How Encryption Type Settings Affect Kerberos Key Generation, Negotiation and Crypto Usage

Will Fiveash

Software Engineer   (updated Aug 30 2005)

Solaris Security Group

Introduction

- DES is not secure enough to protect sensitive information
- Kerberos in Solaris has recently been enhanced to provide more secure crypto algorithms (AES,DES3,RC4,SHA1)
- System administrators, developers and testers should know how to control the encryption used by Kerberos for better security and interoperability

Enctypes Explained

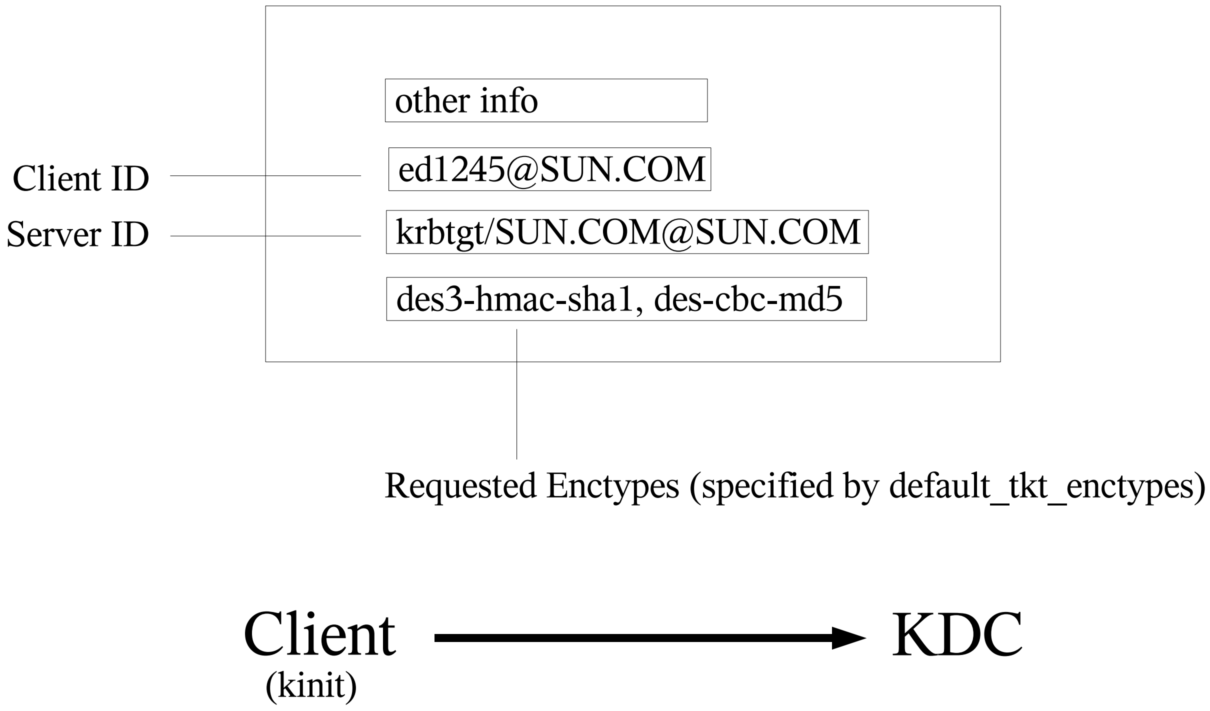
- A Kerberos encryption type (enctype) is an identifier specifying encryption algorithm, mode and hash algorithms. Example: des-cbc-md5.
- Keys in Kerberos have an associated enctype to identify the cryptographic algorithm and mode to be used when performing cryptographic operations with the key.
- Typically each principal has an associated secret key for each unique enctype supported in the local Kerberos realm.

Enctype Negotiation

- Typically a Kerberos client requests one or more enctypes along with client and server principal IDs and other info in an AS request sent to the KDC. Usually the AS_REQ is for a ticket-granting ticket (TGT) from the Ticket Granting Server (TGS). This is the initial credential used to acquire other tickets.
- It is important that the enctypes requested by the client are actually supported on the system hosting the client. This is the case if the defaults controlling enctypes are not overridden.
- The server principal is also called a service principal in Kerberos terminology.

