Oracle 8*i* Security: New Features and Secure Solutions

November 1999

INTRODUCTION

The prevalence of e-commerce and the ubiquity of the Internet change the way organizations do business, the means by which people communicate, and the shape of industry at the turn of the century. With these changes comes the shift of technologies that drive business decisions. Security moves from an under-the-covers solution set to a principal requisite for e-business implementations.

Resourceful organizations are harnessing the Internet to streamline operations and communicate directly with suppliers, partners, internal users, and customers alike. These capabilities, however, bring with them varied challenges. Different classes of users need to access data in different ways. A partner must see certain restricted data, and an employee should view company-confidential information, while a customer should see his—and only his—account information. Moreover, the scalability of Oracle products along with the expansiveness of the Internet exponentially increase the number of users accessing data stored in database servers. In short, the user-specific challenges consist of knowing the users, setting their data access rights, and keeping confidential data private. The technology challenges include interoperating within the standards that complementary organizations adopt, and managing the users, their access rights, and the databases themselves within various domains.

Administrators and integrators can overcome these Internet security challenges with the functionality of Oracle8*i* and Oracle Advanced Security, which provide a security-rich environment to host Internet-based, enterprise-wide solutions. This paper covers five security-based areas available in the new release of Oracle8*i*:

- Public Key Infrastructure, including Oracle's Secure Sockets Layer (SSL) implementation and the key management tools that integrate with leading PKI solutions
- Directory-enabling Oracle enterprises to achieve high degrees of security, centralized user administration, and reduced total costs of ownership
- Java implementations that secure transactions using the Java Database Connectivity (JDBC) interface
- Leading-edge authentication resulting from integration of the RADIUS protocol
- Virtual Private Databases, offering data server-enforced granular access control

In today's intranet- and extranet-focused applications, security issues drive technology decisions and implementations. The new Oracle8*i* release 8.1.6 security features make Oracle the most secure, reliable, and flexible-choice database for Internet-driven e-commerce.

PUBLIC KEY INFRASTRUCTURE

Public Key Infrastructure (PKI) is emerging as the foundation for secure electronic commerce and Internet security by providing the cornerstones of security: *authentication*, *encryption*, *integrity*, and *non-repudiation*. The importance of *authentication*, verifying the identity of users and machines, becomes crucial when an organization opens its doors to the Internet. Strong authentication mechanisms ensure that persons and machines are the entities they claim to be. *Encryption* algorithms are employed to secure communications and ensure the privacy of data sent from one computer to another. Data *integrity* often coincides with encryption, because the algorithms check the order in which data packets are received to protect against attacks such as modification of data, replay attempts, and removal of data packets. Secure infrastructures employ techniques to ensure *non-repudiation*,which proves that a specific user performed certain operations at a given time. Together, these elements combine to provide a secure, non-breakable environment for deploying e-commerce and a reliable environment for building virtually any type of electronic transactions, from corporate intranets to Internet-based e-business applications.

The principle components of a public key infrastructure are:

- *Digital Certificates*, which identify users and machines, and are securely stored in wallets
- *Public and Private Keys*, that form the basis of a PKI for secure communications based on a secret private key and a mathematically related public key
- *Secure Sockets Layer (SSL)*, the Internet-standard underlying secure protocol
- *Certificate Authority (CA)*, that acts as a trusted, independent provider of digital certificates

Additional but nonetheless important components that enable a PKI deployment are secure storage of certificates and keys, management tools to request certificates, access wallets and administer users, and a directory service acting as a centralized repository for user and machine identification and authorization.

Digital Certificates

The most widely-used public key certificates comply with the X.509 format, and the X.509 version 3 certificate is the current industry standard format. A public key infrastructure relies on X.509 certificates, also called digital certificates or public-key certificates, for public-key authentication. Authenticating a person or machine with a certificate is analogous to identifying oneself with a driver's license or a passport; servers and clients prove their identities to one another by showing their identifying credentials. Use of certificates in an Oracle environment also provides single sign-on, allowing a user to authenticate once, and then connect to multiple applications and databases without providing additional credentials. Single sign-on improves system security and ease-of-use because users don't have to remember multiple passwords and administration is limited to one password per user or machine, providing centralized security.

