Oracle Advanced Networking Option^{TM}

Administrator’s Guide

Release 8.0

December 1997

Part No. A58229-01

ORACLE®

Enabling the Information Age

Oracle Advanced Networking Option Administrator’s Guide

Release 8.0

Part No. A58229-01

Primary Author: Gilbert Gonzalez
Contributing Authors: Laura Ferrer, Patricia Markee, Kendall Scott, Sandy Venning, Rick Wong
Contributors: Andre Srinivasan, Richard Wessman, Lisa-ann Wilkinson

The programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be licensee's responsibility to take all appropriate fail-safe, back up, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle disclaims liability for any damages caused by such use of the Programs.

This Program contains proprietary information of Oracle Corporation; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright patent and other intellectual property law. Reverse engineering of the software is prohibited.

Portions of Oracle Advanced Networking Option have been licensed by Oracle Corporation from RSA Data Security.

RSA Secure™

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free.

If this Program is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

Restricted Rights Legend Programs delivered subject to the DOD FAR Supplement are 'commercial computer software' and use, duplication and disclosure of the Programs shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are 'restricted computer software' and use, duplication and disclosure of the Programs shall be subject to the restrictions in FAR 52..227-14, Rights in Data -- General, including Alternate III (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

Contents

Part I Oracle Advanced Networking Option Security and Single Sign-On

1 Network Security and Single Sign-On

2 Configuring Encryption and Checksumming

3 Configuring the CyberSAFE Authentication Adapter

4 Configuring the Kerberos Authentication Adapter

5 Configuring Oracle for Use with the SecurID Adapter

7 Choosing and Combining Authentication Services

8 Configuring the DCE GSSAPI Authentication Adapter

Part II Oracle Advanced Networking Option and Oracle DCE Integration

9 Overview of Oracle DCE Integration

10 Configuring DCE for Oracle DCE Integration

11 Configuring Oracle for Oracle DCE Integration

12 Connecting to an Oracle Database in DCE

13 DCE and Non-DCE Interoperability

A Encryption and Checksum Parameters

B Authentication Parameters

Glossary

Index

Preface

The Oracle Advanced Networking Option is an optional product that provides enhanced functionality to SQL*Net and Net8. Its set of features provides enhanced security and authentication to your network and enables integration with a Distributed Computing Environment (DCE). This guide provides generic information on all these features of the Advanced Networking Option.

For information about installation of the Oracle Advanced Networking Option and platform-specific details of the configuration and use of its features, refer also to your Oracle platform-specific documentation.

How This Manual Is Organized

This manual is divided into two parts: Security and Single Sign-On and DCE Integration. Each part describes a different set of Oracle Advanced Networking Option features.

Part I Security and Single Sign-On

Chapter 1, “Network Security and Single Sign-On”

This chapter provides an overview of the security and single sign-on features of the Oracle Advanced Networking Option. It includes an brief overview of the authentication adapters available with this release, and it describes how to disable the use of the authentication adapters when you want to use username/password authentication instead. These features include:

- network security
  - data encryption
  - data integrity checking
- token authentication
- single sign-on

Note: These features were previously packaged as the Secure Network Services product.

Chapter 2, “Configuring Encryption and Checksumming”

This chapter provides a brief overview of the authentication adapters available with this release. It describes how to disable the use of the authentication adapters when you want to use username/password authentication instead. It also describes how to configure multiple authentication adapters on clients and servers.This chapter tells you how to install the encryption and checksumming software and tells you how to configure encryption and checksumming into your existing SQL*Net release 8.0.3 network using Oracle Net8 Assistant.

Chapter 3, “Configuring the CyberSAFE Authentication Adapter”

This chapter discusses how to configure Oracle for use with CyberSAFE, and provides a brief overview of steps to configure CyberSAFE to authenticate Oracle users.

Chapter 4, “Configuring the Kerberos Authentication Adapter”

This chapter discusses how to configure Oracle for use with MIT Kerberos, and provides a brief overview of steps to configure Kerberos to authenticate Oracle users.

Chapter 5, “Configuring Oracle for Use with the SecurID Adapter”

This chapter discusses how to configure the SecurID authentication adapter in combination with the Oracle server and Oracle clients. It includes system requirements and known limitations. It also contains troubleshooting information if you experience problems while configuring the SecurID authentication adapter.

Note: For a complete list of Advanced Networking Option error messages see the Oracle Network Products Troubleshooting Guide.

Chapter 6, “Configuring and Using the Identix Biometric Authentication Adapter”

This chapter describes how to configure and use the the Oracle Biometric authentication adapter, which enables the use of the Identix fingerprint authentication device.

Chapter 7, “Choosing and Combining Authentication Services”

This chapter discusses how to use the SecurID authentication adapter in combination with the Oracle client tools.

Chapter 8, “Configuring the DCE GSSAPI Authentication Adapter”

This chapter describes how to configure the Oracle DCE GSSAPI authentication adapter to provide DCE authentication even if you are not using other DCE services in your network.

Part II DCE Integration

Chapter 9, “Overview of Oracle DCE Integration”

This chapter provides a brief discussion of OSF’s DCE and Oracle’s DCE Integration.

Chapter 10, “Configuring DCE for Oracle DCE Integration”

This chapter describes what you need to do to configure DCE to use Oracle DCE Integration. It also describes how to configure the DCE CDS naming adapter.

Chapter 11, “Configuring Oracle for Oracle DCE Integration”

This chapter describes the DCE parameters that you need to add to the SQL*Net configuration files to enable clients and servers to access Oracle7 and Oracle8 servers in the DCE environment. It also describes some Oracle Server configuration that you need to perform, such as setting up DCE groups to map to external roles. Additionally, it describes how to configure clients to use the DCE CDS naming adapter.

Chapter 12, “Connecting to an Oracle Database in DCE”

This chapter discusses how to connect to an Oracle database in a DCE environment.

Chapter 13, “DCE and Non-DCE Interoperability”

This chapter discusses how clients outside of DCE can access Oracle databases using another protocol such as TCP/IP.

Appendices

Appendix A, “Encryption and Checksum Parameters”

This appendix shows examples of the Oracle Advanced Networking Option encryption and checksumming configuration parameters. You can use the Oracle Net8 Assistant to create, modify, or delete these parameters. When the configuration files are generated, the parameters appear in a profile. These parameters are described in Chapter 2, “Configuring Encryption and Checksumming”.

Appendix B, “Authentication Parameters”

This appendix shows examples of the Oracle Advanced Networking Option authentication configuration file parameters.

Notational Conventions

The following syntax conventions are used in this guide:

italic	Italic characters indicate that the parameter, variable, or expression in the command syntax must be replaced by a value that you provide. Italics may also indicate emphasis or the first mention of a technical term.

`Monospace Text` Monospace font indicates something the computer displays.
Note: In some cases, brackets surround certain words (for example, `<pin><passcode>`) to more clearly separate words in a command.

`Monospace Text` Bolded monospace font indicates text you need to enter exactly as shown.
Note: In some cases, angle brackets surround certain words (for example, `<pin><passcode>`) to more clearly separate words in a command.

Punctuation Punctuation other than brackets and vertical bars must be typed as shown.

[ ] Brackets enclose optional items. Do not type the brackets.

( ) Parentheses enclose all SQL*Net and Net8 Keyword-Value pairs in connect descriptors. They must be entered as part of the connect descriptor, as in (KEYWORD=value).

| A vertical bar represents a choice of two or more options. You must type one of the options separated by the vertical bar. Do not type the vertical bar.

UPPERCASE Uppercase characters within the text represent command names, file names, and directory names.

Related Publications

To install and configure Advanced Networking Option software on your particular platform, refer to the Oracle platform-specific documentation.

In addition, see the following documents for detailed information about Oracle network products that applies across platforms:

- *Oracle Net8 Administrator's Guide*
- *Oracle8 Distributed Database Systems*

For information on roles and privileges, see:

- *Oracle Security Server Guide*

For third-party vendor documentation on security and single sign-on features see:

- Security Dynamics' ACE/Server Installation Manual, release 1.3

- Security Dynamics' ACE/Server Version 1.3 Administration Manual
- ACE/Server Version 2.0 Client for UNIX
- CyberSAFE Challenger Release Notes, release 5.2.6
- CyberSAFE Challenger Administrator's Guide, release 5.2.6
- CyberSAFE Challenger Navigator Administrator's Guide, release 5.2.6
- CyberSAFE Challenger UNIX User's Guide, release 5.2.6
- CyberSAFE Challenger Windows and Windows NT User's Guide, release 5.2.6

For information on MIT Kerberos see:

- CyberSAFE Challenger documentation
- Notes on building and installing Kerberos from Kerberos V5 source distribution
- CNS (Cygnus Network Security) documentation from http://www.cygnus.com/library-dir.html

For additional information about the OSF Distributed Computing Environment (DCE), refer to the following OSF documents published by Prentice Hall, Inc.:

- OSF DCE User's Guide and Reference
- OSF DCE Application Development Guide
- OSF DCE Application Development Reference
- OSF DCE Administration Guide
- OSF DCE Administration Reference
- OSF DCE Porting and Testing Guide
- Application Environment Specification/Distributed Computing
- OSF DCE Technical Supplement

For information about Identix products, refer to the following Identix documentation.

Client side documentation:

- Identix TouchNet II User's Guide

Server side documentation:

- Identix TouchNet II System Administrator's Guide

Send Us Your Comments

Oracle Advanced Networking Option™ Administrator's Guide

Release 8.0

Part No. A58229-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the chapter, section, and page number (if available).

You can send comments to us in the following ways

- electronic mail - infodev@us.oracle.com
- FAX - 650- 506-7226. Attn: Server Technologies Documentation Manager
- postal service
  Oracle Corporation
  500 Oracle Parkway
  Redwood City, CA 94065
  USA

If you would like a reply, please give your name, address, and telephone number below.

Part I

Oracle Advanced Networking Option Security and Single Sign-On

The following chapters of the *Oracle Advanced Networking Option Administrator's Guide* provide generic information on the security related features of the Advanced Networking Option.

- Chapter 1, "Network Security and Single Sign-On"
- Chapter 2, "Configuring Encryption and Checksumming"
- Chapter 3, "Configuring the CyberSAFE Authentication Adapter"
- Chapter 4, "Configuring the Kerberos Authentication Adapter"
- Chapter 5, "Configuring Oracle for Use with the SecurID Adapter"
- Chapter 7, "Choosing and Combining Authentication Services"
- Chapter 6, "Configuring and Using the Identix Biometric Authentication Adapter"
- Chapter 8, "Configuring the DCE GSSAPI Authentication Adapter"

Part I of this document includes information on how to configure security and authentication into your existing Net8 release 8.0.3 network. Refer also to the port-specific documentation on how to install and configure the Advanced Networking Option.

In addition to the features described in this section, the Oracle Advanced Networking Option includes the following feature:

- DCE Integration

  Refer to Part II "Oracle Advanced Networking Option and Oracle DCE Integration" for detailed information.

  The following chapters provide Oracle DCE Integration information:

- Chapter 9, “Overview of Oracle DCE Integration”
- Chapter 10, “Configuring DCE for Oracle DCE Integration”
- Chapter 11, “Configuring Oracle for Oracle DCE Integration”
- Chapter 12, “Connecting to an Oracle Database in DCE”
- Chapter 13, “DCE and Non-DCE Interoperability”

1

Network Security and Single Sign-On

The proliferation of distributed computing has been matched by an increase in the amount of information that organizations now place on computers. Employee records, financial records, product testing information, and other sensitive or critical data have moved from filing cabinets into file structures. The volume of critical or sensitive information on computers has increased the value of data that may be compromised, and the increase in distributed computing, in particular, has increased the vulnerability of this data.

The principal challenges in distributed environments are:

- *data integrity*—ensuring that data is not modified during transmission
- *data privacy*—ensuring that data is not disclosed during transmission
- *authentication*—having confidence that users', hosts', and clients' identities are correctly known
- *authorization*—giving permission to a user, program, or process to access an object or set of objects

The Oracle Advanced Networking Option ensures data integrity through cryptographic checksums using the MD5 algorithm. It also ensures data privacy through encryption. Release 8.0 provides 40-bit, 56-bit, and 128-bit RSA RC4 algorithms as well as 40-bit and 56-bit DES algorithms.

Establishing user identity is also of primary concern in distributed environments; otherwise, there can be little confidence in limiting privileges by user. For example, unless you have confidence in user authentication mechanisms, how can you be sure that user Smith connecting to Server A from Client B really *is* user Smith? Furthermore, you need to have confidence in the way clients and servers are made known to one another over the network, so that you have assurance not only that user Smith is who she says she is, but that Client B and Server A are also what they claim to be. The Oracle Advanced Networking Option release 8.0 provides this authentication ability through Oracle authentication adapters that support third-party authentication services such as Kerberos, CyberSAFE Challenger (a Kerberos-based authentication server), SecurID, and Identix TouchNet II. These adapters are described later in this chapter.

Note: User authentication and authorization are already standard features of Oracle8; however, they are significantly enhanced in the Oracle Advanced Networking Option release 8.0.

1.1 What's Covered in this Chapter

The first part of this chapter contains an introduction to the Oracle Advanced Networking Option encryption and checksumming features. These services are available to network products that use Net8, including the Oracle8 Server, Designer 2000, Developer 2000, and any other Oracle or third-party products that support Net8. For a comparison of the benefits of using one encryption algorithm over another, see Chapter 2.2, "Benefits of the Oracle Advanced Networking Option Encryption and Checksum Algorithms".

The second part of this chapter contains a discussion of how the Oracle Advanced Networking Option release 8.0 supports network user authentication in distributed environments through the use of Oracle authentication adapters.

1.2 Authentication Adapters Supported

For this release of the Oracle Advanced Networking Option, the following adapters are supported:

- Kerberos
- CyberSAFE Challenger
- SecurID
- Identix TouchNet II

This release of the documentation only provides configuration instructions for Kerberos, CyberSAFE Challenger, SecurID, and Identix authentication adapters.

1.2.1 System Requirements

The Oracle Advanced Networking Option is an add-on product to standard Net8 which makes getting Net8 licenses a prerequisite. The Oracle Advanced Networking Option is an extra cost item, and to be functional, must be purchased on both the client and the server.

The Oracle Advanced Networking Option must be installed with the Oracle Installer (tapes, CDs, and floppies) on all clients and servers where the Oracle Advanced Networking Option is required.

- The Oracle Advanced Networking Option release 8.0 work or later
- Oracle 8.0 or later

> **Note:** The Oracle Advanced Networking Option release 8.0 will provide secure communication when used with earlier releases (such as 1.0 and 1.1); however, the security functionality will default to that provided by the earlier release.

1.2.1.1 CyberSAFE Challenger Authentication Adapter Requirements

To use the CyberSAFE Challenger Authentication Adapter you need to have:

- CyberSAFE Application Security Toolkit version 1.0.4 or later
- This must be installed on both the machine that runs the Oracle client and on the machine that runs the Oracle server.
- CyberSAFE Challenger release 5.2.5 or later
- This must be installed on a physically secure machine that will run the authentication server.
- CyberSAFE Challenger Client
- This must be installed on the machine that runs the Oracle client.

1.2.1.2 Kerberos Authentication Adapter Requirements

To use the Kerberos Authentication Adapter you need to have:

- Kerberos 5.4.2

  The Kerberos authentication server must be installed on a physically secure machine.

1.2.1.3 SecurID Authentication Adapter Requirements

To use the SecurID Authentication Adapter you need to have:

- ACE/Server 1.2.4 or higher running on the authentication server.

1.2.1.4 Identix TouchNet II

To use the Identix TouchNet II Authentication Adapter you need to have:

- Identix hardware installed on each Biometric Manager station and client.
- Identix driver installed (it is supplied by both the Oracle Enterprise Manager and NT media.

1.3 Protection from Tampering and Unauthorized Viewing

Organizations around the world are deploying distributed databases and client/server applications in record numbers, often on a national or global scale, based on Net8 and the Oracle8 Server. Along with the increased distribution of data in these environments comes increased exposure to theft of data through eavesdropping. In Wide Area Network (WAN) environments, both public carriers and private network owners often route portions of their network through either insecure land lines or extremely vulnerable microwave and satellite links, leaving valuable data open to view for any interested party. In Local Area Network (LAN) environments within a building or campus, the potential exists for insiders with access to the physical wiring to view data not intended for them. Even more dangerous is the possibility that a malicious third party can execute a computer crime by actually tampering with data as it moves between sites. Oracle Advanced Networking Option protects against these possibilities in distributed environments containing confidential or otherwise sensitive data.

1.3.1 Verification of Data Integrity

To ensure that data has not been modified, deleted, or replayed during transmission, the Oracle Advanced Networking Option optionally generates a cryptographically secure message digest and includes it with each packet sent across the network.

1.3.2 High-Speed Global Data Encryption

To protect data from unauthorized viewing, the Oracle Advanced Networking Option includes an encryption module that uses the RSA Data Security RC4™ encryption algorithm. Using a secret, randomly-generated key for every session, all network traffic is fully safeguarded (including all data values, SQL statements, and stored procedure calls and results). The client, server, or both, can request or require the use of the encryption module to guarantee that data is protected. Oracle's optimized implementation provides a high degree of security for a minimal performance penalty. For the RC4 algorithm, Oracle provides encryption key lengths of 40 bits, 56 bits, and 128 bits.

Since the Oracle Advanced Networking Option RSA RC4 40-bit implementation meets the U.S. government export guidelines for encryption products, Oracle provides an export version of the media and exports it to all but a few countries, allowing most companies to safeguard their entire worldwide operations with this software.

1.3.3 Standards-Based Encryption

For financial institutions and other organizations that are required to use the U.S. Data Encryption Standard (DES), the Oracle Advanced Networking Option for Domestic Use offers a standard, optimized 56-bit key DES encryption algorithm. Due to current U.S. government export restrictions, standard DES is initially available only to customers located in the U.S.A. and Canada. For customers located outside the U.S.A. and Canada, the Oracle Advanced Networking Option for Export Use also offers DES40, a version of DES which combines the standard DES encryption algorithm with the international availability of a 40-bit key. Selecting the algorithm to use for network encryption is a user configuration option, allowing varying levels of security and performance for different types of data transfers.

1.3.4 Data Security Across Protocols

The Oracle Advanced Networking Option is fully supported by the Connection Manager, making secure data transfer a reality across network protocol boundaries. Clients using LAN protocols such as NetWare (SPX/IPX), for instance, can now securely share data with large servers using different network protocols such as LU6.2, TCP/IP, or DECnet. To eliminate potential weak points in the network infrastructure and to maximize performance, Connection Manager passes encrypted data from protocol to protocol without the cost and exposure of decryption and re-encryption.

1.3.5 The Oracle Advanced Networking Option is Not Yet Supported by Some Oracle Products

The Oracle Advanced Networking Option requires Net8 to transmit data securely. Accordingly, the Oracle Advanced Networking Option's authentication features are not currently supported by *some* parts of Oracle Financial, Human Resource, and Manufacturing Applications when they are running on the MS-Windows platform. The portions of these products that use Oracle Display Manager (ODM) can not yet take advantage of the Oracle Advanced Networking Option, since ODM does not currently use Net8. A maintenance version of Release 10 will allow the Oracle Advanced Networking Option to be used in all parts of these applications.

1.4 How Encryption and Checksumming are Activated

In any network connection, it is possible that both ends (client and server) may support more than one encryption algorithm and more than one cryptographic checksumming algorithm. When each connection is made, the server decides which algorithm to use, if any, based on which algorithms are available on each end of the connection and on what preferences have been specified in the Net8 configuration files.

When the server is trying to find a match between the algorithms it has made available and the algorithms the client has made available, it picks the first algorithm in its own list that also appears in the client's list. If one side of the connection does not specify a list of algorithms, all the algorithms that are installed on that side are acceptable.

1.4.1 Encryption and Checksumming Configuration

Encryption and checksumming parameters are defined by modifying a profile for the clients and servers on your network. Refer to Appendix A, "Encryption and Checksum Parameters" for an example of a profile (SQLNET.ORA) for the client and server nodes in a network using encryption and checksumming.

1.5 The Oracle Advanced Networking Option Provides Enhanced Client/ Server Authentication

Oracle servers and the Oracle Advanced Networking Option together provide the enhanced client/server authentication required in distributed, heterogeneous environments.

1.5.1 Why Single Sign-On?

In a distributed system, users may need to remember multiple passwords for the different applications and services that they use. To use a software development organization as an example, a developer may have access to an application in development on a workstation, a production system on a mini-computer, a PC for creating documents, and several mini-computers or workstations for testing, reporting bugs, configuration management, and so on. Administration of all these accounts and passwords is complex and time-consuming.

Users generally respond to multiple accounts in one of two ways: if they can choose their own passwords, they may standardize them so that they are the same on all machines (which results in a potentially large exposure in the event of a compromised password) or use passwords with slight variations (which may be easily guessed from knowing one password). Users with complex passwords may just write them down or forget them, either of which severely compromises password secrecy and service availability.

Providing a single sign-on, so that users can access multiple accounts and applications with a single password, eliminates the need for multiple passwords for users and simplifies management of user accounts and passwords for system administrators.

1.6 How Oracle Authentication Adapters Provide Enhanced Security

Among the types of authentication mechanisms that can be used in networked environments are the following:

- “Network Authentication Services” (such as the ACE/Server, Kerberos and CyberSAFE), provide secure, centralized authentication of users and servers.
- “Token Cards”. Token cards (such as SecurID) provide one-time passwords.
- “Biometric Authentication Adapter” provides centralized management of biometrically identified users and of database servers that authenticate them.

These authentication mechanisms are discussed in more detail in the following sections.

1.6.1 Network Authentication Services

In distributed environments, unless you can physically secure all connections in a network, which may be either physically or economically impossible, malefactors may hijack connections. For example, a transaction that should go from the Personnel system on Server A to the Payroll system on Server B may be intercepted in transit and routed instead to a terminal masquerading as Server B.

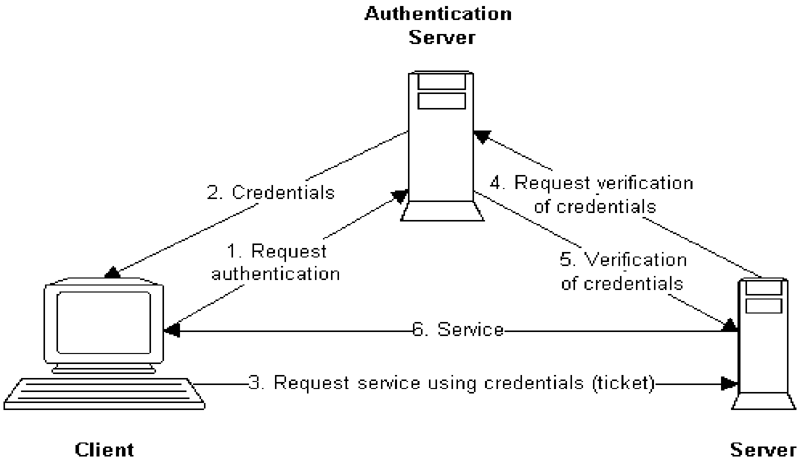
This threat may be addressed by having a central facility authenticate all members of the network (clients to servers, servers to servers, users to both clients and servers), rather than relying on parties identifying themselves to one another directly. By having a centralized, secure authentication service, you can have high confidence in the identity of users, clients, and servers in distributed environments. Network authentication services also can provide the benefit of single sign-on for users (refer to Section 1.5.1, “Why Single Sign-On?”).

1.6.2 Centralized Authentication

Figure 1–1, “How a Network Authentication Service Works” illustrates how a network authentication service typically operates, while the steps below describe each operation.

1. A user (client) requests authentication services, providing some identification that he is who he claims to be, such as a token or password.
2. After authenticating the user, the authentication server passes a ticket or credentials back to the client. (This ticket may include an expiration time.)
3. The client can now take these credentials and pass them to the server while asking for a service, such as connection to a database.
4. The server, to verify that the credentials are valid, sends them back to the authentication server.
5. If the authentication server accepts the credentials, it notifies the server.
6. The server provides the requested service to the user. If the credentials are not accepted, the requested service is denied.

Figure 1–1 How a Network Authentication Service Works

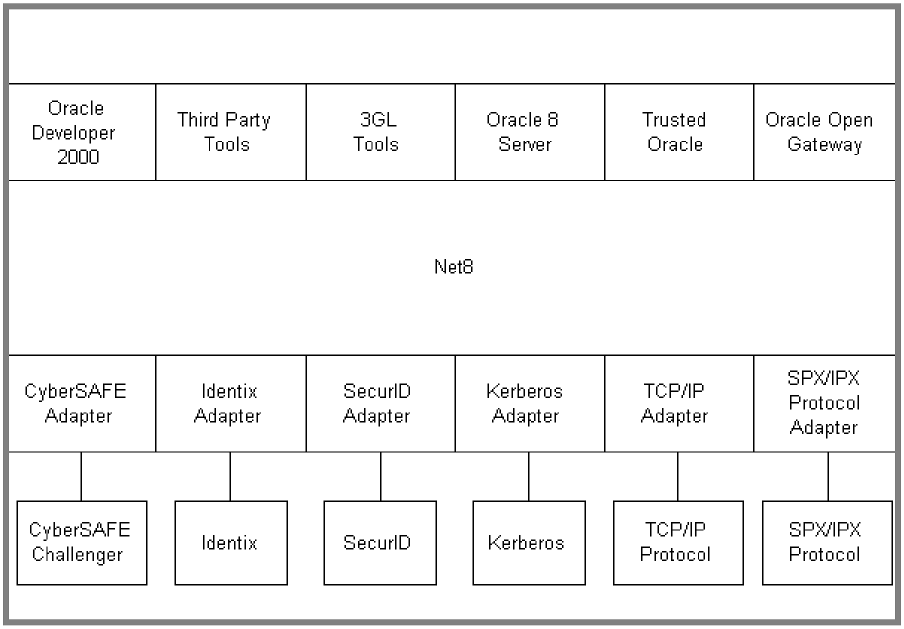

Authentication Server
2. Credentials
1. Request authentication
4. Request verification of credentials
5. Verification of credentials
6. Service
3. Request service using credentials (ticket)
Client
Server

1.6.3 Kerberos and CyberSAFE Support

Attention: The Oracle Authentication Adapter for Kerberos provides database link authentication (also called “proxy authentication”). CyberSAFE and SecurID do *not* provide support for proxy authentication.

The Oracle Advanced Networking Option support for Kerberos and CyberSAFE provides the benefits of single sign-on and centralized authentication in an Oracle environment. As shown in Figure 1–2, “Net8 with authentication adapters”, support for authentication services is provided through authentication adapters, which are very much like the existing Net8 protocol adapters. Authentication adapters integrate below the Net8 interface and allow existing applications to take advantage of new authentication systems transparently, without any changes to the application.

Figure 1–2 Net8 with authentication adapters



Kerberos is a trusted third-party authentication system that relies on shared secrets. It assumes that the third party is secure. It provides single sign-on capabilities, centralized password storage, database link authentication, and enhanced PC security.

Support for Kerberos is provided in the Oracle Advanced Networking Option in two ways:

- through the Kerberos Authentication Adapter
- through the CyberSAFE Challenger: an Authentication Adapter

Note: Oracle Corporation does not provide centralized authentication servers—only support for the authentication services provided through other vendors' security services or third-party Kerberos-based servers such as CyberSAFE. Oracle Corporation does provide a distributed authentication mechanism based on X.509 v1 certificates through the Oracle Security Server.

1.6.4 Token Cards

Token cards can provide improved ease-of-use for users through several different mechanisms. Some token cards offer one-time passwords that are synchronized with an authentication service. The server can verify the password provided by the smart card at any given time by contacting the authentication service. Other token cards operate on a challenge-response basis, in which the server offers a challenge (a number) and the user types the challenge into a token card, which provides another number (cryptographically-derived from the challenge), which the user then offers to the server.

Token cards provide the following benefits:

- ease of use for users, who need only remember, at most, a personal identification number (PIN) instead of multiple passwords
- ease of password management (one smart card rather than multiple passwords)
- enhanced password security, since to masquerade as a user, a malefactor would have to have the smart card as well as the PIN required to operate it
- enhanced accountability through a stronger authentication mechanism

1.6.5 SecurID Token Card

The Oracle Advanced Networking Option supports the Security Dynamics' SecurID card. SecurID provides two-factor user identification. Factor one is something the user knows: a PIN. The second factor is something the user possesses: the SecurID card. Single-use access codes change automatically every 60 seconds, and no two cards ever display the same number at the same time. The Oracle Advanced Networking Option support for SecurID provides the convenience of token cards in an Oracle environment.

1.6.6 Biometric Authentication Adapter

The Oracle Advanced Networking Option provides support for the Oracle Biometric Authentication adapter. Oracle Biometric Authentication adapters are used on both the clients and on the database servers to communicate biometric authentication data between the authentication server and the clients.

1.6.7 Oracle Parameters that Must be Configured for Network Authentication

For clients and servers to be able to use an Oracle Authentication Adapter, the following parameter must be in a profile:

```
SQLNET.AUTHENTICATION_SERVICES=(oracle_authent_adapter)
```

For example, the following parameter must be set in a profile on all clients and servers that use the Kerberos Authentication Adapter to authenticate users:

```
SQLNET.AUTHENTICATION_SERVICES=(KERBEROS5)
```

1.6.7.1 Set REMOTE_OS_AUTHENT to False

It is strongly recommended that when configuring the Oracle authentication adapters, you add the following parameter to the initialization file used for the database instance:

```
REMOTE_OS_AUTHENT=FALSE
```

Attention: Setting REMOTE_OS_AUTHENT to TRUE may create a security hole, because it allows someone using a non-secure protocol (for example, TCP) to perform an operating system-authorized login (formerly referred to as an OPS$ login).

If REMOTE_OS_AUTHENT is set to FALSE, and the server cannot support any of the authentication services requested by the client, the authentication service negotiation will fail, and the connection will be terminated.

If the following parameter is set in the SQLNET.ORA file on either the client or server side:

```
SQLNET.AUTHENTICATION_SERVICES=(NONE)
```

the database will attempt to use the provided username and password to log the user in. However, if REMOTE_OS_AUTHENT is set to FALSE, the connection will fail.