Enctypes and the AS Request

Overview of AS Request Message



other info

Client ID — ed1245@SUN.COM

Server ID — krbtgt/SUN.COM@SUN.COM

des3-hmac-sha1, des-cbc-md5

Requested Enctypes (specified by default_tkt_enctypes)

Client (kinit) → KDC

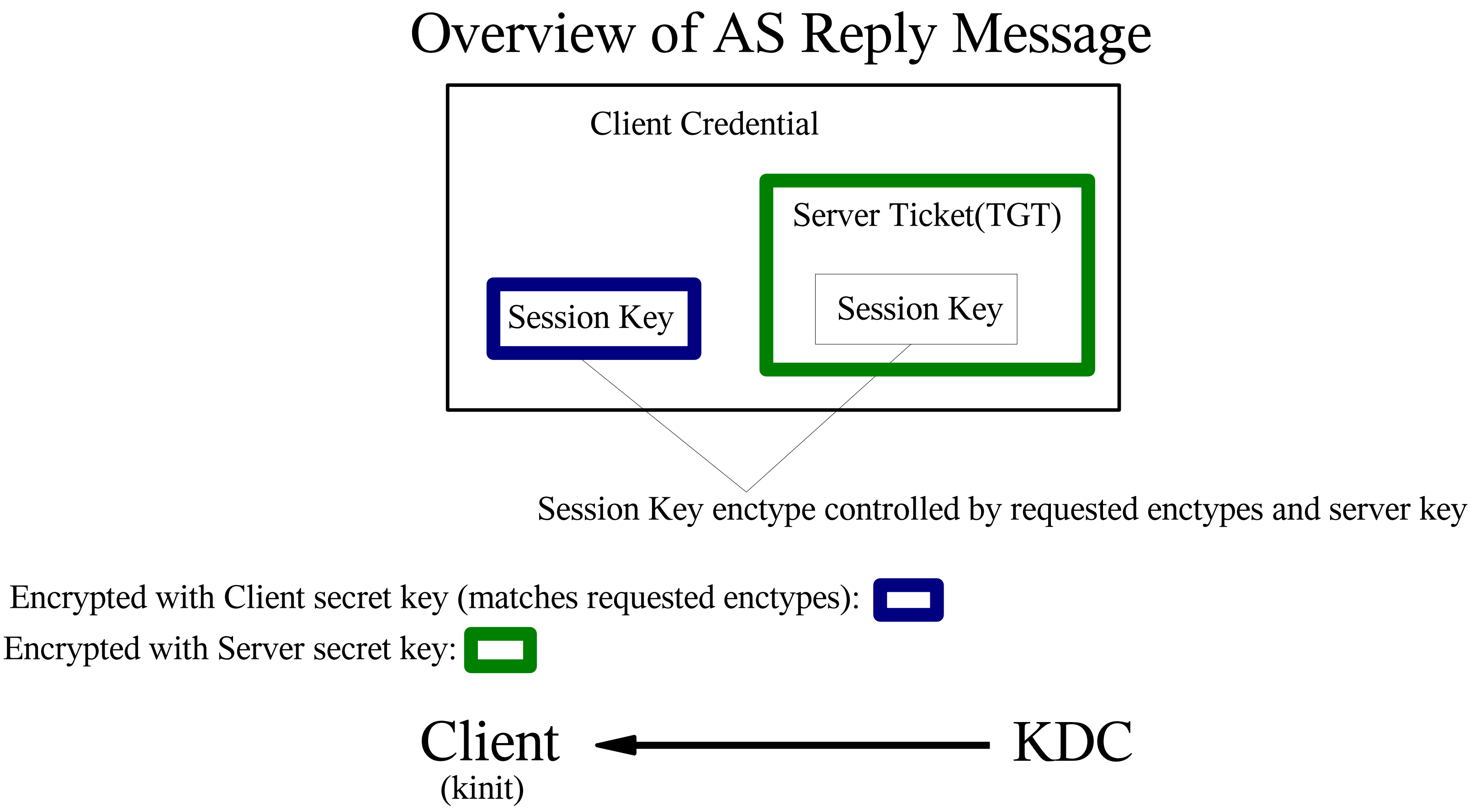
Enctype Negotiation cont.

