record in DNS that maps the name of a service to the DNS name of a server that offers that service. **domain controllers (DCs)** advertise their capabilities by publishing **SRV records** in DNS.

ticket: A record generated by the **key distribution center (KDC)** that helps a client authenticate to a service. It contains the client's identity, a unique cryptographic key for use with this ticket (the **session key**), a time stamp, and other information, all sealed using the service's **secret key**. It only serves to authenticate a client when presented along with a valid authenticator.

ticket-granting service (TGS): A service that issues **tickets** for admission to other services in its own domain or for admission to the ticket-granting service in another domain.

ticket-granting service (TGS) exchange: The Kerberos subprotocol in which the **key distribution center (KDC)** distributes a session key and a ticket for the service requested by the client, as specified in [RFC4120] section 3.3. This exchange is initiated when the client sends the **KDC** a KRB_TGS_REQ message.

ticket-granting ticket (TGT): A special type of **ticket** that can be used to obtain other **tickets**. The TGT is obtained after the initial authentication in the **Authentication Service (AS) exchange**; thereafter, users do not need to present their credentials, but can use the TGT to obtain subsequent tickets.

user principal name (UPN): A user account name (sometimes referred to as the user logon name) and a domain name that identifies the domain in which the user account is located. This is the standard usage for logging on to a Windows domain. The format is: someone@example.com (in the form of an email address). In **Active Directory**, the userPrincipalName attribute (2) of the account object, as described in [MS-ADTS].

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

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.

[C706] The Open Group, "DCE 1.1: Remote Procedure Call", C706, August 1997, <https://www2.opengroup.org/ogsys/catalog/c706>

[FIPS140] FIPS PUBS, "Security Requirements for Cryptographic Modules", FIPS PUB 140, December 2002, <http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf>

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

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

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

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

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

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

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

[MS-ERREF] Microsoft Corporation, "Windows Error Codes".

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

[MS-KKDCP] Microsoft Corporation, "Kerberos Key Distribution Center (KDC) Proxy Protocol".

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

[MS-NRPC] Microsoft Corporation, "Netlogon Remote Protocol".

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

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

[MS-RRP] Microsoft Corporation, "Windows Remote Registry Protocol".

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

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

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

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

[MS-WKST] Microsoft Corporation, "Workstation Service Remote Protocol".

[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.rfc-editor.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.rfc-editor.org/rfc/rfc2119.txt>

[RFC2279] Yergeau, F., "UTF-8, a transformation format of ISO 10646", RFC 2279, January 1998, <http://www.rfc-editor.org/rfc/rfc2279.txt>

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

[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.rfc-editor.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>

[RFC6113] Hartman, S., and Zhu, L., "A Generalized Framework for Kerberos Pre-Authentication", RFC 6113, April 2011, <http://www.ietf.org/rfc/rfc6113.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>

[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>

1.2.2 Informative References

[ADDL] 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".

[MS-GPOD] Microsoft Corporation, "Group Policy Protocols Overview".

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

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

[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.rfc-editor.org/rfc/rfc2396.txt>

[UNICODE] The Unicode Consortium, "The Unicode Consortium 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>

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~~The implementer has to study ~~must be made, particularly of~~carefully 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.

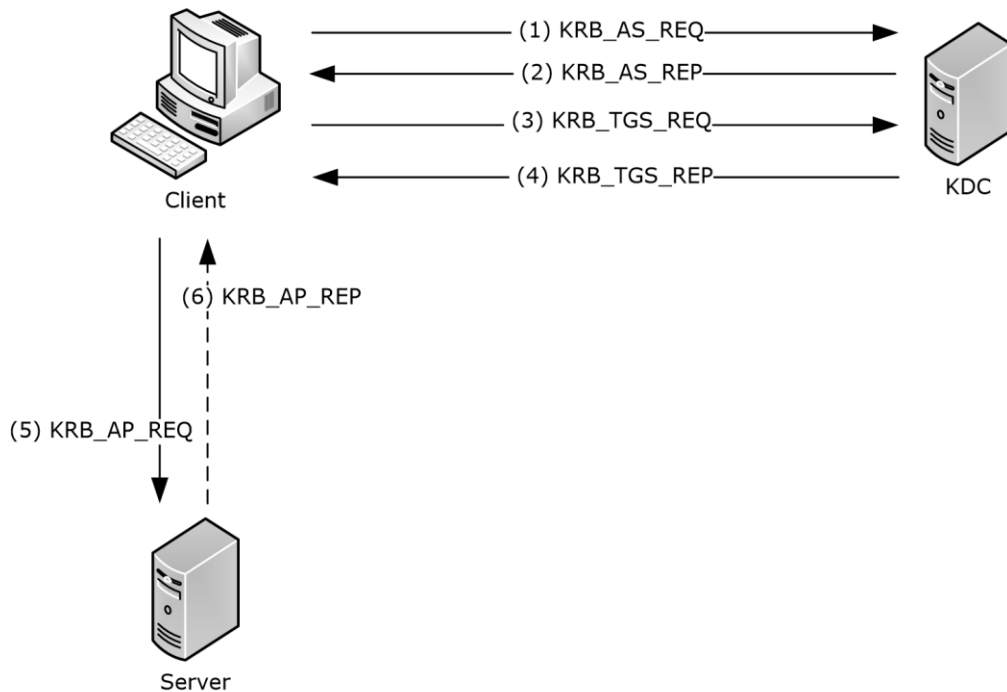


Figure 1: Kerberos V5 Exchanges

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 [message](#) (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 [message](#) (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 (TGS) request [message](#) (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), a **Kerberos authenticator** ([RFC4120] section 5.5.1), and the **Service Principal Name (SPN)**.
- Kerberos ticket-granting service (TGS) response [message](#) (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 **message** (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 **message** (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 FAST

Flexible Authentication Secure Tunneling (FAST) provides a protected channel between the client and the Key Distribution Center (KDC). FAST is only available for Authentication Service (AS) and ticket-granting service (TGS) exchanges.

FAST armor uses a ticket-granting ticket (TGT) for the computer to protect Authentication Service (AS) exchanges with the KDC, so the computer's AS exchange is not armored. The user's TGT is used to protect its TGS exchanges with the KDC.

1.3.4 Compound Identity

KILE extends FAST to support compound identity in the following manner. The client sends a **compound identity TGS-REQ** which is a FAST TGS-REQ by using explicit armoring with the computer's TGT. When a KDC receives a compound identity TGS-REQ for an application server which supports compound identity, then the KDC adds the computer's authorization data to the **privilege attribute certificate (PAC)**. By providing authorization data for the computer in the PAC, the application server can create a compound identity for the caller which is a combination of the user's and computer's authorization data.

1.3.5 KILE Synopsis

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

- Group membership
- **Claims**
- Interactive logon information
- Integrity levels

By extending FAST, KILE provides the server with additional information such as:

- Group membership and claims for the computer on which the client is running

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 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 Network Authentication Service (V5) Extensions do 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.

The Generalized Framework for Kerberos Pre-Authentication ([RFC6113]) includes several areas for vendor extension.

KILE does not provide vendor extensibility beyond what is specified in [RFC4120] and [RFC6113].

1.9 Standards Assignments

Assignment of Kerberos V5 IANA numbers is as specified in [RFC4120] section 9 and [RFC6113] sections 6 and 7. 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 exchange 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 6.3.2.3.

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-EXT-ERROR

This structure is a Windows-specific structure returned by the KDC to provide extended error information.

```
typedef struct KERB_EXT_ERROR {
    unsigned long status;
    unsigned long reserved;
    unsigned long flags;
} KERB_EXT_ERROR;
```

Status: An NTSTATUS value. For details about NTSTATUS values, see [MS-ERREF] section 2.3.

Reserved: SHOULD be set to zero and MUST be ignored on receipt.

Flags: SHOULD be set to 0x00000001.

2.2.2 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, and by the KDC for extended errors.

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

Data-type: This value SHOULD be as follows.

Integer Value	Meaning
2 KERB_AP_ERR_TYPE_SKEW_RECOVERY	Clock skew recovery was attempted.
3 KERB_ERR_TYPE_EXTENDED	The Data-value field contains extended, implementation-specific error information.

Data-value: This value SHOULD be as follows.

Data Type	Data Value
KERB_AP_ERR_TYPE_SKEW_RECOVERY	NULL.
KERB_ERR_TYPE_EXTENDED	A KERB-EXT-ERROR structure (section 2.2.1).

2.2.3 KERB-PA-PAC-REQUEST

This structure is a padata 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
}
```

2.2.4 KERB-LOCAL

The KERB-LOCAL structure contains implementation-specific data used when the Kerberos client and application server are on the same host.<3> Its structure is defined using ASN.1 notation, and the syntax is as follows.

```
KERB-LOCAL ::= OCTET STRING -- Implementation-specific data which MUST be
                             -- ignored if Kerberos client is not local.
```

2.2.5 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.

2.2.6 KERB-AD-RESTRICTION-ENTRY

The KERB-AD-RESTRICTION-ENTRY structure specifies additional restrictions for the client. 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.

2.2.7 Supported Encryption Types Bit Flags

The data in the **msDS-SupportedEncryptionTypes** attribute ([MS-ADA2] section 2.458463), 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.

1										2										3						
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1					
0	0	0	0	0	0	0	0	0	0	0	0	I	H	G	F	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
F	FAST-supported<6>
G	Compound-identity-supported<7>
H	Claims-supported<8>
I	Resource-SID-compression-disabled<9>

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

2.2.8 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 2.2.7).<10>

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

2.2.9 OCTET STRING

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

2.2.10 PA-PAC-OPTIONS

The PA-PAC-OPTIONS structure specifies explicitly requested options in the PAC. Its structure is defined using ASN.1 notation. The syntax is as follows:<11>

```
PA-PAC-OPTIONS ::= SEQUENCE {
  KerberosFlags
  -- Claims (0)
  -- Branch Aware (1)
  -- Forward to Full DC (2)
}
Note: KerberosFlags ::= BIT STRING (SIZE (32..MAX))
      -- minimum number of bits shall be sent, but no fewer than 32
```

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 msDS-SupportedEncryptionTypes servicePrincipalName userAccountControl userPrincipalName <u>sAMAccountName</u>

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 [exchange](#) requests.

3.1 Common Details

3.1.1 Abstract Data Model

Kerberos V5 specifies the abstract data model for common elements.

~~KILE key version numbers (as defined in [RFC4120] section 5.2.9) are signed 32-bit integers.~~

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 [RFC4120] section 5.5.2.) 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 [RFC4120] section 5.5.1).

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. <12>

3.1.1.4 Machine ID

KILE implements a 32-byte binary random string machine ID. <13>

3.1.1.5 SupportedEncryptionTypes

KILE implements a 32-bit unsigned integer that contains a combination of flags that specify what encryption types (section 2.2.7) are supported by Kerberos. <14> The default is 0000001C. <15>

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 padata 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 supports the following pre-authentication types added in [RFC6113] section 7.1:

- PA-FX-COOKIE [133]
- PA-FX-FAST [136]
- PA-FX-ERROR [137]
- PA-ENCRYPTED-CHALLENGE [138]

KILE adds the following pre-authentication types:

- PA-SUPPORTED_ENCTYPES [165] (section 2.2.8)<16>
- PA-PAC-OPTIONS [167] (section 2.2.10)<17>

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:<18>

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

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

and MAY<19> support the other following encryption types, which are listed in order of relative strength:

- RC4-HMAC [23] [RFC4757] <20>
- RC4-HMAC-EXP [24] [RFC4757] <21>
- DES-CBC-MD5 [3] [RFC3961] <22>
- DES-CBC-CRC [1] [RFC3961] <23>

Kerberos V5 encryption type assigned numbers are specified in [RFC3961] section 8, [RFC4757] section 5, and [RFC3962] section 7.<24>

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 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 flag.
- 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 3.2.5.8).
- KERB_AUTH_DATA_TOKEN_RESTRICTIONS (sections 3.2.5.8 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).<25>

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 key version numbers (as defined in [RFC4120] section 5.2.9) are unsigned encoded and decoded as signed 32-bit integers.

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

The key version number consists of 32 bits. The first 16 bits, including the most significant bit, are an unsigned 16-bit number which SHOULD identify the RODC. 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 Network Authentication Service (V5) 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 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 [RFC1964] section 2.1.1 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 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 a target name which SHOULD be an SPN <27> for their service applications for Kerberos authentication.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

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

The KILE client has the following configuration setting for claims, compound authentication, and FAST:

EnableCBACandArmor: A Boolean setting that indicates that the Kerberos client is claims-, compound authentication-, and FAST-aware. The default is FALSE. Implementations that use the Windows registry to persistently store and retrieve the EnableCBACandArmor variable SHOULD use the following registry path:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\System\Kerberos\Parameters
```

The KILE client has the following configuration setting for FAST:

RequireFAST: A Boolean setting that indicates that the Kerberos FAST client MUST enforce FAST. The default is FALSE. Implementations that use the Windows registry to persistently store and retrieve the RequireFast variable SHOULD use the HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\System\Kerberos\Parameters registry path.