1.6.7.2 Set OS_AUTHENT_PREFIX to a Null Value

Authentication service-based user names can be long, and Oracle user names are limited to 30 characters. So, it is strongly recommended that you enter a null value for the OS_AUTHENT_PREFIX parameter in the initialization file used for the database instance:

```
OS_AUTHENT_PREFIX=“”
```

Note: The default value for OS_AUTHENT_PREFIX is OPS$; however, you can set it to any string.

Attention: If a database already has the OS_AUTHENT_PREFIX set to a value other than null (**““**) do not change it, since it could result in previously created externally-identified users not being able to connect to the Oracle server.

The command to create a user is:

```
create user <os_authent_prefix><username> identified externally;
```

When OS_AUTHENT_PREFIX is set to a null value (“”), you would create the user “king” with the following command:

```
create user king identified externally;
```

The advantage of creating a user in this way is that the administrator no longer needs to maintain different usernames for externally-identified users.

Note: This applies to creating Oracle users for use with all Oracle authentication adapters.

2

Configuring Encryption and Checksumming

This chapter includes the following sections:

- Section 2.1, “Where to Get Information on Installing the Oracle Advanced Networking Option”
- Section 2.2, “Benefits of the Oracle Advanced Networking Option Encryption and Checksum Algorithms”
- Section 2.3, “Diffie-Hellman-Based Key Management”
- Section 2.4, “Overview of Encryption and Checksumming Configuration Parameters”
- Section 2.5, “Using Oracle Net8 Assistant to Configure Servers and Clients to Use Encryption and Checksumming”

The configuration instructions assume that your Net8 network software has already been installed and is running. For more information about Net8, refer to the *Oracle Net8 Administrator’s Guide.*

> **Note:** Refer to Appendix A, “Encryption and Checksum Parameters” for examples of encryption and checksumming parameters in configuration files.

2.1 Where to Get Information on Installing the Oracle Advanced Networking Option

You can install and configure the Oracle Advanced Networking Option with other Oracle networking products and configure everything at once, or you can add the Oracle Advanced Networking Option to an already existing network.

This guide contains generic information on how to configure your already-existing Net8 network to use the Oracle Advanced Networking Option. It is meant to be used in conjunction with the guide that describes how to install and configure the Oracle Advanced Networking Option on your particular platform.

2.2 Benefits of the Oracle Advanced Networking Option Encryption and Checksum Algorithms

This release of the Oracle Advanced Networking Option provides support for 128-bit encryption with the RSA RC4 algorithm. This feature provides very strong encryption security for transmitted data. Following is a discussion of the benefits of using one algorithm over another.

2.2.1 DES Algorithm Provides Standards-Based Encryption

The Oracle Advanced Networking Option for Domestic Use provides the DES (Data Encryption Standard) algorithm for customers with specialized encryption needs. DES has been a U.S. government standard for many years and is sometimes mandated in the financial services industry. In most specialized banking systems today, DES is the algorithm used to protect large international monetary transactions. The Oracle Advanced Networking Option allows this high-security system to be used to protect any kind of application, without any custom programming.

In a secure cryptosystem, the plaintext (a message that has not been encrypted) can not be recovered from the ciphertext (the encrypted message) except by using the secret decryption key. In a "symmetric cryptosystem", a single key serves as both the encryption and the decryption key. DES is a secret-key, symmetric cryptosystem: both the sender and the receiver must know the same secret key, which is used both to encrypt and decrypt the message. DES is the most well-known and widely-used cryptosystem in the world. It has never been broken, despite the efforts of researchers over the last 15 years.

2.2.2 DES40 Algorithm is Provided for International Use

The DES40 algorithm, available internationally, is a variant of DES in which the secret key is preprocessed to provide 40 effective key bits. It is designed for use by customers outside the USA and Canada who want to use a DES-based encryption algorithm. This feature gives commercial customers a choice in the algorithm they use, regardless of their geographic location.

2.2.3 RSA RC4 is a Highly Secure, High Speed Algorithm

The RC4 algorithm, developed by RSA Data Security Inc., has quickly become the de-facto international standard for high-speed data encryption. Despite ongoing attempts by cryptographic researchers to "crack" the RC4 algorithm, the only feasible method of breaking its encryption known today remains brute-force, systematic guessing, which is generally infeasible. RC4 is a stream cipher that operates at several times the speed of DES, making it possible to encrypt even large bulk data transfers with minimal performance consequences.

2.2.4 RC4_56 and RC4_128 Can be Used by Domestic Customers

RC4 is a variable key-length stream cipher. The Oracle Advanced Networking Option for Domestic Use, release 8.0, offers an implementation of RC4 with 56 bit and 128 bit key lengths. This provides strong encryption with no sacrifice in performance when compared to other key lengths of the same algorithm.

2.2.5 RC4_40 Can be Used by Customers Outside the US and Canada

Oracle has obtained special license to export the RC4 data encryption algorithm with a 40-bit key size to virtually all destinations where other Oracle products are available. This makes it possible for international corporations to safeguard their entire operations with fast, strong cryptography.

2.3 Diffie-Hellman-Based Key Management

The secrecy of encrypted data is dependent on the existence of a secret key shared between the communicating parties. Providing and maintaining such secret keys is known as "key management". In a multi-user environment, secure key distribution may be difficult; public-key cryptography was invented to solve this problem. The Oracle Advanced Networking Option uses the public-key based Diffie-Hellman key negotiation algorithm to perform secure key distribution for both encryption and crypto-checksumming.

When encryption is used to protect the security of encrypted data, keys should be changed frequently to minimize the effects of a compromised key. For this reason, the Oracle Advanced Networking Option key management facility changes the session key with every session.

2.3.1 Overview of Site-Specific Diffie-Hellman Encryption Enhancement

The Oracle Advanced Networking Option includes the Diffie-Hellman key negotiation algorithm to choose keys used both for encryption and for crypto-checksumming.

A key is a secret shared by both sides of the connection and by no one else. Without the key, it is extremely difficult to decrypt an encrypted message or to tamper undetectably with a crypto-checksummed message. Diffie-Hellman is subject to a particular computationally-expensive table-based attack. Site-specific Diffie-Hellman, on the other hand, lowers the effectiveness of this attack by enabling the Diffie-Hellman parameters at each site to be changed frequently.

The system administrator can lessen the consequences of this attack by running a parameter generation program called naegen to change the default Diffie-Hellman parameters. The Oracle Advanced Networking Option server will then use the modified parameters to establish a Diffie-Hellman session key with the Oracle Advanced Networking Option client. If the Diffie-Hellman parameters do not exist, the Oracle Advanced Networking Option server will use its default parameters.

2.3.1.1 How to Generate the Diffie-Hellman Parameters with naegen

You can use the naegen utility to generate the new Diffie-Hellman parameters. naegen takes as an argument either zero or an integer argument in the range of 256 to 512. For example:

```
naegen 300
```

This argument represents the number of bits in those parameters. If you do not provide an argument to `naegen`, `naegen` generates 512-bit parameters. If a number lower than 256 is provided as the argument, `naegen` will generate 256-bit parameters. Once it has generated the parameters, `naegen` stores them in `snsdh.ora` which is then read by the Oracle Advanced Networking Option server to be used in key negotiation. Note that every time the administrator runs naegen, the values in the snsdh.ora file will be different.

Note: The naegen utility uses the snsdh.ora parameter file, whose location may vary depending on your platform. For example, the default file location for the snsdh.ora file on the UNIX platform is $ORACLE_HOME/network/admin.

If you are using a 40-bit key such as that used by RC4_40, you should provide naegen argument of 300 or greater. If you are using a 56-bit key such as DES, you should provide an argument of 512.

Although using different Diffie-Hellman parameters for each connection is preferred for better security, it is not feasible because naegen can take up to 4 minutes to generate the necessary parameters, depending on the parameter size. Therefore, it is recommended that network administrators generate the parameters once a day. Optionally, you could generate the parameters once a week or once a month.

2.3.2 Overview of Authentication Key Fold-in Encryption Enhancement

The purpose of the Authentication Key Fold-in encryption enhancement is to defeat a possible “middle-man attack” on the Diffie-Hellman key negotiation. It strengthens the session key significantly by combining a shared secret (which is known only to both the client and the server), with the original session key negotiated by Diffie-Hellman.

The client and the server begin communicating using the session key generated by Diffie-Hellman. When the client authenticates itself to the server, there is a shared secret that is only known to both sides. The Oracle Advanced Networking Option then combines the shared secret and Diffie-Hellman session key to generate a stronger session key that would defeat the middle-man, who has no way of knowing the shared secret.

2.3.2.1 Authentication Key Fold-in Feature Requires no Configuration

The authentication key fold-in encryption enhancement feature is included in the Oracle Advanced Networking Option and requires no configuration by the system or network administrator.

2.3.3 The MD5 Message Digest Algorithm

Encryption of network data provides data privacy, so no unauthorized party is able to view the plaintext data as it passes over the network. The Oracle Advanced Networking Option also provides protection against two other forms of attack: Data Modification Attack and Replay Attack.

In a data modification attack, an unauthorized party on the network intercepts data in transit and changes portions of that data before retransmitting it. An example of this would be to change the dollar amount of a banking transaction.

In a replay attack, an entire set of valid data is repeatedly interjected onto the network. An example would be to repeat a valid bank account transfer transaction.

The Oracle Advanced Networking Option uses a keyed, sequenced implementation of the MD5 message digest algorithm to protect against both of these forms of active attack. This protection is activated independently from the encryption features provided.

2.3.4 Domestic and Export Versions

Due to export controls placed on encryption technology, the Oracle Advanced Networking Option is available in two versions: an Export version and a Domestic version.

The Oracle Advanced Networking Option for Export Use contains the Diffie-Hellman key negotiation algorithm, MD5 message digest algorithm, and DES40 and RC4_40 encryption algorithms.

The Oracle Advanced Networking Option for Domestic Use contains the Diffie-Hellman key negotiation algorithm, MD5 message digest algorithm, and DES40, DES, RC4_40, RC4_56, and RC4_128 encryption algorithms.

In certain circumstances, a special license may be obtained to export the domestic version. Licenses are generally available to wholly owned subsidiaries of US corporations. Special licenses can be obtained to allow banks to have the export version updated to include DES. Export and import regulations vary from country to country and change from time to time, so it is important to check on current restrictions in your area.

2.4 Overview of Encryption and Checksumming Configuration Parameters

As a network administrator, you set the encryption and checksumming configuration parameters. For information about configuring your existing Net8 network to use the Oracle Advanced Networking Option, see Section 2.5, “Using Oracle Net8 Assistant to Configure Servers and Clients to Use Encryption and Checksumming”.

The profile (SQLNET.ORA) on clients and servers using encryption and checksumming must contain some or all of the parameters listed below. See Appendix A, “Encryption and Checksum Parameters” for sample SQLNET.ORA configuration files for clients or servers using the Oracle Advanced Networking Option.

2.4.1 Negotiating Encryption and Checksumming

To negotiate whether to turn on encryption or checksumming, you can specify four possible values for four of the Oracle Advanced Networking Option configuration parameters: “ACCEPTED”, “REJECTED”, “REQUESTED”, “REQUIRED”. Each of these four values is listed below followed by a brief one-sentence explanation. This explanation is followed by a paragraph or two containing detailed explanations of its meaning and behavior. The default value for these four parameters is ACCEPTED. If you do not specify a value for a parameter, it defaults to ACCEPTED.

ACCEPTED

Turn on the security service if the other side wants it.

My side of the connection does not desire the security service, but it will be allowed if the other side asks with a setting of REQUIRED or REQUESTED. If the other side is set to REQUIRED or REQUESTED, and an algorithm match is found, the connection will continue without error and with the security service turned on. If the other side is set to REQUIRED and no algorithm match is found, the connection will terminate with error message ORA-12650.

If the other side is set to REQUESTED and no algorithm match is found, or if the other side is set to ACCEPTED or REJECTED, the connection will continue without error and without the security service enabled.

REJECTED

Do not turn on the security service even if the other side wants it.

My side of the connection specifies that the security service is not allowed. If the other side specifies REQUIRED, the connection will terminate with error message ORA-12650. If the other side is set to REQUESTED, ACCEPTED, or REJECTED, the connection will continue without error and without the security service enabled.

REQUESTED

Turn on the security service if the other side allows it.

My side of the connection specifies that the security service is desired, but not required. The security service will be active if the other side specifies ACCEPTED, REQUESTED, or REQUIRED. There must be a matching algorithm available on the other side, otherwise the service will not be activated. If the other side specifies REQUIRED and there is no matching algorithm, the connection fails.

REQUIRED

Turn on the security service or do not make the connection.

My side of the connection specifies that the security service must be activated. The connection will fail if the other side specifies REJECTED or if there is no compatible algorithm on the other side.

Table 2–1, "Encryption and Checksumming Negotiation Scheme" below shows whether or not the security service will be turned on based on a combination of client and server configuration parameters. If either the server or client has specified REQUIRED, lack of a common algorithm will cause the connection to fail. Otherwise, if the service would be on, lack of a common service algorithm will result in the service being turned off.

Table 2–1 Encryption and Checksumming Negotiation Scheme

			Client		
		Accepted	*Rejected*	*Requested*	*Required*
Server	*Accepted*	**OFF**	**OFF**	**ON**	**ON**
	Rejected	**OFF**	**OFF**	**OFF**	**Connection will fail**
	Requested	**ON**	**OFF**	**ON**	**ON**
	Required	**ON**	**Connection will fail**	**ON**	**ON**

Note: If Table 2-1, indicates that a service is ON, but a common algorithm is not available to perform the service, the service will not be used. In this case, if either side had specified that the service was REQUIRED, the connection will fail.

2.4.2 What the Encryption and Checksumming Parameters Do

There are nine parameters used to turn on encryption and checksumming. These parameters are described in the following sections:

Section 2.4.2.1, "Server Encryption Level Setting"

Section 2.4.2.2, "Client Encryption Level Setting"

Section 2.4.2.3, "Server Encryption Selected List"

Section 2.4.2.4, "Client Encryption Selected List"

Section 2.4.2.5, "Server Checksum Level Setting"

Section 2.4.2.6, "Client Checksum Level Setting"

Section 2.4.2.7, "Server Checksum Selected List"

Section 2.4.2.8, "Client Checksum Selected List"

Section 2.4.2.9, "Client Profile Encryption"

Note: Use the Oracle Net8 Assistant to edit the SQLNET.ORA file.

Refer to Section 2.4.1, "Negotiating Encryption and Checksumming" for descriptions of the possible values you can specify for the four level setting parameters listed above.

2.4.2.1 Server Encryption Level Setting

```
SQLNET.ENCRYPTION_SERVER = valid_value
```

This parameter specifies the desired behavior when a client (or a server acting as a client) is connecting to this server. The behavior of the server will depend in part on the SQLNET.ENCRYPTION_CLIENT setting at the other end.

Possible values: ACCEPTED, REJECTED, REQUESTED, REQUIRED

Default value: ACCEPTED

2.4.2.2 Client Encryption Level Setting

```
SQLNET.ENCRYPTION_CLIENT = valid_value
```

This parameter specifies the desired behavior when this client (or this server acting as a client) is connecting to a server. The behavior of the client will depend in part on the value set for SQLNET.ENCRYPTION_SERVER at the other end of the connection.

Possible values: ACCEPTED, REJECTED, REQUESTED, REQUIRED

Default value: ACCEPTED

2.4.2.3 Server Encryption Selected List

```
SQLNET.ENCRYPTION_TYPES_SERVER = (valid_encryption_algorithm
[,valid_encryption_algorithm])
```

where *valid_encryption_algorithm* is one of the following:

RC4_40	RSA RC4 (40-bit key size)
	Domestic & International
RC4_56	RSA RC4 (56-bit key size)
	Domestic only
RC4_128	RSA RC4 (128-bit key size)
	Domestic only
DES	Standard DES (56-bit key size)
	Domestic only
DES40	DES40 (40-bit key size)
	Domestic & International

This parameter specifies a list of encryption algorithms this server is allowed to use when acting as a server in the order of desired use. Type the most desired algorithm first. This list is used to negotiate a mutually acceptable algorithm with the other end of the connection. Each algorithm will be checked against the list of client algorithm types available until a match is found. If an algorithm that is not installed is specified on this side, the connection will terminate with error message ORA-12650.

Default value: All installed algorithms will be used in a negotiation if no algorithms are defined in the SQLNET.ORA file.

Domestic version: If you are using the Domestic version, all five algorithms are installed: RC4_40, RC4_56, RC4_128, DES, and DES40. If no algorithms are specified, the installed algorithms will be used in that order to negotiate a mutually acceptable algorithm with the other end of the connection.

Export version: If you are using the Export version, the following algorithms are installed: RC4_40 and DES40. If no algorithms are specified, the installed algorithms will be used in that order to negotiate a mutually acceptable algorithm.

You can specify multiple encryption algorithms, that is, either a single value or a list of algorithm names. For example, either of the following encryption parameters is acceptable:

```
SQLNET.ENCRYPTION_TYPES_SERVER=(RC4_40)
SQLNET.ENCRYPTION_TYPES_SERVER=(DES,RC4_56,RC4_128,DES40)
```

2.4.2.4 Client Encryption Selected List

```
SQLNET.ENCRYPTION_TYPES_CLIENT = (valid_encryption_algorithm
[,valid_encryption_algorithm])
```

where *valid_encryption_algorithm* is one of the following:

RC4_40	RSA RC4 (40-bit key size) Domestic & International
RC4_56	RSA RC4 (56-bit key size) Domestic only
RC4_128	RSA RC4 (128-bit key size) Domestic only
DES	Standard DES (56-bit key size) Domestic only
DES40	DES40 (40-bit key size) Domestic & International

This parameter specifies a list of encryption algorithms this client (or this server acting as a client) is allowed to use when connecting to a server. This list is used to negotiate a mutually acceptable algorithm with the other end of the connection. The parameters can be listed in any order. If an algorithm that is not installed is specified on this side, the connection will terminate with error message ORA-12650.

Default value: All installed algorithms will be used if no algorithms are defined in the SQLNET.ORA file.

Domestic version: If you are using the Domestic version, all five algorithms are installed: RC4_40, RC4_56, RC4_128, DES, and DES40. If no algorithms are defined in the SQLNET.ORA file, the installed algorithms will be used in that order to negotiate a mutually acceptable algorithm with the other end of the connection.

Export version: If you are using the Export version, the following algorithms are installed: RC4_40 and DES40. If no algorithms are defined in the SQLNET.ORA file, the installed algorithms will be used in that order to negotiate a mutually acceptable algorithm.

You can specify multiple encryption algorithms, that is, either a single value or a list of algorithm names. For example, either of the following encryption parameters is acceptable:

```
SQLNET.ENCRYPTION_TYPES_CLIENT=(DES,DES40,RC4_56,RC4_40)
SQLNET.ENCRYPTION_TYPES_CLIENT=(RC4_40)
```

2.4.2.5 Server Checksum Level Setting

```
SQLNET.CRYPTO_CHECKSUM_SERVER = valid_value
```

This parameter specifies the desired checksum behavior when a client (or another server acting as a client) is connecting to this server. The resulting behavior will depend in part on the SQLNET.CRYPTO_CHECKSUM_CLIENT setting at the other end.

Possible values: ACCEPTED, REJECTED, REQUESTED, REQUIRED

Default value: ACCEPTED

2.4.2.6 Client Checksum Level Setting

```
SQLNET.CRYPTO_CHECKSUM_CLIENT = valid_value
```

This parameter specifies the desired checksum behavior when this client (or this server acting as a client) is connecting to a server. The resulting behavior will depend in part on the SQLNET.CRYPTO_CHECKSUM_SERVER setting at the other end of the connection.

Possible values: ACCEPTED, REJECTED, REQUESTED, REQUIRED

Default value: ACCEPTED

2.4.2.7 Server Checksum Selected List

```
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER = (crypto_checksum_algorithm)
```

where *crypto_checksum_algorithm* is:

```
MD5
```

Currently, the only supported crypto-checksum algorithm choice is RSA Data Security's MD5 algorithm. Other algorithms may be supported in future releases.

This parameter specifies a list of the checksumming algorithms this server is allowed to use, in order of desired use with the most desired algorithm first, when acting as a server to a client or another server. This list is used to negotiate a mutually acceptable algorithm with the remote end. Each algorithm will be checked against the list of client algorithm types available until a match is found. The first algorithm match will be the one that is used. If an algorithm is specified that is not installed on this side, the connection will terminate with error message ORA-12650.

Default value: MD5 (currently the only valid value)

2.4.2.8 Client Checksum Selected List

Currently, the only supported crypto-checksum algorithm choice is RSA Data Security's MD5 algorithm. Other algorithms may be supported in future releases.

This parameter specifies a list of checksumming algorithms this client (or this server acting as a client) is allowed to use when connecting to a server. This list is used to negotiate a mutually acceptable algorithm with the remote end. The order in which the algorithms are listed is not important. If an algorithm that is not installed on this side is specified, the connection will terminate with error message ORA-12650.

Default value: MD5 (currently the only valid value)

2.4.2.9 Client Profile Encryption

```
SQLNET.CRYPTO_SEED = "10-70 random characters"
```

The characters that form the value for this parameter are used when generating cryptographic keys. The more random the characters entered into this field are, the stronger the keys are. You set this parameter by entering from 10 to 70 random characters into the above statement.

> **Note:** It is recommended that you enter as many characters as possible (up to 70), since the resulting key will be more random and therefore stronger.

This parameter must be present in the profile (SQLNET.ORA) whenever encryption or checksumming is turned on.

2.5 Using Oracle Net8 Assistant to Configure Servers and Clients to Use Encryption and Checksumming

Use the Oracle Net8 Assistant to configure clients and servers in your network to use encryption and checksumming. Oracle Net8 Assistant automatically updates your profile (SQLNET.ORA) with the parameters you select. Refer to the Oracle Net8 Assistant HELP system for detailed configuration information. Refer to Appendix A, “Encryption and Checksum Parameters” for sample configuration files using encryption and checksumming.

2.5.1 Configure Servers and Clients to Use Encryption

The following instructions assume you are using the Oracle Net8 Assistant configuration tool. Configure Servers to use encryption as follows. Refer to Figure 2–1, “Oracle Net8 Assistant Profile Encryption Tab”.

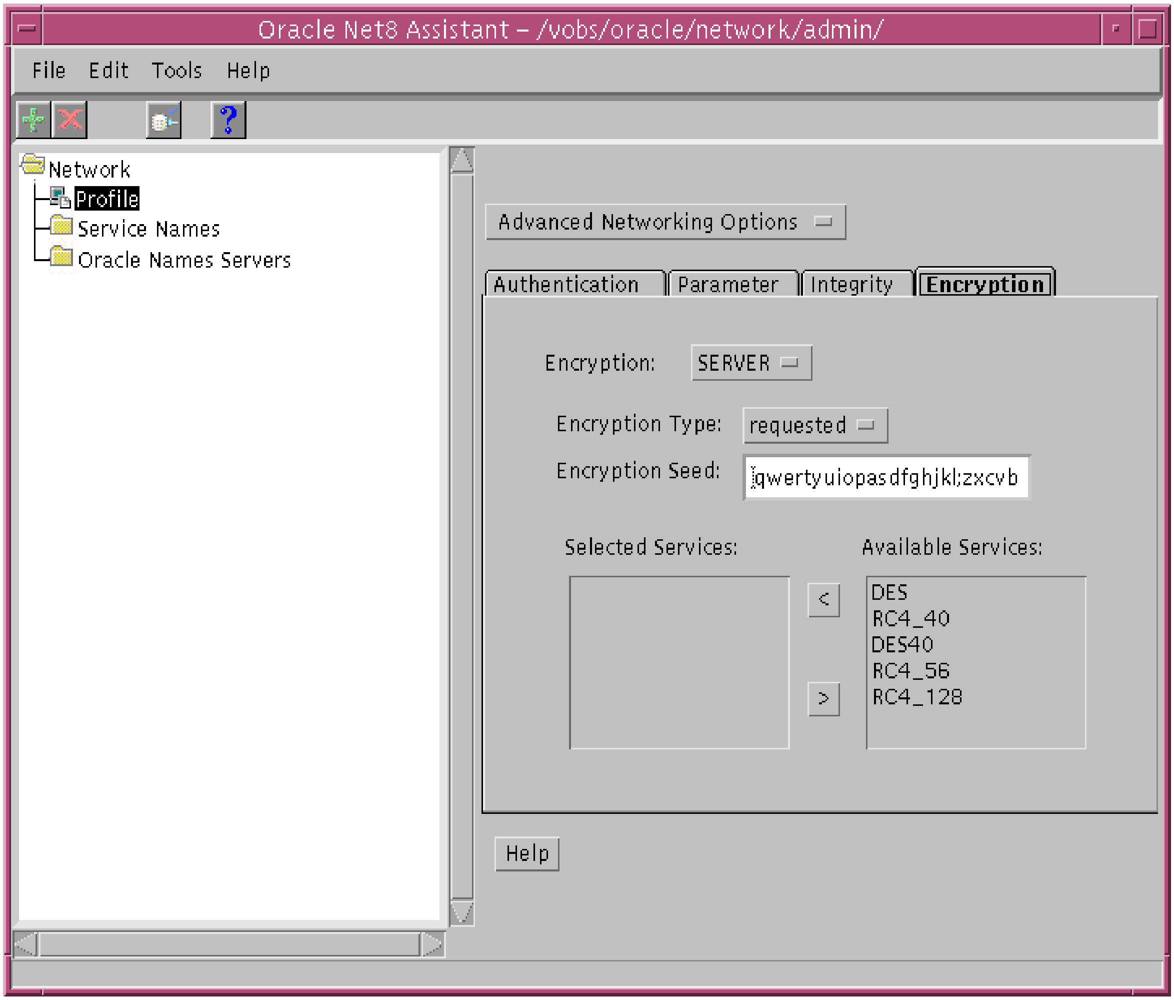
1. Click the Profile folder.
2. Select Advanced Networking Options from the drop-down list box.
3. Click the Encryption tab.
4. Click the Encryption drop-down list box, and click SERVER.
5. Click the Encryption Type drop-down list box, and click one of the following values: requested, required, accepted, rejected.

6. Type between 10 and 70 random characters for the Encryption Seed.
7. Move services to and from the Available Services and Selected Services lists by selecting a service and clicking the arrow keys.

Configure Clients to use encryption as follows:

1. Click the Profile folder.
2. Select Advanced Networking Options from the drop-down list box.
3. Click the Encryption tab.
4. Click the Encryption drop-down list box, and click CLIENT.
5. Click the Encryption Type drop-down list box, and click one of the following values: requested, required, accepted, rejected.
6. Type between 10 and 70 random characters for the Encryption Seed.
7. Move services to and from the Available Services and Selected Services lists by selecting a service and clicking the arrow keys.

Figure 2–1 Oracle Net8 Assistant Profile Encryption Tab



Encryption: SERVER

Encryption Type: requested

Encryption Seed: qwertyuiopasdfghjkl;zxcvb

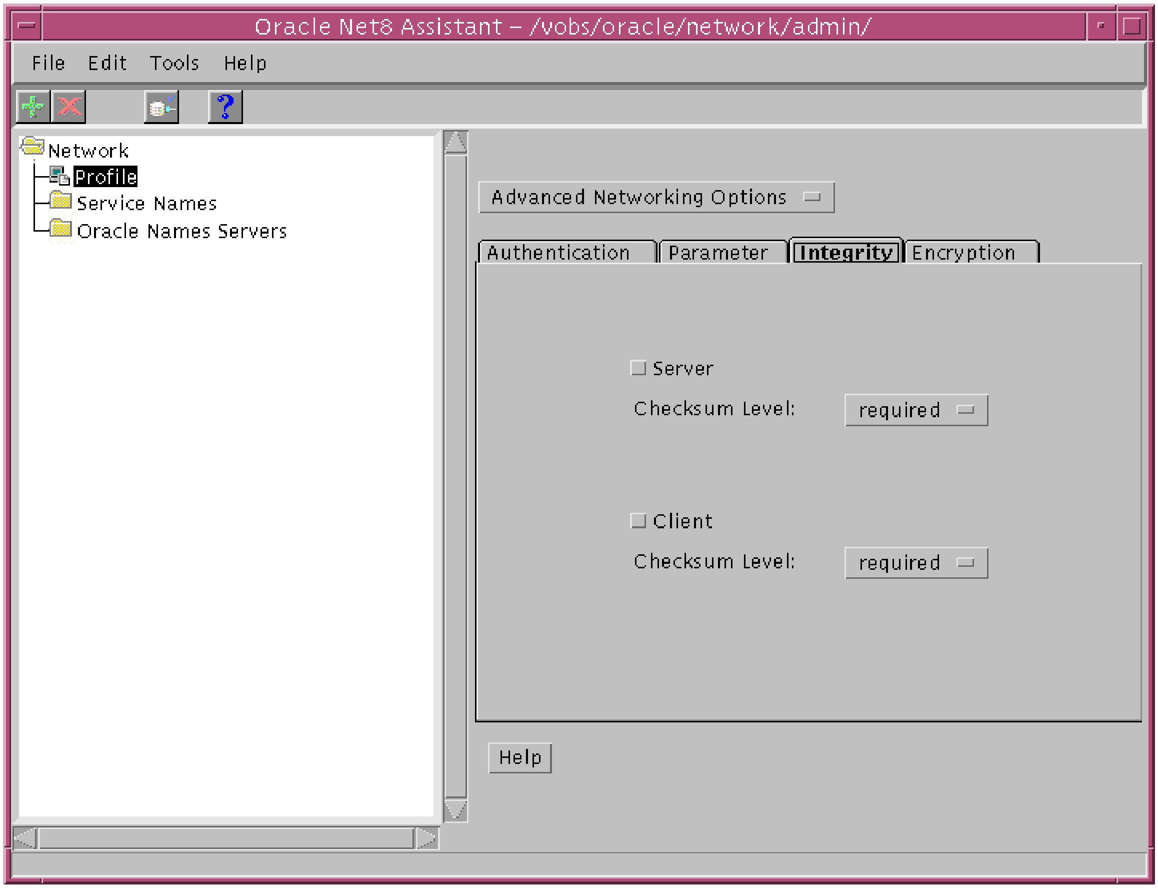
Available Services:
DES
RC4_40
DES40
RC4_56
RC4_128

2.5.2 Configure Servers and Clients to Use Checksumming

Configure Servers and Clients to use checksumming as follows. Refer to Figure 2–2, “Oracle Net8 Assistant Profile Integrity Tab”.