X.509v3 certificates contain a user's identifying information:

- Certificate owner's distinguished name (DN), which uniquely identifies the owner
- Distinguished name of the certificate's issuer (a certificate authority), which uniquely identifies the issuer

- Certificate owner's public key
- Issuer's signature
- Dates in which the certificate is valid
- Serial number, which is unique to each certificate

Secure Storage of Private Keys and Certificates

In order to authenticate a user, SSL must have a private key and a certificate provided to it. Therefore, in an Oracle environment, the client or server requires some method of storing and providing this information to SSL, namely the Oracle wallet. The wallet stores the X.509 certificate, the private key, and additional data such as trusted certificates, which are processed by SSL. These credentials are used to authenticate the user to multiple services such as data servers and application servers. The user must remember only one password, which is used to unlock her wallet.

Oracle Wallet Manager manages the wallet and requests certificates from a certificate authority. It gives users and database administrators control over the contents of their wallets. The administrator can centrally manage wallet information about applications and databases. In addition, Oracle provides Oracle Enterprise Login Assistant, an easy-to-use tool for end users to access their wallets. This tool allows users to achieve single sign-on, simply and transparently, using certificates for authentication. The wallet and management tools are used together to securely store and manage certificates, private keys, and requests to certificate servers.

The SSL Protocol

The Secure Sockets Layer protocol is widely used over the Internet to give users established digital identities and to prevent eavesdropping, tampering with, or forging messages. SSL support in Oracle Advanced Security encrypts network traffic and provides integrity checking, authenticates Oracle clients and servers, and brings public key-based single sign-on to the Oracle environment. SSL provides encryption and data integrity through the use of cipher suites, which are sets of authentication, encryption and data integrity types. The client and server each have a list of cipher suites they support and they negotiate which one is to be used during connection.

An example of a cipher suite is RSA for authentication with 3DES for encryption and SHA-1 for data integrity. Among encryption algorithms provided by SSL to Oracle Advanced Security option are RC4, DES, and Triple DES. The Triple DES (3DES) algorithm is an extremely strong means of protecting data because it uses more than one 56-bit key employed by standard DES. Triple DES is increasingly being used by organizations such as banks and financial institutions that require strong security. SHA (Secure Hashing Algorithm) provides a means of data integrity checking new to the Oracle environment. It generates a hash to protect data transmissions and ensure that packets have not been modified or tampered with during transmission.

The SSL Handshake

Using the SSL protocol, when a client attempts to connect to a server, it initiates a handshake to establish the SSL session. The client process sends its X.509 certificate to the server process, and the two engage in a handshake that verifies the identity of the client to the server. The handshake is transparent to the user. If the handshake is successful, the client is granted access, and the session is protected by the encryption and data integrity chosen during the handshake.

Once the client(s) and server(s) are configured to use SSL as the protocol for communications, the following process takes place from a Net8 client perspective:

1. The wallet is installed, at which point the user must provide Oracle Wallet Manager a password. The password is used to create a public- and private-key pair. The private key is stored in the wallet.
2. Oracle Wallet Manager requests the certificate from the certificate server. The certificate then gets downloaded and stored locally in the client's wallet, which is in the file system.
3. A Net8 connection using SSL is initiated. The SSL handshake takes place as part of the connection.
4. If the SSL handshake succeeds, the connection is allowed. The outcome is that the server gets the identity of the user from SSL, and the server validates that identity.

Support for the SSL protocol strengthens the security and interoperability of Oracle Advanced Security, from propagating overall security services to permitting stronger data protection.

Entrust Integration

Entrust Technologies, Inc. is a market-leading provider of Public Key Infrastructure solutions, through their Entrust/PKI software. Entrust/PKI includes many products, such as Entrust Profile, which secures users' PKI credentials, and Entrust Authority, Entrust's certificate authority product.

Oracle is making specific product modifications to Oracle Advanced Security to enable customers of both Oracle and Entrust to incorporate Entrust-based single sign-on into their Oracle applications. By integrating with Entrust/PKI, Oracle enhances its ability to provide X.509-based single sign-on to large customers who require the extensive key management, certificate revocation, and other features that Entrust provides.