- Upon receipt of an AS_REQ the KDC creates session keys using the first common enctype between the AS_REQ requested enctypes and the server principal keys. The server keys are located in the KDC principal DB (KDB) using the server ID in the AS_REQ (usually the server in a AS_REQ is krbtgt).
- The KDC then encrypts the server ticket using the first server key in the KDB and associated enctype.

Enctype Negotiation cont.

- Using the client ID, the KDC finds the client KDB record and then chooses the key that matches one of the requested enctypes and encrypts part of the AS_REQ reply (including the server ticket) with that key and enctype.
- Again, the enctype chosen by the KDC must be supported by the client system in order for the client to access the session key.

Enctypes and the AS Reply



Overview of AS Reply Message

Session Key enctype controlled by requested enctypes and server key

Encrypted with Client secret key (matches requested enctypes): 🟦

Encrypted with Server secret key: 🟩

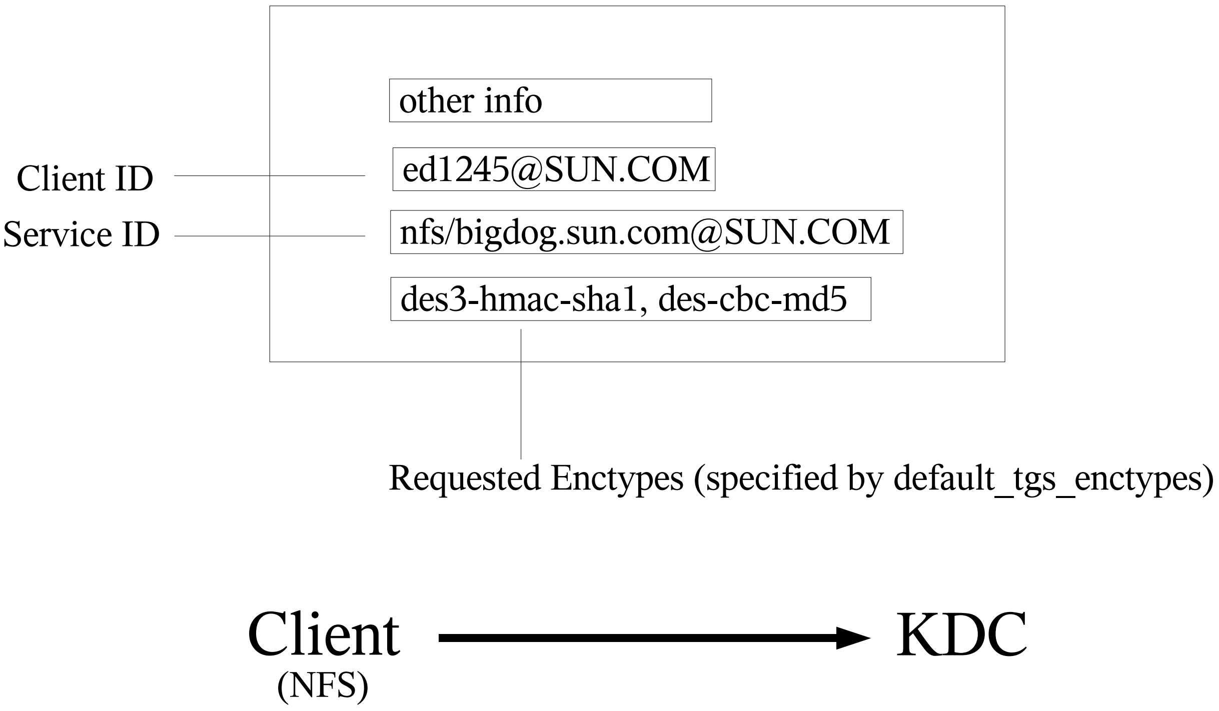
Enctype Negotiation Cont.