1. Click the Integrity folder.
2. Click the Server radio button to configure Server checksumming.
3. Click the Checksum Level drop-down list box to select one of the following server checksum level values: required, requested, accepted, rejected.
4. Click the Client radio button to configure Client checksumming.
5. Click the Client Checksum Level drop-down list box to select one of the following client checksum level values: required, requested, accepted, rejected.

Figure 2–2 Oracle Net8 Assistant Profile Integrity Tab



□ Server

Checksum Level: required

□ Client

Checksum Level: required

3 Configuring the CyberSAFE Authentication Adapter

This chapter contains information on how to configure Oracle for use with CyberSAFE, as well as a brief overview of the steps you need to follow to configure CyberSAFE to authenticate Oracle users. This information includes the following:

- Section 3.1, "Steps to Perform to Enable CyberSAFE Authentication"
- Section 3.2, "CyberSAFE Configuration Parameters Required on the Oracle Server and Client"
- Section 3.3, "Troubleshooting the Configuration of the CyberSAFE Authentication Adapter"

3.1 Steps to Perform to Enable CyberSAFE Authentication

This section contains information on the following tasks:

Important: You should perform these tasks in the order listed.

1. ”Install the CyberSAFE Server on the Machine that will Act as the Authentication Server”
2. ”Install the CyberSAFE Challenger Client on the Same Machine that Runs the Oracle Server and the Client”
3. ”Install the CyberSAFE Application Security Toolkit on the Client and on the Server”
4. ”Configure a Service Principal for an Oracle Server”
5. ”Extract the Service Table from CyberSAFE”
6. ”Install an Oracle Server”
7. ”Install the Oracle Advanced Networking Option”
8. ”Configure Net8 and Oracle8 on your Server and Client”
9. ”Configure the CyberSAFE Authentication Adapter using the Net8 Assistant”
10. ”Create a CyberSAFE User on the Authentication Server”
11. ”Create an Externally Authenticated Oracle User on the Oracle Server”
12. ”Use kinit on the Client to Get the Initial Ticket for the Kerberos / Oracle User”
13. ”Connect to an Oracle Server Authenticated by CyberSAFE”

3.1.1 Install the CyberSAFE Server on the Machine that will Act as the Authentication Server

For information on how to install the CyberSAFE Challenger Master Server on your machine, refer to the CyberSAFE documentation listed in the ”Related Publications” section of the Preface of this guide.

3.1.2 Install the CyberSAFE Challenger Client on the Same Machine that Runs the Oracle Server and the Client

For information on installing the CyberSAFE Challenger Client on clients, refer to the CyberSAFE documentation listed in the ”Related Publications” section of the Preface of this guide.

3.1.3 Install the CyberSAFE Application Security Toolkit on the Client and on the Server

Install the CyberSAFE Application Security Toolkit on the Oracle client and Oracle server machines.

3.1.4 Configure a Service Principal for an Oracle Server

For the Oracle server to validate the identity of clients, you need to configure a service principal for an Oracle server on the machine running the CyberSAFE Challenger Master Server. Also configure a realm if necessary.

The name of the principal should have the following format:

```
kservice/kinstance@REALM
```

where kservice is a string that represents the Oracle service. This may or may not be the same as the database service name; kinstance is typically the fully-qualified name of the machine on which Oracle is running, and *REALM* is the domain of the server.

Note: kservice is case-sensitive, and REALM must always be capitalized.

Note: The utility names in this section are actual programs that you run. However, the CyberSAFE user name “cyberuser” and realm “SOMECO.COM” are examples only—the actual names will vary.

For example, if kservice is "oracle", and the fully-qualified name of the machine on which Oracle is running is "dbserver.someco.com", and the realm is "SOMECO.COM", the principal name would be:

```
oracle/dbserver.someco.com@SOMECO.COM
```

Note: It is a common convention to use the DNS domain name as the name of the realm.

Run kdb5_edit as root to create the service principal.

```
# cd /krb5/admin
# ./kdb5_edit
```

To add a principal called "oracle / dbserver.someco.com@SOMECO.COM" to the list of server principals known by CyberSAFE, from kdb5_edit type the following:

```
kdb5_edit:  ark oracle/dbserver.someco.com@SOMECO.COM
```

3.1.5 Extract the Service Table from CyberSAFE

You need to extract a service table from CyberSAFE and copy it to both the Oracle server and CyberSAFE Challenger client machines. For example, to extract a service table for dbserver.someco.com, type the following from kdb5_edit:

```
kdb5_edit:  xst dbserver.someco.com oracle
'oracle/dbserver.someco.com@SOMECO.COM' added to keytab
'WRFILE:dbserver.someco.com-new-srvtab'
kdb5_edit:  exit
# /krb5/bin/klist -k -t dbserver.someco.com-new-srvtab
```

Note: If you do not enter a REALM (in the example, SOMECO.COM) when using *xst*, kdb5_edit uses the realm of the current host and displays it in the command output, as shown above.

After the service table has been extracted, verify that the new entries are in the table in addition to the old entries. If the new entries are not in the service table, or if you need to add additional new entries, use kdb5_edit to append the additional entries.

At this point, you need to move the CyberSAFE service table to the CyberSAFE Challenger client machine. If the service table is on the same machine as the CyberSAFE client, you can simply move it (using a command such as that shown below). If the service table is on a different machine from the CyberSAFE Challenger client, you must transfer the file with a program like FTP. For example, to move it, type the following:

```
# mv dbserver.someco.com-new-srvtab /krb5/v5srvtab
```

Remember to transfer the file in binary mode when you use FTP.

3.1.5.1 Ensure that the Oracle Server Can Read the Service Table

Make sure that the owner of the Oracle Server executable can read the service table (in the previous example, /krb5/v5srvtab). Set the file owner to the Oracle user or make the file readable by the group to which Oracle belongs. Do not make the file readable to all users, since this would allow a security breach.

3.1.6 Install an Oracle Server

Install an Oracle server on the same machine that is running the CyberSAFE Challenger client. Refer to your operating system-specific documentation for information.

3.1.7 Install the Oracle Advanced Networking Option

Install the Oracle Advanced Networking Option on your Oracle client and Oracle server machines. Refer to your operating system-specific documentation.

3.1.8 Configure Net8 and Oracle8 on your Server and Client

For information on how to configure Net8 and Oracle8 on servers and clients, see your operating system-specific documentation.

3.1.9 Configure the CyberSAFE Authentication Adapter using the Net8 Assistant

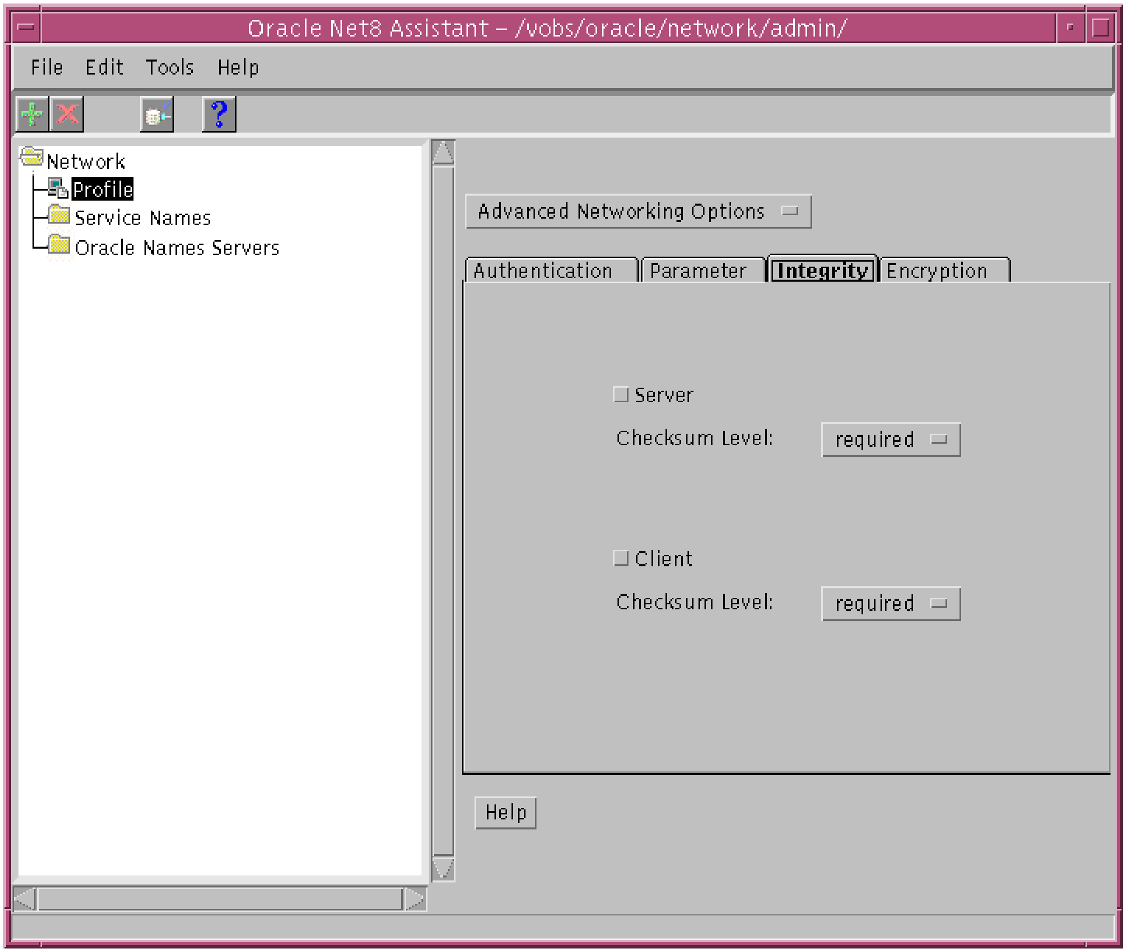
The following steps show you how to use the Net8 Assistant to configure the CyberSAFE authentication adapter. Refer also to the Net8 Assistant on-line HELP system for instructions on how to configure the CyberSAFE Authentication adapter.

Configure Clients, and Servers, to use encryption as follows. Refer to Figure 3–1, “Oracle Net8 Assistant Profile Encryption Tab”.

1. Click the Profile folder.
2. Select Advanced Networking Options from the drop-down list box.
3. Click the Encryption tab.

4. Click the Encryption drop-down list box, and click CLIENT or SERVER.
5. Click the Encryption Type drop-down list box, and click one of the following values: requested, required, accepted, rejected.
6. Type between 10 and 70 random characters for the Encryption Seed.
7. Move services to and from the Available Services and Selected Services lists by selecting a service and clicking the arrow keys.

Figure 3–1 Oracle Net8 Assistant Profile Encryption Tab



□ Server

Checksum Level: required

□ Client

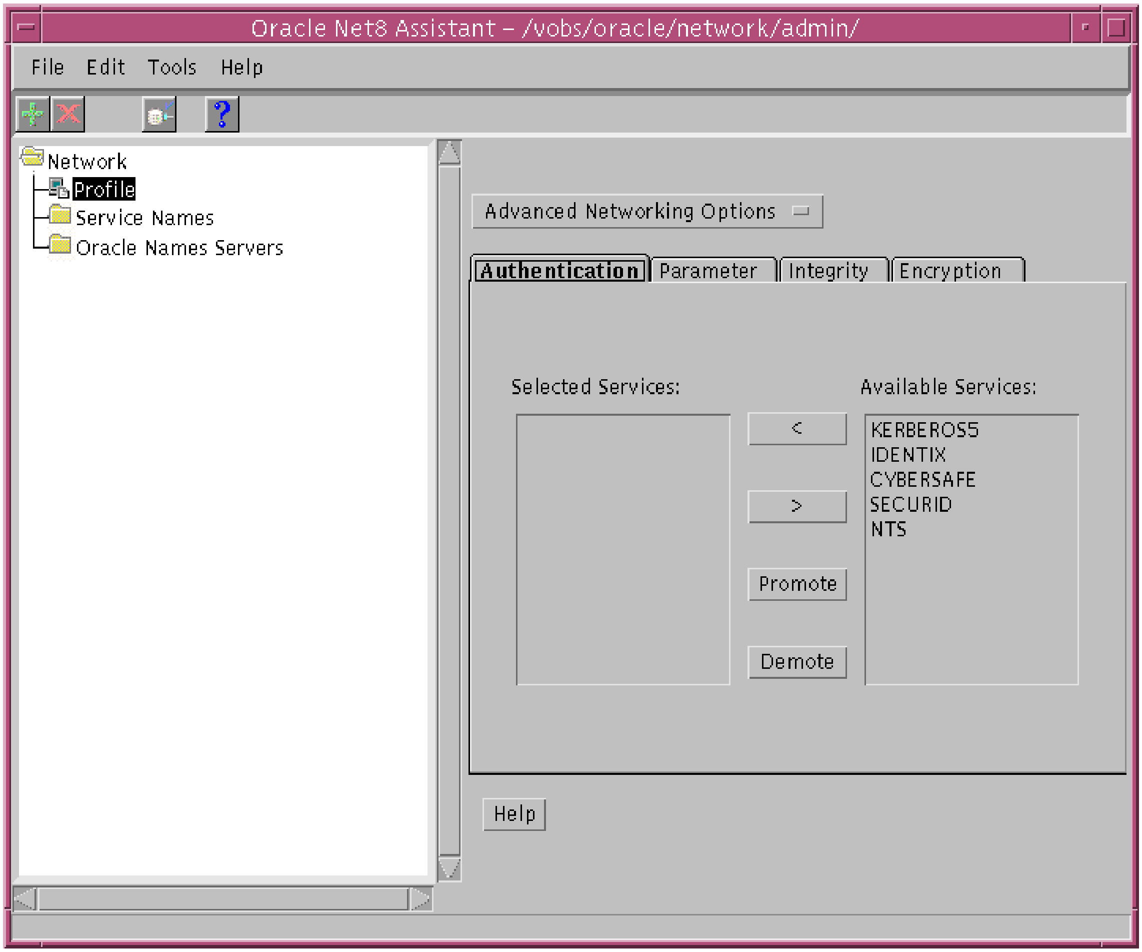
Checksum Level: required

Next, you must configure an authentication service on your network. Refer to Figure 3–2, “Oracle Net8 Assistant Profile Authentication Tab”.

1. Click the Profile folder.
2. Click the Authentication tab.

3. Click to select the authentication service you want from the Available Services list.
4. Click the [<] button to move the service over to the Selected Services list.
5. Repeat steps 4 and 5, above, until you have selected all of your required authentication services.
6. Arrange the selected services in order of desired use. Click on a service to select it, then click [Promote] or [Demote] to arrange the services in the list. For example, put SECURID at the top of the list if you want that service to be the first one used.

Figure 3–2 Oracle Net8 Assistant Profile Authentication Tab



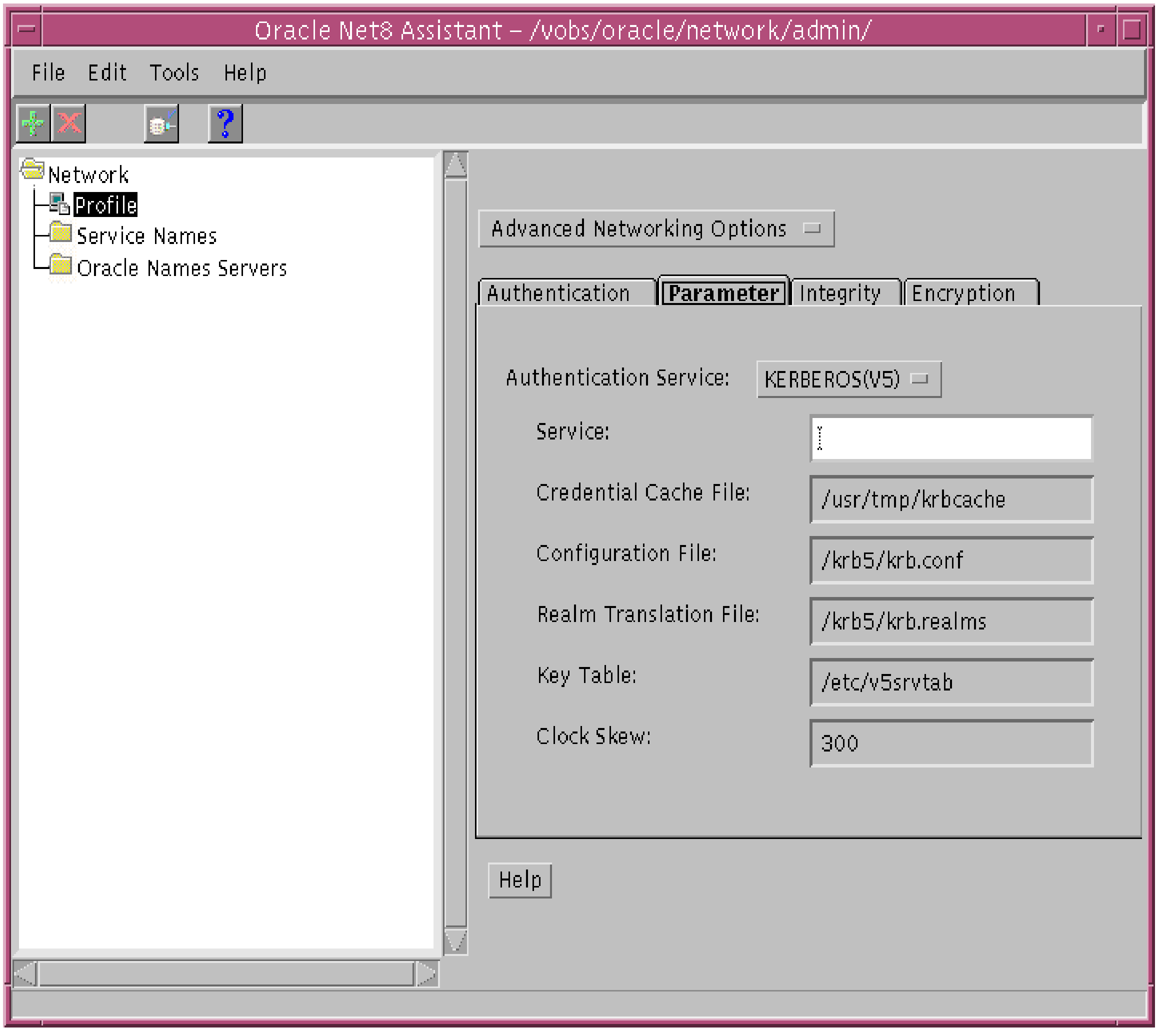
You now must configure the authentication parameters. Refer to Figure 3–3, “Oracle Net8 Assistant Profile Parameter Tab”. You must provide the value for only one parameter: GSSAPI Service.

1. Click the Profile folder.
2. Click the Parameter tab.
3. Click the Authentication Service drop-down list box, and select CYBERSAFE.

4. Type the name of the GSSAPI Service in the following format:

```
oracle/dbserver.someco.com@SOMECO.COM
```

Figure 3–3 Oracle Net8 Assistant Profile Parameter Tab



Authentication Service: KERBEROS(V5)

Service:

Credential Cache File: /usr/tmp/krbcache

Configuration File: /krb5/krb.conf

Realm Translation File: /krb5/krb.realms

Key Table: /etc/v5srvtab

Clock Skew: 300

3.1.10 Create a CyberSAFE User on the Authentication Server

Perform the following steps to create Oracle users, so they can be authenticated by the CyberSAFE adapter:

> **Note:** Perform these steps on the authentication server (where the administration tools are installed).

It is assumed that the realm already exists. (Refer to the CyberSAFE documentation listed in the ”Preface” if the realm needs to be created.)

> **Note:** The utility names in this section are actual programs that you run. However, the CyberSAFE user name “cyberuser” and realm “SOMECO.COM” are examples only; these may vary among systems.

Run /krb5/admin/kdb5_edit as root on the authentication server to create the new CyberSAFE user, that is, "cyberuser". Type the following:

1. # kdb5_edit
2. kdb5_edit: ank cyberuser
3. Enter password: <password not echoed to screen>
4. Re-enter password for verification: <password...>
5. kdb5_edit: quit

3.1.11 Create an Externally Authenticated Oracle User on the Oracle Server

Run Server Manager to create the Oracle user that corresponds to the CyberSAFE user, and perform the following commands on the Oracle server machine:

```
SVRMGR> connect internal;
SVRMGR> create user “CYBERUSER@SOMECO.COM” identified externally;
SVRMGR> grant create session to “CYBERUSER@SOMECO.COM”;
```

In this example, OS_AUTHENT_PREFIX is set to:

```
“”
```

When you create the Oracle user, the name must be in upper case and double-quoted. For example:

“CYBERUSER@SOMECO.COM”

3.1.12 Use kinit on the Client to Get the Initial Ticket for the Kerberos/Oracle User

Before users can connect to the database, they need to run kinit on the clients for an initial ticket.

```
% kinit (user name)
Password for CYBERUSER@US.ORACLE.COM:
<password not echoed to screen>
```

3.1.12.1 Use klist on the Client to Display Credentials

Users should run klist on the clients to list the tickets currently owned.

```
% klist


Creation Date     Expiration Date   Service
 11-Aug-95 16:29:51  12-Aug-95 00:29:21   krbtgt/SOMECO.COM@SOMECO.COM
 11-Aug-95 16:29:51  12-Aug-95 00:29:21   oracledbserver.someco.com@SOMECO.COM
```

3.1.13 Connect to an Oracle Server Authenticated by CyberSAFE

After running kinit to get an initial ticket, users can connect to an Oracle Server without using a username or password. Enter a command like the following:

```
%  sqlplus /@service_name
```

where *service_name* is a Net8 service name.

For example:

```
% sqlplus /@npddoc_db
```

Refer to Chapter 1, “Network Security and Single Sign-On” and to the *Oracle8 Distributed Database Systems* for more information on external authentication.

3.2 CyberSAFE Configuration Parameters Required on the Oracle

Server and Client

This section describes the parameters that need to exist in configuration files on Oracle servers and clients to enable CyberSAFE to authenticate users.

Note: Use the Oracle Net8 Assistant to configure these files.

3.2.1 Oracle Client Configuration Parameters

3.2.1.1 Required SQLNET.ORA Parameters

Make sure the following line is present in the SQLNET.ORA file on the client:

```
SQLNET.AUTHENTICATION_SERVICES=(CYBERSAFE)
```

3.2.2 Oracle Server Configuration Parameters

3.2.2.1 Required SQLNET.ORA Parameters

Make sure the following lines are present in the SQLNET.ORA file on the server.

```
sqlnet.authentication_services=(CYBERSAFE)
sqlnet.authentication_gssapi_service=oracle/dbserver.someco.com@SOMECO.COM
```

Note: You must insert the principal name, using the format described in Section 3.1.4, “Configure a Service Principal for an Oracle Server”.

3.2.2.2 Required INIT.ORA Parameters

It is strongly recommended that you add the following parameter to the INIT<SID>.ORA file used for the database instance:

```
REMOTE_OS_AUTHENT=FALSE
```

where <SID> is the database system identifier.

Attention: Setting REMOTE_OS_AUTHENT to TRUE may create a security hole because it allows someone using a non-secure protocol (for example, TCP) to perform an operating system-authorized login (formerly referred to as an OPS$ login).

CyberSAFE user names can be long and Oracle user names are limited to 30 characters, so it is strongly recommended that you use the following null value for the value of OS_AUTHENT_PREFIX:

```
OS_AUTHENT_PREFIX=""
```

Restart the Oracle server after modifying the configuration files, so the changes will take effect. (For information on how to restart the Oracle server refer to your operating system-specific documentation and to the *Oracle8 Administrator’s Guide*.)

3.3 Troubleshooting the Configuration of the CyberSAFE Authentication Adapter

Following are some common configuration problems and tips to help resolve them:

If you cannot get your ticket-granting ticket using kinit:

- Make sure the default realm is correct by looking at krb.conf.
- Make sure the Challenger Master Server is running on the host specified for the realm.
- Make sure that the Master Server has an entry for your user principal and that the passwords match.
- Make sure the krb.conf and krb.realms files are readable by Oracle.

If you have an initial ticket, but still cannot connect:

- After trying to connect, check for a service ticket.
- Check that the profile (SQLNET.ORA) on the server side has a service name that corresponds to a service known to the CyberSAFE Master Server.
- Check that the clocks on all the involved machines are within a few minutes of each other.

If you have a service ticket and you still cannot connect:

- Check the clocks on the client and server.
- Check that the v5srvtab exists in the correct location and is readable by Oracle.
- Check that the v5srvtab has been generated for the service named in the profile (SQLNET.ORA) on the server side.

If everything seems to work fine, but then you issue another query and it fails:

- Check that the initial ticket is forwardable. (You must have been obtained the initial ticket by running kinit -f.)
- Check the expiration date on the credentials.
- If your credentials have expired, close your connection and run kinit to get a new initial ticket.

4Co

nfiguring the Kerberos Authentication Adapter

This chapter contains information on how to configure Oracle for use with the Kerberos authentication adapter. Also included are brief descriptions of the steps to follow to configure Kerberos to authenticate Oracle users.

This information includes the following:

Section 4.1, “Steps to Perform to Enable Kerberos Authentication”

Section 4.2, “Configure the Kerberos Authentication Adapter Using the Oracle Net8 Assistant”

Section 4.3, “Description of Configuration File Parameters on Oracle Server and Client”

Section 4.4, “Troubleshooting the Configuration of the Kerberos Authentication Adapter”

4.1 Steps to Perform to Enable Kerberos Authentication

The following tasks are required to enable Kerberos authentication. Perform the following tasks in the order listed.

1. "Install Kerberos on the Machine that will Act as the Authentication Server"
2. "Configure a Service Principal for an Oracle Server"
3. "Extract a Service Table from Kerberos"
4. "Install an Oracle Server and an Oracle Client"
5. "Install Net8"
6. "Configure Net8 and Oracle on the Oracle Server and Client"
7. "Create a Kerberos User on the Kerberos Authentication Server"
8. "Create an Externally-Authenticated User on the Oracle Database"
9. "Get an Initial Ticket for the Kerberos / Oracle User"

4.1.1 Install Kerberos on the Machine that will Act as the Authentication Server

For information on how to install Kerberos on your machine, refer to the Kerberos documentation listed in the "Preface" of this guide.

4.1.2 Configure a Service Principal for an Oracle Server

For the Oracle Server to be able to validate the identity of clients that authenticate themselves using Kerberos, you must first create a service principal for Oracle.

The name of the principal should have the following format:

```
kservice/kinstance@REALM
```

where

String	Definition
kservice	a string that represents the Oracle service. This may or may not be the same as the database service name. It is case-sensitive.
kinstance	typically the fully-qualified name of the machine on which Oracle is running.
REALM	the domain of the server. It must always be capitalized.

Note: The utility names in this section are actual programs that you run. However, the Kerberos user name “krbuser” and realm “SOMECO.COM” are examples only: the actual names may vary among systems.

For example, if kservice is "oracle", and the fully-qualified name of the machine on which Oracle is running is "dbserver.someco.com", and if the realm is "SOMECO.COM", the principal name would be:

```
oracle/dbserver.someco.com@SOMECO.COM
```

It is a common convention to use the DNS domain name as the name of the realm.

To create the service principal, run kdb5_edit. The following example is UNIX specific.

```
# cd /krb5/admin
# ./kdb5_edit
```

To add a principal called "oracle / dbserver.someco.com@SOMECO.COM" to the list of server principals known by Kerberos, type the following:

```
kdb5_edit:ark oracle/dbserver.someco.com@SOMECO.COM
```

4.1.3 Extract a Service Table from Kerberos

You now need to extract the service table from Kerberos and copy it to the Oracle server / Kerberos client machine.

For example, to extract a service table for dbserver.someco.com, do the following:

```
kdb5_edit:  xst dbserver.someco.com oracle
'oracle/dbserver.someco.com@SOMECO.COM' added to keytab
'WRFILE:dbserver.someco.com-new-srvtab'
kdb5_edit:  exit
oklist -k -t dbserver.someco.com-new-srvtab
```

After the service table has been extracted, verify that the new entries are in the table in addition to the old ones. If they are not, or you need to add more, use kdb5_edit to append the additional entries.

If you do not enter a realm (for example, SOMECO.COM) when using xst, it uses the realm of the current host and displays it in the command output, as shown above.

If the Kerberos service table is on the same machine as the Kerberos client, you can simply move it. If the service table is on a different machine from the Kerberos client, you must transfer the file with a program like binary FTP. The following example is UNIX specific.

```
# mv dbserver.someco.com-new-srvtab /etc/v5srvtab
```

The default name of the service file is /etc/v5srvtab. If a different name is used, then that name should be substituted for the default name.

4.1.3.1 Ensure that the Oracle Server Can Read the Service Table

Verify that the owner of the Oracle Server executable can read the service table (in the above example, /etc/v5srvtab). To do that, set the file owner to the Oracle user or make the file readable by the group to which Oracle belongs.

WARNING: Do not make the file readable to all users. This may allow a security breach.

4.1.4 Install an Oracle Server and an Oracle Client

Install an Oracle Server and an Oracle Client. Refer to your operating system-specific documentation for information.

4.1.5 Install Net8

Install Net8 on the Oracle server and client machines.

4.1.6 Configure Net8 and Oracle on the Oracle Server and Client

For information on how to configure the Oracle server and client machines, see your operating system-specific documentation. Also refer to the *Oracle Net8 Administrator’s Guide.*

4.1.7 Create a Kerberos User on the Kerberos Authentication Server

Perform the following steps on the Kerberos authentication server, where the administration tools are installed, to create Oracle users so that they can be authenticated by the Kerberos adapter.

It is assumed that the realm already exists. Refer to the Kerberos documentation listed in the “Preface” if the realm needs to be created.

Note: The utility names in this section are actual programs that you run. However, the Kerberos user name "krbuser" and realm "SOMECO.COM" are examples only; these may vary among systems.

Run /krb5/admin/kdb5_edit as root to create the new Kerberos user, for example, "krbuser". The following example is UNIX specific.

```
# ./kdb5_edit
kdb5_edit: ank krbuser
Enter password:  <password not echoed to screen>
Re-enter password for verification:  <password...>
kdb5_edit: quit
```

4.1.8 Create an Externally-Authenticated User on the Oracle Database

Run Server Manager on the Oracle server to create the Oracle user that corresponds to the Kerberos user. In the following example, OS_AUTHENT_PREFIX is set to ““.

```
SVRMGR> connect internal;
SVRMGR> create user "KRBUSER@SOMECO.COM" identified externally;
SVRMGR> grant create session to "KRBUSER@SOMECO.COM";
```

Remember that the Oracle user name must be in upper-case and double-quoted: for example, "KRBUSER@SOMECO.COM".