The KILE client has the following configuration setting for non-KILE realms:

RealmCanonicalize: SHOULD be initialized in an implementation specific way. Implementations that use the Windows registry to persistently store and retrieve the RealmCanonicalize variable SHOULD use the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Domains\ registry path,

which is the name of the realm, and RealmFlags key bit 0x8 is set when the non-KILE realm supports canonicalization.

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]).

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 Active Directory 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 [exchange](#) 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.

NOTE: The KRB_SAFE and KRB_PRIV messages are part of the KRB_SAFE exchange and KRB_PRIV exchange, respectively.

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, and [Referrals-11] section 3). For non-KILE realms, if RealmCanonicalize is not set for the realm, the client SHOULD NOT set the canonicalize flag ([RFC4120] section 5.4.1).

The client SHOULD NOT set the PROXY or PROXIABLE option ([RFC4120] section 2.5).

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).

If the Kerberos client does not have network access to the KDC and KKDCP is supported, the Kerberos client SHOULD call ProxyMessage() ([MS-KKDCP] section 3.1.5.1) where:

- `kerb-message` contains the KRB_AS_REQ or KRB_TGS_REQ [message](#).
- `target-domain` contains the realm field of the KRB_AS_REQ or KRB_TGS_REQ message ([RFC4120] section 5.4.1).
- `dlocator-hint` is the *Flags* parameter ([MS-NRPC] section 3.5.4.3.1) the client used to find a domain controller for the Kerberos message to determine that a KDC was not accessible.

If `Output_kerb_message` is returned, then process the KRB_AS_REP, KRB_TGS_REP, or KRB_ERROR [message](#) contained in `Output_kerb_message.kerb-message`. Otherwise, the Kerberos client SHOULD fail.

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_DELEG_FLAG ([RFC4121] section 4.1.1.1).

ExtendedError: GSS_C_EXTENDED_ERROR_FLAG ([RFC4757] section 7.1).

Identify: GSS_C_IDENTIFY_FLAG ([RFC4757] section 7.1); set in the GSS_Init_sec_context call ([RFC1964] section 1.1.1).

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 Locate a DS_BEHAVIOR_WIN2012 DC

When a DS_BEHAVIOR_WIN2012 domain controller (DC) is required, DsrGetDcNameEx2 ([MS-NRPC] section 3.5.4.3.1) is called where:

- AccountName is the client account name.
- AllowableAccountControlBits has bits A, B, C, D, E, and F set.
- DomainName is the client domain name.
- Flags has bits G, H, and U set.
- All other fields are set to NULL.

The IP address of the DS_BEHAVIOR_WIN2012 DC is returned in DomainControllerInfo.DomainControllerAddress.

3.2.5.4 Using FAST When the Realm Supports FAST

In addition to the RFC behavior ([RFC6113]), the Kerberos client SHOULD use the PA-SUPPORTED-ENCTYPES from the TGT obtained from a realm to determine if a realm supports FAST. <28>

1. If the client does not have a TGT for the realm and is creating an:
 - AS-REQ: the client SHOULD obtain a TGT for the computer principal from the user principal's domain.
 - TGS-REQ: the client SHOULD obtain a referral TGT for the user principal for the target domain.
 - Compound identity TGS-REQ: the client SHOULD obtain a user principal TGT and computer principal TGT for the target domain with the same key version numbers (section 3.1.5.8).

If a TGT for the required principals cannot be obtained and RequireFAST is:

- TRUE: the client SHOULD fail the request.
- FALSE: the client SHOULD continue without FAST.

2. When processing the AS_REP or TGS_REP, if the FAST-supported bit in the in PA-SUPPORTED-ENCTYPES of the TGT received in step 1 is:

- Not set and RequireFAST is TRUE: the client SHOULD fail the request.
- Not set and RequireFAST is FALSE: the client SHOULD continue without FAST.
- Set: the client SHOULD find a DC that supports FAST and use FAST:

Locate a DS_BEHAVIOR_WIN2012 DC (section 3.2.5.3).

If a DS_BEHAVIOR_WIN2012 DC is not found and RequireFAST is:

- TRUE: the client SHOULD fail the request.
- FALSE: the client SHOULD continue without FAST.

If a DS_BEHAVIOR_WIN2012 DC is found, the client SHOULD use the TGT obtained in step 1 to armor the message it is creating ([RFC6113] sections 5.4.2, 5.4.3 and 5.4.4) to the DS_BEHAVIOR_WIN2012 DC. If the request fails without an authenticated Kerberos error message ([RFC6113] section 5.4.4) and RequireFAST is TRUE, then the client SHOULD fail the request.

3.2.5.5 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 padata type when generating an AS-REQ message. The PAC is specified in [MS-PAC].

If EnableCBACandArmor is TRUE, the client SHOULD<29> behave as follows:

1. When sending the AS_REQ, add a PA-PAC-OPTIONS [167] (section 2.2.10) padata type with the Claims bit set in the AS_REQ to request claims authorization data.
2. When receiving the AS_REP, if the Claims bit is set in PA-SUPPORTED-ENCTYPES [165], and not set in PA-PAC-OPTIONS [167], the client SHOULD locate a DS_BEHAVIOR_WIN2012 DC (section 3.2.5.3) and go back to step 1.

If EnableCBACandArmor is TRUE, the principal is not the computer account, and the client is running on a domain-joined computer, the Kerberos client SHOULD use FAST [RFC6113] when the principal's Realm supports FAST (section 3.2.5.4).<30>

3.2.5.6 Forwardable TGT Request

When the client requests a forwardable TGT ([RFC4120] Section 2.6) for the application server, the client SHOULD:<31>

- 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.

3.2.5.7 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 2.2.4) in an AD-IF-RELEVANT element ([RFC4120] section 5.2.6.1) in the enc-authorization-data field ([RFC4120] section 5.2.6).<32>

The Kerberos client SHOULD add a PA-PAC-OPTIONS [167] (section 2.2.10) padata type with the Branch Aware bit set to the TGS REQ. If a server principal unknown with a substatus of NTSTATUS STATUS_NO_SECRETS message ([MS-ERREF] section 2.3.1) is returned, the client SHOULD send an AS-REQ adding a PA-PAC-OPTIONS [167] (section 2.2.10) padata type, with the Forward to Full DC bit set, to a full DC, and then send a new TGS_REQ using this TGT to the full DC.

If EnableCBACandArmor is TRUE, the Kerberos client SHOULD add a PA-PAC-OPTIONS [167] (section 2.2.10) padata type with the Claims bit set in the TGS REQ to notify the KDC that the client is claims aware.<33>

If EnableCBACandArmor is TRUE, the Kerberos client SHOULD use FAST [RFC6113] when the realm supports FAST (section 3.2.5.4).<34>

If EnableCBACandArmor is TRUE and the application server's realm TGT's PA-SUPPORTED-ENCTYPES Compound Identity bit is set, the Kerberos client SHOULD send a compound identity TGS-REQ by using FAST with explicit armoring, using the computer's TGT.<35>

3.2.5.8 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 2.2.4).
- KERB_AUTH_DATA_TOKEN_RESTRICTIONS (141), containing the KERB-AD-RESTRICTION-ENTRY structure (section 2.2.6).<36>

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 variable 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 2.2.2) 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.<37>

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:

- **Minimum lifetime** ([RFC4120] section 8.2): 0 minutes.
- **MaxRenewAge**: A 64-bit signed integer containing the maximum renewable lifetime ([RFC4120] section 8.2). KILE implementations, which use the LSAD for the configuration database, SHOULD directly access the **MaxRenewAge** field in the Kerberos Policy Information ([MS-LSAD] section 3.1.1.1).
- **MaxClockSkew**: A 64-bit signed integer containing the Acceptable clock skew ([RFC4120] section 8.2). KILE implementations, which use the LSAD for the configuration database, SHOULD directly access the **MaxClockSkew** field in the Kerberos Policy Information ([MS-LSAD] section 3.1.1.1).

The maximum ticket lifetime ([RFC4120] section 8.2) is configured separately for TGTs and service tickets:

- **MaxServiceTicketAge:** A 64-bit signed integer containing the maximum service ticket lifetime. KILE implementations, which use the LSAD for the configuration database, SHOULD directly access the **MaxServiceTicketAge** field in the Kerberos Policy Information ([MS-LSAD] section 3.1.1.1). The default is 10 hours.
- **MaxTicketAge:** A 64-bit signed integer containing the maximum TGT lifetime. KILE implementations, which use the LSAD for the configuration database, SHOULD directly access the **MaxTicketAge** field in the Kerberos Policy Information ([MS-LSAD] section 3.1.1.1). The default is 10 hours.

KILE also adds the following new KDC configuration setting:

- **AuthenticationOptions:** A 32-bit unsigned integer containing the POLICY_KERBEROS_VALIDATE_CLIENT flag. KILE implementations, which use the LSAD for the configuration database, SHOULD directly access 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.

The KDC configuration setting is a registry key, ClaimsCompIdFASTSupport. This is a 32-bit unsigned integer, used as follows: <38>

- If set to 0, there are no new behaviors.
- If set to 1, the KDC supports claims, compound identity, and FAST and other KDCs in the domain do not.
- If set to 2, all KDCs in the domain support claims, compound identity, and FAST.
- If set to 3, all KDCs in the domain support claims and compound identity and enforce FAST.

Implementations that use the Windows registry to persistently store and retrieve this variable SHOULD use the following:

- RegistryValueName: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\KDC\Parameters
- RegistryValueType: 4
- RegistryValue: CbacAndArmorLevel

The implementation SHOULD also expose the key and value at the specified registry path using the Windows Remote Registry Protocol ([MS-RRP]). For each abstract data model element that is loaded from the registry, there is one instance that is shared between the Windows Remote Registry Protocol and any protocols that use the abstract data model element. Any changes made to the registry keys will be reflected in the abstract data model elements when a PolicyChange event is received ([MS-GPOD] section 2.8.2) or on KDC start up.

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-WKST] section 3.2.1.2).
- **NetbiosDomainName:** The NetBIOS domain name for the domain to which the server belongs. This Abstract Data Model element is shared with **DomainName.NetBIOS** ([MS-WKST] section 3.2.1.6).
- **DomainSid:** A **security identifier (SID)** for the domain. This Abstract Data Model element is shared with **DomainSid** ([MS-WKST] section 3.2.1.6).

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 account database SHOULD use the **supplementalCredentials** attribute ([MS-ADA3] section 2.287).
- **KerbSupportedEncryptionTypes:** A 32-bit unsigned integer that contains a combination of flags that specify what encryption types (section 2.2.7) are supported by the application server, and whether compound identity is supported. <39> KILE implementations that use an Active Directory for the account database SHOULD use the **msDS-SupportedEncryptionTypes** attribute ([MS-ADA2] section 2.458463).

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 2.2.16) NA flag. The default is FALSE.
- **AssignedPolicy:** A link to the policy. KILE implementations that use an Active Directory for the account database SHOULD use the msDS-AssignedAuthNPolicy attribute ([MS-ADA2] section 2.216).
- **AssignedSilo:** A link to the silo. KILE implementations that use an Active Directory for the account database SHOULD use the msDS-AssignedAuthNPolicySilo attribute ([MS-ADA2] section 2.218).
- **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 2.2.16) 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 2.2.16) 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 2.2.16) 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 2.2.16) 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 2.2.16) 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 2.2.16) TD flag. The default is FALSE.
- **UseDESOOnly:** 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 2.2.16) 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 2.2.16):

- D flag: Disabled
- DK flag: UseDESOOnly
- 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 LsarQueryDomainInformationPolicy method ([MS-LSAD] section 3.1.4.4.7), and the *InformationClass* parameter SHOULD be set to the value of PolicyDomainKerberosTicketInformation in order to retrieve the current values. The KDC SHOULD set its configuration settings as follows:

- **MaxRenewAge** (section 3.3.1) to the value of the **MaxRenewAge** field.
- **MaxClockSkew** (section 3.3.1) to the value of the **MaxClockSkew** field.
- **MaxServiceTicketAge** (section 3.3.1) to the value of the **MaxServiceTicketAge** field.
- **MaxTicketAge** (section 3.3.1) to the value of the **MaxTicketAge** field.
- **AuthenticationOptions** (section 3.3.1) to the value of the **AuthenticationOptions** field.