Oracle will implement support for Entrust/PKI in Oracle Advanced Security release 8.1.6, enabling customers to use Entrust Profile, Entrust's "wallet" mechanism, for storage of certificate and private keys, and for secure credential management. Instead of accessing user credentials (private key and certificate) from an Oracle wallet, Oracle Advanced Security accesses a user's Entrust Profile for authentication and single sign-on. Entrust integration will require both release 8.1.6 of Oracle Advanced Security and Entrust Authority 5. Production use of this feature will be available shortly after general availability of Oracle Advanced Security release 8.1.6.

PKI Support for Secure Single Sign-On

As the public key infrastructure is deployed more frequently to secure such applications as email and electronic commerce, PKI is one of the most important investments companies are making. Because all clients, application servers and data servers can authenticate themselves to one another, PKI provides an important security infrastructure to a network.

SSL secures not only Net8, but also other protocols such as IIOP (Internet Inter-ORB Protocol). By capitalizing on Java support, Oracle Advanced Security secures IIOP connections, giving Oracle the ability to work with thin clients and Enterprise JavaBeans (EJB).

Support for SSL in Oracle Advanced Security closes the loop for secure end-to-end communications between any client, a web server or application server, and any Oracle8*i* RDBMS. For example, when a user wants to connect to her financial institution to transfer funds, she must be able to verify beyond a doubt that she is providing sensitive information such as passwords and account numbers to the proper server. With SSL and public-key authentication, the server can verify its identity to her browser, and the

client can identify itself to the server. Now that organizations are implementing application servers and firewalls to protect their networks, the connection process expands. Using the same example, the financial information can be stored in an Oracle8 dataserver secured behind a firewall. The user connects to the database using SSL to connect over the Internet and to the application server, which passes the connect request over Net8 (still protected with SSL) through a firewall and to the secured Oracle8*i* server with her financial account information.

Certificates not only authenticate clients to servers, but it also authenticates servers to other servers. This expands the security of the entire system with secure database links for mutual authentication of servers. With SSL deployment, all clients and all servers, including database servers and application servers, have credentials that identify them to all other machines and services with which they communicate.

The complete package that Oracle delivers provides standards-based methods to prevent eavesdropping, tampering with, or forging messages sent over the network, while providing single sign-on and strong authentication of clients and server in the network and over the Internet. A public key infrastructure paves the way for secure electronic commerce and e-business in the Information Age.

DIRECTORY-ENABLING ORACLE ENTERPRISES

ENTERPRISE USER MANAGEMENT

Enterprises today face tremendous challenges in managing information about users, keeping user information current, and securing access to all the information in an enterprise. Each user may have multiple accounts on different databases, requiring her to remember passwords for each of these accounts. Users not only have too many passwords, but there are too many accounts for administrators to manage. Furthermore, the lack of centralization is a security risk, because old or unused accounts and privileges can be misused.

To address these challenges, Release 8.1.6 introduces enterprise user management. Enterprise users and their authorizations are managed in Oracle Internet Directory, a Lightweight Directory Access Protocol (LDAP)-based directory service, using Oracle Enterprise Security Manager, a tool accessible through Oracle Enterprise Manager.

Enterprise users can be assigned enterprise roles, which are containers of database-specific global roles that determine their access privileges in databases. For example, the enterprise role MANAGER could contain the global role HRMANAGER on the Human Resources database and the global role ANALYST on the Payroll database. An enterprise role can be granted or revoked to one or more enterprise users. For example, an administrator could grant the enterprise role MANAGER to a number of enterprise users who hold the same job. This information about users and roles is protected in the directory through Access Control Lists (ACLs), ensuring that only a privileged administrator can manage users and grant and revoke roles.

USER/SCHEMA SEPARATION

In general, users do not need their own accounts - or their own schemas - in a database; they merely need to access an application schema. For example, users Jane, Cindy and Rakesh are all users of the Payroll application and need access to the Payroll schema on the Finance database. None of them needs to create his or her own objects in the database; in fact, they only need to access Payroll objects.

Release 8.1.6 allows the separation of users from schemas, allowing many enterprise users to access a single, shared application schema. Instead of creating a user account (that is, a user schema) in each database a user needs to access, the administrator only needs to create an enterprise user in the directory and "point" the user at a shared schema that other enterprise users can also access. For example, if Jane, Cindy and Rakesh all access the Sales database, you can create a single schema, such as ‘sales_application,’ which all three users can access, instead of creating an account for each user on the Sales database.