4.1.9 Get an Initial Ticket for the Kerberos/Oracle User

Users need to run the following:

```
okinit (user name)
```

on the client to ask the Key Distribution Center (KDC) for an initial ticket before they can connect to the database. If, when making a database connection, a reference such as

```
sqlplus /@oracle
```

will follow a database link, you must use the forwardable flag (-f option). Executing okinit -f enables credentials that can be used across database links. You should be on the Oracle client before running the following commands.

```
% okinit -f
Password for krbuser@SOMECO.COM:<password not echoed to screen>
```

4.1.10 Utilities to Use with the Kerberos Authentication Adapter

The following three utilities are shipped with the Oracle Kerberos authentication adapter. You should be on the Oracle client before running these commands.

- `okinit`
  gets an initial ticket
- `oklist`
  displays a list of currently-owned tickets
- `okdstry`
  removes all tickets from the credentials cache

These utilities are intended for customers who are running an Oracle client with an Oracle Kerberos authentication adapter installed.

> **Note:** (The following applies to UNIX only.) Solaris is shipped with Kerberos version 4. Make sure that the Kerberos version 5 utilities are in your path so that the version 4 utilities are not inadvertently used.

4.1.10.1 Use okinit to Obtain the Initial Ticket

`okinit` obtains and caches Kerberos tickets. okinit is typically used to obtain your ticket-granting ticket, using a password entered by the user to decrypt the credential from the key distribution center (KDC). The ticket-granting ticket is then stored in the user's credential cache. The following options are available with okinit.

Available okinit Options

`-f` Ask for a forwardable ticket-granting ticket. This option is necessary to follow database links.

`-l` Specify the lifetime of the ticket-granting ticket and all subsequent tickets. By default, the ticket-granting ticket is good for eight (8) hours, but shorter or longer-lived credentials may be desired. Note that the KDC can ignore this option or put site-configured limits on what can be specified. The lifetime value is a string that consists of a number qualified by ‘w’ (weeks), ‘d’ (days), ‘h’ (hours), ‘m’ (months), or ‘s’ (seconds).

For example,

```
okinit -l 2wld6h20m30s
```

means ask for a ticket-granting ticket that has a lifetime of 2 weeks, 1 day, 6 hours, 20 minutes, and 30 seconds.

`-c` Specify an alternative credential cache. For UNIX, the default is /tmp/krb5cc_<*uid*>. You can also specify the alternate credential cache by using the SQLNET.KERBEROS5_CC_NAME parameter in the SQLNET.ORA file.

`-?` List command line options.

4.1.10.2 Use oklist to Display Credentials

Users can run oklist to display the list of tickets they hold. The show flag option (-f) displays additional information.

```
% oklist -f
27-Jul-1995 21:57:51   28-Jul-1995 05:58:14
krbtgt/SOMECO.COM@SOMECO.COM
Flags: FI
```

Available oklist Options

`-f`	Show flags with credentials. The important ones for Oracle are 'I' (credential is a ticket-granting ticket), 'F' (credential is forwardable), and 'f' (credential is forwarded).
`-c`	Specify an alternative credential cache. For UNIX, the default is /tmp/krb5cc_<*uid*>. The alternate credential cache can also be specified by using the SQLNET.KERBEROS5_CC_NAME parameter in the SQLNET.ORA file.
`-k`	List the entries in the service table (default /etc/v5srvtab) on UNIX. The alternate service table can also be specified by using the SQLNET.KERBEROS5_KEYTAB parameter in the SQLNET.ORA file.

4.1.10.3 Use okdstry to Remove Credentials from Cache File

Use okdstry to remove credentials from the credentials cache file.

```
$ okdstry -f
```

Available okdstry Options

`-f`	Specify an alternative credential cache. For UNIX, the default is /tmp/krb5cc_<*uid*>. You can also specify the alternate credential cache by using the SQLNET.KRB5_CC_NAME parameter in a profile (SQLNET.ORA).

4.1.11 Connecting to an Oracle Server Authenticated by Kerberos

You can now connect to an Oracle Server without using a username or password. Enter a command like the following:

```
$ sqlplus /@service_name
```

where service_name is a Net8 service name. For example:

```
$ sqlplus /@oracle_dbname
```

Refer to Chapter 1, "Network Security and Single Sign-On" and to *Oracle8 Server Distributed Systems* for more information on external authentication.

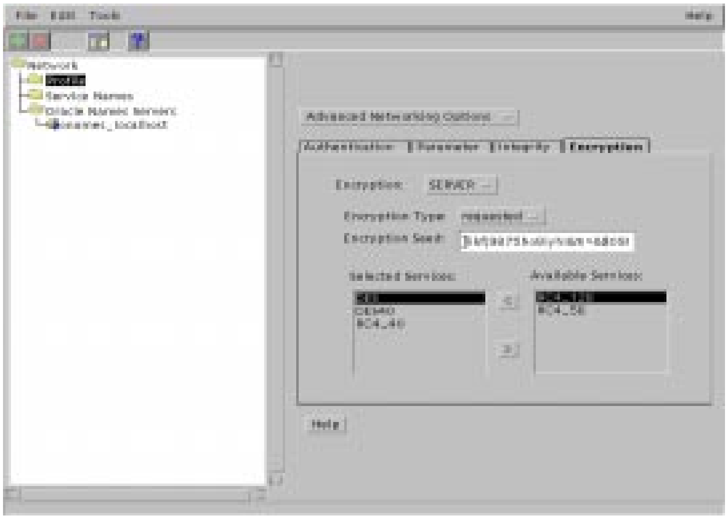
4.2 Configure the Kerberos Authentication Adapter Using the Oracle Net8 Assistant

The following steps show you how to use the Net8 Assistant to configure the Kerberos authentication adapter. Refer also to the Net8 Assistant on-line HELP system for instructions on how to configure the Kerberos Authentication adapter.

Configure Clients, and Servers, to use encryption as follows. Refer to Figure 4–1, “Oracle Net8 Assistant Profile Encryption Tab”.

1. Click the Profile folder.
2. Select Advanced Networking Options from the drop-down list box.
3. Click the Encryption tab.
4. Click the Encryption drop-down list box, and click CLIENT or SERVER.
5. Click the Encryption Type drop-down list box, and click one of the following values: requested, required, accepted, rejected.
6. Type between 10 and 70 random characters for the Encryption Seed.
7. Move services to and from the Available Services and Selected Services lists by selecting a service and clicking the arrow keys.

Figure 4–1 Oracle Net8 Assistant Profile Encryption Tab



Encryption: SERVER

Encryption Type: requested

Encryption Seed: [illegible]

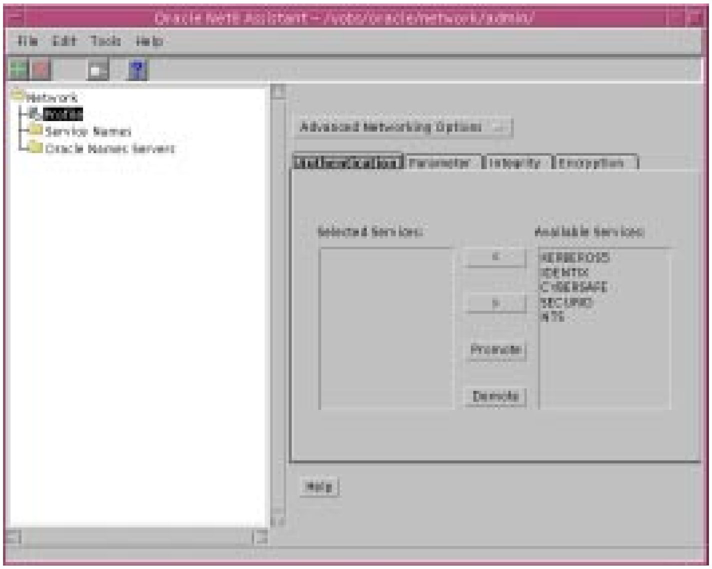
Selected Services: DES, DES40, RC4_40

Available Services: RC4_128, RC4_56

Next, you must configure an authentication service on your network. Refer to Figure 4–2, “Oracle Net8 Assistant Profile Authentication Tab”.

1. Click the Profile folder.
2. Click the Authentication tab.
3. Click to select the authentication service you want from the Available Services list.
4. Click the [<] button to move the service over to the Selected Services list.
5. Repeat steps 4 and 5, above, until you have selected all of your required authentication services.
6. Arrange the selected services in order of desired use. Click on a service to select it, then click [Promote] or [Demote] to arrange the services in the list. For example, put KERBEROS5 at the top of the list if you want that service to be the first one used.

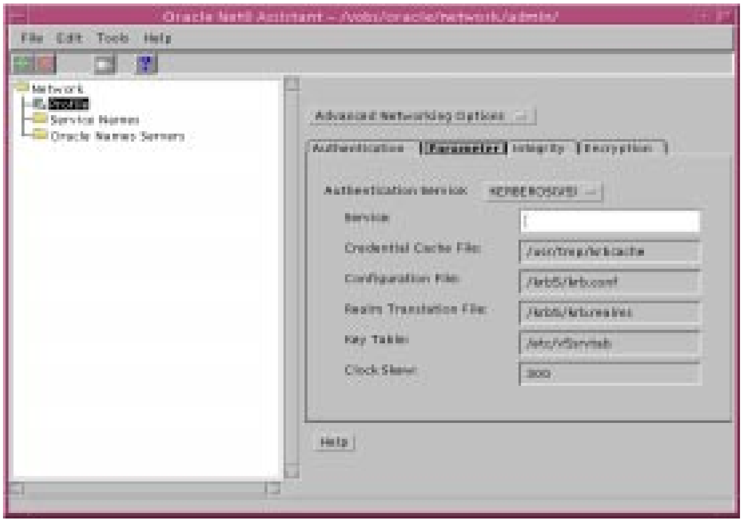
Figure 4–2 Oracle Net8 Assistant Profile Authentication Tab



You now must configure the authentication parameters. Refer to Figure 4–3, “Oracle Net8 Assistant Profile Parameter Tab”. You must provide the value for the following parameters.

- Service
- Credential Cache File
- Configuration File
- Realm Translation File
- Key Table
- Clock Skew

Figure 4–3 Oracle Net8 Assistant Profile Parameter Tab



Authentication Service: KERBEROS(V5)

Service:

Credential Cache File: /usr/tmp/krbcache

Configuration File: /krb5/krb.conf

Realm Translation File: /krb5/krb.realms

Key Table: /etc/v5srvtab

Clock Skew: 300

4.3 Description of Configuration File Parameters on Oracle Server and Client

This section describes the parameters that need to exist in configuration files on Oracle servers and clients for Kerberos to authenticate users.

4.3.1 Oracle Client Configuration Parameters

4.3.1.1 Required Profile Parameters

Make sure the following line is present in the profile (SQLNET.ORA) on the client:

```
sqlnet.authentication_services=(KERBEROS5)
```

4.3.2 Oracle Server Configuration Parameters

4.3.2.1 Required Profile Parameters

Make sure the following parameters are present in the profile (SQLNET.ORA) on the server:

```
sqlnet.authentication_services=(KERBEROS5)
sqlnet.authentication_kerberos5_service=kservice
```

Note: The second parameter specifies the name of the service Oracle will use to obtain a Kerberos service ticket. You must substitute a value for the kservice part of the service name.

Example:

```
sqlnet.authentication_kerberos5_service=oracle
```

There is no default; you must define one.

4.3.2.2 Required Initialization Parameters

You must add the following parameter to the INIT.ORA file used for the database instance:

```
REMOTE_OS_AUTHENT=FALSE
```

Attention: Setting REMOTE_OS_AUTHENT to TRUE may create a security hole, because it allows someone using a non-secure protocol (for example, TCP) to perform an operating system-authorized login (formerly referred to as an OPS$ login).

Because Kerberos user names can be long and Oracle user names are limited to 30 characters, it is strongly recommended that the following null value be used for the value of OS_AUTHENT_PREFIX:

```
OS_AUTHENT_PREFIX=""
```

Setting OS_AUTHENT_PREFIX to a null value overrides the default value of `OPS$`.

After modifying the configuration files, restart the Oracle server so that the changes will take effect. (For information on how to restart the Oracle server refer to your operating system-specific documentation and to the *Oracle8 Administrator's Guide*.)

4.3.2.3 Optional Profile Parameters

In addition to the above required parameters, you can optionally set the parameters described below on the client or server. The string:

```
SQLNET.KERBEROS5_CC_NAME=pathname_to_credentials_cache_file
```

Specifies the complete pathname to the Kerberos credentials cache (CC) file. The default value is operating system-dependent. For UNIX, it is /tmp/krb5cc_user id.

For example:

```
SQLNET.KERBEROS5_CC_NAME=/usr/tmp/krbcache
```

Note: You can also set this parameter by using the KRB5CCNAME environment variable.

The value set for the SQLNET.KERBEROS5_CC_NAME parameter in the SQLNET.ORA file takes precedence over the value set in the KRB5CCNAME environment variable.

```
SQLNET.KERBEROS5_CLOCKSKEW=number_of_seconds_accepted_as_network_delay
```

This parameter specifies how many seconds can pass before a Kerberos credential is considered out-of-date. It is used when a credential is actually received by either a client or a server. It is also used by an Oracle server to decide if a credential needs to be stored to protect against a replay attack. The default is 300 seconds. For example:

```
SQLNET.KERBEROS5_CLOCKSKEW=1200
```

SQLNET.KERBEROS5_CONF=*pathname_to_Kerberos_configuration_file*

This parameter specifies the complete pathname to the Kerberos configuration file. The configuration file contains the realm for the default KDC (key distribution center) and maps realms to KDC hosts. The default is operating system-dependent. For UNIX, it is /krb5/krb.conf. For example:

```
SQLNET.KERBEROS5_CONF=/krb5/krb.conf
```

SQLNET.KERBEROS5_KEYTAB=*pathname_to_Kerberos_principal/key_table*

This parameter specifies the complete pathname to the Kerberos principal/secret key mapping file. It is used by the Oracle server to extract its key and decrypt the incoming authentication information from the client. The default is operating system-dependent. For UNIX, it is /etc/v5srvtab. For example:

```
SQLNET.KERBEROS5_KEYTAB=/etc/v5srvtab
```

SQLNET.KERBEROS5_REALMS=*pathname_to_Kerberos_realm_translation_file*

This parameter specifies the complete pathname to the Kerberos realm translation file. The translation file provides a mapping from a host name or domain name to a realm. The default is operating system dependent. For UNIX, it is /etc/krb.realms. For example:

```
SQLNET.KERBEROS5_REALMS=/krb5/krb.realms
```

4.4 Troubleshooting the Configuration of the Kerberos Authentication Adapter

Some common configuration problems are listed below followed by tips on how to resolve them.

If you cannot get your ticket-granting ticket using okinit:

- Make sure the default realm is correct by looking at krb.conf.
- Make sure the KDC is running on the host specified for the realm.
- Make sure that the KDC has an entry for your user principal and that the passwords match.
- Make sure the krb.conf and krb.realms files are readable by Oracle.

If you have an initial ticket, but still cannot connect:

- After trying to connect, check for a service ticket.
- Check that the SQLNET.ORA file on the server side has a service name that corresponds to a service known by Kerberos.
- Check that the clocks on all machines involved are within a few minutes of each other (or change the sqlnet.kerberos5_clockskew parameter in the SQLNET.ORA file).

If you have a service ticket and you still cannot connect:

- Check the clocks on the client and server.
- Check that the v5srvtab exists in the correct location and is readable by Oracle (remember the SQLNET.ORA parameters).
- Check that the v5srvtab has been generated for the service named in the SQLNET.ORA file on the server side.

If everything seems to work fine, but then you issue another query and it fails:

- Check that the initial ticket is forwardable. (You must have obtained the initial ticket by running okinit -f.)
- Check the expiration date on the credentials.
- If your credentials have expired, you must close your connection and run okinit to get a new initial ticket.

5

Configuring Oracle for Use with the SecurID Adapter

This chapter describes how to configure and use the SecurID authentication adapter with the Oracle server and clients. It assumes that you are familiar with the Security Dynamics ACE/Server and that the ACE/Server is installed and running. Refer to the "Preface" for a list of related publications to read.

The following topics are discussed:

- Section 5.1, "System Requirements"
- Section 5.2, "Known Limitations"
- Section 5.3, "Steps to Perform to Enable SecurID Authentication"
- Section 5.4, "Configure the SecurID Authentication Adapter using the Net8 Assistant"
- Section 5.5, "Creating Users for the SecurID Adapter"
- Section 5.6, "Troubleshooting the Configuration of the SecurID Authentication Adapter"
- Section 5.7, "Using the SecurID Authentication Adapter"
- Section 5.8, "Configure the Oracle Client to Use the SecurID Authentication Adapter"

5.1 System Requirements

To use the SecurID authentication adapter included in the Oracle Advanced Networking Option release 8.0.3, you need the following:

- SQL*Net release 8.0.3 or higher
- Oracle 8.0.3 or higher
- ACE/Server 1.2.4 or higher
- The Oracle server machine must be running UDP/IP and TCP/IP protocols, because Oracle needs to communicate with the ACE/Server. Even though the client uses SQL*Net or Net8 to connect to Oracle, Oracle needs UDP to connect to the ACE/Server.

5.2 Known Limitations

The SecurID authentication adapter does not support database links, also known as "proxy authentication." This is a direct consequence of the fact that the SecurID card codes can only be used once.

When using the SecurID authentication adapter, password encryption is disabled. This means that the SecurID card code (and, if you use standard cards, the PIN), are sent over to the Oracle server in clear text. This could be a security problem, so Oracle recommends that you turn on the Oracle Advanced Networking Option datastream encryption, which ensures that the PIN is encrypted when sent to the Oracle server. For information on how to use datastream encryption, see Chapter 2, “Configuring Encryption and Checksumming”.

5.3 Steps to Perform to Enable SecurID Authentication

This section contains information on the following tasks:

- Section 5.3.1, “Register Oracle as a SecurID Client (ACE/Server Release 1.2.4)”
- Section 5.3.2, “Ensure that Oracle Can Find the Correct UDP Port (ACE/Server Release 1.2.4)”
- Section 5.3.3, “Install the Oracle Advanced Networking Option on the Oracle Server and Client”
- Section 5.3.4, “Configure Oracle as a SecurID Client (for ACE/Server Release 1.2.4)”
- Section 5.3.5, “Configure Oracle as a SecurID Client (Release ACE/Server 2.0)”

5.3.1 Register Oracle as a SecurID Client (ACE/Server Release 1.2.4)

Register the machine on which the Oracle Server resides as a SecurID client with the ACE server. You can do this with the Security Dynamics tool sdadmin. From the Client menu, choose Create Client (ACE/Server 1.2.4) or Add Client (ACE/Server 2.0), to create a client.

Refer to the Security Dynamics ACE/Server Instruction manual, version 1.2.4, or to the Security Dynamics ACE/Server version 2.0 Administration manual for more detailed information.

5.3.2 Ensure that Oracle Can Find the Correct UDP Port (ACE/Server Release 1.2.4)

First verify that the ACE/Server, the Oracle server, and the Oracle Advanced Networking Option are installed.

Make sure that the Oracle server can discover what the correct UDP port for contacting the ACE/Server is. These port numbers are typically stored in a file called services. On the UNIX operating system, this file is typically in the /etc directory. If you are using NIS (Network Information Services) as a naming service, make sure that the services map contains the correct entries for SecurID.

Note: You can verify which port the ACE server is using by running the Security Dynamics tool Kitconts (for ACE/Server 1.2.4) or sdinfo (for ACE/Server 2.0).

5.3.3 Install the Oracle Advanced Networking Option on the Oracle Server and Client

Install the Oracle Advanced Networking Option on the Oracle server and Oracle client using the Oracle Installer.

5.3.4 Configure Oracle as a SecurID Client (for ACE/Server Release 1.2.4)

5.3.4.1 Install the SecurID configuration files on the Oracle server machine.

You can obtain the SecurID configuration files from any other SecurID client or from the machine that runs the ACE/Server.

Note: The information in the following sections is UNIX-specific.

These files are typically stored in /var/ace. On the Oracle server machine, create this directory and copy the configuration files to it. At the minimum, you need the file `sdconf.rec`. The configuration files are used by both Oracle and the standard SecurID tools. Because the SecurID tools run setuid root, there can be a problem with the access permissions on the directory /var/ace and the files in this directory. Make sure that the owner of the Oracle executable (for example, the user "oracle8") is able to read all the files in /var/ace and can create new files in this directory.

Attention: Do not attempt to overcome this by running Oracle setuid root. It is not necessary, and it is dangerous to do so.

There are two ways to reach this goal without compromising security. Both ways work, but it is recommended that you use method #1. Both methods allow you to use Oracle with the SecurID authentication adapter and still continue using the other SecurID tools.

Method #1 The owner of the Oracle executable should also own the /var/ace directory and the files in /var/ace. For example, if the owner of the Oracle executable is the user "oracle8," perform the following steps, as root:

```
# chown oracle8 /var/ace
# chmod 0770 /var/ace
# chown oracle8 /var/ace/*
# chmod 0660 /var/ace/*
```

Method #2 The other option is to have root own the `/var/ace` directory and the files in `/var/ace`, but give the Oracle group read and write access. If the Oracle group is “dba”, you need to perform the following steps, as root:

```
# chown root /var/ace
# chmod 0770 /var/ace
# chgrp dba /var/ace
# chown root /var/ace/*
# chmod 0660 /var/ace/*
# chgrp dba /var/ace/*
```

5.3.5 Configure Oracle as a SecurID Client (Release ACE/Server 2.0)

The Oracle process will act as an ACE server client. For this reason, you need to install the ACE client software on the Oracle server machine. For information on how to install an ACE client, refer to the *ACE/Server Version 2.0 Client for UNIX* manual.

Note the following:

- The VAR_ACE environment variable is not supported. You have to store the configuration data in the /var/ace directory. If you currently have the ACE configuration data in a different location, you should create a symbolic link using the following command:

  ```
  # ln -s $VAR_ACE /var/ace
  ```

- Oracle needs to be able to read and write the ACE configuration data. This data is stored in the directory /var/ace (or $VAR_ACE if you use the symbolic link shown above).

Whether Oracle can read the configuration data depends on how you installed the ACE client software on the Oracle server. During the installation of the ACE client software, you can specify which administrator should own the configuration files.

Attention: Whether you use Method 1 or Method 2, below, make sure that you do not install Oracle as root.

5.3.5.1 Method #1

If root is the owner of the ACE server configuration data files, you will have to change the UNIX file permissions so that the owner of the oracle executable can read and write to these files. For example, the following commands give Oracle access to the files, and all the Security Dynamics tools that run as setuid root will still be able to access the files.

```
# chown oracle8 /var/ace
# chown oracle8 /var/ace/*
# chmod 0770 /var/ace
# chmod 0660 /var/ace/*
```

If the environment variable VAR_ACE is set to a different location than /var/ace, you should instead execute the following commands:

```
# ln -s $VAR_ACE /var/ace
# chown oracle8 $VAR_ACE
# chown oracle8 $VAR_ACE/*
# chmod 0770 $VAR_ACE
# chmod 0660 $VAR_ACE/*
```

5.3.5.2 Method #2

If the ACE files are not owned by root, you have two options:

- Install the ACE client or server and Oracle under the same UNIX account. (You have to install the ACE software as root, but you can specify which administrator should own the files. Specify the same user as the owner of the Oracle executable, typically “oracle8”).
- Add the owner of the Oracle executable to the ACE administrators' group.

> **Note:** Make sure the owner of the oracle executable remains a member of the DBA group; otherwise you will not be able to control your database.

For the change to take effect, do the following:

1. Log out, and log in again as the Oracle owner.
2. Restart your Network listener.
3. Restart your Oracle server.

5.4 Configure the SecurID Authentication Adapter using the Net8 Assistant

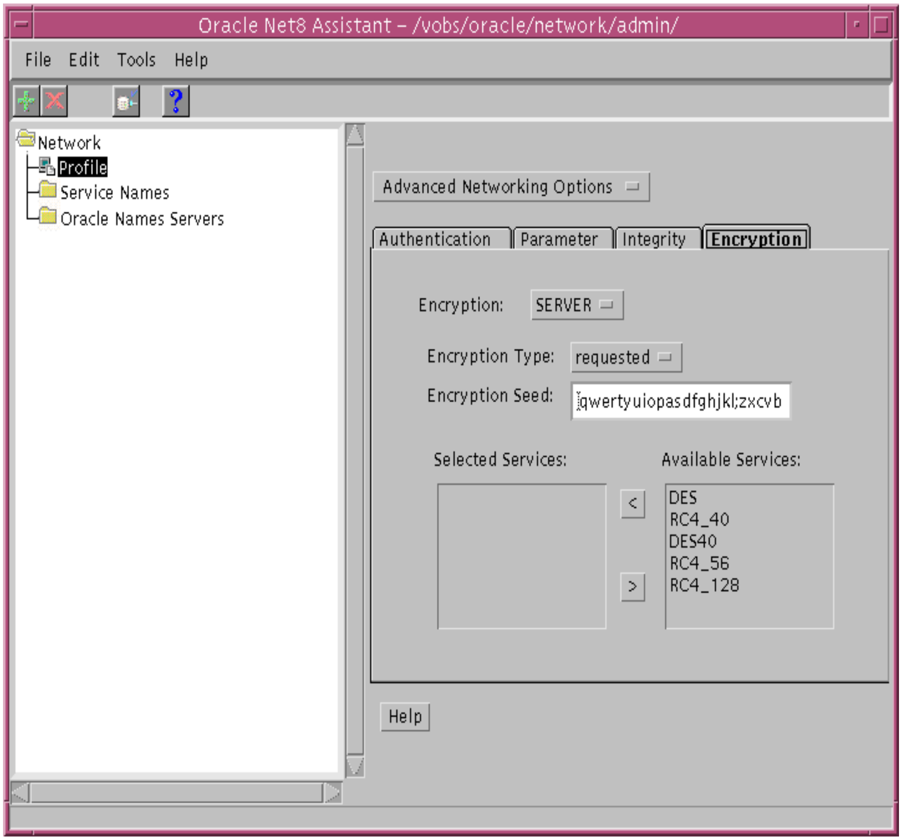
The following steps show you how to use the Net8 Assistant to configure the SECURID authentication adapter. Refer also to the Net8 Assistant online HELP system for instructions on how to configure the SECURID Authentication adapter.

Configure Clients, and Servers, to use encryption as follows. Refer to Figure 5–1, “Oracle Net8 Assistant Profile Encryption Tab”.

1. Click the Profile folder.
2. Select Advanced Networking Options from the drop-down list box.
3. Click the Encryption tab.

4. Click the Encryption drop-down list box, and click CLIENT or SERVER.
5. Click the Encryption Type drop-down list box, and click one of the following values: requested, required, accepted, rejected.
6. Type between 10 and 70 random characters for the Encyption Seed.
7. Move services to and from the Available Services and Selected Services lists by selecting a service and clicking the arrow keys.

Figure 5–1 Oracle Net8 Assistant Profile Encryption Tab



Oracle Net8 Assistant – /vobs/oracle/network/admin/

Encryption: SERVER

Encryption Type: requested

Encryption Seed: qwertyuiopasdfghjkl;zxcvb

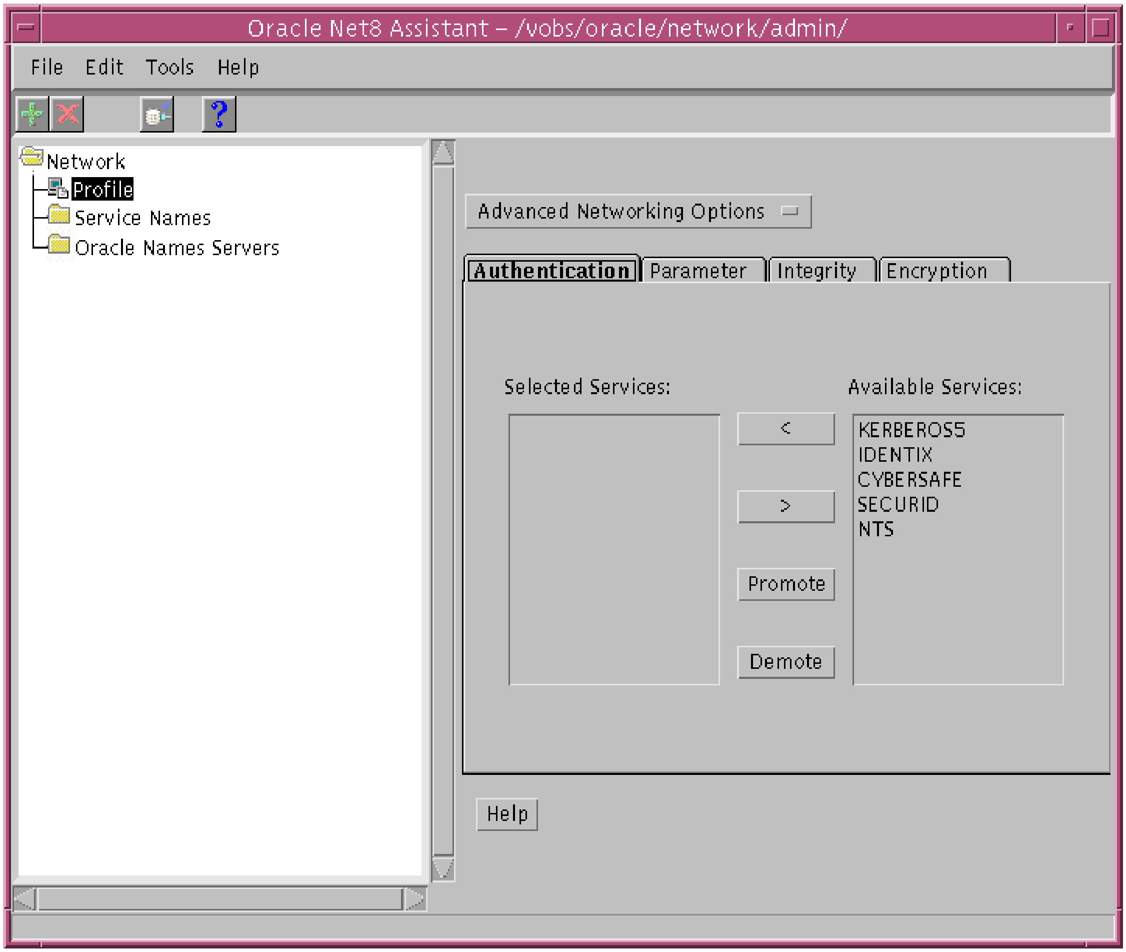
Selected Services:

Available Services:
DES
RC4_40
DES40
RC4_56
RC4_128

Next, you must configure an authentication service on your network. Refer to Figure 5–2, “Oracle Net8 Assistant Profile Authentication Tab”.