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

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.

If the KDC supports: <40>

- **FAST:** the KDC SHOULD set the FAST-supported bit on the krbtgt account's **KerbSupportedEncryptionTypes**.
- **Claims:** the KDC SHOULD set the Claims-supported bit on the krbtgt account's **KerbSupportedEncryptionTypes**.

3.3.4 Higher-Layer Triggered Events

For KILE implementations which use the LSAD for the configuration database, a KDC ConfigurationChange event ([MS-LSAD] section 3.1.4.4.8) 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 LsarQueryDomainInformationPolicy method ([MS-LSAD] section 3.1.4.4.7). The *InformationClass* parameter SHOULD be set to the value of PolicyDomainKerberosTicketInformation in order to retrieve the current values. The KDC SHOULD set its configuration settings as follows:

- **MaxRenewAge** (section 3.3.1) to the value of the **MaxRenewAge** field.
- **MaxClockSkew** (section 3.3.1) to the value of the **MaxClockSkew** field.
- **MaxServiceTicketAge** (section 3.3.1) to the value of the **MaxServiceTicketAge** field.
- **MaxTicketAge** (section 3.3.1) to the value of the **MaxTicketAge** field.
- **AuthenticationOptions** (section 3.3.1) to the value of the **AuthenticationOptions** 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.

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 (or if domainControllerFunctionality returns a value ≥ 6 ([MS-ADTS] section 3.1.1.3.2.25) and the principal is a member of

PROTECTED_USERS ([MS-DTYP] section 2.4.2.4)), 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).

If **ClaimsCompIdFASTSupport** is set to:<41>

- 0: The KDC SHOULD respond as if it does not process FAST.
- 1, and a KDC_ERR_PREAUTH_REQUIRED is returned in the KRB_ERROR: The KDC SHOULD NOT return PA-FX-FAST in the KRB_ERROR.
- 1, 2, or 3 and an armored AS-REQ is received: The KDC SHOULD process per FAST ([RFC6113]).
- 1 or 2, and an unarmored AS-REQ is received: The KDC SHOULD continue without FAST.
- 3, and an AS-REQ is received: If the principal is a computer account, then the KDC SHOULD continue without FAST. Otherwise, the KDC SHOULD return KDC_ERR_PREAUTH_REQUIRED and return PA-FX-FAST ([RFC6113] section 5.4.2).<42>

3.3.5.1.1 Server Principal Lookup

This section is relevant only for KILE implementations that use Active Directory for the account database.

Note Some of the data types in the following procedures are defined in [RFC4120] section 5.2.

If the Name Type ([RFC4120] section 6.2) is NT-PRINCIPAL, NT-SRV-HST, or NT-SRV-INST, then the KDC SHOULD:

1. If the KerberosString[0] element of name-string of the PrincipalName is "krbtgt" and there are only two KerberosString elements in name-string, then call GetUserLogonInfoByAttribute ([MS-ADTS] section 3.1.1.13.6) where:

- SearchKey is set to KerberosString[1].
- Attribute is set to the **sAMAccountName** attribute ([MS-ADA3] section 2.222).

2. Otherwise:

1. Call GetUserLogonInfoByAttribute where:

- SearchKey is set to KerberosString[0] + "/" + the concatenation of the remaining KerberosString elements in order.
- Attribute is set to the **userPrincipalName** attribute ([MS-ADA3] section 2.349).

2. If STATUS NOT FOUND or STATUS NO SUCH USER is returned ([MS-ERREF] section 2.3.1) and there is only one KerberosString element in name-string, then:

1. Call GetUserLogonInfoByAttribute where:

- SearchKey is set to KerberosString[0].
- Attribute is set to **sAMAccountName**.

2. If STATUS NOT FOUND or STATUS NO SUCH USER is returned, then call GetUserLogonInfoByAttribute where:

- SearchKey is set to KerberosString[0] + "\$".

- Attribute is set to **sAMAccountName**.

3. If STATUS_NOT_FOUND or STATUS_NO_SUCH_USER is returned, then the KDC MUST return KDC_ERR_S_PRINCIPAL_UNKNOWN ([RFC4120] section 7.5.9).

If the Name Type ([RFC4120] section 6.2) is NT-ENTERPRISE, then the KDC SHOULD:

1. Set local variable *UPNServerName* to the contents of the **sname** field of the request before the @ character.
2. If there is only one KerberosString element in name-string, then call *GetUserLogonInfoByAttribute* where:
 - SearchKey is set to *KerberosString[0]*.
 - Attribute is set to the **servicePrincipalName** element.
3. If STATUS_NOT_FOUND or STATUS_NO_SUCH_USER is returned, then call *GetUserLogonInfoByAttribute* where:
 - SearchKey is set to *UPNServerName*.
 - Attribute is set to **sAMAccountName**.
4. If ERROR_SUCCESS is returned and the account has no SPNs registered, then the KDC MUST return KDC_ERR_S_PRINCIPAL_UNKNOWN.
5. Or if STATUS_NOT_FOUND or STATUS_NO_SUCH_USER is returned, then call *GetUserLogonInfoByAttribute* where:
 - SearchKey is set to *UPNServerName* + "\$".
 - Attribute is set to **sAMAccountName**.
6. If STATUS_NOT_FOUND or STATUS_NO_SUCH_USER is returned, then the KDC MUST return KDC_ERR_S_PRINCIPAL_UNKNOWN.

In all cases, if the call succeeds, the Active Directory account for the requested principal was found.

~~3.3.5.1~~ **3.3.5.1.2 Canonicalization of Server Principals**

For initial TGTs and referral TGTs, KILE KDCs SHOULD return the krbtgt/FQDN for the server principal.

If the canonicalize flag ([RFC4120] section 5.4.1) is set, KILE KDCs SHOULD canonicalize other server principals unless:

- The server principal is kadmin/changepw.
- The server principal's account has UseDESOnly set to TRUE.

3.3.5.2 User Account Objects Without UPN

If the user account object does not have the **userPrincipalName** attribute ([MS-ADA3] section 2.349) 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. <43>

3.3.5.3 PAC Generation

In either of the following two cases, a PAC [MS-PAC] MUST be generated and included in the response by the KDC when the client has requested that a PAC be included. The request to include a PAC is expressed through the use of a KERB-PA-PAC-REQUEST (section 2.2.3) padata type that is set to TRUE:

- During an Authentication Service (AS) request that has been validated with pre-authentication and for which the account has `AuthorizationDataNotRequired` set to FALSE.
- During a TGS request that results in a service ticket unless the NA bit is set in the `UserAccountControl` field in the `KERB_VALIDATION_INFO` structure ([MS-PAC] section 2.5).

Otherwise, the response will not contain a PAC.

Note Population of the PAC is covered in the corresponding KDC details sections.

3.3.5.4 Determining Authentication Policy Silo Membership

If `domainControllerFunctionality` returns a value < 6 ([MS-ADTS] section 3.1.1.3.2.25), the KDC SHOULD set `BelongsToSilo` to FALSE.<44>

Note The `BelongsToSilo` variable is a Boolean variable that is used for illustrative purposes in the processing rules of this section and section 3.3.5.5. The value of `BelongsToSilo` is not persisted across client requests.

If `domainControllerFunctionality` returns a value ≥ 6 , the KDC SHOULD check whether the account is a member of an Authentication Policy Silo:

- If the `AssignedSilo` (section 3.3.1.1) is NULL, the KDC SHOULD set `BelongsToSilo` to FALSE.
- If the `AssignedSilo` is not NULL and `AssignedSilo.msDS-AuthNPolicySiloMembers` does not contain the account, the KDC SHOULD set `BelongsToSilo` to FALSE.
- If the `AssignedSilo` is not NULL and `AssignedSilo.msDS-AuthNPolicySiloMembers` contains the account, the KDC SHOULD set `BelongsToSilo` to TRUE.

3.3.5.5 Determining Authentication Policy Settings

If `domainControllerFunctionality` returns a value < 6 ([MS-ADTS] section 3.1.1.3.2.25), the KDC SHOULD set `PolicyName` to NULL.<45>

If `domainControllerFunctionality` returns a value ≥ 6 , the KDC SHOULD check whether the account has an Authentication Policy:

- If `BelongsToSilo` == TRUE (section 3.3.5.4) for the account, the account belongs to a Silo. In this case, when the account is of type:
 - User ([MS-ADSC] section 2.268): the KDC SHOULD set:
 - `PolicyName` to `AssignedSilo.msDS-UserAuthNPolicy.RDN`.
 - `Enforced` to `AssignedSilo.msDS-AuthNPolicyEnforced`
 - `TGTLifetime` to `AssignedSilo.msDS-UserAuthNPolicy.msDS-UserTGTLifetime`
 - `AllowedToAuthenticateTo` to `AssignedSilo.msDS-UserAuthNPolicy.msDS-UserAllowedToAuthenticateTo`

- **AllowedToAuthenticateFrom** to AssignedSilo.msDS-UserAuthNPolicy.msDS-UserAllowedToAuthenticateFrom
- ManagedServiceAccount ([MS-ADSC] sections 2.139 and 2.141): the KDC SHOULD set:
 - **PolicyName** to AssignedSilo.msDS-ServiceAuthNPolicy.RDN.
 - **Enforced** to AssignedSilo.msDS-AuthNPolicyEnforced
 - **TGTLifetime** to AssignedSilo.msDS-ServiceAuthNPolicy.msDS-ServiceTGTLifetime
 - **AllowedToAuthenticateTo** to AssignedSilo.msDS-ServiceAuthNPolicy.msDS-ServiceAllowedToAuthenticateTo
 - **AllowedToAuthenticateFrom** to AssignedSilo.msDS-ServiceAuthNPolicy.msDS-ServiceAllowedToAuthenticateFrom
- Computer ([MS-ADSC] section 2.21): the KDC SHOULD set:
 - **PolicyName** to AssignedSilo.msDS-ComputerAuthNPolicy.RDN.
 - **Enforced** to AssignedSilo.msDS-AuthNPolicyEnforced
 - **TGTLifetime** to AssignedSilo.msDS-ComputerAuthNPolicy.msDS-ComputerTGTLifetime
 - **AllowedToAuthenticateTo** to AssignedSilo.msDS-ComputerAuthNPolicy.msDS-ComputerAllowedToAuthenticateTo
 - **AllowedToAuthenticateFrom** to NULL
- If the account does not belong to a Silo (**BelongsToSilo** == ~~TRUE~~FALSE (section 3.3.5.4)) and AssignedPolicy (section 3.3.1.1) is NULL, the KDC SHOULD set **PolicyName** to NULL and **Enforced** to FALSE.
- If the account does not belong to a Silo (**BelongsToSilo** == ~~TRUE~~FALSE (section 3.3.5.4)) and the AssignedPolicy is not NULL, the KDC SHOULD set **PolicyName** to AssignedPolicy.RDN, **Enforced** to AssignedPolicy.msDS-AuthNPolicyEnforced, and when the account is of type:
 - User: the KDC SHOULD set:
 - **TGTLifetime** to AssignedPolicy.msDS-UserAuthNPolicy.msDS-UserTGTLifetime
 - **AllowedToAuthenticateTo** to AssignedPolicy.msDS-UserAuthNPolicy.msDS-UserAllowedToAuthenticateTo
 - **AllowedToAuthenticateFrom** to AssignedPolicy.msDS-UserAuthNPolicy.msDS-UserAllowedToAuthenticateFrom
 - ManagedServiceAccount: the KDC SHOULD set:
 - **TGTLifetime** to AssignedPolicy.msDS-ServiceAuthNPolicy.msDS-ServiceTGTLifetime
 - **AllowedToAuthenticateTo** to AssignedPolicy.msDS-ServiceAuthNPolicy.msDS-ServiceAllowedToAuthenticateTo
 - **AllowedToAuthenticateFrom** to AssignedPolicy.msDS-ServiceAuthNPolicy.msDS-ServiceAllowedToAuthenticateFrom
 - Computer: the KDC SHOULD set:
 - **TGTLifetime** to AssignedPolicy.msDS-ComputerAuthNPolicy.msDS-ComputerTGTLifetime

- **AllowedToAuthenticateTo** to AssignedPolicy.msDS-ComputerAuthNPolicy.msDS-ComputerAllowedToAuthenticateTo
- **AllowedToAuthenticateFrom** to NULL

3.3.5.6 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.