- Typically the client uses the TGT to request tickets for non-TGS servers (NFS for example) from the KDC. default_tgs_enctypes (krb5.conf parameter described later) controls the enctypes requested in the TGS req. sent by the client.

- If default_tgs_enctypes is unset the client system will request all enctypes supported on the system.

Enctypes and the TGS Request

Overview of TGS Request Message



other info

Client ID — ed1245@SUN.COM

Service ID — nfs/bigdog.sun.com@SUN.COM

des3-hmac-sha1, des-cbc-md5

Requested Enctypes (specified by default_tgs_enctypes)

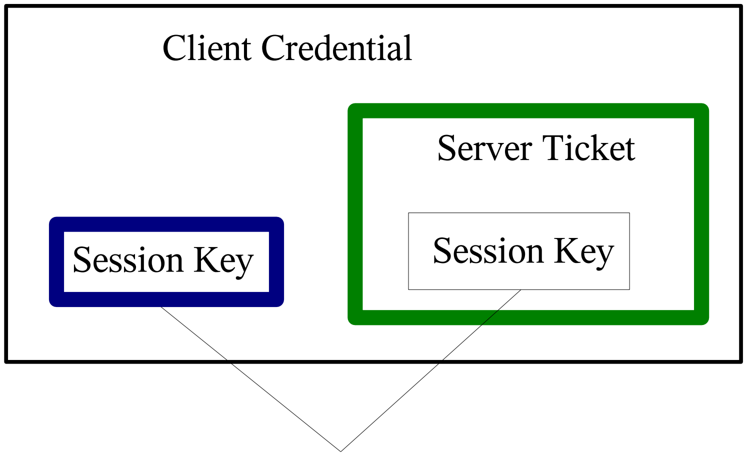
Client (NFS) → KDC

Enctype Negotiation Cont.

- The KDC uses the first match between the TGS req enc-types and the server's keys to determine the enctype of the session key for the server ticket.
- The KDC uses the first server KDB key and enctype to encrypt the server ticket.
- The KDC encrypts parts of the TGS rep. with the TGT session key created from the earlier AS exchange.

Enctypes and the TGS Reply

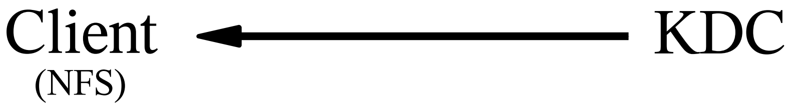
Overview of TGS Reply Message



Session Key enctype controlled by requested enctypes and server key

Encrypted with TGT session key (shared between client and KDC): 🟦

Encrypted with Server secret key: 🟩



Enctype Negotiation Cont.

- The client then sends the server ticket in a AP req message to the server along with other data to authenticate to the server.
- The server must access the proper key in the local file /etc/krb5/krb5.keytab to decrypt the server ticket in the AP req message.
- The server never communicates with the KDC. This is why the KDC must only find server principal keys that are supported on the server when encrypting the server ticket.

Enctype Negotiation Cont.

- To recap: the first key (and associated enctype) for the server principal protects the server ticket.
- The first enctype in the client's requested enctypes to match one of the server keys specifies the enctype of the session key shared by the client and server.
- The client key that matches one of the requested enctypes in the AS/TGS request is used to protect the AS/TGS reply sent back to the client from the KDC.

New Enctypes

- Prior to Solaris 10, Kerberos (SEAM) supports the des-cbc-crc and des-cbc-md5 enctypes (both indicate the use of 1DES crypto).
- In Solaris 10, Kerberos additionally supports the des3-hmac-sha1, rc4-hmac-md5, aes128-cts-hmac-sha1-96 and aes256-cts-hmac-sha1 enctypes. Note, support of aes256-cts-hmac-sha1 requires the unbundled SUNW-cry package (strong crypto) be installed.