1. Click the Profile folder.
2. Click the Authentication tab.
3. Click to select the authentication service you want from the Available Services list.
4. Click the [<] button to move the service over to the Selected Services list.
5. Repeat steps 4 and 5, above, until you have selected all of your required authentication services.
6. Arrange the selected services in order of desired use. Click on a service to select it, then click [Promote] or [Demote] to arrange the services in the list. For example, put SECURID at the top of the list if you want that service to be the first one used.

Figure 5–2 Oracle Net8 Assistant Profile Authentication Tab

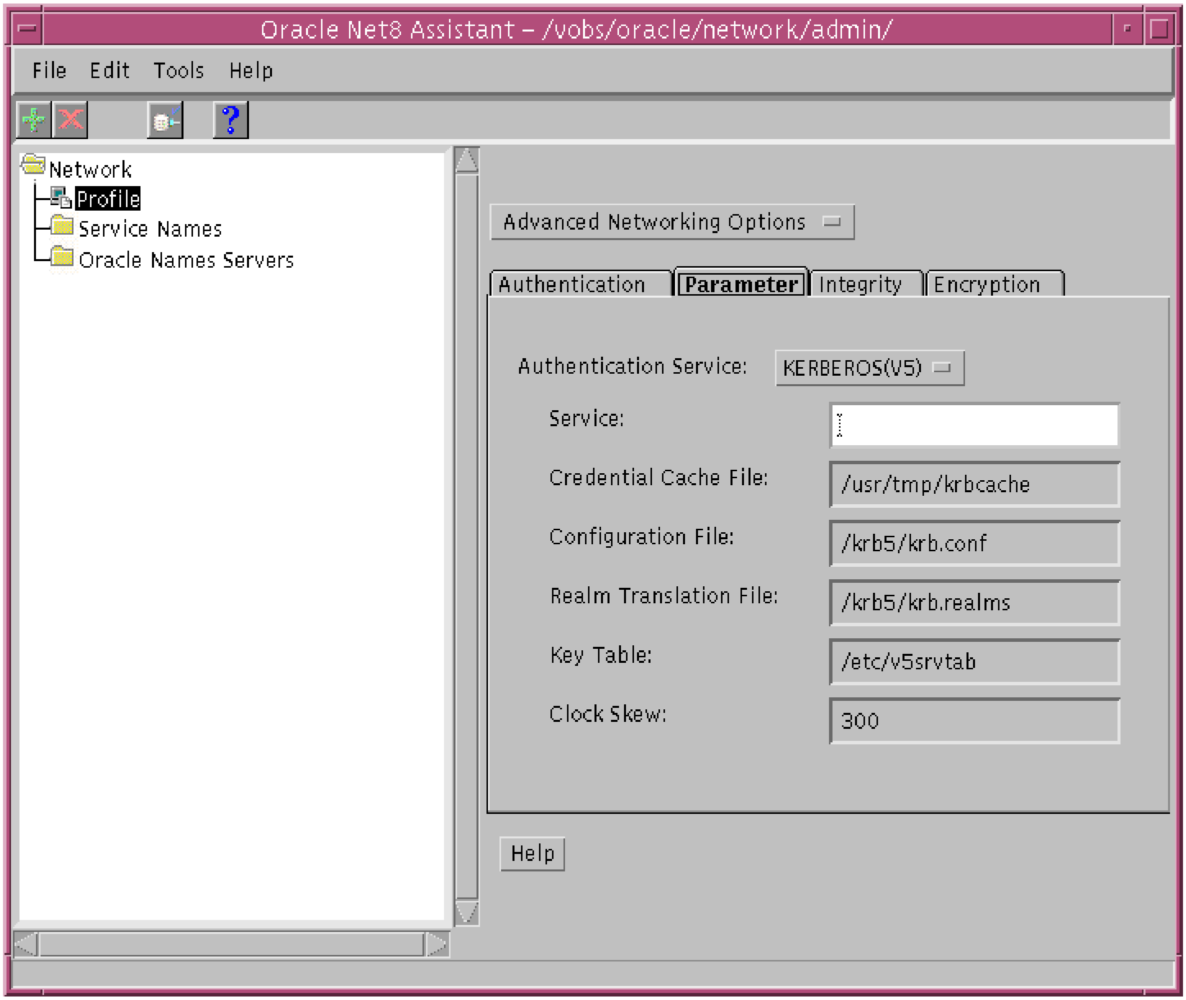


You now must configure the authentication parameters. Refer to Figure 5–3, “Oracle Net8 Assistant Profile Parameter Tab”. You do not provide any additional parameter for the SECURID authentication service.

1. Click the Profile folder.

2. Click the Parameter tab.
3. Click the Authentication Service drop-down list box, and select SECURID.
4. No additional parameters are required.

Figure 5–3 Oracle Net8 Assistant Profile Parameter Tab



Oracle Net8 Assistant – /vobs/oracle/network/admin/

Authentication Service: KERBEROS(V5)

Service:

Credential Cache File: /usr/tmp/krbcache

Configuration File: /krb5/krb.conf

Realm Translation File: /krb5/krb.realms

Key Table: /etc/v5srvtab

Clock Skew: 300

5.5 Creating Users for the SecurID Adapter

To create users for the SecurID authentication adapter, perform the following steps:

1. Assign a card to a person, using the Security Dynamics sdadmin program. When the sdadmin tool asks for a login name when creating a new user, fill in the same name you will use later to create the Oracle user. Refer to the Security Dynamics documentation for information on how to do this.

   If you want the user to be able to specify a new PIN to the card using the Oracle tools, choose the option that allows the user to make up his or her own PIN. If you do not allow this, the user will have to use the Security Dynamics tools to generate a PIN if the card is in new-PIN mode. Activate the user on the Oracle Server. (The Oracle Server should already be registered as a SecurID client.)

2. Create an Oracle Server account for this user. You can do this by using Server Manager connected as a user with the create user database role. Use the following syntax to create an account:

   ```
   SVRMGRL> connect system/manager
   SVRMGRL> create user os_authent_prefix username identified externally
   ```

   The OS_AUTHENT_PREFIX is an Oracle Server initialization parameter (for example, in INIT.ORA). The OS_AUTHENT_PREFIX default value is OPS$. The username should be the same as the name you assigned to the card in step 1 above.

   > **Note:** Because user names can be long and Oracle user names are limited to 30 characters, it is strongly recommended that `OS_AUTHENT_PREFIX` be set to a null value:
   >
   > `OS_AUTHENT_PREFIX=””`
   >
   > At this point, an Oracle user with *username* should not yet exist.

   Example: Assuming you have assigned a card to the user "king", and assuming that os_authent_prefix has been set to a null value (""), at this point you should create an Oracle user account using the following syntax:

   ```
   SQLDBA> create user king identified externally;
   ```

3. You may want to give this user some database privileges. At the minimum, the user should have the "create session" privilege.

```
SQLDBA> grant create session to king;
```

The user “king” can now connect to Oracle using his or her SecurID card.

For information on how to log into an Oracle server after the SecurID adapter has been installed and configured, see Section 6.1.1, “Log into the Oracle Server”.

5.6 Troubleshooting the Configuration of the SecurID Authentication Adapter

This section lists some things to verify if you experience problems while configuring the SecurID Adapter.

- The services map should have an entry for the Security Dynamics ACE server. The service name is typically securid, but the SecurID administrator can choose any name.

  Use the SecurID tool kitconts (for ACE/Server 1.2.4) or sdinfo (for ACE/Server 2.0) to verify the name of the authentication service and the port numbers that SecurID is expecting to use. Verify that these port numbers match those in /etc/services, or the services map if you are using NIS.

  (Applies to ACE/Server release 1.2.4 only) Verify that the /var/ace/sdconf.rec file is present on the machine running the Oracle server. Also verify that the permissions on the /var/ace/sdconf.rec file and the directory /var/ace are set so that the Oracle process can read and write in the directory.

  (Applies to ACE/Server release 2.0 only) Make sure the ACE configuration data is in the /var/ace directory. Use of the VAR_ACE environment variable is not supported. Also make sure that the owner of the oracle executable can read and write the files in this directory.

- Check to see if the Oracle server machine is registered as a SecurID client. You can do this by using the Security Dynamics tool sdadmin.

- The user who is trying to connect to Oracle should be activated on the Oracle Server, either as a direct user or as part of a group of users. Verify this using the SecurID tool sdadmin.

- Security Dynamics has developed a few logging facilities that can help you find problems. By using sdadmin, you can see a log of the recent system activities, including failed authentication with the reason for the failure. You can also use sdlogmon to get a similar log listing.

- Turn on tracing by adding the following line to the SQLNET.ORA file on the Oracle side:

  ```
  trace_level_server = admin
  ```

  Turning tracing on at the client side is less informative, because all interaction between the Oracle server and the ACE server happens at the Oracle server side of the SQL*Net connection. Be sure to turn off tracing when you have completed your check.

- Make sure that the user has been created in the Oracle database as an externally-identified user with the correct prefix (which defaults to OPS$). When connected as system, enter:

  ```
  SQL> select * from all_users;
  ```

  to get a list of all database users.

- When you connect to Oracle as a non-externally identified user, the SecurID log file will indicate a warning. For example, if you connect as 'system' using:

  ```
  sqlplus system/manager@oracle_dbname
  ```

  the SecurID log file displays:

  ```
  03/24/95   10:04  User not on client machinename
  ```

  This is not an error. Since the Oracle client and server negotiated to use SecurID because of the SQLNET.AUTHENTICATION_SERVICES line in SQLNET.ORA, Oracle will contact the ACE/Server to validate 'system'. When validation fails, Oracle will validate the password internally. If the password is valid, you will be able to connect.

  The only way to eliminate the warning message is to disable the SecurID authentication adapter. To do so, change the SQLNET.ORA file on the Oracle client to:

  ```
  SQLNET.AUTHENTICATION_SERVICES=(NONE)
  ```

  Setting this parameter to this value disables the SecurID authentication adapter. You will no longer be able to connect to Oracle using the SecurID card.

5.7 Using the SecurID Authentication Adapter

This section describes how to use the Oracle SecurID authentication adapter with the Oracle client tools. This chapter assumes that you are already familiar with SecurID concepts, and that you have configured Oracle for use with the SecurID adapter. (See Section 5.4, “Configure the SecurID Authentication Adapter using the Net8 Assistant” for information.) Also refer to the “Preface” of this guide for a list of publications to read.

5.8 Configure the Oracle Client to Use the SecurID Authentication Adapter

Before you can use the SecurID authentication adapter to verify passwords, make sure the following things have been done:

- The SecurID authentication adapter has been installed and linked into the SQL*Net configuration.
- Oracle has been configured for use with the ACE/Server (that is, it can act as a SecurID client).
- The client and server have been configured with the necessary parameters so that database passwords can be verified by the central SecurID authentication server.
- Users have been configured for use with the SecurID adapter as described in Section 5.4, “Configure the SecurID Authentication Adapter using the Net8 Assistant”.

5.8.1 Log into the Oracle Server

The SecurID authentication adapter allows you to log into the Oracle server with the PASSCODE that is generated by the SecurID card. The PASSCODE replaces the password in the Oracle connect statement.

There are two types of SecurID cards:

- standard (model SD200)
- PINPAD (model SD520)

Depending on the type of card, you type the PIN

- directly onto the card

or

- or as part of the Oracle connect statement.

5.8.1.1 Using Standard Cards

The standard cards generate and display a PASSCODE. When logging in to Oracle, you need to specify your username, your PIN and the current PASSCODE, using the following syntax:

```
SQL>connect <username>/<pin><passcode>@<service_name>
```

For example, if the card is assigned to user *king*, the PIN is ”3511,” and the card shows the number “698244,” this is how you would log into Oracle using SQL*Plus:

```
% sqlplus king/3511698244@oracle_database
```

or,

```
% sqlplus king@oracle_database
% password: 698244
```

Note: The Security Dynamics tools support the following characters as delimeters between the PIN and the PASSCODE:

“ ” <tab> \ / ; :

You should not use these characters, because Oracle will interpret these characters differently.

5.8.1.2 Using PINPAD Cards

If you have a PINPAD card, you first have to type in your PIN on the card and generate a new PASSCODE. You would then use this PASSCODE to connect to Oracle using the following syntax:

```
SQL>connect <username>/<passcode>@<service_name>
```

For example, if the card is assigned to user “king”, first generate a PASSCODE by typing the PIN on the PINPAD card. (Refer to the Security Dynamics documentation on how to do this.) For example, if the generated PASSCODE is “698244”, to connect to Oracle using SQL*Plus, you would type:

```
% sqlplus king/698244@oracle_dbname
```

5.8.2 Assign a New PIN to a SecurID Card

If you are logging in for the first time, or the administrator has put your card in the new-PIN mode, you have to assign a PIN to the card. You can tell that this is the case if, while trying to connect to Oracle, you get the following error message:

```
ORA-12681 “Login failed: the SecurID card does not have a pincode yet”
```

Assigning a PIN to a card is easy and can be done by connecting to the Oracle Server using a special syntax. First, you need to select a PIN, which is typically 4 to 8 digits long. Depending on the type of SecurID card you have, you may be able to use letters too.

The syntax while connecting to the Oracle database is:

```
SQL>connect <username>/“+<pincode>+<passcode>”@oracle_dbname
SQL>connect
```

For the passcode, enter the cardcode that is currently displayed on your SecurID card’s LCD. If you have a PINPAD card, do not enter the PIN on the card.

Note: You must add the two “+” characters in the connect string, because they tell Oracle that this is an attempt to assign a PIN to the card. Also, they separate the new PIN from the passcode.

You must also enclose the PIN / passcode combination in double quotes. Some Oracle tools such as Server Manager truncate the password string (PIN / passcode) just before the plus (“+”) character. Surrounding the password string (PIN / passcode) in double quotes (“”) prevents the password string from being truncated.

For example, if the card is assigned to user “king”, your new PIN is ”45618”, and the SecurID card currently displays number ”564728”, you would type:

```
% sqlplus king/“+45618+564728”@oracle_dbname
% passwd:<passcode>
```

If the new PIN is accepted, you will be connected to Oracle. The next time you want to connect to Oracle you should use the procedure described in ”Logging into the Oracle Server”. If the new PIN is rejected, you will get the following error:

```
ORA-12688 “Login failed: the SecurID server rejected the new pincode”
```

5.8.2.1 Possible Reasons Why a PIN Would be Rejected

Following are some possible reasons why a PIN would be rejected:

- The new PIN is less than 4 or more than 8 characters long.
- The PIN contains invalid characters. Valid characters are numeric digits, and for some SecurID cards, the letters "a" through "z".
- You are not allowed to make up your own PIN. The Security Dynamics ACE/Server can be configured in such a way that you cannot make up your own PIN. If this is the case, you will have to use one of the Security Dynamics tools to generate a new PIN for your card.

5.8.3 Log in When the SecurID Card is in "Next Code" Mode

As an additional safety step, the ACE/Server sometimes asks for the next card code, to ensure that the person who is trying to log in actually has the card in his or her possession. This is the case if you get the following error message when you try to log into Oracle:

```
ORA-12682, "Login failed: the SecurID card is in next PRN mode"
```

The next time you want to log in to Oracle, you will have to specify the next two card codes. The syntax you use to log into Oracle depends on the kind of SecurID card you have (Standard versus PINPAD).

5.8.3.1 Log in with a Standard Card

If you have a standard card, specify the following:

1. your PIN
2. the current card code
3. a "+" character and the next card code

Steps 1, 2, and 3 above replace the password. The "+" character is important, because it separates the first card code (passcode) from the second one. Use the following syntax:

```
SQL>connect <username>/ "<pincode><passcode>+<next passcode>"@<service_name>
```

Note: You must enclose the PIN / passcode / next passcode combination in double quotes. Some Oracle tools such as Server Manager truncate the password combination just before the plus ("+") character. Surrounding the PIN and passcode in double quotes ("") prevents the password combination from being truncated.

For example, if the card is assigned to user "king", the PIN is "3511", and the card first shows the number "698244" and the next number is "563866", you would type:

```
% sqlplus king/"3511698244+563866"@oracle_database
```

This connects you to the Oracle server and puts the card back into normal mode. The next time you want to log in to the Oracle server, use the procedure described in Section 5.8.1, "Log into the Oracle Server".

5.8.3.2 Log in with a PINPAD Card

If you have a PINPAD card, do the following to log on to the Oracle server:

1. Type in your PIN on the card to generate the first PASSCODE.
2. Clear your card’s memory by pressing P, then wait for the next PASSCODE.
3. Log into the Oracle server with these two passcodes, separated by a “+” character. Use the following syntax:

```
SQL>connect <username>/ “<first passcode>+<second passcode>”@service_name
```

For example, if the card is assigned to user “king”:

1. Type the PIN on the PINPAD card to generate a passcode: e.g., “231003”.
2. Clear the card’s memory. The next displayed number might be “831234”.
3. To log in, use the following syntax, typing the two passcodes generated in steps 1 and 2:

```
% sqlplus king/“231003+831234”@oracle_dbname
```

This connects you to Oracle and puts the card back into normal mode. The next time you want to log in to Oracle, use the procedure described in Section 5.8.1, “Log into the Oracle Server”.

6Co

nfiguring and Using the Identix Biometric Authentication Adapter

This chapter contains information on how to configure Oracle for use with the Identix Biometric Authentication Adapter. The following topics are discussed:

- Section 6.1, “Overview”
- Section 6.2, “Architecture of the Biometric Authentication Service”
- Section 6.3, “Prerequisites”
- Section 6.4, “Configuring the Biometric Authentication Service”
- Section 6.5, “Configuring the Oracle Biometric Authentication Service using the Oracle Net8 Assistant”
- Section 6.6, “Administering the Oracle Biometric Authentication Service”
- Section 6.7, “Authenticating Users With the Oracle Biometric Authentication Service”
- Section 6.8, “Using the Biometric Manager”
- Section 6.9, “Troubleshooting”

6.1 Overview

The Oracle Biometric Authentication Service uses the Identix Biometric Authentication Adapter to provide tamper-proof biometric authentication of users using secret-key MD5 hashing, centralized management of biometrically identified users, and centralized management of those database servers that authenticate biometrically identified users.

Following is an overview of how the Oracle Biometric Authentication Service works in a client-server environment. Refer to Figure 6–1, “Typical Oracle Biometric Authentication Service Configuration” for an illustration of the components and the configuration of the Oracle Biometric Authentication Service.

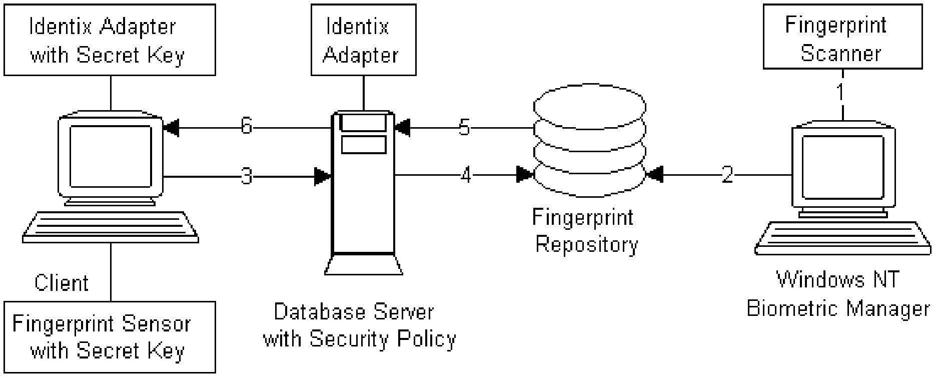
- The fingerprint repository has one administrator who is responsible for enrolling multiple users’ fingerprints and defining the DEFAULT policy that will be in force for all databases that subscribe to the fingerprint server for authentication.
- The Fingerprint Security Service Administrator uses a desktop fingerprint scanner to read user fingerprints and sends them with measured accuracies to the Oracle Biometric Authentication Service which stores them in the fingerprint repository: an Oracle database. The measured accuracy of a fingerprint is an estimate of how reliable a comparison can be made between the stored fingerprint and the user’s fingerprint that is entered later for authentication. The enrollment quality is expressed as a percent score between 0 and 100. For example, a user may have an enrollment quality of 72%.
- The Fingerprint Security Service Administrator also defines one security policy named DEFAULT for all of the database servers that accept biometrically identified users. The security policy is enforced for all clients serviced by that database server. It contains a secret key and three types of threshold levels for fingerprints: verification, false finger, and high security.
- At the client, before any authentication can occur, the Fingerprint Security Service Administrator stores the secret key in the fingerprint sensor for each client. The secret key stored in the fingerprint sensor will be compared against the secret key stored in the security policy.
- At the client, in response to the user’s request for authentication, the database server enforces on the client the set of values that it obtains from the DEFAULT security policy in its fingerprint server. The three threshold levels (values) are:
  - verification threshold
  - false finger threshold

- high security threshold

Please refer to the Identix documentation for detailed information on these threshold levels.

- At the client, the Oracle Biometric Authentication Service fulfills the request for authentication by "reading" the user's fingerprint, the three threshold values, and the secret key from the sensor and creating a hash from them. This hash is then compared with the hash constructed from the repository's copy of the secret key, threshold, and stored fingerprint in order to determine whether this user may access the system.

Figure 6–1 Typical Oracle Biometric Authentication Service Configuration


Identix Adapter with Secret Key
Identix Adapter
Fingerprint Scanner
1
6
5
3
4
2
Fingerprint Repository
Client
Fingerprint Sensor with Secret Key
Database Server with Security Policy
Windows NT Biometric Manager

6.2 Architecture of the Biometric Authentication Service

The Oracle Biometric Authentication Service consists of the following Oracle modules:

- The Oracle Biometric Manager, which the administrator uses to enter the security policy and fingerprints, is an Oracle Enterprise Manager Database tool based on and delivered with the Oracle Enterprise Manager. In the remainder of this document, the Oracle Biometric Manager will also be referred to as the manager.
- The Oracle Biometric Authentication Server (fingerprint repository), which stores the security policies and fingerprints, is a specially configured version of a production Oracle Database Server. In the remainder of this document, the

Oracle Biometric Authentication Server will also be referred to as the authentication server.

- The Oracle Advanced Networking Option Identix Authentication Adapters are used on both the clients and the database servers to communicate biometric authentication data between the authentication server and the clients in order to authenticate a database user. In the remainder of this document, the Oracle Advanced Networking Option Identix Authentication Adapter will also be referred to as the adapter.

Both the manager and the client-side adapter interface with Identix products: TouchNet II Software Libraries, the TouchNet II Hardware Interface, and the TouchNet II Desktop Sensor. Please refer to “Related Publications” in the Preface of this manual for a list of Identix documentation that describe these Identix products.

6.2.1 Administration Architecture

The Fingerprint Security Server Administrators use the manager to scan user fingerprints, measure the accuracy of the fingerprints, and establish security policies for database servers. The manager sends this information to the authentication server which stores the data in the repository.

The administrator, or someone who can be trusted, uses the Identix TouchNet II Software to store the secret key in the client PC. This key must match the key stored in the DEFAULT security policy before authentication can occur.

6.2.2 Authentication Architecture

Each user who wants to use the system must place a fingerprint on a TouchNet II Desktop Sensor. The client-side adapter sends an authentication request to the server-side adapter which uses the previously enrolled fingerprint stored in the authentication server for comparison. For each authentication request from a client, the authentication server retrieves and sends the user’s fingerprint and the database server’s security policy back to the client-side adapter via the server-side adapter.

The user’s authentication request causes the Oracle Advanced Networking Option Identix Authentication Adapter (client-side) to send the request to the Biometric Authentication Adapter (server-side), which looks up the user’s fingerprint in the Authentication Server, which returns the stored fingerprint and the associated security policy.

Using threshold level values from the associated security policy, the adapter (client-side) uses the TouchNet II Software Libraries to set threshold values on the Touch-

Net II Desktop Sensor. It then prompts for the placing of the user's finger on the TouchNet II Desktop Sensor. The adapters on the client and the database server work together to compare the user's fingerprint, the secret key, and the threshold levels against the administrator-entered security policy stored in the authentication server repository. If this data matches, the user is then authenticated.

6.3 Prerequisites

- The Windows NT machine that is to become the manager PC must be running the Oracle Enterprise Manager 1.3.5 or above.
- Each Windows NT or Windows 95 machine that is to become a client PC must be running Net8.
- The authentication server and each database server must be running Oracle8 Server Version 8.0.3 or higher.
- Before proceeding with the installation of the Oracle Advanced Networking Option, you must make sure that each Windows NT and Windows 95 client has Net8 connectivity with its associated database server.

6.3.1 Oracle Biometric Manager PC

The Oracle Biometric Manager installation process automatically installs the necessary TouchNet II software and automatically configures the device if requested. On the manager PC:

1. Install the Identix hardware and the Identix driver firmware and configure the Identix variables and devices. See the Identix Readme file for additional information.
2. Install and test the Identix TouchNet II (Encrypt) 1.5 from the Oracle Enterprise Manager disk. Please see your platform-specific installation documentation. Follow the instructions in the Identix manual to verify that the module works with the Identix demonstration program. This demonstration program must work on the PC before any other Oracle products can be loaded onto the PC. Refer to the Identix Readme file for additional information.
3. Install the Oracle Biometric Manager on top of the Oracle Enterprise Manager.

6.3.2 Client PC

On each client PC:

1. Install the Identix hardware and the Identix driver firmware and configure the Identix variables and devices. Refer to the Identix Readme file for additional information.
2. Install and test the Identix TouchNet II (Encrypt) 1.4 from the Oracle Enterprise Manager disk. Please see your platform-specific installation documentation. Follow the instructions in the Identix manual to verify that the module works with the Identix demonstration program. This demonstration program must work on the PC before any other Oracle products can be loaded onto the PC.
3. Install the Oracle Advanced Networking Option Identix Authentication Adapter following the instructions in your platform-specific documentation. Refer also to the Identix Readme file.

6.3.3 Database Server

The Biometric authentication adapter must be installed on each production database that will use Biometric services for its authentication. Install the Biometric authentication adapter following the instructions in your platform-specific documentation. Do not install the adapter on the database housing the Biometric Authentication Service unless you want to have the Biometric Service Administrator authenticate using the adapter. Refer also to the Identix Readme file.

6.3.4 Biometric Authentication Service

The Biometric Authentication Service is the database that houses both the user and fingerprint information. This database can be any Oracle 8.0.3 or later production database. It should be on a secure, trusted system with strict security and access controls. The adapter should not be installed on this database.

6.4 Configuring the Biometric Authentication Service

Configure the Oracle Biometric Authentication Service by following these instructions:

1. Configure the database server that is to become the authentication server:
   a. Connect to the database server as SYSTEM/MANAGER (or whatever your system password is).

b. Copy the naui...sql scripts from your Oracle Enterprise Manager install to the authentication server.

c. Test the connection by connecting as:

```
ofm_admin/ofm_admin
```

2. In the database server's local profile (SQLNET.ORA), set the following parameters:

```
sqlnet.identix_fingerprint_database= service_name
sqlnet.identix_fingerprint_database_user= username
sqlnet.identix_fingerprint_database_password= password
sqlnet.identix_fingerprint_method= oracle
sqlnet.authentication_services= (identix)
```

where

- *service_name* is the name of your authentication server
- *username* is the well-known username: ofm_client
- *password* is the well-known password: ofm_client

Note: The samples directory contains a file that show how to set these parameters.

Note: The ofm_client username and password are set up by running NAUICAT.SQL. You should not change ofm_client.

3. In the database server's local initialization file (INIT.ORA), set the following parameters:

```
remote_os_authent = false
os_authent_prefix = ""
```

Note: The local naming configuration file (TNSNAMES.ORA) on the database server should contain the service name of the fingerprint repository. If they are on the same database, use the following with the service name:

```
(security=(authentication_service=none))
```

4. Establish a service name and connect descriptor for the fingerprint repository server in the database server's local naming configuration file. The service name must be the same as that used in the local profile. Use the Oracle Net8 Assistant or the Service Names Wizard to construct this parameter.

   ```
   service_name =(DESCRIPTION =
                   (ADDRESS_LIST =
                       (ADDRESS =
                           . . .
   ```

5. Configure the adapter (client-side):

   a. Verify that the address of the database server is accessible to the client, either through a local naming configuration file or a naming service. For more information, refer to the *Oracle Net8 Administrator's Guide*.

   b. Modify the client's local profile, by adding identix to the list of authentication services:

      ```
      sqlnet.authentication_services = (identix)
      ```

6. Configure the manager PC by setting the local naming configuration file (TNSNAMES.ORA) to connect to the authentication server. Please refer to the *Oracle Net8 Administrator's Guide* .

6.5 Configuring the Oracle Biometric Authentication Service using the Oracle Net8 Assistant

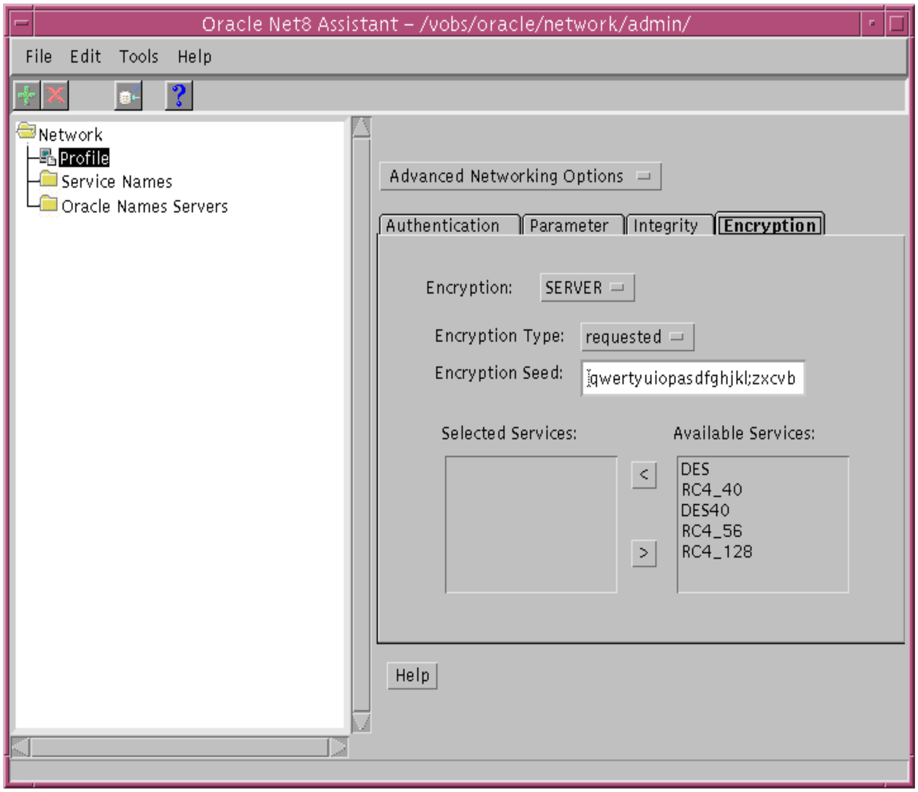
The following steps show you how to use the Net8 Assistant to configure the IDENTIX authentication adapter. Refer also to the Net8 Assistant online HELP system for instructions on how to configure the SECURID Authentication adapter.