If DES is used for pre-authentication, the KDC MUST: <46>

- If UseDESOnly is not set: the KDC MUST return KDC_ERR_ETYPE_NOTSUPP.
- Otherwise, if the account is:
 - krbtgt: the KDC MUST return KDC_ERR_ETYPE_NOTSUPP.
 - The computer account of a KDC: the KDC MUST return KDC_ERR_ETYPE_NOTSUPP.

The KDC SHOULD<47> return in the encrypted part of the AS-REP message a PA-DATA structure with padata-type set to PA-SUPPORTED-ENCTYPES (165), to indicate what encryption types are supported by the KDC, and whether Claims or FAST are supported.<48>

If domainControllerFunctionality returns a value ≥ 6 ([MS-ADTS] section 3.1.1.3.2.25), the KDC MUST check whether the account is a member of PROTECTED_USERS ([MS-DTYP] section 2.4.2.4). If it is a member of PROTECTED_USERS, then:<49>

- If pre-authentication used DES or RC4, the KDC MUST return KDC_ERR_ETYPE_NOTSUPP.
- MaxRenewAge (section 3.3.1) for the TGT is 4 hours unless specified by policy.
- MaxTicketAge (section 3.3.1) for the TGT is 4 hours unless specified by policy.

If domainControllerFunctionality returns a value ≥ 6 , the KDC MUST determine whether an Authentication Policy is applied to the account (section 3.3.5.5). If **Enforced** is TRUE, then:<50>

- If TGTLifetime is not 0: MaxRenewAge for the TGT is TGTLifetime.
- If TGTLifetime is not 0: MaxTicketAge for the TGT is TGTLifetime.
- If AllowedToAuthenticateFrom is not NULL, the PAC of the armor TGT MUST be used to perform an access check for the CTRL_DS_CONTROL_ACCESS right against the AllowedToAuthenticateFrom. If the access check fails, the KDC MUST return KDC_ERR_POLICY.

The KDC SHOULD check whether the domainControllerFunctionality ([MS-ADTS] section 3.1.1.3.2.25) returns a value:

- < 3 : the KDC SHOULD, in the encrypted pre-auth data part ([Referrals-11], Appendix A) of the AS-REP message, include a PA-DATA structure with padata-type set to PA-SUPPORTED-ENCTYPES (165), and padata-value set to 0x7 (section 2.2.7).
- ≥ 3 : the KDC SHOULD, in the encrypted pre-auth data part ([Referrals-11], Appendix A) of the AS-REP message, include a PA-DATA structure with padata-type set to PA-SUPPORTED-ENCTYPES (165), and padata-value set to 0x1F (section 2.2.7).

3.3.5.6.1 Client Principal Lookup

This section is relevant only for KILE implementations that use Active Directory for the account database.

If the Name Type ([RFC4120] Section 6.2) is NT-PRINCIPAL, then the KDC SHOULD:

1. If the **realm** field is not present in the request or is the DC's domain name, call GetUserLogonInfoByAttribute ([MS-ADTS] section 3.1.1.13.6) where:
 - SearchKey is set to the **cname** field of the request.
 - Attribute is set to the **sAMAccountName** attribute ([MS-ADA3] section 2.222).
2. If STATUS_NOT_FOUND or STATUS_NO_SUCH_USER is returned ([MS-ERREF] section 2.3.1), then if **realm** is not present or is the DC's domain name, call GetUserLogonInfoByAttribute where:
 - SearchKey is set to **cname** + "\$".
 - Attribute is set to **sAMAccountName**.
3. If STATUS_NOT_FOUND or STATUS_NO_SUCH_USER is returned, then call GetUserLogonInfoByUPNOrAccountName ([MS-ADTS] section 3.1.1.13.7) where UPNOrName is set to:
 - If **realm** is present, cname@realm.
 - Otherwise, cname@DC's domain name.
4. If STATUS_NOT_FOUND or STATUS_NO_SUCH_USER is returned and:
 - If no preauthentication data was provided, then call IDL_DRSCrackNames ([MS-DRSR] section 4.1.4) where:
 - **pmsgIn.dwFlags** is set to GC and TR.
 - **pmsgIn.formatOffered** is set to DS_USER_PRINCIPAL_NAME_AND_ALTSECID.
 - **pmsgIn.cNames** is set to 1.
 - **pmsgIn rpNames** is set to:
 - If **realm** is present, cname@realm.
 - Otherwise, cname@DC's domain name.
 - If preauthentication data was provided, then call IDL_DRSCrackNames where:
 - **pmsgIn.dwFlags** is set to GC and TR.
 - **pmsgIn.formatOffered** is set to DS_USER_PRINCIPAL_NAME.
 - **pmsgIn.cNames** is set to 1.
 - **pmsgIn rpNames** is set to:
 - If **realm** is present, cname@realm.
 - Otherwise, cname@DC's domain name.
5. If DS_NAME_ERROR_NOT_FOUND is returned ([MS-DRSR] section 4.1.4.1.8), then the KDC MUST return KDC_ERR_C_PRINCIPAL_UNKNOWN ([RFC4120] section 7.5.9).

If the Name Type is NT-ENTERPRISE, then the KDC SHOULD:

1. Set local variable *UPNClientName* to the contents of **cname** before the @ character.
2. Set local variable *UPNDomainName* to the contents of **cname** after the @ character.
3. Call *GetUserLogonInfoByUPNOrAccountName* where *UPNOrName* is set to **cname**.
4. If STATUS_NOT_FOUND or STATUS_NO_SUCH_USER is returned and *UPNDomainName* is the same as the DC's domain name, then call *GetUserLogonInfoByAttribute* where:
 - *SearchKey* is set to *UPNClientName*.
 - *Attribute* is set to **sAMAccountName**.
5. If STATUS_NOT_FOUND or STATUS_NO_SUCH_USER is returned and *UPNDomainName* is the same as the DC's domain name, then call *GetUserLogonInfoByAttribute* where:
 - *SearchKey* is set to *UPNClientName* + "\$".
 - *Attribute* is set to **sAMAccountName**.
6. If STATUS_NOT_FOUND or STATUS_NO_SUCH_USER is returned and:
 - If no preauthentication data was provided, then call *IDL_DRSCrackNames* where:
 - **pmsgIn.dwFlags** is set to GC and TR.
 - **pmsgIn.formatOffered** is set to DS_USER_PRINCIPAL_NAME_AND_ALTSECID.
 - **pmsgIn.cNames** is set to 1.
 - **pmsgIn.rpNames** is set to **cname**.
 - If preauthentication data was provided, then call *IDL_DRSCrackNames* where:
 - **pmsgIn.dwFlags** is set to GC and TR.
 - **pmsgIn.formatOffered** is set to DS_USER_PRINCIPAL_NAME.
 - **pmsgIn.cNames** is set to 1.
 - **pmsgIn.rpNames** is set to **cname**.
7. If STATUS_NOT_FOUND or STATUS_NO_SUCH_USER is returned, then the KDC MUST return KDC_ERR_C_PRINCIPAL_UNKNOWN.

In all cases, if the call succeeds, the Active Directory account for the requested principal was found.

3-3-5-6-13.3.5.6.2 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>

3-3-5-6-23.3.5.6.3 Check Account Policy for Every TGT 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 on the KDCs and the KDC MUST verify that the account and return the following errors:

- 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 the current time is not within **LogonHours**, then the KDC MUST return `KDC_ERR_CLIENT_REVOKED`.
- If **PasswordMustChange** is in the past, then the KDC MUST return `KDC_ERR_KEY_EXPIRED`.
- If **PasswordMustChange** is zero, then the KDC MUST return `KDC_ERR_KEY_EXPIRED`.
- If the KILE implementation uses an Active Directory for the account database and the **userAccountControl** attribute ([MS-ADTS] section 2.2.16) SR flag is set to TRUE, because this is a password-based logon the KDC MUST return `STATUS_SMARTCARD_LOGON_REQUIRED`.

~~3.3.5.6.33.3.5.6.4~~ **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). The KDC MUST ensure that the PAC structure specified in [MS-PAC] does not end with a zero-length buffer.

~~3.3.5.6.3-13.3.5.6.4.1~~ **KERB_VALIDATION_INFO Structure**

~~Note: Some of the information in this section is subject to change because it applies to a preliminary product version, and thus may differ from the final version of the software when released. All behavior notes that pertain to the preliminary product version contain specific references to it in the Product Behavior appendix.~~

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 computed and set as follows:
 1. Convert the local machine time into an offset from the beginning of the week (as defined in [MS-SAMR] section 2.2.7.5). This conversion must use the same granularity as the **UnitsPerWeek** field of the `Buffer.SAMPR_USER_ALL_INFORMATION.LogonHours` of the **SamrQueryInformationUser2** ([MS-SAMR] section 3.1.5.5.5) response message.
 2. Starting at the offset determined in step 1, examine the remaining entries in the `Buffer.SAMPR_USER_ALL_INFORMATION.LogonHours`. If the value at the initial offset is disabled for authentication, the KDC MUST return Kerb Error `KDC_ERROR_CLIENT_REVOKED` with status code `STATUS_INVALID_LOGON_HOURS`. If none of the remaining entries are

disabled, use the time stamp value 0x7FFFFFFFFFFFFFFF. Otherwise, compute a time stamp by adding the offset of the next disabled authentication unit to the current time.

3. Set the **LogoffTime** field to the lesser of the value determined in step 2 and the value of the **Buffer.SAMPR_USER_ALL_INFORMATION.AccountExpires** field 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 MUST retrieve the following attributes from the local directory service instance using the processing rules defined in the **GetUserLogonInfo()** procedure ([MS-ADTS] section 3.1.1.13.3). The KDC populates the returned KERB_VALIDATION_INFO structure ([MS-PAC] section 2.5) as follows: <51>:

- The **GroupCount** field SHOULD be set to the count of SIDs returned in the *ExpandedSids* parameter of the **GetUserLogonInfo()** procedure.
- The **GroupIds** field SHOULD be set to the set of SIDs returned in the *ExpandedSids* parameter of the **GetUserLogonInfo()** procedure.

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

- 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 SHOULD contain the number of SIDs in the **ExtraSids** field. The **ExtraSids** field SHOULD contain the AUTHENTICATION_AUTHORITY_ASSERTED_IDENTITY SID ([MS-DTYP] section 2.4.2.4), and the D bit SHOULD be set in the **UserFlags** field. <5251>
- The **ResourceGroupDomainSid** field MUST be set to NULL.
- The **ResourceGroupCount** field SHOULD contain the number of SIDs in the **ResourceGroupIds** field.
- The **ResourceGroupIds** field MUST be set to NULL.

3.3.5.6.3.23.3.5.6.4.2 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.

3.3.5.6.3.33.3.5.6.4.3 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<5352> 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.

~~3.3.5.6.3.43~~ 3.3.5.6.4.4 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.

~~3.3.5.6.3.53~~ 3.3.5.6.4.5 UPN_DNS_INFO Structure

The KDC populates the returned UPN_DNS_INFO structure ([MS-PAC] section 2.10) fields<5453> 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 2.349) 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.

~~3.3.5.6.3.63~~ 3.3.5.6.4.6 PAC_CLIENT_CLAIMS_INFO Structure

If **ClaimsCompIdFASTSupport** is set to:

- 0: The KDC SHOULD NOT insert into the returned PAC a PAC_CLIENT_CLAIMS_INFO structure ([MS-PAC] section 2.11).
- 1: If a PA-PAC-OPTIONS [167] (section 2.2.10) padata type with the Claims bit set is in the AS-REQ, the KDC SHOULD behave as noted in the next step, "2 or 3". Otherwise, the KDC SHOULD NOT provide a PAC_CLIENT_CLAIMS_INFO structure ([MS-PAC] section 2.11).
- 2 or 3: The KDC SHOULD<5554>

- Add the CLAIMS_VALID_SID ([MS-DTYP] section 2.4.2.4) to **KERB_VALIDATION_INFO.ExtraSids**.
- Increment **SidCount**.
- Add a **PAC_CLIENT_CLAIMS_INFO** structure as follows:

For KILE implementations that use an Active Directory for the account database, KDCs SHOULD retrieve the claims from the local directory service instance with the same processing rules as defined in **GetClaimsForPrincipal()** ([MS-ADTS] section 3.1.1.11.2.1) message processing. The KDC populates the returned **PAC_CLIENT_CLAIMS_INFO** structure fields as follows:

- The **Claims** field SHOULD be set to the ClaimsBlob.