Config Parameters and enctypes

- There are several Kerberos configuration parameters that specify the enctypes to use for certain operations:
  - default_tkt_enctypes
  - default_tgs_enctypes
  - permitted_enctypes
  - supported_enctypes
  - master_key_type

Config Parameters and enctypes

- First, a strong suggestion: if control over the enctypes used by Kerberos is desired, use the supported_enctypes parameter (described later) for this purpose as this is less likely to cause interoperability problems.
- Be sure that all systems using Kerberos authentication support the supported_enctypes specified for the realm.

default_tkt_enctypes

- default_tkt_enctypes (in [libdefaults] section in krb5.conf) specifies the requested enctypes in the AS_REQ. Generally useful in restricting the session key enctype shared by the client and the TGS (ticket granting server/KDC).
- In Solaris 10 the default list is aes256-cts-hmac-sha1*, aes128-cts-hmac-sha1-96, rc4-hmac-md5, des3-hmac-sha1, des-cbc-md5, des-cbc-crc. *The SUNWcry package must be installed to get aes256-cts-hmac-sha1 support.

default_tgs_enctypes

- default_tgs_enctypes ([libdefaults] section in krb5.conf) controls the set of session key enctypes the client requests in a TGS_REQ. Generally useful in restricting the session key enctype shared by the client and the application server (for example secure NFS or rlogin). Must include enctype of TGT session key.
- In Solaris 10 the default list is aes256-cts-hmac-sha1*, aes128-cts-hmac-sha1-96, rc4-hmac-md5, des3-hmac-sha1, des-cbc-md5, des-cbc-crc.

permitted_enctypes

- permitted_enctypes ([libdefaults] section in krb5.conf) controls the set of ticket enctypes that will be accepted by the server. This is used by the TGS (KDC) and Kerberized servers. Essentially, this limits enctypes permitted for using in the service ticket and session key encryption.
- In Solaris 10 the default list is aes256-cts-hmac-sha1*, aes128-cts-hmac-sha1-96, rc4-hmac-md5, des3-hmac-sha1, des-cbc-md5, des-cbc-crc.
- Must include the strongest key enctype in the KDB entry for service principals on system (KDC issues tickets using service principals strongest key).