Now, administrators can create an enterprise user only once in the directory. The enterprise user, nonetheless, can access multiple databases using only the privileges she needs to perform her job, thus lowering the cost of managing users in an enterprise.

Oracle’s LDAP version 3-compliant directory server, Oracle Internet Directory, is fully integrated with Oracle8*i* and supports "off-the-shelf" enterprise user management. Other LDAP directories, including Novell Directory Service (NDS) and Microsoft’s Active Directory for Windows 2000 will be certified to operate with Release 8.1.6.

The separation of users from schemas is truly the payoff for deploying a directory service. Thousands of users can connect to a database, be known to the database (and audited in the database), with specific privileges in the database, without being created in the database. Now, you can truly create an enterprise user once, in the directory *a single enterprise user account* who nonetheless can access multiple databases, using only the privileges she needs to perform her job.

Enterprise user management thus offers the following benefits:

- Fewer User Accounts—Enterprise users no longer need to be database users nor have identified schemas.
- Internet Scalability—You can support hundreds of thousands of users, who are known to multiple databases, accountable (and audited in) multiple databases, without creating thousands of database user accounts.
- Easily-Enforced Security—If a user changes jobs or leaves, his privileges can be altered or removed, everywhere, merely by changing his user entry in Oracle Internet Directory. Organizations no longer need to worry about old, unused accounts or out-of-date privileges, which consume valuable system resources and are targets for hackers.
- Reduced Cost of Ownership—Organizations save significant resources by managing a single enterprise user account and assigning enterprise roles once, instead of creating multiple user accounts with multiple passwords, each having multiple authorizations.

PRODUCT SUMMARY

With the continued growth of distributed systems, the problem of user authentication and user management is now acute. Users have too many passwords; consequently, they write them down or choose the same password for all accounts. Organizations must manage multiple accounts for each user. As a result, they devote significant resources to user administration, or invest in network authentication services, many of which promise single sign-on and centralized authorization management. Common information used by multiple applications—such as username, user's office location and phone number—is often fragmented across the enterprise, leading to data that is redundant, inconsistent, and expensive to manage. The lack of centralization is a security risk since old or unused accounts and privileges can be misused.

Oracle Advanced Security addresses the above needs for strong security, single sign-on and centralized user management by offering integrated security and directory services; specifically, by storing and managing user information in a directory which supports the Lightweight Directory Access Protocol (LDAP). Multiple Oracle applications can rely on a common, centralized definition of a user to determine which applications, services, and databases a user may access, and with what privileges.

The benefits of integrated security and directory services include:

- Single sign-on to multiple Oracle8*i* databases throughout the enterprise
- Single enterprise user account, instead of multiple accounts per user
- Reduced total cost of ownership through single station administration (SSA)
- Well-integrated, standards-based public key infrastructure (PKI)
- Stronger security through centralized authorization management and strong authentication.

Oracle Advanced Security's integrated security and directory services incorporates multiple Oracle components, including:

- *Oracle Wallet Manager*—A tool which is used to protect and manage user certificates, keys, and trusted certificates
- *Oracle Enterprise Login Assistant*—An easy-to-use tool which enables single sign-on for users
- *Oracle Internet Directory*—An LDAPv3-compliant directory service, built on the Oracle8*i* database, which stores information about enterprise users and enterprise roles
- *Oracle Enterprise Security Manager*—An administration tool available through Oracle Enterprise Manager, which allows administrators to manage enterprise users and enterprise roles in Oracle Internet Directory, and across multiple Oracle8*i* databases, from a single console
- *Oracle8i*—A database server which retrieves users' enterprise roles from Oracle Internet Directory and authenticates users over SSL

Oracle Advanced Security's integrated security and directory services also requires the following non-Oracle component:

- *Certificate Authority (CA)*—An X.509-compliant certificate authority which creates digital certificates.