3.3.5.7 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 TGT received is encrypted with DES and not a referral TGT from a realm that only supports DES, then the KDC MUST return KDC_ERR_ETYPE_NOTSUPP.<5655>

If the server or service has a **KerbSupportedEncryptionTypes** populated with supported encryption types,<5756> then the KDC SHOULD<5857> return in the encrypted part ([Referrals-11] Appendix A) of TGS-REP message, a PA-DATA structure 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<5958> 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 a PA-DATA structure with padata-type set to PA-SUPPORTED-ENCTYPES (165), and padata-value set to 0x3 (section 2.2.7).
- 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 a PA-DATA structure with padata-type set to PA-SUPPORTED-ENCTYPES (165), and padata-value set to 0x7 (section 2.2.7).
 - If the account is krbtgt, and domainControllerFunctionality returns greater than or equal to 3: the KDC SHOULD, in the encrypted pre-auth data part ([Referrals-11], Appendix A) of the TGS-REP message, include a PA-DATA structure with padata-type set to PA-SUPPORTED-ENCTYPES (165), padata-value set to 0x1F (section 2.2.7), the Claims-supported bit if claims is supported, and the FAST-supported bit if FAST is supported.<6059>
 - DES MUST NOT be used to protect the service ticket. If DES is the only configured etype, the KDC MUST return KDC_ERR_ETYPE_NOTSUPP.<6160>

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 Application Server's service account does not have a registered SPN, the KDC MUST return KDC_ERR_MUST_USE_USER2USER.

If the OTHER_ORGANIZATION_SID ([MS-DTYP] section 2.4.2.4) is in **KERB_VALIDATION_INFO.ExtraSids**, the PAC MUST be used to perform an access check for the Allowed-To-Authenticate right ([MS-ADTS] section 6.1.1.2.7.41) 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.

If domainControllerFunctionality returns a value ≥ 6 ([MS-ADTS] section 3.1.1.3.2.25) and the account is not also the application service account, the KDC MUST determine whether an Authentication Policy is applied to the server or service (section 3.3.5.5); if Enforced is TRUE then: <6261>

- If AllowedToAuthenticateTo is not NULL, the PAC of the user and the PAC of the armor TGT MUST be used to perform an access check for the ACTRL_DS_CONTROL_ACCESS right against the AllowedToAuthenticateTo. If the access check fails, the KDC MUST return KDC_ERR_POLICY.

If there are no claims in the PAC and the PA-PAC-OPTIONS [167] (section 2.2.10) padata type does not have the Claims bit set, then the KDC SHOULD NOT call the TransformClaimsOnTrustTraversal procedure ([MS-ADTS] section 3.1.1.11.2.11). Otherwise the KDC SHOULD call this procedure. <6362>

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 2.2.4) as an AD-IF-RELEVANT to the end of authorization data in the service ticket. <6463>

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 3.3.5.6.34.3) and KDC Signature (section 3.3.5.6.34.4) which replace the existing signature fields in the PAC. The KDC MUST ensure that the PAC structure specified in [MS-PAC] does not end with a zero-length buffer.

3.3.5.7.1 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 on the account KDCs. When this property is set on the account KDC for the client's domain, and the TGT is older than an implementation-specific time <6564>, the account 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 the KDC receiving the **session** ticket request is not in the user account's domain, then the check cannot be made.

- 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.

3.3.5.7.2 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.

3.3.5.7.3 Domain Local Group Membership

~~Note: Some of the information in this section is subject to change because it applies to a preliminary product version, and thus may differ from the final version of the software when released. All behavior notes that pertain to the preliminary product version contain specific references to it in the Product Behavior appendix.~~

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 the **GetResourceDomainInfo()** procedure ([MS-ADTS] section 3.1.1.13.4) where ~~<66>~~:

- *InputSids* is an array of SIDs that identify the user and also the groups, contained in *GroupIds* ([MS-PAC] section 2.5), that the user is a member of.

Note that each SID is calculated as follows:

- For the user, the SID contains the SID of the user created by concatenating **LogonDomainId** ([MS-PAC] section 2.5) and **UserId** ([MS-PAC] section 2.5).
- For each account domain group, the SID contains the SID of the group created by concatenating **LogonDomainId** ([MS-PAC] section 2.5) and **GroupIds.RelativeID** ([MS-PAC] section 2.2.2).
- For each group in other domains, the SID contains **ExtraSids.Sid** ([MS-PAC] section 2.2.2).

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 that are returned by the **GetResourceDomainInfo()** procedure ([MS-ADTS] section 3.1.1.13.4) to the existing fields as follows:

- If the `Resource-SID-compression-disabled` bit is NOT set in the Application Server's service account's `KerbSupportedEncryptionTypes` and NOT set in the `krbtgt`'s account's `KerbSupportedEncryptionTypes`: ~~<6765>~~
 - The **ResourceGroupDomainSid** field contains the SID for the domain.
 - The **ResourceGroupCount** field contains the number of groups in the **ResourceGroupIds** field.
 - The **ResourceGroupIds** 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:
 - **RelativeId** ([MS-PAC] section 2.2.2) contains the RID of the value from the *ResourceSids* parameter ([MS-ADTS] section 3.1.1.13.4).
 - **Attributes** ([MS-PAC] section 2.2.2) has the A, B, C, and E bits set to 1, and all other bits set to zero.

- Otherwise:
 - 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 from the *ResourceSids* parameter ([MS-ADTS] section 3.1.1.13.4).
 - **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.

3.3.5.7.4 Compound Identity

If a compound identity TGS-REQ (FAST TGS-REQ explicitly armored with the computer's ticket-granting ticket (TGT)) is received and a Compound-Identity-supported bit is set in the application server's service account's *KerbSupportedEncryptionTypes*, the KDC SHOULD add to the privilege attribute certificate (PAC) a *PAC_DEVICE_INFO* structure ([MS-PAC] section 2.12) and *PAC_DEVICE_CLAIMS_INFO* structure ([MS-PAC] section 2.13) with the group membership and claims for the computer. <6866>

The armor key for an explicitly armored TGT is generated as follows:

```
explicit_armor_key = KRB-FX-CF2(armor_subkey, ticket_session_key, "subkeyarmor",
"ticketarmor" )
```

The *armor_subkey* is the *ap-req* subkey in the armor ticket. Then the explicit armor key is used to create the armor key, which is used per [RFC6113].

```
armor_key = KRB-FX-CF2( explicit_armor_key, subkey, " explicitarmor", " tgsarmor" )
```

The KDC SHOULD add the *COMPOUNDED_AUTHENTICATION_SID* ([MS-DTYP] section 2.4.2.4) to **KERB_VALIDATION_INFO.ExtraSids** and increment **SidCount**.

The KDC SHOULD populate the following *PAC_DEVICE_INFO* structure ([MS-PAC] section 2.12) fields by using the following fields from the *KERB_VALIDATION_INFO* structure from the computer's TGT:

- **UserID**: from the **UserID** field
- **PrimaryGroupId**: from the **PrimaryGroupId** field
- **AccountDomainId**: from the **LogonDomainId** field
- **AccountGroupCount**: from the **GroupCount** field
- **AccountGroupIds**: from the **GroupIds** field.

The non-account domain fields MUST be initialized as follows:

- **SidCount** field set to zero
- **ExtraSids** field is NULL
- **DomainCount** field set to zero
- **DomainGroup** field is NULL

The KDC MUST call IDL_DRSGetMemberships ([MS-DRSR] section 4.1.8) to obtain the Domain Local Group Membership as described in section 3.3.5.7.3 using the computer TGT. If **ExtraSids.Sid** in the Domain Local Group Membership (section 3.3.5.7.3) is the only SID from a domain, then **ExtraSids** SHOULD be used:

- Add one to the **SidCount** field.
- The **ExtraSids** field SHOULD be populated with the value of the **ExtraSids** field in the Domain Local Group Membership (section 3.3.5.7.3), using the computer principal.

For the rest of the **ExtraSids.Sid**, **DomainGroup** SHOULD be used:

- The **DomainCount** field contains the number of domains with **DomainGroup** populated.
- The **DomainGroup** field SHOULD be populated for each domain where:
 - The **DomainId** field contains the SID for the domain.
 - The **GroupCount** field contains the number of groups in **GroupIds** field.
 - For each **ExtraSids.Sid** in the DomainId domain, the **GroupIds** field SHOULD be populated with the value of the **ResourceGroupIds** field in the Domain Local Group Membership (section 3.3.5.7.3) using the computer principal.

The KDC SHOULD populate the following **PAC_DEVICE_CLAIMS_INFO** structure ([MS-PAC] section 2.13) fields using the following fields from the **PAC_CLIENT_CLAIMS_INFO** structure from the computer's TGT:

- **Claims: Claims** field.

3.3.5.7.5 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]. DES MUST NOT be used unless no other etype is supported. <6967>

If the TRUST_ATTRIBUTE_CROSS_ORGANIZATION flag is set in the TrustAttributes field ([MS-ADTS] section 6.1.6.7.9), the OTHER_ORGANIZATION SID ([MS-DTYP] section 2.4.2.4) MUST be added to **KERB_VALIDATION_INFO.ExtraSids** and the **SidCount** field MUST be incremented in 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 Active Directory account object,
- the client principal MUST be that of the client user,
- and the requested access MUST be CTRL_DS_CONTROL_ACCESS.

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

If the TRUST_ATTRIBUTE_CROSS_ORGANIZATION_NO_TGT_DELEGATION flag is set in the **trustAttributes** field ([MS-ADTS] section 6.1.6.7.9), the KDC MUST return a ticket with the ok-as-delegate flag not set in TicketFlags.

3.3.5.7.6 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. <7068> See section 3.1.5.2 for the relative strengths of KILE-supported encryption types.

3.3.5.7.7 Read-only Domain Controller (RODC)

When a Key Distribution Center (KDC) which is a read-only domain controller (RODC) receives:

- An AS-REQ message with a PA-PAC-OPTIONS [167] (section 2.2.10) padata type with the forward to full DC bit set, the RODC SHOULD forward the AS-REQ to a full DC.
- A TGS-REQ message with a PA-PAC-OPTIONS [167] (section 2.2.10) padata type with the Branch Aware bit set, and the application server (SNAME) is not in its database, the RODC SHOULD return server principal unknown with the substatus message of NTSTATUS STATUS_NO_SECRETS ([MS-ERREF] section 2.3.1).

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. <7169>

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

- **ImpersonationAccessToken** (Public): A Token/Authorization Context (see [MS-DTYP] section 2.5.2).

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 2.458463) 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 3.4.5.2.9) with the *Level* parameter set to 1 and

WkstaBuffer.WorkstationInfo.KerberosSupportedEncryptionTypes set to zero.<7270>If the **WkstaBuffer.WorkstationInfo.KerberosSupportedEncryptionTypes** returned is not equal to SupportedEncryptionTypes (section 3.1.1.5), then LDAP is used to update the setting:<7371>

1. Establish an LDAP connection with server information set to NULL ([MS-ADTS] section 7.1).
2. Perform an LDAP modify operation to set the msDS-SupportedEncryptionTypes attribute ([MS-ADA2] section 2.458463) 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 2.2.2) in the e-data field of the KRB-ERROR message ([RFC4120] section 5.9.1).

When the checksum field is not present, the application server SHOULD process the requests as though none of the flags ([RFC4121] section 4.1.1.1) are set and SHOULD NOT check channel binding information ([RFC4121] section 4.1.1.2) as it is likewise not present.