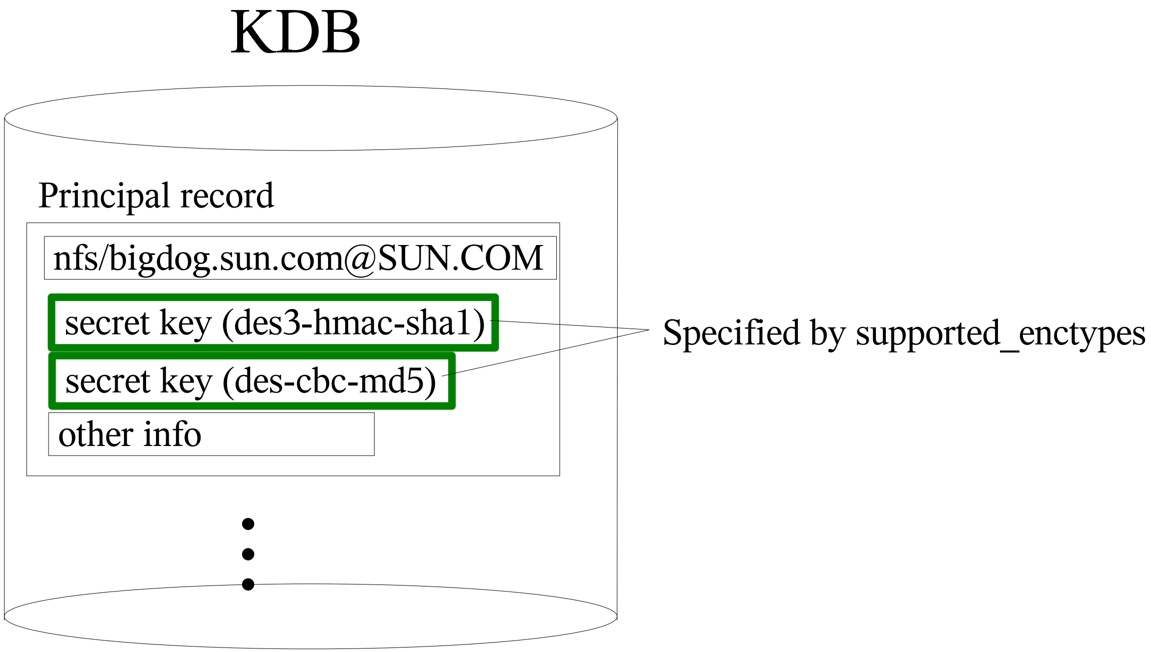
supported_enctypes

- supported_enctypes (in [realm] section in kdc.conf) a list of enctype and salt pairs that controls the default types of keys generated for principals in that realm.
- Basically controls keys generated by kadmin.
- In Solaris 10 the default list is aes256-cts-hmac-sha1*, aes128-cts-hmac-sha1-96, rc4-hmac-md5, des3-hmac-sha1, des-cbc-md5, des-cbc-crc.

master_key_type

- master_key_type (in [realm] section in kdc.conf) specifies the enctype of the KDB master key. This determines the type of crypto. used to encrypt the keys in the KDB.
- Defaults to des-cbc-crc. Can be set to any of the supported enctypes to encrypt the keys in the KDB.
- If using a non-default enctype make sure all slave KDC's support that enctype otherwise the slave KDC will not be able to read the keys in the KDB. Note, this should not be changed if the KDB already exists (at this time).

Enctypes and the KDB



KDB

Principal record

nfs/bigdog.sun.com@SUN.COM

secret key (des3-hmac-sha1)

secret key (des-cbc-md5)

other info

Specified by supported_enctypes

Encrypted with master key (enctype specified by master_key_type): 🟩

For Admins

- Do not create server keytab entries using enctypes that the Kerberized server doesn't support! If necessary specify the enctype when creating (kadmin ktadd -e “enctype:salt”) the server principal keytab entry if the supported_enctypes parameter includes enctypes not supported by the server.
- The default for supported_enctypes is all enctypes supported by the installed Kerberos code.

For Admins cont.

- The Solaris 10 KDC kadmind daemon will automatically limit the enctypes for the service principal to those supported on the system where the kadmin ktadd command is run. This will help prevent enctype compatibility problems for servers that use Kerberos auth.

For Admins cont.

- Also, the keys in a server's keytab must stay in sync with the keys for that server principal in the KDB. If the keys need to be changed, use kadmin ktadd on the server so the keys in the keytab and KDB are identical.

Telnet Hard-coded Enctypes

- In Telnet, the requested session key enctypes are determed by the intersection of a hard-coded list and default_tgs_enctypes.
- The hard-coded list is des-cbc-md5, des-cbc-crc. This limits Telnet to 1DES enctypes.
- The host service principal must have 1DES keys (in addition to any other keys) in the KDB so the KDC will generate a 1DES session key for the telnet session.

Rcmds and Enctypes

- The rcmds such as rsh, rlogin, rexec use the Kerberos session key enctype which is determined by the intersection of the default_tgs_enctypes and the host service principal keys for the remote host.
- The enctype of the Kerberos session key determines the cryptographic algorithm used if the rcmd session is encrypted (see the -x flag).

For Developers

- Do not hard-code enctype usage in your application, let Kerberos deal with this (this can restrict the encryption used by your application in the future).

Conclusion

- The encryption used in Kerberos depends on:
  - > The enctypes the client requests.
  - > The enctype of the keys.
  - > The parameters in the Kerberos config files.
- It's useful to understand the interaction between these things if control is desired over the encryption used to protect: the keys in the KDB, communication between the client and the KDC and communication between the client and the server.

Conclusion cont.

- Solaris 10 now supports AES, RC4 and 3DES enctypes in addition to the current, less secure 1DES enctypes.

- **Controlling the enctypes used in a realm is best done using the supported_enctypes and permitted_enctypes config parameters on the KDC and use the defaults for the other config parameters.**

How Encryption Type Settings Affect Kerberos Key Generation, Negotiation and Crypto Usage

Will Fiveash

william.fiveash@sun.com