Configure Clients, and Servers, to use encryption as follows. Refer to Figure 6–2, "Oracle Net8 Assistant Profile Folder Encryption Tab".

1. Click the Profile folder.
2. Select Advanced Networking Options from the drop-down list box.
3. Click the Encryption tab.
4. Click the Encryption drop-down list box, and click CLIENT or SERVER.

5. Click the Encryption Type drop-down list box, and click one of the following values: requested, required, accepted, rejected.
6. Type between 10 and 70 random characters for the Encryption Seed.
7. Move services to and from the Available Services and Selected Services lists by selecting a service and clicking the arrow keys.

Figure 6–2 Oracle Net8 Assistant Profile Folder Encryption Tab



Encryption: SERVER

Encryption Type: requested

Encryption Seed: qwertyuiopasdfghjkl;zxcvb

Selected Services:

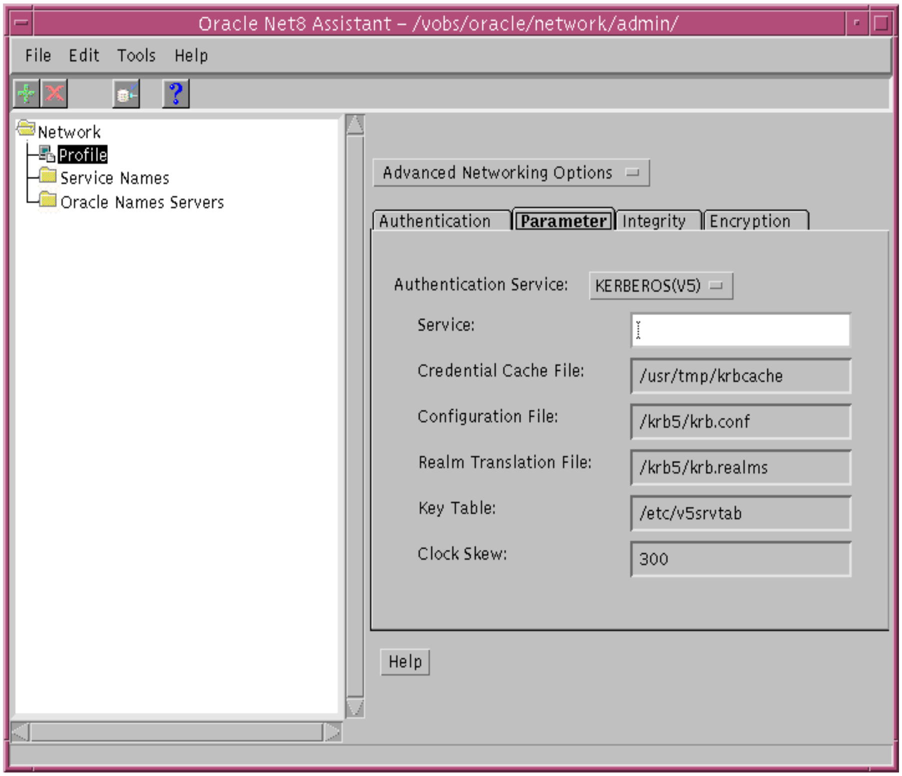
Available Services:
DES
RC4_40
DES40
RC4_56
RC4_128

Next, you must configure an authentication service on your network. Refer to Figure 6–3, “Oracle Net8 Assistant Profile Folder Authentication Tab”.

1. Click the Profile folder.
2. Click the Authentication tab.

3. Click to select the authentication service you want from the Available Services list.
4. Click the [<] button to move the service over to the Selected Services list.
5. Repeat steps 4 and 5, above, until you have selected all of your required authentication services.
6. Arrange the selected services in order of desired use. Click on a service to select it, then click [Promote] or [Demote] to arrange the services in the list. For example, put IDENTIX at the top of the list if you want that service to be the first one used.

Figure 6–3 Oracle Net8 Assistant Profile Folder Authentication Tab



Authentication Service: KERBEROS(V5)

Service:

Credential Cache File: /usr/tmp/krbcache

Configuration File: /krb5/krb.conf

Realm Translation File: /krb5/krb.realms

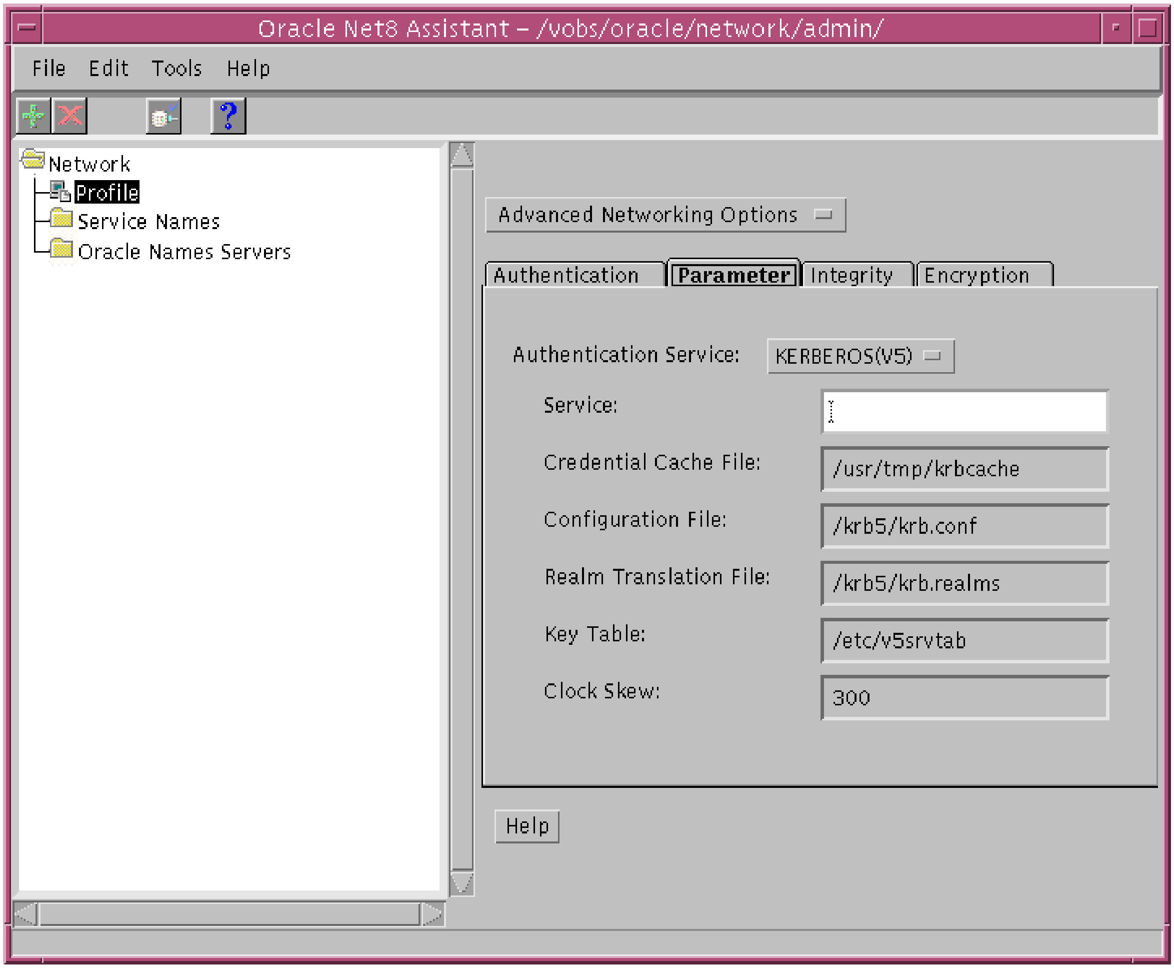
Key Table: /etc/v5srvtab

Clock Skew: 300

You now must configure the authentication parameters. Refer to Figure , “”.

1. Click the Profile folder.
2. Click the Parameter tab.
3. Click the Authentication Service drop-down list box, and select IDENTIX.
4. Type the name of the fingerprint server you want to use.

Figure 6–4 Oracle Net8 Assistant Profile Folder Parameter Tab



Authentication Service: KERBEROS(V5)

Service:

Credential Cache File: /usr/tmp/krbcache

Configuration File: /krb5/krb.conf

Realm Translation File: /krb5/krb.realms

Key Table: /etc/v5srvtab

Clock Skew: 300

6.6 Administering the Oracle Biometric Authentication Service

Add a security policy called “DEFAULT” to the manager using the Biometric Manager on the Oracle Enterprise Manager. Refer to Oracle Biometric Manager online Help for task oriented procedures.

6.6.1 Create a Hashkey on each of the Clients

Use the Identix Setkey utility to configure a hexadecimal hashkey on each of the clients: e.g., FF30EE. This key must be the same for each client and must match the DEFAULT Policy hashkey. This key can range from 1 to 32 hexadecimal digits.

6.6.2 Create Users for the Biometric Authentication Adapter

To create a user for the adapter, execute the following steps:

1. On the client use the Windows NT User Manager to create a username. (This username must match the username used in the next step.)
2. On the database server, restart the database and create an Oracle Server account for the user. Use SVRMGRL if using the Oracle Enterprise Manager or Server Manager connected as a user with the create user database role. Use the following syntax to create an account:

```
SVRMGRL> connect system/manager
SVRMGRL> create user os_authent_prefix username identified externally;
```

3. The *os_authent_prefix* is an Oracle Server initialization parameter. The default value for os_authent_prefix is OPS$. The username in this step should match the *username* created at the client. If you reset os_authent_prefix, you must stop and restart your database.

Note: Oracle user names are limited to 30 characters and user names can be long, so it is strongly recommended that os_authent_prefix be set to a null value:

```
os_authent_prefix=””
```

Note: An Oracle user with *username* should not yet exist.

4. Example: If you create the user “king,” and set os_authent_prefix to a null value (““), you should create an Oracle user account using the following syntax:

```
SQLDBA> create user king identified externally;
```

5. At the minimum, you should give the user the “create session” privilege:

   ```
   SQLDBA> grant create session to king;
   ```

6. Use the manager to enroll the user in the Oracle Biometric Authentication Service.
7. The user “king” can now be biometrically authenticated to Oracle.

For information on how to log on to a database server once the adapter has been installed and configured, see Section 6.7, “Authenticating Users With the Oracle Biometric Authentication Service”. Store the secret key in the client according to the directions in the Identix documentation.

6.7 Authenticating Users With the Oracle Biometric Authentication Service

To authenticate a user, first make sure that the Biometric Authentication Service has been installed and configured and the steps in Section 6.6, “Administering the Oracle Biometric Authentication Service” have been executed.

The user should follow these instructions:

1. Log on as the *username* assigned by the database administrator.
2. Set the System Environment Variable. The following variable is based on the 10 port setting on your TouchNet II firmware.

   ```
   ETSII_IOPORT = 0X280
   ```

3. Double click Svrmgr 2.3. (Authentication is not limited to Svrmgr, but may be implemented through other front ends.)
4. Type the name of your database server when Svrmgr displays the prompt:

   ```
   Svrmgr>connect /@service_name
   ```

   where, *service_name* is the name of the database server.

5. Wait for the beep that announces the SQL*Net Native Authentication dialog box.

> **Note:** On some systems the dialog box is displayed behind the current window. The beep alerts you when it is displayed.

6. Click OK in the SQL*Net Native Authentication dialog box.
7. When a message appears telling you to place your finger on the desktop fingerprint sensor, use the same finger as you and the administrator entered into the authentication server repository.
8. Remove your finger at the prompt. Another prompt tells you whether you've been authenticated or not.

If the authentication fails, and the message, "Access Denied," appears, try one of the following recovery methods:

- Restart the authentication process. See Section 6.7, "Authenticating Users With the Oracle Biometric Authentication Service".
- Have the security administrator lower the threshold value to 80.
- Have the security administrator reenroll you. Refer to Oracle Biometric Manager online Help for task oriented procedures.

6.8 Using the Biometric Manager

The Oracle Biometric Authentication Service is administered using the Biometric Manager which is based on the Oracle Enterprise Manager. It provides a graphical user interface (GUI) which enables the administrator to:

- log on to the Fingerprint Authentication Server
- browse the Oracle Biometric Authentication Service data for current users and security policies
- enroll / delete a user to / from the database
- create / modify a user's fingerprint
- add / delete the default security policy to / from the database

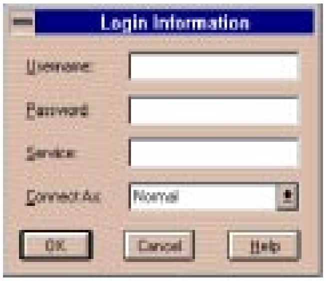
Refer to Oracle Biometric Manager online Help for task oriented procedures.

Note: Once the Biometric Manager has been installed, the first action taken must be that of adding a security policy called "DEFAULT" to the database.

6.8.1 Logging On

Figure 6–5, "Login Information Window", appears after you click on the Oracle Biometric Manager icon in the Oracle Enterprise Manager window.

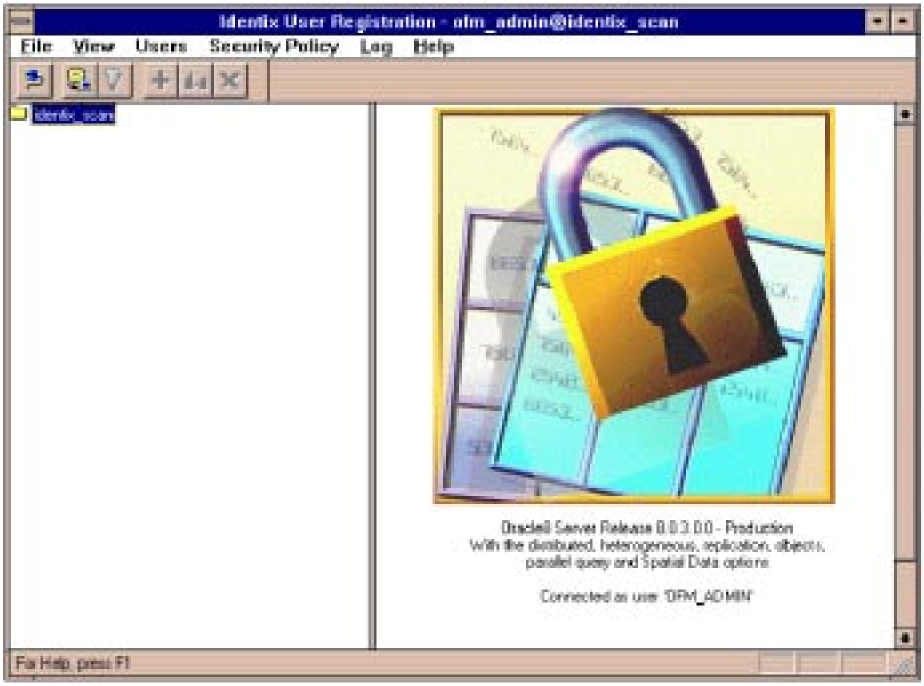
Figure 6–5 Login Information Window



Connect As: Normal

1. Type, or select, the following information to log on to the Oracle Biometric Manager.
   - username
   - password
   - service_name
     where *service_name* is the name of the authentication server
   - Connect As
     leave this field blank
2. Click [OK] to continue, click [Cancel] to return to the Oracle Enterprise Manager, or click [Help] for Oracle Enterprise Manager help.
3. Figure 6–6, "Indentix User Registration Window", appears after you click [OK].

Figure 6–6 Indentix User Registration Window


Identix User Registration - olm_admin@identix_scan
File View Users Security Policy Log Help
identix_scan

Oracle8 Server Release 8.0.3.0.0 - Production
With the distributed, heterogeneous, replication, objects,
parallel query and Spatial Data options

Connected as user 'DFM_ADMIN'

For Help, press F1

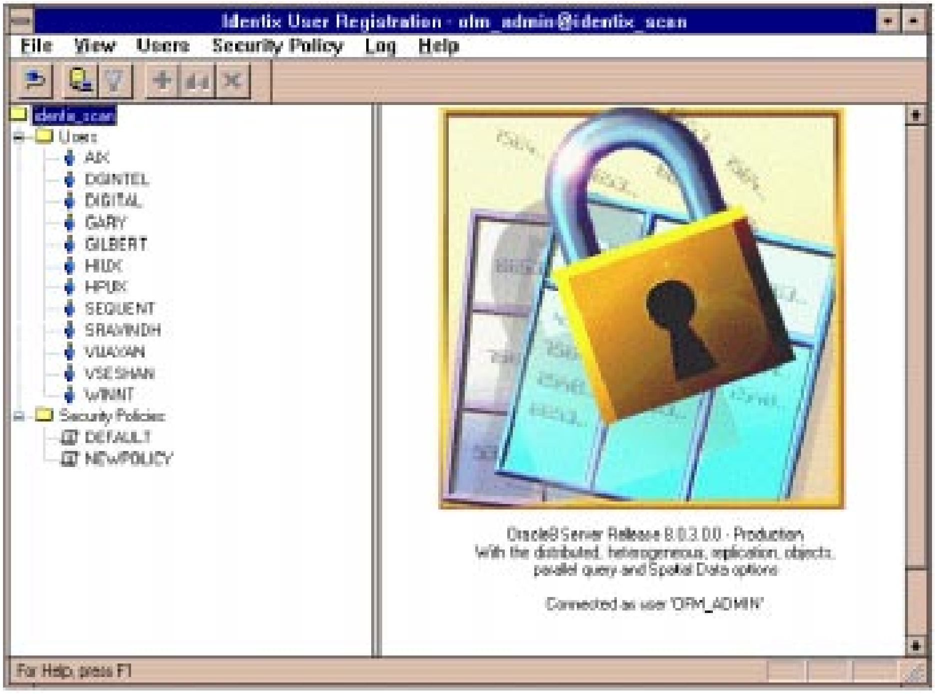
6.8.2 Displaying Oracle Biometric Authentication Service Data

The Oracle Enterprise Manager displays the Oracle Biometric Authentication Service database schema in two windows: the Object Tree window and the Properties window.

6.8.2.1 The Object Tree Window

The object tree window is located on the left side of the screen. It displays the Oracle Biometric Authentication Service database schema in a tree-like structure. This tree-like structure is composed of a series of folders that contain objects. These objects, in turn, may also contain folders that contain additional objects. See Figure 6–7, “Identix User Registration Window with Expanded Object Tree”.

Figure 6–7 Identix User Registration Window with Expanded Object Tree



Oracle8 Server Release 8.0.3.0.0 - Production
With the distributed, heterogeneous, replication, objects,
parallel query and Spatial Data options

Connected as user 'OFM_ADMIN'

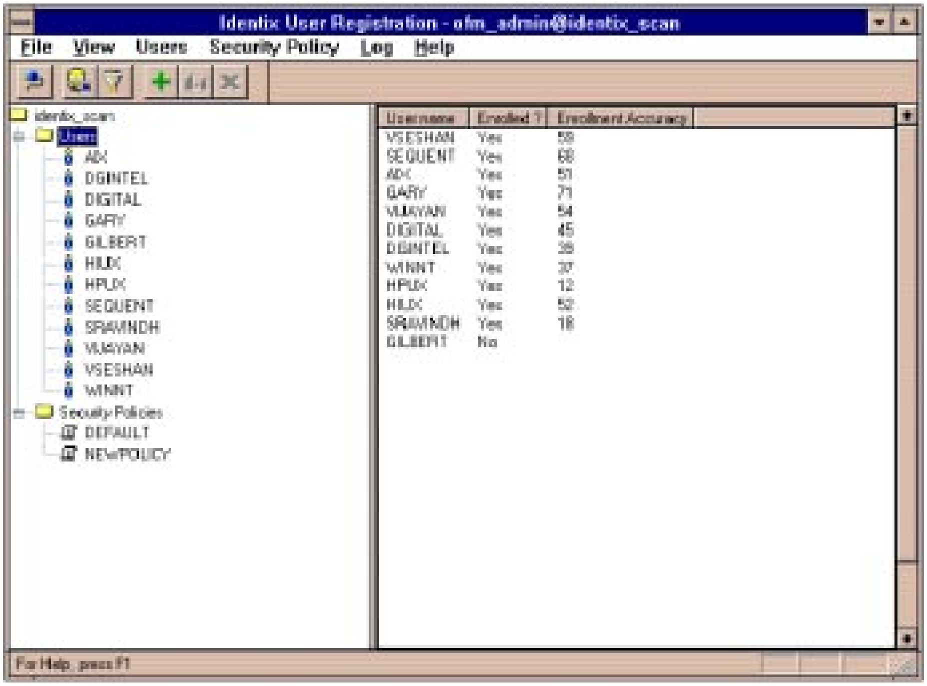
For Help, press F1

Double-click the *identix_scan* folder to expand the object tree. Two folders will appear under the *Identix_scan* folder: *Users* and *Security Policies*. You can expand or contract the object tree or any of its folders by clicking the [+] or [-] boxes, respectively.

6.8.2.2 The Properties Window

The Properties window is located on the right side of the screen. It initially displays a graphic along with application and user information. The contents of this window will change depending on what you select on the object tree. The Properties window can display summary or detail information on a folder's contents when you click on a folder in the Object Tree window. See Figure 6–8, "Properties Window with Summary Information", or Figure 6–9, "Properties Window with Detail Information".

Figure 6–8 Properties Window with Summary Information

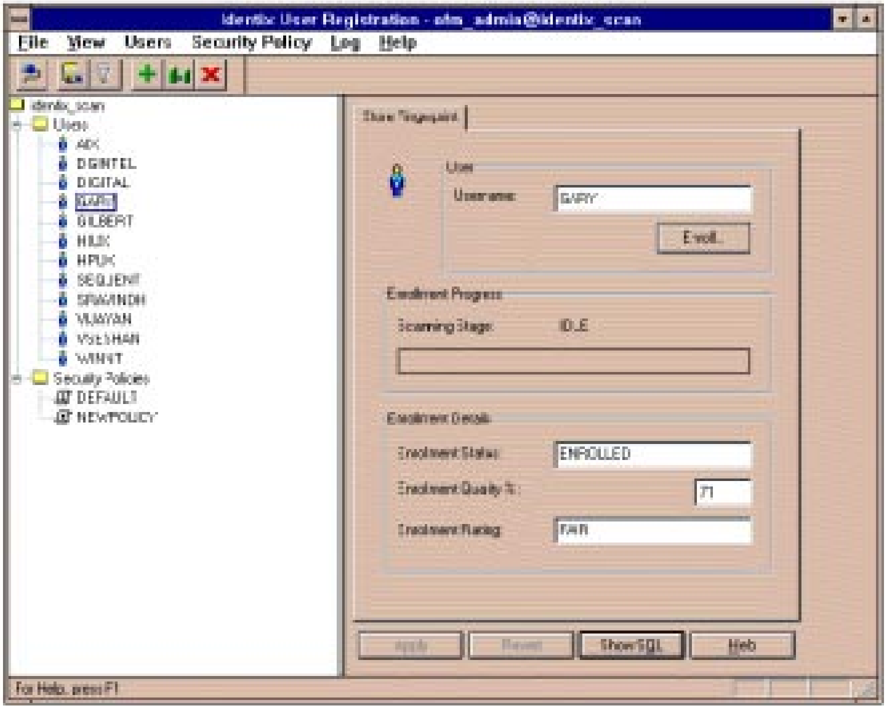


Username	Enrolled ?	Enrollment Accuracy
VSESHAN	Yes	59
SEQUENT	Yes	68
AIX	Yes	51
GARY	Yes	71
VIJAYAN	Yes	54
DIGITAL	Yes	45
DGINTEL	Yes	38
WINNT	Yes	37
HPUX	Yes	12
HIUX	Yes	52
SRAVINDH	Yes	18
GILBERT	No	

6.8.2.2.1 Sorting Summary Data in the Properties Window The Properties window with summary information contains a list of items that can be sorted by clicking on each column heading. For example:

- Click *User Names* to sort the items alphabetically by name
- Click *Enrolled ?* to sort the items alphabetically by Yes / No
- Click *Enrollment Accuracy* (fingerprint accuracy) to sort the items numerically by number

Figure 6–9 Properties Window with Detail Information



Username: GARY

Scanning Stage: IDLE

Enrollment Status: ENROLLED

Enrollment Quality %: 71

Enrollment Rating: FAIR

6.9 Troubleshooting

Check the following if you encounter any problems while installing or using the Biometric Authentication Adapter.

1. Ensure that the Identix Set Key utility hash key exactly matches the Biometric manager DEFAULT Policy hash key.
2. The NT user name must exactly match the externally defined user name in the database server and the user name used when adding the user with the Biometric Manager.

3. Domain naming must be consistent. For example, if the local naming configuration (TNSNAMES.ORA) uses .world as an appendix to the service name, then the profile (SQLNET.ORA) must reflect this naming convention for the service name. For example:

```
TNSNAMES.ORA
biometrics.world = (DESCRIPTION =
                    (ADDRESS_LIST =
                     (ADDRESS =
                       ...
SQLNET.ORA
sqlnet.identix_fingerprint_database=biometrics.world
```

4. It is possible to use one database for both the biometric authentication service and the production database; however, this is not recommended. If you do this, add the following line of code to the local naming configuration fiel (TNSNAMES.ORA) on the server and on each PC client.

```
(security = (Authentication_service = NONE))
```

7

Choosing and Combining Authentication Services

This chapter describes how to use conventional username/password authentication even if you have configured another authentication service. It also discusses how to configure your network to use one or more authentication services in your network using the Oracle Advanced Networking Option and how to set up more than one authentication service on a client or on a server.

Authentication adapters available with this release include the following:

- CyberSAFE
  Refer to Chapter 3, “Configuring the CyberSAFE Authentication Adapter”.
- Kerberos
  Refer to Chapter 4, “Configuring the Kerberos Authentication Adapter”.
- SecurID
  Refer to Chapter 5, “Configuring Oracle for Use with the SecurID Adapter”.
- Biometric (Identix)
  Refer to Chapter 6, “Configuring and Using the Identix Biometric Authentication Adapter”.
- DCE GSSAPI
  Refer to Chapter 8, “Configuring the DCE GSSAPI Authentication Adapter”.

Refer to the individual chapters and the platform-specific documentation listed above for details of configuring these adapters.

> **Note:** Use the Oracle Net8 Assistant to edit client and server SQLNET.ORA files for CyberSAFE, Kerberos, SecurID, Biometric, and DCE GSSAPI adapters.

7.1 Connect with a Username/Password When Authentication Has Been Configured

To connect to an Oracle server using a username and password when an Oracle authentication adapter has been configured, you need to configure No Authentication in your profile (SQLNET.ORA). Use the Oracle Net8 Assistant to configure the profile (SQLNET.ORA).

7.1.1 Configure No Authentication

Configure the profile for no authentication when you want to disable authentication. For example, for users to be able to log into an Oracle database server using *username/password*, you must disable authentication by defining this value. If you do, the profile appears as follows:

```
SQLNET.AUTHENTICATION_SERVICES = (NONE)
```

A user can now connect to a database using the following username/password format:

```
% sqlplus username/password@service_name
```

For example:

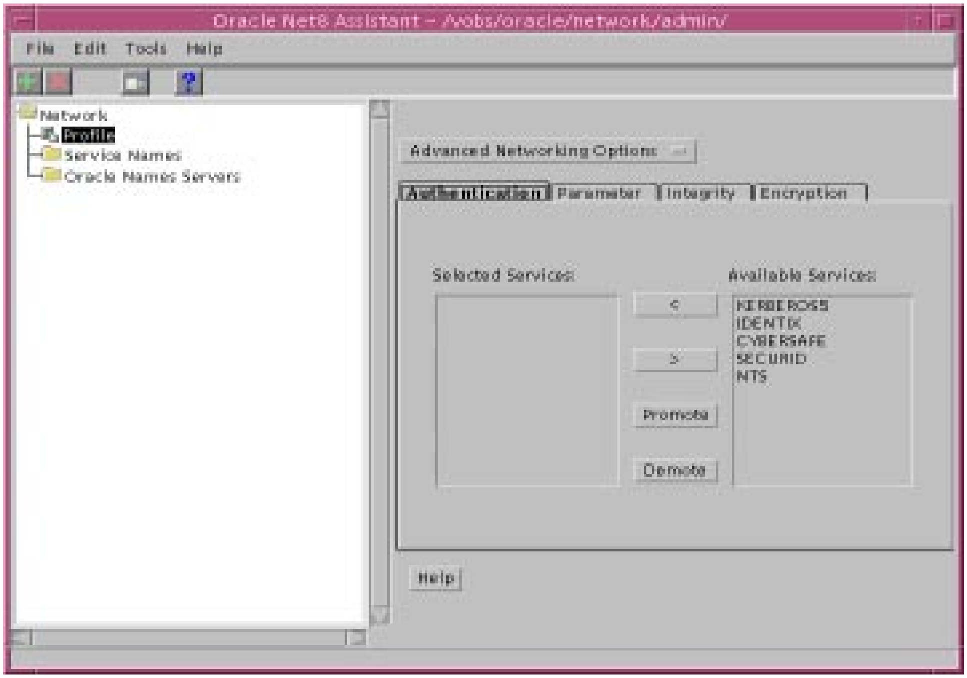
```
% sqlplus scott/tiger@emp
```

Refer to Figure 7–1, “Select No Authentication”, for an example of how you use the Oracle Net8 Assistant to configure No Authentication. To configure No Authentication:

1. Click the Profile folder on the Oracle Net8 Assistant Object Tree.
2. Click the Authentication tab in the right Properties window.
3. Click a service listed in the Selected Services area.