When the server receives AP [exchange](#) requests for SPNs with the serviceclass string 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. <7472>

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 [exchange](#) 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. <7573>

If the service ticket received for the computer's principal is encrypted with DES, the KILE server MUST return KRB_AP_ERR_MODIFIED regardless of supporting DES. <7674>

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 [exchange](#) 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 [exchange](#) 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 [exchange](#) 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 2.2.6) 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. <7775>
- Otherwise, the server MUST ignore the KERB_AUTH_DATA_TOKEN_RESTRICTIONS [141] Authorization Data Type, the KERB-AD-RESTRICTION-ENTRY structure (section 2.2.6), the KERB-LOCAL (142), and the containing KERB-LOCAL structure (section 2.2.4). <7876>

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.
- If a **PAC_CLIENT_CLAIMS_INFO** structure ([MS-PAC] section 2.11) and CLAIMS_VALID SID ([MS-DTYP] section 2.4.2.4) are in **KERB_VALIDATION_INFO.ExtraSids**, then the server SHOULD set the **ImpersonationAccessToken.UserClaims** field to the value of the **Claims** field. <7977>
- If a **PAC_DEVICE_INFO** structure ([MS-PAC] section 2.12) and COMPOUNDED_AUTHENTICATION SID ([MS-DTYP] section 2.4.2.4) are in **KERB_VALIDATION_INFO.ExtraSids**, then the server SHOULD populate the User SID and

Security Group SIDs in the **ImpersonationAccessToken.DeviceSids** array (section 3.4.1) as follows: <8078>

- Concatenate the **AccountDomainId** ([MS-PAC] section 2.12) and **PrimaryGroupId** ([MS-PAC] section 2.12) fields, add the result to the **ImpersonationAccessToken.DeviceSids** array, and set the **ImpersonationAccessToken.DevicePrimaryGroup** field to the index of the newly added SID.
- For each **AccountGroupIds** ([MS-PAC] section 2.5), concatenate **AccountDomainId** ([MS-PAC] section 2.12) and **AccountGroupIds.DevieRelativeID** ([MS-PAC] section 2.2.2) and add to the **ImpersonationAccessToken.DeviceSids** array.
- For each **ExtraSids** ([MS-PAC] section 2.5), add the **ExtraSids.Sid** ([MS-PAC] section 2.5) to the **ImpersonationAccessToken.DeviceSids** array.
- For each **DomainGroup**: for each **DomainGroup.DomainId** ([MS-PAC] section 2.2.3), concatenate **DomainGroup.DomainId** ([MS-PAC] section 2.2.3) and **DomainGroup.GroupIds.RelativeID** ([MS-PAC] section 2.2.2) and add to the **ImpersonationAccessToken.DeviceSids** array.
- If **CLAIMS_VALID SID** ([MS-DTYP] section 2.4.2.4) is in **PAC_DEVICE_INFO.ExtraSids** and **COMPOUNDED_AUTHENTICATION SID** ([MS-DTYP] section 2.4.2.4) is in **KERB_VALIDATION_INFO.ExtraSids**, then the server SHOULD set **ImpersonationAccessToken.DeviceClaims** to **Claims**. <8179>

The server SHOULD call **GatherGroupMembershipForSystem** ([MS-DTYP] section 2.5.2.1.1) where **InitialMembership** contains the **ImpersonationAccessToken.Sids** array and set **ImpersonationAccessToken.Sids** array to **FinalMembership**.

The server SHOULD call **AddPrivilegesToToken** ([MS-DTYP] section 2.5.2.1.2) where **Token** contains **ImpersonationAccessToken**.

Other SIDs **maycan** be added to the **ImpersonationAccessToken** following authentication (see [MS-DTYP] section 2.7.1).

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 [RFC1964] section 1.2.2.1 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 [RFC1964] section 1.1 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 [RFC4757] section 7.3 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

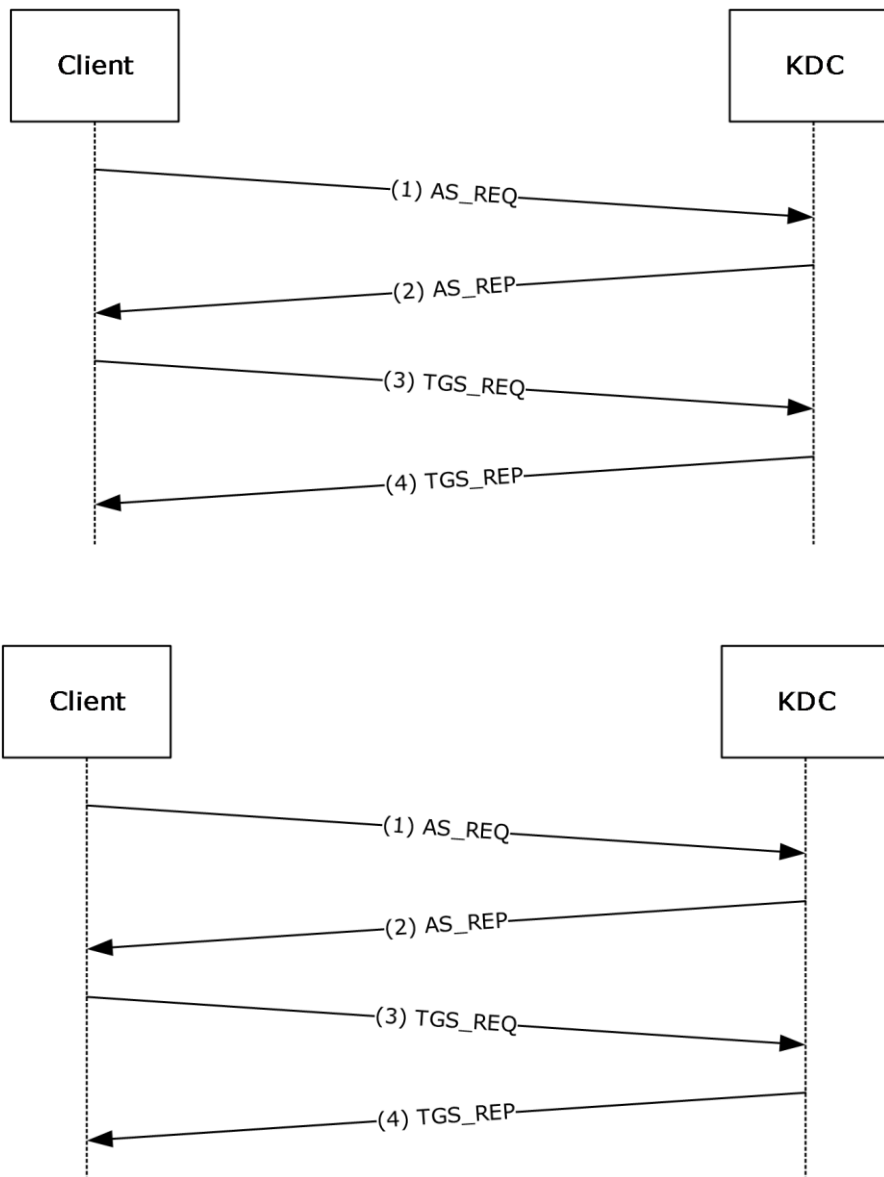


Figure 2: Interactive logon that uses passwords

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 3.3.5.6). 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 3.3.5.7). The service ticket is then interpreted by the Kerberos runtime within the local workstation.

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

- **LogonTime**
- **LogoffTime**
- **KickOffTime**
- **PasswordLastSet**
- **PasswordCanChange**
- **EffectiveName**
- **FullName**
- **LogonScript**
- **ProfilePath**
- **HomeDirectory**
- **HomeDirectoryDrive**
- **LogonCount**
- **BadPasswordCount**
- **LogonServer**
- **LogonDomainName**
- **UserAccountControl**

4.2 Network Logon

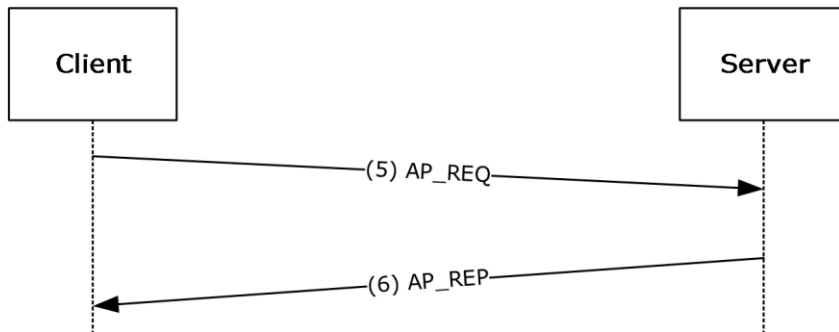


Figure 3: Network Logon

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 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.

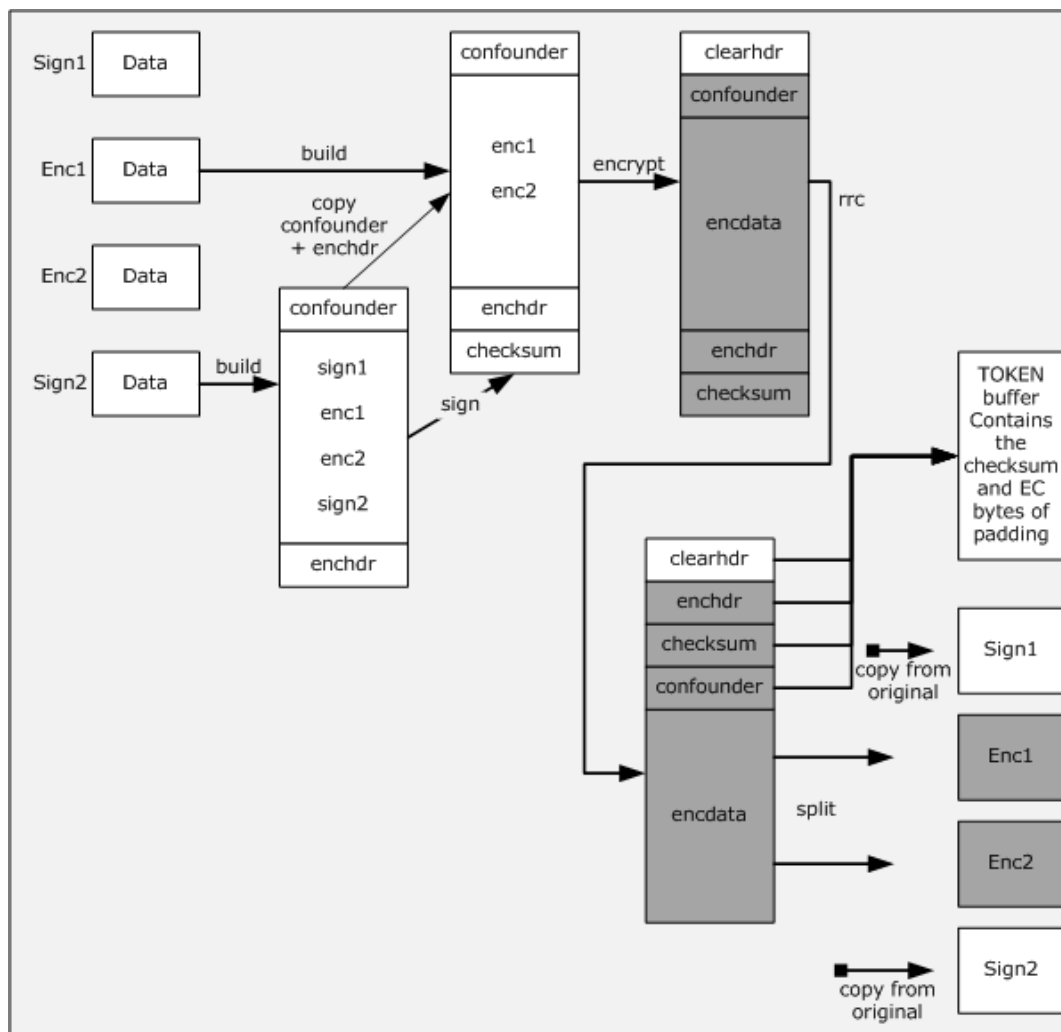


Figure 4: Example of RRC with output message with 4 buffers

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 [RFC4121] section 4.2.4.