Oracle Wallet Manager

An advantage of X.509 certificates is that they may be used to uniquely identify an individual within an organization, and thus enable strong authentication. Oracle Wallet Manager provides secure management of PKI-based user credentials. Oracle Wallet Manager creates a private and public key pair for a user, and issues a Public Key Certificate Standard (PKCS) #10 certificate signing request that can be fulfilled by any X.509 v3-compliant CA. After the CA issues an X.509 certificate, the user can load the certificate into his wallet. Oracle Wallet Manager stores and manages PKI credentials for users as well as servers, such as Oracle Internet Directory and Oracle8*i*.

Oracle Wallet Manager also manages user trusted certificates, the list of root certificates that the user trusts, and is pre-configured with root certificates from PKI vendors such as VeriSign and GTE. Wallets are protected using password-based, strong encryption, and are stored on the client.

PKI-based Single Sign-On

Oracle Advanced Security and its predecessor, Advanced Networking Option, support multiple forms of single sign-on for database users, among them Kerberos, CyberSafe and the Distributed Computing Environment (DCE). Release 8.1.6 of Oracle Advanced Security enhances the single sign-on options available by providing support for SSL-based single sign-on.

A user can easily access his wallet using Oracle Enterprise Login Assistant, an easy-to-use login tool that hides the complexity of a private key and certificate from users. The user provides a password to Oracle Enterprise Login Assistant, which then opens the user's encrypted wallet. The certificate and private key contained in an Oracle wallet are used to authenticate the user to multiple databases, which no longer need to store and manage local passwords for every user. Furthermore, authentication is transparent to the user, who need not provide any additional passwords once his wallet has been opened.

Oracle Advanced Security offers enhanced PKI-based single sign-on through use of interoperable X.509 version 3 certificates for authentication over Secure Sockets Layer, the Internet standard for

authentication. In addition to strong user authentication, SSL also provides network data confidentiality through encryption and data integrity for multiple types of connections: LDAP , IIOP (Internet Intra-ORB Protocol), and Net8.

Moreover, PKI-based single sign-on over SSL can be used alone, that is, in the absence of a directory server, or in conjunction with enterprise user management.

Single Station Administration (SSA)

Managing thousands of user accounts is one of the largest administration challenges facing large organizations. Creating user accounts and assigning privileges is often a multi-step process, requiring multiple tools.

Significant new functionality has been added in Release 8.1.6 to address this need. Oracle Enterprise Security Manager (an extension to Oracle Enterprise Manager) provides single station administration. From a single console, an administrator can perform the following:

- Create enterprise users in Oracle Internet Directory.
- Create shared schemas in databases
- Map enterprise users to shared schemas
- Create enterprise roles that span multiple databases.
- Assign one or more enterprise roles to a user.
- Configure Access Control Lists on directory objects

Oracle Enterprise Security Manager provides one tool for enterprise user management, resulting in a lower cost of user administration throughout the enterprise. Another benefit of single station administration is that if security is easy to administer, organizations are more likely to implement strong security throughout the enterprise.

Robust Directory Services

Oracle Internet Directory is a native LDAP version 3 implementation that combines the mission-critical strength of Oracle's database technology with the flexibility of the Internet standard. Oracle Internet Directory is the default data repository for accessing Oracle8*i* enterprise user information, including enterprise roles and shared schema information.

Oracle Internet Directory offers flexible, highly-granular access control mechanisms which protect the sensitivity of common application information as well as Oracle's enterprise user information. Oracle Internet Directory's Access Control Lists (ACLs) can be used to specify, at the attribute level, the users entitled to access and the type of access permitted. For example, a user's manager may have read and write access to a user's salary attribute, the user can read his own salary, but no other user has access to it. Oracle Internet Directory also offers configurable default, guest, and super-user permissions, to enable users to have "least privilege"—just the privileges they need to perform their jobs, and no more. Oracle Internet Directory can be automatically configured with the schema and required Access Control Lists (ACLs) the Oracle8*i* database needs for enterprise user management,

Oracle Internet Directory exploits the built-in availability and performance of the underlying Oracle8*i* data server, as well as offering bulk loading, deletion, and updates of directory entries, high-speed backup and recovery tools, the ability to add or delete directory nodes without service disruption, and other high availability features.