4. Click [>] to transfer the selected service to the Available Services area.
5. Repeat steps 3 and 4 above until all services are removed from the Selected Services area.

Figure 7–1 Select No Authentication



7.2 Set Up an Oracle Server With Multiple Authentication Services

Many networks use more than one authentication service on a single security server. For this reason, the Oracle Advanced Networking Option allows you to configure your network so that Oracle clients can use a specific authentication service and Oracle Servers can accept any service specified.

This section describes how to set up an Oracle server that uses multiple authentication adapters. Depending on which authentication adapter the client is using, the server will pick one from the list of configured adapters. Following are examples of profiles (SQLNET.ORA) using multiple authentication adapters.

Server Side

The profile for the Oracle server that uses either SecurID or CyberSAFE for authentication must contain the line:

```
SQLNET.AUTHENTICATION_SERVICES=(SECURID,CYBERSAFE)
```

Client Side Using SecurID

The profile for the Oracle client that uses SecurID must contain the line:

```
SQLNET.AUTHENTICATION_SERVICES=(SECURID)
```

Using this configuration, the Oracle server will accept connections from clients using SecurID for the authentication service. This gives you flexibility in your network configuration.

Client Side Using CyberSAFE

The profile for the Oracle client that uses CyberSAFE must contain the line:

```
SQLNET.AUTHENTICATION_SERVICES=(CYBERSAFE)
```

Using this configuration, the Oracle server will accept connections from clients using CyberSAFE for the authentication service. This gives you flexibility in your network configuration.

7.3 Set Up an Oracle Client to Use Multiple Authentication Services

This section describes how to set up clients to use multiple authentication adapters. Depending on which authentication adapter the server is configured to use, the client will pick one from the list of configured adapters. The following is an example of a profile using multiple authentication adapters.

Attention: Use the Oracle Net8 Assistant to modify your profile.

Client Side

The profile for the Oracle client that uses either SecurID or CyberSAFE for authentication must contain the line:

```
SQLNET.AUTHENTICATION_SERVICES=(SECURID,CYBERSAFE)
```

Server Side Using SecurID

The profile for the Oracle server that uses SecurID to authenticate users must contain the line:

```
SQLNET.AUTHENTICATION_SERVICES=(SECURID)
```

Server Side Using CyberSAFE

The profile for the Oracle server that uses CyberSAFE to authenticate users must contain the line:

```
SQLNET.AUTHENTICATION_SERVICES=(CYBERSAFE)
```

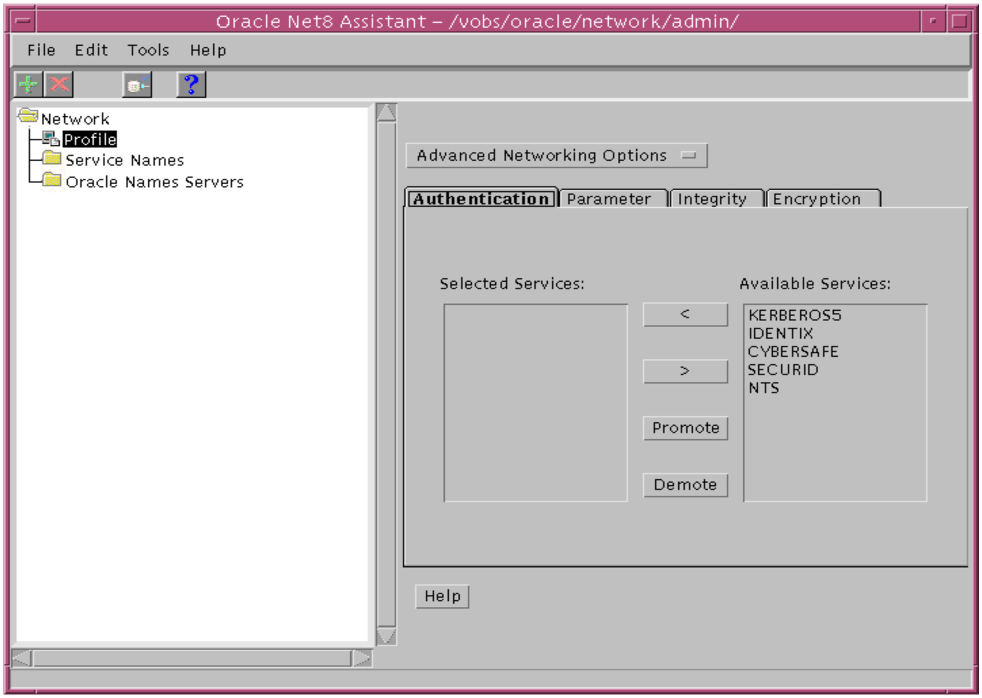
Using this configuration, the Oracle client can connect to multiple Oracle servers using different authentication services.

7.4 Use the Oracle Net8 Assistant to Set Up Multiple Authentication Services

You can use the Oracle Net8 Assistant to set up multiple authentication services on both client and server machines. Refer to Figure 7–2, “Set Up Multiple Authentication Services Using Oracle Net8 Assistant”, for a sample Oracle Net8 Assistant window you use to set up multiple authentication services on clients and servers. The following instructions apply to both clients and servers.

1. Click the Profile folder on the Oracle Net8 Assistant Object Tree.
2. Click the Authentication tab in the right Properties window.
3. Click a service listed in the Available Services area.
4. Click [<] to transfer the selected service to the Selected Services area.
5. Repeat steps 3 and 4 above until you have added all your required services to the Selected Services area.
6. Arrange the authentication services in the order of desired use by clicking a service and clicking either [Promote] or [Demote].
7. Authentication will occur starting with the first service listed at the top of the Selected Services list.

Figure 7–2 Set Up Multiple Authentication Services Using Oracle Net8 Assistant



Selected Services:

Available Services:
KERBEROS5
IDENTIX
CYBERSAFE
SECURID
NTS

8

Configuring the DCE GSSAPI Authentication Adapter

The DCE GSSAPI authentication adapter enables you to use DCE authentication even if you do not use other portions of the Oracle DCE Integration product in your environment.

Note: If you are already using Oracle DCE Integration, you do not also have to use the DCE GSSAPI authentication adapter. The Oracle DCE Integration product described in Part II, “Oracle Advanced Networking Option and Oracle DCE Integration”, includes DCE authentication.

This chapter describes how to configure and use the DCE GSSAPI authentication adapter. It describes the following four steps:

1. “Create the DCE Principal”
2. “Set Up Parameters to Use the New DCE Principal, and Turn On DCE GSSAPI Authentication”
3. “Set Up the Account You Will Use to Authenticate to the Database”
4. “Connect to an Oracle Server Using DCE GSSAPI Authentication”

Note: The instructions in this chapter assume that you are familiar with DCE terminology. For more information about DCE, refer to Part II,"Oracle Advanced Networking Option and Oracle DCE Integration", in this guide, your operating system-specific DCE administration guide, and the documentation listed in the "Preface".

8.1 Create the DCE Principal

To create the DCE principal used by the Oracle Server to validate authentication, type the commands below shown in **bold** typeface. These instructions assume the Oracle Server principal is named "oracle_server". Type the following commands on the database server.

```
% su
  password:(root password is not echoed)
# dce_login cell_admin cell_admin_password
# rgy_edit
Current site is: registry server at
    /.../cellname/subsys/dce/sec/master
rgy_edit=> do p
Domain changed to: principal
rgy_edit=> add oracle_server
rgy_edit=> do a
Domain changed to: account
rgy_edit=> add oracle_server -g none -o none -pw oracle_server_password -mp
cell_admin_password
rgy_edit=> ktadd -p oracle_server -pw oracle_server_password
rgy_edit=> quit
bye
```

8.2 Set Up Parameters to Use the New DCE Principal, and Turn On DCE GSSAPI Authentication

The following instructions assume that the Oracle Server principal is named "oracle_server".

Add the following lines to the SQLNET.ORA file. (This file is probably found in <ORACLE_HOME> / NETWORK / ADMIN.)

```
SQLNET.AUTHENTICATION_GSSAPI_SERVICE=/.../cellname/oracle_server
SQLNET.AUTHENTICATION_SERVICES=(DCEGSSAPI)
```

Note: The Oracle Server principal name used above must be a fully qualified name, including the cell name.

8.3 Set Up the Account You Will Use to Authenticate to the Database

Create the DCE principal used by the Oracle client to connect to the database. The following instructions assume the Oracle client principal is named ”oracle”.

```
% dce_login cell_admin cell_admin_password
% rgy_edit
Current site is : registry server at /.../cellname/subsys/dce/sec/master
rgy_edit=> do p
Domain changed to: principal
rgy_edit=> add oracle
rgy_edit=> do a
Domain changed to: account
rgy_edit=> add oracle -g none -o none -pw oracle_client_password -mp
cell_admin_password
rgy_edit=> quit
bye
```

Create the Oracle database user account. The following instructions show how to use the Oracle Server Manager to do this.

```
% svrmgrl

Oracle Server Manager Release 2.3.3.0.0 -Production
Copyright (c) Oracle Corporation 1994,1995. All rights reserved.
Oracle8 Server Release 8.0.3.0.0 -Production Release
With the distributed, heterogeneous, replication, objects, parallel query,
Parallel Server and Spatial Data options
PL/SQL Release 8.0.3.0.0 - Production

SVRMGR> connect internal
Connected

SVRMGR> create user ”/.../CELLNAME/ORACLE” identified externally;
Statement processed.

SVRMGR> grant connect to ”/.../CELLNAME/ORACLE”;
Statement processed.

SVRMGR> exit
```

```
Server Manager complete.
```

Note: The Oracle client principal name must be a fully qualified principal (including full cell designation), must be in uppercase, and must be enclosed within quotes.

8.4 Connect to an Oracle Server Using DCE GSSAPI Authentication

The following instructions assume the Oracle Server principal is ”oracle_server”, the Oracle client principal is ”oracle”, and the database service name is ”sales”.

1. If your DCE authentication is not already encapsulated into the operating system authentication, log in:

   ```
   % dce_login <oracle_client_principal> <oracle_client_password>
   ```

   For example:

   ```
   % dce_login oracle oraclnt
   ```

2. Connect to the Oracle database using DCE GSSAPI authentication.

   ```
   % sqlplus /@<database_service_name>
   ```

   For example:

   ```
   % sqlplus /@sales
   ```

Part II

Oracle Advanced Networking Option and Oracle DCE Integration

The following chapters of the *Oracle Advanced Networking Option Administrator's Guide* describe Oracle Distributed Computing Environment (DCE) Integration.

- Chapter 9, "Overview of Oracle DCE Integration"
- Chapter 10, "Configuring DCE for Oracle DCE Integration"
- Chapter 11, "Configuring Oracle for Oracle DCE Integration"
- Chapter 12, "Connecting to an Oracle Database in DCE"
- Chapter 13, "DCE and Non-DCE Interoperability"

In addition to the features described in this section, the Oracle Advanced Networking Option includes the following features:

- Security and single sign-on

  Refer to Part I, "Oracle Advanced Networking Option Security and Single Sign-On" , for detailed information.

9

Overview of Oracle DCE Integration

This chapter provides brief descriptions of the Distributed Computing Environment (DCE) and the Oracle DCE Integration product. For more detailed information, see the list of related books and papers in “Related Publications” in the Preface of this guide.

This information includes the following:

- Section 9.1, “System Requirements”
- Section 9.2, “Backward Compatibility”
- Section 9.3, “Overview of Distributed Computing Environment (DCE)”
- Section 9.4, “Overview of Oracle DCE Integration”

9.1 System Requirements

Oracle DCE Integration requires Net8 or higher and Oracle8.0.3. It enables Oracle applications and tools to access Oracle8 servers in a DCE environment.

Note: Oracle DCE Integration is based on the Open Software Foundation (OSF) DCE V1.0 and V1.1, and will be compatible with OSF's future DCE releases.

Note: OSF recently merged with another standards group, X/OPEN, to form The Open Group. This group will continue to support DCE.

9.2 Backward Compatibility

Oracle servers running DCE Integration 2.3.2 and later are backward compatible with clients running SQL*Net/DCE 2.1.6 or 2.2.3; however, the 2.1.6 clients will not be able to take advantage of external roles.

A DCE Integration 2.3.2 or later client will *not* be able to connect to a SQL*Net/DCE 2.1.6 or 2.2.3 server. A DCE Integration release 2.3.2 or later client requires a 2.3.2 or later server in order to connect to a database.

9.3 Overview of Distributed Computing Environment (DCE)

The Distributed Computing Environment (DCE) from the Open Software Foundation (OSF) is a set of integrated network services that work across multiple systems to provide a distributed environment. The network services include remote procedure calls (RPCs), directory service, security service, threads, distributed file service, diskless support, and distributed time service.

DCE is the middleware between distributed applications and the operating system/network services and is based on a client/server model of computing. By using the services and tools that DCE provides, users can create, use, and maintain distributed applications that run across a heterogeneous environment.

For more detailed information on DCE, see "Related Publications" in the Preface of this guide.

9.4 Overview of Oracle DCE Integration

Oracle DCE Integration enables users to use Oracle tools and applications to access Oracle8 servers in a DCE environment. Oracle's DCE Integration product is comprised of the following parts:

9.4.1 DCE Communication/Security Adapter

The DCE Communication / Security Adapter component includes:

- **Authenticated RPC**—Oracle DCE Integration provides authenticated RPC (Remote Procedure Call) as the transport mechanism which enables multi-vendor interoperability. RPC also uses some of the other DCE services, including directory and security services, to provide location transparency and secure distributed computing.
- **Integrated Security and Single Sign-On**—Oracle DCE Integration works with the DCE Security service to provide security within DCE cells. It enables a user logged onto DCE to securely access any Oracle database without having to specify a username or password. This is sometimes referred to as *external authentication* to the database. It is also known as *single sign-on*. Clients and servers that are not running DCE authentication services can interoperate with systems that have DCE security by specifying an Oracle password.
- **Data Privacy and Integrity**—Oracle DCE Integration uses the multiple levels of security that DCE provides to ensure data authenticity, privacy and integrity. For example, users have a range of choices from no protection to full encryption for each connection, with a guarantee that no data has been modified in transit.

Note: For parts of your network that do not use DCE, you may want to use the other security and authentication services included with the Oracle Advanced Networking Option. These services (formerly included in Secure Network Services) work with SQL*Net release 2.1 and above. They provide message integrity and data encryption services in non-DCE environments, allowing administrators to ensure that all network traffic is protected against unauthorized viewing or modification, regardless of the start or end point. For more information on these services see Part I, "Oracle Advanced Networking Option Security and Single Sign-On", of this guide.

9.4.2 DCE CDS Native Naming Adapter

The DCE CDS Native Naming adapter component includes:

- **Naming and Location Transparency**—DCE Integration registers Oracle8 connect descriptors in the DCE Cell Directory Service (CDS), allowing them to be transparently accessed across the entire DCE environment. Users can connect to Oracle database servers in a DCE environment using familiar Oracle service names.

  The DCE Cell Directory Service offers a distributed, replicated repository service for name, address and attributes of objects across the network. Because servers register their name and address information in the Cell Directory Service (CDS), Oracle clients can make location-independent connections to Oracle8 servers. Services can be relocated without any changes to the client configuration. An Oracle utility is provided to load the Oracle service names (with corresponding connect descriptors) into CDS. After this is done, Oracle connect descriptors can be viewed from a central location with standard DCE tools.

  For location of services across multiple cells, either of the following options may be used:

  - DCE Global Directory Service (GDS)
  - Internet Domain Naming Service (DNS)

For more information about the DCE CDS Native Naming Adapter see the following:

- To configure DCE to use the CDS naming adapter, see Chapter 10, “Configuring DCE for Oracle DCE Integration”.
- To configure Oracle clients and servers to use CDS, see Chapter 11, “Configuring Oracle for Oracle DCE Integration”.
- To read about how Oracle Native Naming adapters work with other Oracle name services, refer to the *Oracle Net8 Administrator’s Guide*.

9.4.3 Flexible DCE Deployment

Oracle Advanced Networking Option provides you flexibility in your use of DCE services. You have the following options:

- You can use full DCE Integration in your environment to integrate with all the DCE Secure Core services (RPC, directory, security, threads) described in this part of the guide.

- You can choose to use only the DCE directory services by using the DCE CDS Native Naming Adapter, along with any conventional protocol adapter, such as TCP/IP. Configuration of the CDS Native Naming adapter is described in Chapter 10, “Configuring DCE for Oracle DCE Integration” and Chapter 11, “Configuring Oracle for Oracle DCE Integration” in this guide. For an overview of how Native Naming adapters work with other Oracle name services, refer to the *Oracle Net8 Administrator’s Guide.*
- You can use only DCE authentication services by using the DCE GSSAPI authentication adapter described in Chapter 8, “Configuring the DCE GSSAPI Authentication Adapter” of this guide. This requires OSF DCE 1.1.

9.4.4 Limitations in This Release

- Only one listener address that uses the DCE protocol is allowed per node.
- Database links must specify a username and password to connect.
- This release of the DCE Integration adapter does not support the Oracle Multi-Protocol Interchange.
- This release does not work with the Oracle Multi-Threaded Server (MTS).

10

Configuring DCE for Oracle DCE Integration

This chapter describes what you need to do to configure DCE to use Oracle DCE Integration after Oracle DCE Integration has been successfully installed.

This information includes the following:

- Section 10.1, “Overview”
- Section 10.2, “Create New Principals and Accounts”
- Section 10.3, “Install the Key of the Server into a Keytab File”
- Section 10.4, “Configuring DCE CDS for Use by Oracle DCE Integration”

For detailed information on DCE, see the list of books and papers in the “Related Publications” section in the Preface of this guide.

10.1 Overview

Following is a list of steps with examples you need to follow to configure DCE to use DCE Integration. The steps assume that a DCE cell has been configured and the machines being used are part of that cell.

As the DCE cell administrator, you will need to do the following:

- "Create New Principals and Accounts" for all users in your organization
- "Install the Key of the Server into a Keytab File"
- "Create Oracle Directories in the CDS Namespace"
- "Give Servers Permission to Create Objects in the CDS Namespace"
- "Load Oracle Service Names Into CDS"

10.2 Create New Principals and Accounts

First, you need to add server principals using a procedure like the one below:

```
% dce_login cell_admin password
% rgy_edit
Current site is: registry server at
  /.../cell1/subsys/dce/sec/master
rgy_edit=>do p
Domain changed to: principal
rgy_edit=> add oracle
rgy_edit=> do a
Domain changed to: account
rgy_edit=> add oracle -g none -o none -pw oracle_password
  -mp cell_admin_password
rgy_edit=> quit
bye
```

You just created a DCE principal called "oracle". The principal has a corresponding account with password "*password*". The account does not belong to any DCE group or DCE profile.

You only need to do this once after DCE Integration has been installed. Also, you only need to do this procedure for the Oracle database server, not for the client.

10.3 Install the Key of the Server into a Keytab File

In this step by step procedure, you install the key of the server into a keytab file: dcepa.key. This keytab file contains the password of the principal under which the

Net8 listener starts. The Net8 listener reads this file to authenticate itself to DCE. You only need to do this once after DCE Integration has been installed. Also, you only need to do this procedure for the Oracle database server, not for the client.

> **Note:** Remember to substitute the correct full pathname for the $ORACLE_HOME variable. If the specified directories do not already exist, you will need to create it before running the command. Type the following to create the directories.
>
> ```
> mkdir $ORACLE_HOME/dcepa
> mkdir $ORACLE_HOME/dcepa/admin
> ```

Run the following command to generate the keytab file.

```
% dce_login cell_admin password
% rgy_edit
Current site is: registry server at /.../cell1/subsys/dce/sec/master
rgy_edit=> ktadd -p oracle -pw Oracle_password -f
$ORACLE_HOME/dcepa/admin/dcepa.key
rgy_edit=>quit
bye
```

10.4 Configuring DCE CDS for Use by Oracle DCE Integration

The `/.:/subsys/oracle/names` directory contains objects that map Net8 service names to connect descriptors, which are used by the CDS naming adapter.

The `/.:/subsys/oracle/service_registry` directory also contains objects that map the service name in DCE addresses to the network endpoint which is used by both DCE protocol adapter clients and servers.

10.4.1 Create Oracle Directories in the CDS Namespace

You need to perform the steps in this section after installing the DCE Integration Adapter for the first time in a cell.

```
% dce_login cell_admin
Enter Password:    (password not displayed)

$ cdscp
cdscp> create dir /.:/subsys/oracle
```

```
cdscp> create dir /.:/subsys/oracle/names
cdscp> create dir /.:/subsys/oracle/service_registry
cdscp> exit
```

Note: Create these directories on all CDS replicas.

10.4.2 Give Servers Permission to Create Objects in the CDS Namespace

Perform the following steps to add the principal `oracle` to the `cds-server` group.

```
$ dce_login cell_admin
Enter Password:    (password not displayed)
$ rgy_edit
rgy_edit=> domain group
Domain changed to: group
rgy_edit=> member subsys/dce/cds-server -a oracle
rgy_edit=> exit
```

10.4.3 Load Oracle Service Names Into CDS

Refer to Section 11.6, “Configuring Clients to Use the DCE CDS Naming Adapter” for instructions on how to configure clients, and to Section 11.6.3, “Create a TNSNAMES.ORA For Loading Oracle Connect Descriptors into CDS” for information on how to load Oracle service names into CDS.

11

Configuring Oracle for Oracle DCE Integration

This chapter discusses how to configure Oracle and Net8 to use Oracle DCE Integration after it has been successfully installed. The following sections describe the parameters you need to configure for servers and clients.

- Section 11.1, “DCE Address Parameters”
- Section 11.2, “Configuring the Server”
- Section 11.3, “Creating and Naming Externally-Authenticated Accounts”
- Section 11.4, “Setting up DCE Integration External Roles”
- Section 11.5, “Configuring the Client”
- Section 11.6, “Configuring Clients to Use the DCE CDS Naming Adapter”

11.1 DCE Address Parameters

DCE addresses in the LISTENER.ORA and TNSNAMES.ORA configuration files are defined by DCE parameters. These parameters consist of both mandatory and optional fields, which are described below:

```
    ADDRESS=(PROTOCOL=DCE)
             (SERVER_PRINCIPAL=server_name)
             (CELL_NAME=cell_name)
             (SERVICE=dce_service_name))
```

where:

PROTOCOL is a mandatory field that identifies the DCE RPC protocol.

SERVER_PRINCIPAL is a mandatory field for the server and an optional field for the client. The server authenticates itself to DCE as this principal. This field is mandatory in the listener configuration file (LISTENER.ORA) and specifies the principal the server will start under. This field is optional in your local naming configuration file (TNSNAMES.ORA) and specifies the principal of the server the client must connect to. If not specified, then one-way authentication is used. In this case, the client does not care what principal the server is running under.

CELL_NAME is an optional parameter. If present, it specifies the DCE cell name of the database. If this parameter is not set, the cell name defaults to the local cell (useful for single-cell environments). Optionally, the SERVICE parameter (described below) may specify the complete path (including the cell name) to the service, making this parameter unnecessary.

SERVICE is a mandatory field for both server and client. For the server, this is the service registered with CDS. For the client, this is the service name used when querying CDS for the location of the Oracle DCE servers. The default directory for storing service names in CDS is /.../cell_name/subsys/oracle/service_registry. This service name can fully specify the path in CDS.

You can specify a service as:

```
SERVICE=/.../cell_name/subsys/oracle/service_registry/dce_service_name
```

or it can be specified as

```
SERVICE=dce_service_name
```

provided that CELL_NAME=cell_name is also specified.

A third option is to specify SERVICE=dce_service_name, in which case the cell name defaults to the local cell. However, this third way of specifying service names only works well if you are working within a single cell.

Note: The dce_service_name in the service field may or may not be the same as the service name used by Net8. The service name used by Net8 is mapped to the connect descriptor in a local naming configuration file (TNSNAMES.ORA). The dce_service_name is part of the address within the connect descriptor.

Note: In this DCE Integration release, the configuration files LISTENER.ORA, SQLNET.ORA, TNSNAMES.ORA, and PROTOCOL.ORA are located in the $ORACLE_HOME/network/admin directory. The INIT<SID>.ORA file is located in the $ORACLE_HOME/dbs directory.

11.2 Configuring the Server

To configure a server for DCE Integration, you need to configure the following Net8 files with DCE address and parameter information as described in Section 11.1, “DCE Address Parameters” and in the following sections.

Note: Use the Oracle Net8 Assistant to create the necessary configuration files. For explanations of the configuration files, refer to the *Oracle Net8 Administrator’s Guide.*

Note the following prerequisites:

- Listener configuration file (LISTENER.ORA) must be configured with DCE address information for all servers.
- Profile (SQLNET.ORA) and PROTOCOL.ORA need to be configured for servers in distributed systems that will need to make database link connections to other servers.

11.2.1 LISTENER.ORA Parameters

For a database server to receive connections from Net8 clients in a DCE environment, there must be a Net8 listener active on the server platform. A listener listens for connections on a network address that is defined in the listener configuration file (LISTENER.ORA).

The SERVER_PRINCIPAL parameter designates what DCE principal the listener should be running under. In the sample below, the listener is running under principal ”oracle”.

11.2.2 Sample DCE Address in LISTENER.ORA

Below is a sample DCE address as it would appear in the LISTENER.ORA file.

```
LSNR_DCE=
  (ADDRESS=
      (PROTOCOL=DCE)
      (SERVER_PRINCIPAL=oracle)
      (CELL_NAME=cell1)
      (SERVICE=dce_svc))
SID_LIST_LSNR_DCE=
      (SID_DESC=
        (SID_NAME=ORASID)
         (ORACLE_HOME=/private/oracle7))
```

11.3 Creating and Naming Externally-Authenticated Accounts

To use DCE authentication for logging onto the Oracle database, you need to create database accounts that are ”authenticated externally”.

Refer to *Oracle8 Distributed Database Systems* for more information on external authentication.

To enable secure external authentication, do the following:

1. Verify that these lines are in the INIT<*SID*>.ORA file:

   ```
   REMOTE_OS_AUTHENT=FALSE
   OS_AUTHENT_PREFIX=“”
   ```

2. Verify that the INIT<*SID*>.ORA file does not have a multi-threaded server (MTS) entry for DCE. For example, an entry such as the following is *not* allowed:

   ```
   mts_dispatchers=“dce, 3”
   ```

3. Make sure that you are logged in as a member of the DBA group. Restart the database instance for the changes to take effect.

4. At the server manager prompt, define users. Before doing so, decide whether you are, or ever will be, operating in a multi-cell DCE environment in which you will allow Oracle access across cell boundaries. The way you define users depends on whether they will be connecting within a single cell, or across cell boundaries.

   > **Note:** The privileges shown in the remainder of this section are the minimum privileges necessary. The actual set of privileges needed depends on the instance and / or application.

   If users will be connecting within a local cell, use the following format.

   ```
   svrmgr1> create user SERVER_PRINCIPAL identified externally;
   svrmgr1> grant create session to SERVER_PRINCIPAL;
   ```

   For example:

   ```
   svrmgr1> create user oracle identified externally;
   svrmgr1> grant create session to oracle;
   ```

   > **Note:** The entire CELL_NAME / SERVER_PRINCIPAL string must be 15 characters or less.
   >
   > For example:
   >
   > ```
   >         svrmgr1> create user ”CELL1/ORACLE” identified externally;
   > svrmgr1> grant create session to ”CELL1/ORACLE”;
   > ```

   If connecting to the database across multiple cells, specify both the *CELL_NAME* and the *SERVER_PRINCIPAL*.

   ```
   svrmgr1> create user “CELL_NAME/SERVER_PRINCIPAL” identified externally;
   svrmgr1> grant create session to ”CELL_NAME/SERVER_PRINCIPAL”;
   ```

   > **Attention:** You must enclose the externally-identified account name in double quotes, because the slash is a reserved character. Also, if the account (user) name is double-quoted, it must be capitalized.

For example:

```
svrmgr1> create user ”CELL1/ORACLE” identified externally;
svrmgr1> grant create session to ”CELL1/ORACLE”;
```

Note: When using the above format, set the following parameter in PROTOCOL.ORA to FALSE:

```
dce.local_cell_usernames=false
```

Note: References to an Oracle account created in this manner must include the schema / account in the correct format. For example, consider requests for access to tables from another account. When a user references the tables in another account created within a local cell, the command might be:

```
SQL> select * from oracle.emp
```

If a user wants to access tables in an another account created for connections across cells, the command might be:

```
SQL> select * from ”CELL1/ORACLE”.emp
```

11.4 Setting up DCE Integration External Roles

To set up external roles for DCE Integration, do the following:

1. Set the following parameter in the INIT<*SID*>.ORA file.

   ```
   OS_ROLES=TRUE
   ```

   Then restart the database.

2. Make sure that the DCE groups that map to Oracle roles adhere to the following syntax:

   ```
   ORA_<SID>_<ROLE>[_[A][D]]
   ```

   where:

ORA	Designates that this group is used for Oracle purposes
<SID>	Is the Oracle System Identifier
<ROLE>	Is the name of the role, as defined in the data dictionary
A	Optional character indicating that the user has admin privileges for this role.
D	Optional character indicating the role is to be enabled by default at connect time.

> **Note:** For more details on external roles see the *Oracle8 Administrator’s Guide.*

3. DCE authenticate to a user who is a member of a DCE group by performing a `dce_login` and a `klist` command. (Below is some sample output from the `dce_login` and `klist` commands.)

   Note: The DCE group must adhere to the syntax described in step 2.