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 .....
0000010: 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 .....
0000030: 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 .....
0000050: 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 .....
0000070: 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 .....
0000090: 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 .....
00000b0: 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 .....
00000d0: 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 .....

```

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  üóτÉ»ΔττòTíP†·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-η  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  üóτÉ»ΔττòTíP†·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          \T4L
```

Data:

```
0000000: 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff  ·"3DUfwêÖ~H e
```

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  ÄT?·L8·3[rTô||B÷`
```

Signature:

```
0000000: 60 3b 06 09 2a 86 48 86 f7 12 01 02 02 02 01 11  \;··*âHâ≈·····
0000010: 00 10 00 ff ff e2 9e 8b bc 63 48 e7 40 eb aa 61  ··· TÊi↓CHT@δ-a
0000020: 92 44 a1 56 a1 3b 5c f6 5e 3c 21 b9 aa          EDíVî;\÷^<!||~
```

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 3.3.5.7.1) 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 Active Directory 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.

5.1.5 DES Downgrade Protection

Since KILE has the ability to configure a principal as supporting only DES, and unarmored AS exchanges are vulnerable to downgrade attacks, the KDC can protect against DES downgrade attacks by not supporting DES for principals that are not DES-only. Since all KILE KDCs support at least RC4, RC4 can always be used for KDCs and their hosts. Additionally, all KILE hosts support at least RC4, so RC4 can always be used for service tickets to hosts. Thus, DES usage is required only for trusts to non-KILE realms and services using non-KILE servers that do not support RC4 or AES.

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.

~~Note: Some of the information in this section is subject to change because it applies to a preliminary product version, and thus may differ from the final version of the software when released. All behavior notes that pertain to the preliminary product version contain specific references to it as an aid to the reader.~~

- 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
- Windows 8 operating system
- Windows Server 2012 operating system
- Windows 8.1 operating system
- Windows Server 2012 R2 operating system
- Windows 10 operating system
- Windows Server 2016 ~~Technical Preview~~ 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 operating system Service Pack 3 (SP3)	2000 bytes
Windows 2000 operating system Service Pack 4 (SP4)	1465 bytes
Windows XP (initial release), Windows XP operating system Service Pack 1 (SP1)	2000

Windows version	Message size
	bytes
Windows XP operating system Service Pack 2 (SP2)	1500 bytes
Windows Server 2003 (initial release), Windows XP 64-Bit Edition operating system, Windows Server 2003 operating system with Service Pack 1 (SP1), Windows Server 2003 R2 operating system, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview .	1465 bytes

Note Windows NT operating system does not include a Kerberos implementation.

<3> Section 2.2.4: Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 do not support transmitting KERB-LOCAL.

<4> Section 2.2.5: The LSAP_TOKEN_INFO_INTEGRITY structure is not supported in Windows 2000, Windows XP, Windows Server 2003, or Windows Vista.

<5> Section 2.2.6: The KERB-AD-RESTRICTION-ENTRY structure is not supported in Windows 2000, Windows XP, Windows Server 2003, or Windows Vista.

<6> Section 2.2.7: The FAST-supported bit is not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7 or Windows Server 2008 R2.

<7> Section 2.2.7: The Compound-identity-supported bit is not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, or Windows Server 2008 R2.

<8> Section 2.2.7: The Claims-supported bit is not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, or Windows Server 2008 R2.

<9> Section 2.2.7: The Resource-SID-compression-disabled bit is not supported in Windows 2000, Windows Server 2003, Windows Server 2008, or Windows Server 2008 R2 KDCs.

<10> Section 2.2.8: PA-SUPPORTED-ENCTYPES are not supported by Windows 2000, Windows XP, or Windows Server 2003.

<11> Section 2.2.10: PA-PAC-OPTIONS is not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7 or Windows Server 2008 R2.

<12> 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 2000 and Windows XP, TGTs are not automatically renewed. Where supported, renewal attempts begin at 15 minutes prior to expiration (except for Windows Server 2003 which is 10 minutes), unless the renew-till time (see [RFC4120] section 2.3) of the TGT is within five minutes.

<13> Section 3.1.1.4: In Windows 2000, Windows XP, Windows Server 2003, and Windows Vista, a 32-byte binary random string machine ID is not sent on the wire. When sent, this machine ID is not used by KILE.

<14> Section 3.1.1.5: SupportedEncryptionTypes are not supported in Windows 2000, Windows XP, and Windows Server 2003.

- <15> 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.
- <16> Section 3.1.5.1: PA-SUPPORTED-ENCTYPES are not supported by Windows 2000, Windows XP, or Windows Server 2003.
- <17> Section 3.1.5.1: PA-PAC-OPTIONS is not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, or Windows Server 2008 R2.
- <18> Section 3.1.5.2: Not supported in Windows 2000, Windows XP, or Windows Server 2003.
- <19> 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. Otherwise, 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.
- <20> Section 3.1.5.2: Supported in Windows.
- <21> Section 3.1.5.2: Supported in Windows.
- <22> Section 3.1.5.2: Supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008. Windows 7 and subsequent versions of Windows, according to the applicability list at the beginning of this section, do not support DES by default.
- <23> Section 3.1.5.2: Supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008. Windows 7 and subsequent versions of Windows, according to the applicability list at the beginning of this section, do not support DES by default.
- <24> 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 and subsequent versions of Windows, according to the applicability list at the beginning of this section, 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.
- <25> Section 3.1.5.6: IPv6 addresses are not supported in Windows 2000, Windows XP and Windows Server 2003.
- <26> Section 3.1.5.8: RODCs are not supported in Windows 2000 and Windows Server 2003.
- <27> Section 3.1.5.11: Windows 7, Windows Server 2008 R2, Windows 8, and Windows Server 2012 support "RestrictedKrbHost/<hostname>" to allow developer frameworks to enable Kerberos authentication for code written prior to SPN support.
- <28> Section 3.2.5.4: PA-PAC-OPTIONS is not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7 or Windows Server 2008 R2.
- <29> Section 3.2.5.5: Claims is not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7 or Windows Server 2008 R2.
- <30> Section 3.2.5.5: FAST is not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7 or Windows Server 2008 R2.
- <31> Section 3.2.5.6: Not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008.
- <32> Section 3.2.5.7: Windows 2000, Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008 do not support KERB-LOCAL.
- <33> Section 3.2.5.7: The following versions of Windows do not support Claims: Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2.

<34> Section 3.2.5.7: FAST is not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, or Windows Server 2008 R2.

<35> Section 3.2.5.7: Compound Identity and FAST are not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, or Windows Server 2008 R2.

<36> Section 3.2.5.8: No version of Windows uses this field. Windows Vista operating system with Service Pack 1 (SP1), Windows 7, Windows Server 2008, and Windows Server 2008 R2 do not send this field on the wire in anticipation of possible future use; all other versions of Windows do.

<37> 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.

<38> Section 3.3.1: Claims, compound identity, FAST, and mixed mode are not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7 or Windows Server 2008 R2.

<39> Section 3.3.1.1: KerbSupportedEncryptionTypes are not supported in Windows NT, Windows 2000, Windows XP, and Windows Server 2003. Compound identity is not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2.

<40> Section 3.3.3: Claims and FAST are not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, or Windows Server 2008 R2.

<41> Section 3.3.5.1: Claims is not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7 or Windows Server 2008 R2.

<42> Section 3.3.5.1: For Active Directory with the **msDS-Behavior-Version** attribute on a domain NC root object equal to DS_BEHAVIOR_WIN2000, DS_BEHAVIOR_WIN2003_WITH_MIXED_DOMAINS, DS_BEHAVIOR_WIN2003, DS_BEHAVIOR_WIN2008, or DS_BEHAVIOR_WIN2008R2, KDCs continue without FAST.

<43> Section 3.3.5.2: Windows 2000 and Windows Server 2003 KDCs do not support the provisioning of UPNs.

<44> Section 3.3.5.4: Authentication Policy Silos are not supported by Windows 2000, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, or Windows Server 2012 DCs.

<45> Section 3.3.5.5: Authentication Policies are not supported by Windows 2000, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, or Windows Server 2012 DCs.

<46> Section 3.3.5.6: DES downgrade protection is not supported in Windows 2000, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, or Windows Server 2012 KDCs.

<47> Section 3.3.5.6: Not supported in Windows 2000 and Windows Server 2003.

<48> Section 3.3.5.6: Claims and FAST are not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7 or Windows Server 2008 R2.

<49> Section 3.3.5.6: PROTECTED_USERS is not supported in Windows 2000, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, or Windows Server 2012 KDCs.

<50> Section 3.3.5.6: Authentication Policies are not supported by Windows 2000, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, or Windows Server 2012 KDCs.

~~<51> Section 3.3.5.6.3.1: The way in which this processing is represented in this specification was updated for the unreleased preliminary version of Windows Server 2016 Technical Preview.~~

~~<52> Section 3.3.5.6.3~~<51> Section 3.3.5.6.4.1: In Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2, the **ExtraSids** field is NULL and the **UserFlags** field is zero.

~~<5352> Section 3.3.5.6.34.3~~: Active Directory with the **msDS-Behavior-Version** attribute on a domain NC root object equal to DS_BEHAVIOR_WIN2000, DS_BEHAVIOR_WIN2003_WITH_MIXED_DOMAINS, or DS_BEHAVIOR_WIN2003 cannot support AES.

~~<5453> Section 3.3.5.6.34.5~~: Windows 2000 and Windows Server 2003 do not support UPN and DNS information.

~~<5554> Section 3.3.5.6.34.6~~: For Active Directory with the **msDS-Behavior-Version** attribute on a domain NC root object equal to DS_BEHAVIOR_WIN2000, DS_BEHAVIOR_WIN2003_WITH_MIXED_DOMAINS, DS_BEHAVIOR_WIN2003, DS_BEHAVIOR_WIN2008, or DS_BEHAVIOR_WIN2008R2, KDCs will behave as if 1 is set.

~~<5655> Section 3.3.5.7~~: DES downgrade protection is not supported in Windows 2000, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, or Windows Server 2012 KDCs.

~~<5756> Section 3.3.5.7~~: When the account is for a computer object and the value of OperatingSystemVersion ([MS-ADA3] section 2.56) is less than 6, **KerbSupportedEncryptionTypes** is treated as if it were not populated to ensure that newer encryption types are not attempted with Windows NT, Windows 2000, Windows XP, and Windows Server 2003, which do not support setting **KerbSupportedEncryptionTypes**.

~~<57> Section 3.3.5.7: Not supported in Windows 2000 and Windows Server 2003.~~

<58> Section 3.3.5.7: Not supported in Windows 2000 and Windows Server 2003.

~~<59> Section 3.3.5.7: Not supported in Windows 2000 and Windows Server 2003.~~

~~<60<59> Section 3.3.5.7~~: Claims and FAST are not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7 or Windows Server 2008 R2.

~~<6160> Section 3.3.5.7~~: DES downgrade protection is not supported in Windows 2000, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, or Windows Server 2012 KDCs.

~~<6261> Section 3.3.5.7~~: Authentication Policies are not supported in Windows 2000, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, or Windows Server 2012 KDCs.

~~<6362> Section 3.3.5.7~~: The following versions of Windows do not support Claims: Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2.

~~<6463> Section 3.3.5.7~~: Windows 2000, Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008 do not support KERB-LOCAL.