Availability

Integrated security and directory services are available with Release 8.1.6 of Oracle Advanced Security, including Oracle Wallet Manager, Oracle Enterprise Login Assistant, Oracle Enterprise Security Manager, use of SSL for encryption, authentication and single sign-on, user/schema separation and the use of Oracle Internet Directory for enterprise user management.

Oracle Internet Directory may also be purchased separately.

Summary

Oracle Advanced Security's integration of security and directory services offers strong user authentication through standards-based single sign-on, and reduces user difficulty with too many passwords. Administrators spend less time managing user accounts, because they are able to centrally administer users across multiple databases. The Single Enterprise User offers even greater benefits for organizations that need only create one account per user for the entire organization. Oracle environments may store their entire definition of a user, and the user's roles and privileges, within a directory service. The Single Enterprise User enables organizations deploying Internet applications to support thousands of users securely, with reduced cost of ownership, and scale to tens of thousands, hundreds of thousands, or even millions of users over a distributed enterprise.

SECURE JAVA IMPLEMENTATIONS

Because encryption is often regarded as one of the most important elements of a secure environment, a large number of Oracle Advanced Security customers deploy it, at least in part, to encrypt Net8 traffic. Oracle Advanced Security now has enhanced cryptographic capabilities that provide encryption of additional protocols supported by the Oracle8*i* release 8.1.6 database. Specifically, the Java implementation of Oracle Advanced Security allows Thin Java Database Connectivity (JDBC) clients to connect securely to Oracle8*i* databases. The Java implementation provides network encryption and integrity protection for thin JDBC clients communicating with Oracle Advanced Security-enabled Oracle8*i* databases.

JDBC Extensions

JDBC is an industry-standard Java interface that provides a Java standard for connecting to a relational database from a Java program. Sun Microsystems defined the JDBC standard, and Oracle Corporation, as an individual provider, implements and extends the standard with its own JDBC drivers.

Oracle implements two types of JDBC drivers: Thick JDBC drivers built on top of the C-based Net8 client, and Thin (Pure Java) JDBC drivers to support downloadable applets. Oracle's JDBC drivers are used when users create JDBC applications to communicate with Oracle databases. Oracle's extensions to JDBC include the following features:

- Data access and manipulation
- LOB access and manipulation
- Oracle object type mapping
- Object reference access and manipulation
- Array access and manipulation
- Application performance enhancement

Securing Thin JDBC

Because the Thin JDBC driver is designed to be used with downloadable applets used over the Internet, Oracle designed a 100% Java implementation of Oracle Advanced Security encryption and integrity algorithms for use with thin clients. Oracle Advanced Security provides the following features for Thin JDBC:

- Data encryption
- Data integrity checking
- Secure connections from Thin JDBC clients to the Oracle RDBMS
- Ability for developers to build applets that transmit data over a secure communication channel
- Secure connections from middle tier servers with Java Server Pages (JSP) to the Oracle RDBMS
- Secure connections from Oracle8*i* databases to older versions of Oracle Advanced Security-enabled databases

Data Encryption and Integrity Checking

The Oracle JDBC Thin driver implements the Oracle O3LOGON protocol for authentication. The Oracle JDBC Thin driver does not support Oracle Advanced Security third party authentication features such as Kerberos, SecurID, and RADIUS. However, the Oracle JDBC OCI driver support is the same as thick client support, in which all of Oracle Advanced Security features are implemented.

Oracle Advanced Security continues to encrypt and provide integrity checking of Net8 traffic between Net8 clients and Oracle servers using algorithms written in C. The Oracle Advanced Security Java implementation provides Java versions of the following encryption algorithms:

- DES40
- DES56
- RC4_40
- RC4_56

In addition, the implementation provides data integrity checking for Thin JDBC with the MD5 algorithm.

These are the same set of algorithms used to encrypt Net8. It is important to note that the Java implementation of these algorithms—along with MD5 for data integrity checking—is available only in the Export Edition of Oracle Advanced Security and not the Domestic Edition. Consequently, only export-level key lengths are implemented.

The implementation allows Thin JDBC clients and virtually any Java client that uses Oracle's Java implementation to securely transmit data to and from the Oracle8*i* database.