   ```
   % dce_login oracle
   Enter Password:
   % klist
   DCE Identity Information:
           Warning: Identity information is not certified
           Global Principal: /.../ilab1/oracle
           Cell:      001c3f90-01f5-1f72-ba65-02608c2c84f3 /.../ilab1
           Principal: 00000068-0568-2f72-bd00-02608c2c84f3 oracle
           Group:     0000000c-01f5-2f72-ba01-02608c2c84f3 none
           Local Groups:
         0000000c-01f5-2f72-ba01-02608c2c84f3 none
         0000006a-0204-2f72-b901-02608c2c84f3 subsys/dce/cds-server
         00000078-daf4-2fe1-a201-02608c2c84f3 ora_dce222_dba
         00000084-89c8-2fe8-a201-02608c2c84f3 ora_dce222_connect_d
         00000087-8a13-2fe8-a201-02608c2c84f3 ora_dce222_resource_d
         00000080-f681-2fe1-a201-02608c2c84f3 ora_dce222_role1_ad
         .
         .
         .
   ```

4. Connect to the database as usual.

   Following is some sample output showing a connection to a database and a listing of external roles (DBA, CONNECT, RESOURCE, and ROLE1) that have been mapped to DCE groups.

   ```
   % sqlplus /@test_222

   SQL*Plus: Release 3.2.2.0.0 - Production on Thu Aug 31 11:24:12 1995

   Copyright (c) Oracle Corporation 1979, 1994.  All rights reserved.

   Connected to:
   Oracle7 Server Release 7.2.2.3.0 - Production Release
   ```

```
PL/SQL Release 2.2.2.3.0 - Production

SQL> select * from session_roles;

ROLE
------------------------------
CONNECT
RESOURCE
ROLE1

SQL> set role all;

Role set.

SQL> select * from session_roles;

ROLE
------------------------------
DBA
EXP_FULL_DATABASE
IMP_FULL_DATABASE
CONNECT
RESOURCE
ROLE1

6 rows selected.

SQL> exit
Disconnected from Oracle7 Server Release 7.2.2.3.0 - Production Release
PL/SQL Release 2.2.2.3.0 - Production
% logout
```

11.5 Configuring the Client

To configure a client for DCE Integration, you need to configure the following Net8 files with DCE address and parameter information, as described in “Description of the DCE Address Parameters” and below:

- PROTOCOL.ORA
- SQLNET.ORA

Typically, CDS is used for name resolution. Thus, a local naming configuration file (TNSNAMES.ORA) is not used, except when loading names and addresses into CDS. See Section 11.6, “Configuring Clients to Use the DCE CDS Naming Adapter”.

11.5.1 Description of Parameters in PROTOCOL.ORA

In this release of DCE Integration, there are four DCE parameters located in PROTOCOL.ORA. Each parameter begins with the prefix ”DCE**.**” to distinguish it from parameters relevant to other protocols. If default values are used for these four parameters, DCE Integration does not require a PROTOCOL.ORA file. The parameters and their current defaults are as follows:

```
DCE.AUTHENTICATION=dce_secret
DCE.PROTECTION=pkt_integ
DCE.TNS_ADDRESS_OID=1.3.22.1.5.1
DCE.LOCAL_CELL_USERNAMES=TRUE
```

Note: The default for DCE.LOCAL_CELL_USERNAMES is now TRUE. (It was set to FALSE in the DCE Integration 2.1.6 release.)

Configuration parameters are not case-sensitive: you can enter them in either upper-case or lower-case.

Note: If the DCE.AUTHENTICATION entry is not specified, cell-wide default authentication is used.

If the DCE.PROTECTION entry is not specified, cell-wide default protection is used.

DCE.AUTHENTICATION. This parameter is optional. It indicates the authentication value to be used for each DCE RPC. The client’s DCE_AUTHENTICATION value must be the same as the server’s DEC_AUTHENTICATION value. The choices are:

NONE: No authentication.

DCE_SECRET: DCE shared-secret key authentication (Kerberos).

DCE_SECRET: is the default authentication level.

DEFAULT: The cell default.

Note: It is recommended that **DCE_SECRET** be used for this parameter.

DCE.PROTECTION. This is an optional field. It specifies the data integrity protection levels for data transmission. The client's DCE_PROTECTION level must be equal to or greater than the server's DCE_PROTECTION level. The choices are:

NONE: Perform no protection for the current connection.

DEFAULT: Use the default cell-wide protection level.

CONNECT: Perform protection only when the client establishes a relationship with the server.

CALL: Perform protection only at the beginning of each remote procedure call when the server receives the request.

PKT: Ensure that all data received is from the expected client.

PKT_INTEG: Ensure and verify that none of the data transferred between the client and server has been modified.

PRIVACY: Perform protection as specified by all of the previous levels and also encrypt each RPC argument value and all user data in each call.

DCE.TNS_ADDRESS_OID. This optional parameter enables you to specify an alternative to the default DCE.TNS_ADDRESS_OID (shown below):

```
DCE.TNS_ADDRESS_OID=1.3.22.1.x.x
```

For information on how to determine if you need to include this parameter, and how to specify it, see Section 11.6.2, "Modify the CDS Attributes File and Restart the CDS".

DCE.LOCAL_CELL_USERNAMES. This optional parameter defines the format used to specify the principal name (username) either with or without the cell name.

> **Note:** The choice you make for this parameter should be determined by whether users will be making connections across cells, and if so, whether you have naming conventions that assure that users in different cells do not have duplicate names.

The choices are:

TRUE: This is the default. Choose TRUE when using just the SERVER_PRINCIPAL format, without the CELL_NAME. An example of a user specified in this format would be:

```
oracle
```

This choice would be appropriate if users are making connections within a single cell, or if naming conventions in your network assure that users in different cells do not have duplicate names.

FALSE: Choose FALSE when using the CELLNAME/SERVER_PRINCIPAL format. An example of a user specified in this format would be:

```
CELL1/ORACLE
```

This choice would be appropriate if users are making connections across cells and there may be users in different cells with identical names.

11.6 Configuring Clients to Use the DCE CDS Naming Adapter

Clients will typically use CDS to resolve Oracle service names to addresses. Follow the instructions below to configure CDS.

11.6.1 Enable CDS for use in Performing Name Lookup

To use CDS for name resolution, the DCE Integration CDS Naming Adapter must be installed on all clients and servers that will use CDS. Also, the CDS namespace must have been configured for use by DCE Integration. (Refer to the DCE Integration installation instructions and to Section 10.4, “Configuring DCE CDS for Use by Oracle DCE Integration” for instructions on how to install and configure the CDS Naming Adapter.) For example, a service name such as ”ORADCE” and its network address can be stored in DCE’s CDS.

Typically, users can connect to Oracle services using the familiar Oracle service name (if there are no domains or the database is in the user’s default domain): For example:

```
sqlplus /@ORADCE
```

This example assumes that DCE externally-authenticated accounts are in use.

As an alternative name resolution service, you can use a local naming configuration file (TNSNAMES.ORA) when CDS is inaccessible. To do this, you must locate names and addresses of all Oracle servers in the local naming configuration file (TNSNAMES.ORA).

11.6.2 Modify the CDS Attributes File and Restart the CDS

On all DCE machines where the CDS naming adapter will be used, add the object ID for the CDS attribute TNS_Address to the CDS attributes file. (The object ID must be the same across all machines.)

1. Add a line with the following format to the /opt/dcelocal/etc/cds_attributes file.

   ```
   1.3.22.1.5.1    TNS_Address    char
   ```

   If the default TNS_Address OID (Object Identifier) value (1.3.22.1.5.1) already exists in the cds_attributes file, then you need to specify a value for the OID that is not already in use.

   Note: The first four digits of the TNS_Address attribute value (1.3.22.1.x.y) are fixed under DCE-naming conventions.

2. If you are unable to use the default value for the OID, you will need to specify the OID in the PROTOCOL.ORA file on the client.

   If you had to specify a value other than the default (`1.3.22.1.5.1`), then you need to add the following parameter to the PROTOCOL.ORA file:

   ```
   DCE.TNS_ADDRESS_OID=1.3.22.1.x.y
   ```

   Note: Make sure that the OID value in the `cds_attributes` file matches the value specified in the `DCE.TNS_ADDRESS_OID` parameter in the PROTOCOL.ORA file.

3. Restart the CDS on the machine. (The command to restart CDS may vary from platform to platform. For example, on IBM AIX, you may use `smit` to restart the CDS.) The steps on IBM AIX are as follows:

   1. Type: **smit DCE**
   2. Choose Restart DCE/CDS Daemons
   3. Select List
   4. Select all CDS daemons available

11.6.3 Create a TNSNAMES.ORA For Loading Oracle Connect Descriptors into CDS

To load the Oracle service names and addresses into CDS, create or modify a a local naming configurationfile (TNSNAMES.ORA) containing service names (or aliases) and addresses. A sample file is shown below. The local naming configuration file (TNSNAMES.ORA) is used to map service names to addresses for use by Net8.

This section describes the parameters that the administrator needs to include in the TNSNAMES.ORA file. TNSNAMES.ORA contains a list of Oracle service names mapped to connect descriptors of destinations or endpoints in the network. The sample DCE address below shows a network address for an Oracle server with the Oracle service name ”ORADCE”. It is used to connect to the service registered as ”DCE_SVC” in the CDS directory /.../*<cell_name>*/subsys/oracle/names.

```
ORADCE=(DESCRIPTION=
        (ADDRESS=
        (PROTOCOL=DCE)
        (SERVER_PRINCIPAL=oracle)
        (CELL_NAME=cell1)
        (SERVICE=DCE_SVC))
(CONNECT_DATA=
     (SID=ORASID)))
```

Note: In this example, the Oracle service name and the DCE service name are different. However, they are often the same.

The keyword value pair PROTOCOL=DCE is mandatory. It appears in the address section of a listener configuration file (LISTENER.ORA) and in the address section of a local naming configuration file (TNSNAMES.ORA). It must be the same in both places.

The DCE parameter SERVER_PRINCIPAL is optional in a local naming configuration file (TNSNAMES.ORA).

The DCE parameter SERVICE is mandatory. The value given for the DCE parameter (SERVICE= dce_service_name) must be the same in the listener configuration file (LISTENER.ORA) and local naming configuration file (TNSNAMES.ORA).

The Oracle parameter SID is mandatory. It identifies the Oracle system ID; each SID value must be unique on a node. This parameter is strictly local and is not used in DCE CDS. For further information on the local naming configuration file (TNSNAMES.ORA), refer to the *Oracle Net8 Administrator’s Guide*.

11.6.4 Load Oracle Connect Descriptors into CDS

A separate utility called ”tnnfg” is provided with Oracle DCE Integration to load connect descriptors into CDS.

To load the Oracle service names or aliases into CDS, perform the following steps:

```
% dce_login cell_admin
% tnnfg dceload full_pathname_to_TNSNAMES.ORA
% Enter Password:  (password will not display)
```

Note: You must enter the full pathname for the TNSNAMES.ORA file in the previous command.

Also make sure that the SQLNET.ORA file exists in the same directory as the TNSNAMES.ORA file.

This procedure loads the service names in TNSNAMES.ORA into DCE’s CDS.

Note: If you configure a new service name and address in TNSNAMES.ORA, `tnnfg` will add the new service name and address to CDS.

If you change the address for a particular service name, tnnfg will update the address for a particular service name.

11.6.5 Delete or Rename TNSNAMES.ORA File

If you are using SQL*Net 2.2 or earlier, after having loaded the TNSNAMES.ORA file into DCE’s CDS, it is recommended that you rename it to another name—TNSNAMES.BAK, for example. or delete it. Otherwise, TNSNAMES.ORA may be searched instead of CDS to resolve the service name to an address.

If you are using SQL*Net 2.3 or Net8, you can keep TNSNAMES.ORA available as a backup in case CDS becomes unavailable. To assure that CDS will routinely be searched instead of TNSNAMES.ORA, configure the NAMES.DIRECTORY_PATH parameter in a profile (SQLNET.ORA), as described in Section 11.6.6, “Modify SQLNET.ORA Parameter File to Have Names Resolved in CDS”.

11.6.6 Modify SQLNET.ORA Parameter File to Have Names Resolved in CDS

The parameters required in a profile (SQLNET.ORA) depend upon the version of SQL*Net or Net8 you are using.

11.6.6.1 SQL*Net Release 2.2 or Earlier

For a client or server to use the DCE CDS Naming Adapter, the administrator needs to do the following:

- make sure that the CDS Naming Adapter has been installed on that node
- add the following two parameters and values to SQLNET.ORA:

```
native_names.use_native=true
native_names.directory_path=(dce)
```

After these parameters are added to the SQLNET.ORA file, the client's or server's name requests will be resolved in CDS instead of by a local TNSNAMES.ORA file.

Note: A client or server can use CDS to reach services on a network even if some of those services are not also using CDS.

11.6.6.2 SQL*Net Release 2.3 and Later

For a client or server to use the DCE CDS Naming Adapter, the administrator needs to do the following:

- make sure that the CDS Naming Adapter has been installed on that node
- add the following parameter to the SQLNET.ORA file:

```
NAMES.DIRECTORY_PATH=(dce, tnsnames, onames)
```

The first name resolution service listed as a value for this parameter is used. If it is unavailable for some reason, the next name resolution service is used, and so forth.

11.6.7 Connect to Oracle Servers in DCE

For information on how to connect to Oracle databases in a DCE environment, see Chapter 12, "Connecting to an Oracle Database in DCE".

12

Connecting to an Oracle Database in DCE

This chapter shows how to connect to an Oracle database after having installed Oracle DCE Integration and having configured both DCE and Oracle to use Oracle DCE Integration.

This information includes the following:

- Section 12.1, “Starting the Network Listener”
- Section 12.2, “Connecting to an Oracle Database Server in the DCE Environment”

12.1 Starting the Network Listener

To start the Net8 listener, do the following:

1. To start the listener, enter the following commands:

   ```
   % dce_login principal_name password
   %  lsnrctl start listener_name
   ```

   For example, if the listener name is LSNR_DCE in LISTENER.ORA, enter the following to start the listener:

   ```
   % dce_login oracle orapwd
   % lsnrctl start LSNR_DCE
   ```

   To make sure the server registered its binding handler with `rpcd`, enter:

   ```
   % rpccp show mapping
   ```

   In the computer response, look for the line that includes the dce_service_name that is part of the listener address.

2. To make sure the service has been created, search for the dce_service_name as follows:

   ```
   % cdscp show object “/.:/subsys/oracle/service_registry/
   dce_service_name”
   ```

   For example:

   ```
   % cdscp show object “/.:/subsys/oracle/service_registry/dce_svc”
   ```

   This shows you the mapping in the CDS namespace that the listener has chosen for the endpoint. For example:

   ```
                 SHOW
               OBJECT    /.../subsys/oracle/service_registry/dce_svc
                   AT   1995-05-15-17:10:52
   RPC_ClassVersion = 0100
          CDS_CTS = 1995-05-16-00:05:01.221106100/aa-00-04-00-3e-8c
            CDS_UTS = 1995-05-16-00:05:01.443343100/aa-00-04-00-3e-8c
          CDS_Class = RPC_Server
   CDS_ClassVersion = 1.0
         CDS_Towers = :
              Tower = ncacn_ip_tcp:144.25.23.57[]
   ```

12.2 Connecting to an Oracle Database Server in the DCE Environment

To connect to an Oracle server in the DCE environment, do one of the following:

1. After externally-identified accounts have been set up, you can take advantage of DCE authentication to log into Oracle without providing any username/password information. To use this single sign-on capability, just log in to DCE using a command like the following:

   ```
   % dce_login principal_name password
   ```

   For example:

   ```
   % dce_login oracle orapwd
   ```

   Note: You only need to enter the **dce_login** command once. If you are already logged into DCE, you do not need to log in again.

   You can now connect to an Oracle Server without using a username or password. Enter a command like the following:

   ```
   % sqlplus /@service_name
   ```

   where *service_name* is the database service name.

   For example:

   ```
   % sqlplus /@ORADCE
   ```

   Refer to “Creating and Naming Externally-Identified Accounts” for information, and to *Oracle8 Distributed Database Systems* for information on external authentication.

2. From a client, you can still connect with a username/password:

   ```
   % sqlplus username/password@service_name
   ```

   where *service_name* is the Net8 service name.

   For example:

   ```
   % sqlplus scott/tiger@ORADCE
   ```

13

DCE and Non-DCE Interoperability

This chapter describes how clients outside DCE can connect to Oracle servers in DCE and how a local naming configuration file (TNSNAMES.ORA) can be used for name lookup when CDS is accessible.

The following topics are covered:

- Section 13.1, “Connecting Clients Outside DCE to Oracle Servers in DCE”
- Section 13.2, “Sample Parameter Files”
- Section 13.3, “Using TNSNAMES.ORA for Name Lookup When CDS is Inaccessible”

13.1 Connecting Clients Outside DCE to Oracle Servers in DCE

Clients without access to DCE and CDS can still connect to Oracle servers in DCE using TCP/IP or some other protocol if a listener is configured to do this. If a listener has been configured in LISTENER.ORA on the server (see the sample listener configuration file (LISTENER.ORA) in the next section), non-DCE clients can use normal Oracle and Net8 procedures to connect to an Oracle server in DCE.

> **Note:** In this case DCE security would not be available to clients. Also, service names would be located and resolved to network addresses in a TNSNAMES.ORA file on the client, not using the CDS name server. See Section 13.2.1, “LISTENER.ORA” for a sample file. Also see Section 13.2.2, “TNSNAMES.ORA”.

Following are samples of LISTENER.ORA and TNSNAMES.ORA files as they would need to be configured if a client from outside of DCE wanted to connect to Oracle database servers in a DCE environment.

13.2 Sample Parameter Files

At least two Oracle parameter files are needed for successful client/server communications. Create and modify these files using you favorite text editor. The files are as follows:

- “LISTENER.ORA”
- “TNSNAMES.ORA”

13.2.1 LISTENER.ORA

This file resides on the listener node. It defines listener characteristics and the addresses at which the listener listens.

In the following example, each element is laid out on a separate line, so it is easy to see the file’s structure. This is the recommended format. If you must edit a LISTENER.ORA file by hand, you do not have to put each element on a separate line. Be careful, though, to include all the appropriate parentheses and to indent if you must continue an element onto the next line.

This example assumes the UNIX operating system and the TCP/IP protocol for one listener, and the DCE protocol for another listener. A single listener may have multiple addresses too. For example, instead of having two separate listeners for different database instances on a server node, you could have one listener for both, listening on both TCP/IP and on DCE. However, performance will be better with separate listeners.

```
LSNR_TCP=
      (ADDRESS_LIST=
          (ADDRESS=
                 (PROTOCOL=IPC)
                 (KEY=DB1)
          )
          (ADDRESS=
                 (PROTOCOL=tcp)
                 (HOST=rose)
                 (PORT=1521)
          ))

  SID_LIST_LSNR_TCP=
         (SID_DESC=
                (SID_NAME=ORASID)
                (ORACLE_HOME=/usr/jprod/oracle7)
         )
LSNR_DCE=
  (ADDRESS=
     (PROTOCOL=DCE)
     (SERVER_PRINCIPAL=oracle)
     (CELL_NAME=cell1)
     (SERVICE=dce_svc))
   SID_LIST_LSNR_DCE=
     (SID_DESC=
       (SID_NAME=ORASID)
       (ORACLE_HOME=/usr/prod/oracle8))

#For all listeners, the following parameters list sample
#default values.

PASSWORDS_LISTENER=
STARTUP_WAIT_TIME_LISTENER=0
CONNECT_TIMEOUT_LISTENER=10
TRACE_LEVEL_LISTENER=OFF
TRACE_DIRECTORY_LISTENER=/usr/prod/oracle7/neatwork/trace
TRACE File_LISTENER=listener.trc
LOG_DIRECTORY_LISTENER=/usr/prod/oracle7/network/log
LOG_FILE_LISTENER=listener.log
```

13.2.2 TNSNAMES.ORA

This file resides on both the client and the server nodes. It provides a list of the service names and addresses of all services on the network.

The following TNSNAMES.ORA file maps the service name ORATCP to the connect descriptor that includes a TCP/IP address and the service name ORADCE to a connect descriptor that includes a DCE address.

```
ORATCP = (DESCRIPTION=
          (ADDRESS=
            (PROTOCOL=TCP)
            (HOST=rose)
            (PORT=1521)
           )
           (CONNECT_DATA=
             (SID=DB1)
           )
          )
ORADCE=(DESCRIPTION=
         (ADDRESS=
           (PROTOCOL=DCE)
           (SERVER_PRINCIPAL=oracle)
           (CELL_NAME=cell1)
           (SERVICE=dce_svc)
          )
          (CONNECT_DATA=
                 (SID=ORASID)
          )
          )
```

A user who wished to access the DB1 database would use ORATCP to identify the appropriate connect descriptor. For example:

```
SQLPLUS SCOTT/TIGER@ORATCP
```

13.3 Using TNSNAMES.ORA for Name Lookup When CDS is Inaccessible

Typically, names are resolved into network addresses by CDS. Though the main purpose (in the context of Native Naming adapters) of TNSNAMES.ORA is to load Oracle service names and network addresses into CDS, it could be used temporarily as a backup name resolution service if CDS is inaccessible.

13.3.1 SQL*Net Release 2.2 and Earlier

To use TNSNAMES.ORA for name lookup and resolution, remove (or comment out) the “native name” parameters from SQLNET.ORA on the client. To comment out the lines, add a # at the beginning of each line. For example:

```
#native_names.use_native=true
#native_names.directory_path=(dce)
```

13.3.2 SQL*Net Release 2.3 and Net8

You can use TNSNAMES.ORA for name lookup and resolution when DCE CDS is unavailable if you have tnsnames listed as a value for the names.directory_path parameter in the SQLNET.ORA file on the client. For example:

```
names.directory_path=(dce, tnsnames)
```

This parameter enables you to list more than one names resolution method. The methods are tried in order. In this example, dce is attempted first. If it is unsuccessful, tnsnames is tried next.

A

Encryption and Checksum Parameters

This appendix shows an example of a profile (SQLNET.ORA) generated after you perform the network configuration described in Chapter 2, “Configuring Encryption and Checksumming” of this guide. It includes the following:

- Section A.1, “SQLNET.ORA for a Single Community Set of Clients and Servers”

This section contains a sample SQLNET.ORA configuration file for a set of clients with similar characteristics and a set of servers with similar characteristics. Examples of the Oracle Advanced Networking Option encryption and checksumming parameters are included in the file.

A.1 SQLNET.ORA for a Single Community Set of Clients and Servers

Following is a sample SQLNET.ORA file.

```
################
# Filename......: sqlnet.ora
# Date..........: 12-MAY-97 13:12:17
################
AUTOMATIC_IPC = ON
TRACE_LEVEL_CLIENT = OFF
SQLNET.EXPIRE_TIME = 0
NAMES.DEFAULT_DOMAIN = world
SQLNET.CRYPTO_SEED = ”-kdje83KKEP39487dvmlqEPTbxXe702M73”
SQLNET.ENCRYPTION_CLIENT = REQUESTED
SQLNET.ENCRYPTION_TYPES_CLIENT = (RC4_40,DES40)
SQLNET.CRYPTO_CHECKSUM_CLIENT = REQUESTED
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT = (MD5)
SQLNET.CRYPTO_CHECKSUM_CLIENT = REQUESTED
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT = (MD5)
SQLNET.ENCRYPTION_TYPES_SERVER = (RC4_40,DES40)
SQLNET.CRYPTO_CHECKSUM_SERVER = REQUESTED
NAMES.DIRECTORY_PATH = (TNSNAMES,ONAMES)
SQLNET.AUTHENTICATION_SERVICES = (SECURID)
```

Note the following:

- If you do not specify any values for Server Encryption, Client Encryption, Server Checksum, or Client Checksum, the corresponding configuration parameters will not appear in the SQLNET.ORA file. However,the Oracle Advanced Networking Option defaults the value to ACCEPTED.
- If no encryption or checksumming algorithm is specified on the Server Encryption, Client Encryption, Server Checksum, or Client Checksum pages, the server side of the connection uses the first algorithm in its own list of installed algorithms that also appears in the client’s list of installed algorithms.
- Encryption and checksumming function independently of each other; encryption can be activated while checksumming is off, and vice versa.

B

Authentication Parameters

This appendix shows some sample configuration files with the necessary profile (SQLNET.ORA) and database initialization file (INIT.ORA) authentication parameters when using the Kerberos, CyberSAFE, or SecurID Authentication Adapters. The following sections are included.

- Section B.1, “Configuration Files for Clients and Servers using CyberSAFE Authentication”
- Section B.2, “Configuration Files for Clients and Servers using Kerberos Authentication”
- Section B.3, “Configuration Files for Clients and Servers using SecurID Authentication”

B.1 Configuration Files for Clients and Servers using CyberSAFE Authentication

Following is a list of parameters to insert into your configuration files for clients and servers using CyberSAFE.

B.1.1 Profile (SQLNET.ORA)

```
SQLNET.AUTHENTICATION_SERVICES=(CYBERSAFE)
SQLNET.AUTHENTICATION_GSSAPI_SERVICE=oracle/dbserver.someco.com@SOMECO.COM
```

B.1.2 Database Initialization File (INIT.ORA)

```
REMOTE_OS_AUTHENT=FALSE
OS_AUTHENT_PREFIX=“”
```

B.2 Configuration Files for Clients and Servers using Kerberos Authentication

Following is a list of parameters to insert into your configuration files for clients and servers using Kerberos.

B.2.1 Profile (SQLNET.ORA)

```
SQLNET.AUTHENTICATION_SERVICES=(KERBEROS5)
SQLNET.AUTHENTICATION_KERBEROS5_SERVICE=oracle
SQLNET.KERBEROS5_CC_NAME=/usr/tmp/DCE-CC
SQLNET.KERBEROS5_CLOCKSKEW=1200
SQLNET.KERBEROS5_CONF=/krb5/krb.conf
SQLNET.KERBEROS5_REALMS=/krb5/krb.realms
SQLNET.KERBEROS5_KEYTAB=/krb5/v5srvtab
```

B.2.2 Database Initialization File (INIT.ORA)

```
REMOTE_OS_AUTHENT=FALSE
OS_AUTHENT_PREFIX=“”
```

B.3 Configuration Files for Clients and Servers using SecurID Authentication

Following is list of parameters to insert into your configuration files for clients and servers using SecurID.

B.3.1 Profile (SQLNET.ORA)

```
SQLNET.AUTHENTICATION_SERVICES=(SECURID)
```

B.3.2 Database Initialization File (INIT.ORA)

```
REMOTE_OS_AUTHENT=FALSE
OS_AUTHENT_PREFIX=“”
```

Glossary

authentication

The process of verifying the identity of a user, device, or other entity in a computer system, often as a prerequisite to allowing access to resources in a system.

authorization

Permission given to a user, program, or process to access an object or set of objects. In Oracle, authorization is done through the role mechanism. A single person or a group of people can be granted a role or a group of roles. A role, in turn, can be granted other roles.

client

A client relies on a service. A client can sometimes be a user, sometimes a process acting on behalf of the user during a database link (sometimes called a proxy).

cryptographic checksum

A mechanism that computes a value for a message packet, based on the data it contains, and passes it along with the data to authenticate that the data has not been tampered with. The recipient of the data recomputes the cryptographic checksum and compares it with the cryptographic checksum passed with the data; if they match, it is "probabilistic" proof the data was not tampered with during transmission. The important property of a cryptographic checksum is that without knowing the secret key, a malicious interceptor has only an infinitesimally small chance of being able to construct an altered message with a valid corresponding checksum.

DES

The U.S. Data Encryption Standard.

initial ticket

An initial ticket or ticket granting ticket (TGT) is retrieved by running the kinit program and providing a password. When run successfully, the user is granted a ticket granting ticket that identifies them as having the right to ask for additional service tickets. No tickets can be obtained without an initial ticket.

Kerberos

A network authentication service developed under Massachusetts Institute of Technology's Project Athena that strengthens security in distributed environments. Kerberos is a trusted third-party authentication system that relies on shared secrets and assumes that the third party is secure. It provides single sign-on capabilities and database link authentication (MIT Kerberos only) for users, provides centralized password storage, and enhances PC security.

KDC/TGS

Key Distribution Center / Ticket Granting Service. The KDC maintains a list of user principals and is contacted through the `kinit` program for the user's initial ticket—the ticket-granting ticket (TGT). The Ticket Granting Service maintains a list of service principals and is contacted when a user wants to authenticate to a server providing such a service.

The KDC / TGS is a trusted third party that must run on a secure host. It creates ticket-granting tickets and service tickets. The KDC and TGS are usually the same entity.

kinstance

An instantiation or location of a service. This is an arbitrary string, but the host machine name for a service is typically specified.

kservice

An arbitrary name of a Kerberos service object.

message digest

See *cryptographic checksum*.

Net8

An Oracle product that works with an Oracle Server and enables two or more computers that run the Oracle Server or Oracle tools such as Designer/2000 to exchange data through a third-party network. Net8 supports distributed processing and distributed database capability. Net8 is an ”open system” because it is independent of the communication protocol, and users can interface Net8 to many network environments.

network authentication service

A means for authenticating clients to servers, servers to servers, and users to both clients and servers in distributed environments. A network authentication service is a repository for storing information about users and the services on different servers to which they have access, as well as information about clients and servers on the network. An authentication server can be a physically separate machine, or it can be a facility co-located on another server within the system. To ensure availability, some authentication services may be replicated to avoid a single point of failure.

principal

A Kerberos object, consisting of *kservice/kinstance@REALM*. See also *kservice, kinstance*, and *realm*. A uniquely-identified client or server.

realm

A Kerberos object. A set of clients and servers operating under a single key distribution center/ticket-granting service (KDC/TGS). *kservices* that are in different realms but that have the same name are unique.

service

A network resource used by clients; for example, an Oracle database server.

service name

For Kerberos-based authentication, the *kservice* portion of a service principal.

service table

A service table is a list of service principals that exist on a *kinstance*. This information must be extracted from Kerberos and copied to the Oracle server machine before Kerberos can be used by Oracle.

session key

A key shared by at least two parties (usually a client and a server).

server

A provider of a service.

service principal

See “principal”.

smart card

A device external to a server that provides user authentication. Smart cards may operate using challenge-response mechanisms, or by providing one-time passwords. Smart cards providing one-time use passwords are synchronized with a service on the server so that the server expects the same password generated by the smart card. Challenge-response cards operate as follows: a user enters his PIN into the card. The server that he is trying to access offers a challenge in the form of a number. The user enters the number into his smart card, and receives a number back from the card, which he offers to the server. If the number is what the server expects, access is allowed.

service ticket

Trusted information used to authenticate the client. A ticket-granting ticket is also known as the initial ticket, is obtained by directly or indirectly running `kinit` and providing a password, and is used by the client to ask for service tickets. A “service ticket” is used by a client to authenticate to a service.

ticket

A ticket is a piece of information that helps identify who the owner is. See *service ticket.*

Index