~~<6564> Section 3.3.5.7.1~~: Windows uses 20 minutes as the time value at which a TGT is verified to be in good standing.

~~<66> Section 3.3.5.7.3: The way in which this processing is represented in this specification was updated for the unreleased preliminary version of Windows Server 2016 Technical Preview.~~

~~<67<65> Section 3.3.5.7.3~~: Resource SID compression is not supported in Windows 2000, Windows Server 2003, Windows Server 2008, or Windows Server 2008 R2 KDCs.

~~<6866> Section 3.3.5.7.4~~: Compound identity is not supported in Windows 2000, Windows Server 2003, Windows Server 2008, or Windows Server 2008 R2 KDCs.

~~<6967> Section 3.3.5.7.5~~: DES downgrade protection is not supported in Windows 2000, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, or Windows Server 2012 KDCs.

- <~~7068~~> Section 3.3.5.7.6: Not supported in Windows 2000 and Windows Server 2003.
- <~~7169~~> Section 3.4.1: Channel binding is not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008.
- <~~7270~~> Section 3.4.3.1: Not supported in Windows 2000, Windows XP and Windows Server 2003.
- <~~7371~~> Section 3.4.3.1: Not supported in Windows 2000, Windows XP and Windows Server 2003.
- <~~7472~~> Section 3.4.5: SPNs with serviceclass string equal to "RestrictedKrbHost" are not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, or Windows Server 2008.
- <~~7573~~> Section 3.4.5: The ApplicationRequiresCBT parameter is not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, or Windows Server 2008.
- <~~7674~~> Section 3.4.5: DES downgrade protection is not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, or Windows Server 2012.
- <~~7775~~> Section 3.4.5.3: Windows 2000, Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008 do not support KERB-LOCAL.
- <~~7876~~> Section 3.4.5.3: Not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008.
- <~~7977~~> Section 3.4.5.3: Claims is not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7 or Windows Server 2008 R2.
- <~~8078~~> Section 3.4.5.3: Compound identity is not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7 or Windows Server 2008 R2.
- <~~8179~~> Section 3.4.5.3: Compound identity and claims are not supported in Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7 or Windows Server 2008 R2.

7 Change Tracking

This section identifies changes that were made to this document since the last release. 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.
- The removal of a document from the documentation set.

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 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 changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

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.
- 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 dochelp@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
2.3 Directory Service Schema Elements	72632 : Included the sAMAccountName attribute for the user class.	Y	Content update.
3.1.1 Abstract Data Model	6183 : Removed that KILE key version numbers are signed 32-bit integers because that information is in section 3.1.5.8.	Y	Content update.
3.1.1.5 SupportedEncryptionTypes	Content reverted to original text.	N	Product behavior note updated.
3.1.5.8 Key Version Numbers	6183 : Specified that KILE key version numbers are encoded and decoded as signed 32-bit integers.	Y	Content update.
3.3.5.1.1 Server Principal Lookup	72632 : Added the Server Principal Lookup section.	Y	New content added.
3.3.5.5 Determining Authentication Policy Settings	Corrected inspection of the BelongsToSilo field, when account does not belong to a Silo.	Y	Content update.
3.3.5.6.1 Client Principal Lookup	72632 : Added the Client Principal Lookup section.	Y	New content added.
3.3.5.7 TGS Exchange	72588 : Added the condition that triggers the KDC to return KDC_ERR_MUST_USE_USER2USER.	Y	Content update.
3.4.5 Message Processing Events and Sequencing Rules	72749 : Clarified the application server behavior when the checksum field is not present.	Y	Content update.

8 Index

A

- Abstract data model
 - Application Server (section 3.1.1 24, section 3.4.1 57)
 - client (section 3.1.1 24, section 3.2.1 30)
 - KDC (section 3.1.1 24, section 3.3.1 36)
 - server 57
- Addressing 28
- AES 128 key creation example 69
- AP exchange 36
- Applicability 18
- Application Server
 - abstract data model (section 3.1.1 24, section 3.4.1 57)
 - higher-layer triggered events (section 3.1.4 25, section 3.4.4 58)
 - implementing public keys 30
 - initialization (section 3.1.3 25, section 3.4.3 57)
 - local events (section 3.1.7 30, section 3.4.7 64)
 - message processing (section 3.1.5 25, section 3.4.5 58)
 - overview (section 3.1.1 24, section 3.4 57)
 - sequencing rules (section 3.1.5 25, section 3.4.5 58)
 - timer events (section 3.1.6 30, section 3.4.6 64)
 - timers (section 3.1.2 25, section 3.4.2 57)
- AS exchange 35
- Authentication
 - datagram style 59
 - pre-authentication 18
 - services 32
 - three-leg DCE style mutual 59
- Authenticator checksum flags 33
- Authorization data
 - overview 28
 - processing 60

C

- Calls
 - GSS_GetMICEx() 63
 - GSS_UnwrapEx() 63
 - GSS_VerifyMICEx() 64
 - GSS_WrapEx() 61
- Capability negotiation 18
- Case sensitivity 28
- Change tracking 79
- Client
 - abstract data model (section 3.1.1 24, section 3.2.1 30)
 - higher-layer triggered events (section 3.1.4 25, section 3.2.4 32)
 - implementing public keys 30
 - initialization (section 3.1.3 25, section 3.2.3 32)
 - local events (section 3.1.7 30, section 3.2.7 36)
 - message processing (section 3.1.5 25, section 3.2.5 33)
 - other local events 36
 - overview 24
 - sequencing rules (section 3.1.5 25, section 3.2.5 33)
 - timer events (section 3.1.6 30, section 3.2.6 36)
 - timers (section 3.1.2 25, section 3.2.2 32)
- Compound identity 16
- Cryptography 24

D

- Data model - abstract

- Application Server (section 3.1.1 24, section 3.4.1 57)
 - client (section 3.1.1 24, section 3.2.1 30)
 - KDC (section 3.1.1 24, section 3.3.1 36)
 - server 57
- Datagram-style authentication 59
- DCE style mutual authentication - three-leg 59
- Directory service schema elements 22
- Domain controller - locating 34
- DS_BEHAVIOR_WIN2012 domain controller - locating 34

E

- Elements - directory service schema 22
- Encryption checksum types 27
- Encryption types (section 1.7.2 18, section 3.1.5.2 26)
- Encryption types - bit flags (section 2.2.7 21, section 3.1.1.5 25)
- Examples
 - AES 128 key creation 69
 - GSS_WrapEx with AES128-CTS-HMAC-SHA1-96 67
 - interactive logon 65
 - network logon 67
 - overview 65
 - RC4 GSS_WrapEx 70

F

- Fields - vendor-extensible 18
- Flags
 - authenticator checksum 33
 - request 33
- Flexible Authentication Secure Tunneling (FAST)
 - overview 16
 - using when supported by realm 34
- Forwardable TGT request 35

G

- Glossary 7
- GSS_GetMICEx() call 63
- GSS_UnwrapEx() call 63
- GSS_VerifyMICEx() call 64
- GSS_WrapEx with AES128-CTS-HMAC-SHA1-96 example 67
- GSS_WrapEx() call 61

H

- Higher-layer triggered events
 - Application Server (section 3.1.4 25, section 3.4.4 58)
 - client (section 3.1.4 25, section 3.2.4 32)
 - KDC
 - configuration changes 40
 - overview (section 3.1.4 25, section 3.3.4 40)
 - server 58

I

- Implementer - security considerations 72
- Index of security parameters 72
- Informative references 13
- Initial logon 32
- Initialization
 - Application Server (section 3.1.3 25, section 3.4.3 57)
 - client (section 3.1.3 25, section 3.2.3 32)
 - KDC (section 3.1.3 25, section 3.3.3 39)

- server 57
- Interactive logon example 65
- Internationalization 28
- Introduction 7

K

KDC

- abstract data model (section 3.1.1 24, section 3.3.1 36)
- higher-layer triggered events
 - configuration changes 40
 - overview (section 3.1.4 25, section 3.3.4 40)
- implementing public keys 30
- initialization (section 3.1.3 25, section 3.3.3 39)
- local events (section 3.1.7 30, section 3.3.7 57)
- message processing (section 3.1.5 25, section 3.3.5 40)
- overview 24
- sequencing rules (section 3.1.5 25, section 3.3.5 40)
- timer events (section 3.1.6 30, section 3.3.6 57)
- timers (section 3.1.2 25, section 3.3.2 39)
- KERB-AD-RESTRICTION-ENTRY message 21
- KERB-AD-RESTRICTION-ENTRY structure 21
- Kerberos OID 25
- Kerberos V5 synopsis 14
- KERB-ERROR-DATA message 19
- KERB-ERROR-DATA structure 19
- KERB-EXT-ERROR message 19
- KERB-LOCAL message 20
- KERB-PA-PAC-REQUEST message 20
- KERB-PA-PAC-REQUEST structure 20
- Keys
 - public 30
 - usage numbers 29
 - version numbers 29
- KILE synopsis 16

L

Local events

- Application Server (section 3.1.7 30, section 3.4.7 64)
 - client (section 3.1.7 30, section 3.2.7 36)
 - KDC (section 3.1.7 30, section 3.3.7 57)
- Locating DS_BEHAVIOR_WIN2012 domain controller 34
- Logon
 - initial 32
 - interactive - example 65
 - network - example 67
- LSAP_TOKEN_INFO_INTEGRITY message 20
- LSAP_TOKEN_INFO_INTEGRITY structure 20

M

- Machine ID 25
- Message processing
 - addressing 28
 - Application Server (section 3.1.5 25, section 3.4.5 58)
 - authorization data 28
 - case sensitivity 28
 - client (section 3.1.5 25, section 3.2.5 33)
 - encryption checksum types 27
 - encryption types 26
 - internationalization 28
 - KDC (section 3.1.5 25, section 3.3.5 40)
 - key usage numbers 29
 - key version numbers 29

- locating DS_BEHAVIOR_WIN2012 domain controller 34
- naming 29
- PAC generation 43
- pre-authentication data 26
- referrals 29
- server 58
- ticket flag 27
- Messages
 - KERB-AD-RESTRICTION-ENTRY 21
 - KERB-ERROR-DATA 19
 - KERB-EXT-ERROR 19
 - KERB-LOCAL 20
 - KERB-PA-PAC-REQUEST 20
 - LSAP_TOKEN_INFO_INTEGRITY 20
 - OCTET STRING 22
 - PA-PAC-OPTIONS 22
 - PA-SUPPORTED-ENCTYPES 22
 - Supported Encryption Types Bit Flags 21
 - syntax 19
 - transport 19

N

- Naming 29
- Network logon example 67
- Normative references 11

O

- OCTET STRING 22
- OCTET STRING message 22
- OID - Kerberos 25
- Other local events
 - client 36
 - server 64
- Overview (synopsis) 14

P

- PAC generation 43
- PA-PAC-OPTIONS message 22
- PA-PAC-OPTIONS structure 22
- Parameter index - security 72
- Parameters - security index 72
- PA-SUPPORTED-ENCTYPES message 22
- PA-SUPPORTED-ENCTYPES structure 22
- PLSAP_TOKEN_INFO_INTEGRITY 20
- Pre-authentication 18
- Pre-authentication data 26
- Preconditions 17
- Prerequisites 17
- Product behavior 73
- Protocol Details
 - overview 24
- Public keys - implementing
 - Application Server 30
 - client 30
 - KDC 30

R

- RC4_GSS_WrapEx example 70
- References 11
 - informative 13
 - normative 11

- Referrals 29
- Relationship to other protocols 17
- Replay cache 24
- Request flags 33

S

- Schema elements - directory service 22
- Security
 - background 14
 - implementer considerations 72
 - overview 72
 - parameter index 72
- Sequencing rules
 - addressing 28
 - Application Server (section 3.1.5 25, section 3.4.5 58)
 - authorization data 28
 - case sensitivity 28
 - client (section 3.1.5 25, section 3.2.5 33)
 - encryption checksum types 27
 - encryption types 26
 - internationalization 28
 - KDC (section 3.1.5 25, section 3.3.5 40)
 - key usage numbers 29
 - key version numbers 29
 - locating DS_BEHAVIOR_WIN1012 domain controller 34
 - naming 29
 - PAC generation 43
 - pre-authentication data 26
 - referrals 29
 - server 58
 - ticket flag 27
- Server
 - abstract data model (section 3.1.1 24, section 3.4.1 57)
 - higher-layer triggered events (section 3.1.4 25, section 3.4.4 58)
 - implementing public keys 30
 - initialization (section 3.1.3 25, section 3.4.3 57)
 - local events (section 3.1.7 30, section 3.4.7 64)
 - message processing (section 3.1.5 25, section 3.4.5 58)
 - other local events 64
 - overview (section 3.1.1 24, section 3.4 57)
 - sequencing rules (section 3.1.5 25, section 3.4.5 58)
 - timer events (section 3.1.6 30, section 3.4.6 64)
 - timers (section 3.1.2 25, section 3.4.2 57)
- Standards assignments 18
- Supported encryption types (section 2.2.7 21, section 3.1.1.5 25)
- Supported Encryption Types Bit Flags message 21
- Syntax - message 19

T

- TGS exchange 35
- Three-leg DCE style mutual authentication 59
- Ticket cache 25
- Ticket flag 27
- Timer events
 - Application Server (section 3.1.6 30, section 3.4.6 64)
 - client (section 3.1.6 30, section 3.2.6 36)
 - KDC (section 3.1.6 30, section 3.3.6 57)
 - server 64
- Timers
 - Application Server (section 3.1.2 25, section 3.4.2 57)
 - client (section 3.1.2 25, section 3.2.2 32)
 - KDC (section 3.1.2 25, section 3.3.2 39)
 - server 57

- Tracking changes 79
- Transport 19
- Triggered events
 - Application Server (section 3.1.4 25, section 3.4.4 58)
 - client (section 3.1.4 25, section 3.2.4 32)
 - KDC
 - configuration changes 40
 - overview (section 3.1.4 25, section 3.3.4 40)
- Triggered events - higher-layer server 58

V

- Vendor-extensible fields 18
- Versioning 18