Secure Connections for Virtually Any Client

On the server, the negotiation of algorithms and the generation of keys function exactly the same as Oracle Advanced Security Net8 encryption, thus allowing backward and forward compatibility of clients and servers. On the clients, the algorithm negotiation and key generation occur in exactly the same manner as C-based Oracle Advanced Security encryption. The client and server negotiate encryption algorithms, generate random numbers, use Diffie-Hellman to exchange session keys, and use the Oracle Password Protocol, in the same manner as traditional Net8 clients. Thin JDBC contains a complete implementation of a Net8 client in pure Java. Consistent with other encryption implementations, the Java implementation of Oracle Advanced Security prevents access to the cryptographic algorithms, makes it impossible to double encrypt data, and encrypts data as it passes through the network. Users cannot alter the keyspace nor alter the encryption algorithms themselves.

Obfuscation

Code implementing cryptography and written in Java must be obfuscated in order to comply with United States government export regulations. Therefore, this implementation protects Java classes and methods that contain encryption and decryption capabilities with obfuscation software.

Java Byte Code Obfuscation is a process that software vendors often use to protect intellectual property in the form of Java programs. The obfuscation process mixes up Java symbols found in the code. The process leaves the original program structure intact, allowing the program to run correctly, while changing the names of the classes, methods, and variables in order to hide the intended behavior.

Although it is easy to decompile and read non-obfuscated Java code, the obfuscation process renders the target code nearly impossible to interpret once decompiled.

Use of the Secure Thin JDBC Implementation

The Oracle Advanced Security Java implementation gives developers the ability to build applets that transmit data over secure communication channels secured by Oracle Advanced Security. In addition, it provides secure connections from any middle tier server with Java Server Pages (JSP) to the Oracle RDBMS and secure connections from Oracle8*i* databases to older versions of Oracle Advanced Security-enabled databases. This allows developers deploying Oracle and other components to securely transmit a variety of information over a variety of channels.

Other Protocols Using Java

In addition to providing secure connections to the RDBMS from Net8 clients and Thin JDBC clients, Oracle Advanced Security now secures another widely used protocol, Internet Intra-Orb Protocol (IIOP), which is used to access the Oracle8*i* release 8.1.6 database. Oracle builds the capability to secure this protocol, and thus data sent over it, using Secure Sockets Layer (SSL). SSL is licensed as part of Oracle Advanced Security.

Secure Sockets Layer Support for IIOP

The current release of Oracle8*i* supports IIOP connections into the database in addition to Net8 connections. IIOP connections secured by SSL provide the ability for Enterprise JavaBeans (EJBs) to communicate securely with the Oracle RDBMS.

Oracle has enhanced the SSL libraries which support IIOP through the addition of an application programming interface (API). The open interface allows customers to set PKI credentials used in establishing a secure IIOP connection, through provision of an X509v3 certificate and a private key, in Public Key Certificate Standard (PKCS) #5 format.

This enhancement makes it easier for users to access the Object Request Broker (ORB) which is available in the database. This modification enables users to more easily integrate existing IIOP-based applications with the Oracle8*i* database.

Summary: Cross-Protocol Security

In sum, Oracle Advanced Security can be used to secure *every* protocol that communicate with the Oracle8*i* Release 8.1.6 database. Whether an organization chooses to deploy "fat" Net8 clients, EJBs, Thin JDBC clients, or any combination thereof, Oracle Advanced Security ensures the privacy and integrity of the communication foundation for every connection.

LEADING-EDGE AUTHENTICATION SOLUTIONS

User authentication becomes increasingly important in today's distributed enterprises and Internet environments where thousands of users access data from varied locations. Oracle provides four categories of authentication with Release 8.1.6: *internal,* where users are identified by passwords stored inside the data server; *OS authentication,* where the database relies on identity authenticity provided by the underlying operating system; *global,* where users and schemas are stored and retrieved from a central data repository outside the database; *external,* where the RDBMS defers to Oracle Advanced Security for third-party authentication through outside mechanisms such as Kerberos or token cards.

Oracle Advanced Security supports a number of third-party authentication solutions that provide assurance of known user identities and, in some cases, provide the additional benefit of single sign-on. Kerberos, CyberSafe and Distributed Computing Environment (DCE) support allow Oracle8*i* data servers to defer authentication to a third party server whose sole purpose is authentication, thus providing a centralized authentication model and eliminating the need for password administration on every database. Oracle8*i* allows strong authentication with token cards and biometric devices by supporting Security Dynamics and Identix, respectively. With the introduction of RADIUS support in Oracle8*i*,external authentication extends to standard-based, flexible solutions.

The RADIUS (Remote Authentication Dial-In User Service) protocol is a standard for remote authentication and controlled access to networks. RADIUS is a lightweight protocol for user authentication, authorization, and accounting between a network client and an authentication server. This Internet Engineering Task Force (IETF) standard was created by Lucent Technologies (formerly Livingston Enterprises) as an open security standard designed to be flexible, open and scaleable.

RADIUS has gained widespread acceptance in the industry, and has been implemented by nearly all organizations that allow users to access the network remotely. Virtually every user who has dialed in to a network remotely has most likely used RADIUS to do so. Many enterprises have standardized on RADIUS because of its acceptance by the industry, its flexibility, and its ability to centralize all user information, easing and reducing the total cost of user administration.

RADIUS Support in Oracle Advanced Security

Oracle has built support for RADIUS into Oracle Advanced Security to satisfy the requirement of integrating into complex environments and expanding security features for Oracle users. In particular, support for RADIUS provides three principal solutions to Oracle users:

- Any RADIUS-compliant token, smart card, or biometric device can authenticate Oracle users
- Integration into existing environments
- Enhanced features including challenge-response and accounting

RADIUS-Compliant Devices Authenticate Oracle Users

With RADIUS support built in, Oracle Advanced Security extends the number of supported authentication methods. RADIUS allows Oracle to support one authentication protocol on the Oracle8*i* server, while authentication server vendors supply the specific authentication servers. With minimal development effort required by the authentication server provider, more authentication solutions – such as smart cards – are available with enhanced features, such as accounting and challenge-response.

Providers of token cards, smart cards, biometrics, and other authentication devices can communicate with Oracle by supporting this standard. Authentication server providers can also enhance authentication by employing accounting and challenge-response. Many token and smart card vendors support RADIUS today.

When the authentication device uses challenge-response mode, a graphical user interface (GUI) on the client prompts the user first for a password, then for additional data such as a dynamic password. An authentication server provider can easily implement this GUI by customizing the Java interface class that ships with Oracle Advanced Security. This adds to the interoperability of Oracle Advanced Security by allowing an organization to integrate into Oracle whichever strong authentication methods they currently use.

Integration into Existing Environments

Many organizations allow their users remote access to the network. While this provides numerous benefits to dial-in users and network administrators, it also provides potential security threats. By opening a network to remote access, that network is also open to access by outside parties not intended to access data in the network. It is therefore imperative that the security of remote access is strictly managed. Because RADIUS has gained acceptance as the major solution for secure remote access, the RADIUS implementation allows Oracle to readily integrate into existing systems.

At the time of a connection request by a remote user, the client has dialed into a network access server (NAS) to gain access to the network. Generally the Network Access Server or router is a RADIUS client, which authenticates its links and user information to a single, central authentication server. User information, such as authentication information and access requirements, is centrally stored in a RADIUS server. The primary responsibility of the RADIUS server is to receive connection requests, authenticate the user, and return the required configuration information. Communications between the RADIUS client and RADIUS server are authenticated with a shared secret.

In this implementation, with an Oracle Advanced Security-enabled Oracle8*i* server configured as the RADIUS client, Oracle passes user information to the same centralized RADIUS server. The RADIUS server either authenticates the user locally or passes the user information to the designated authentication server. Both the RADIUS server and the authentication server can be on the same host or on separate systems. When an Oracle user requests access to an Oracle8*i* server and the user is relying on RADIUS for authentication, the Oracle8*i* server passes the access request to the central RADIUS server. The RADIUS server passes the authentication request to an authentication server, and once properly authenticated, access to the Oracle8*i* server(s) is accepted.

Challenge-Response and Accounting

By delivering RADIUS support, Oracle Advanced Security provides additional enhancements, such as challenge-response support for stronger authentication and accounting of user sessions. Challenge-response mode strengthens security by adding a step to secure the authentication process. Accounting features of RADIUS allow connect time to be logged and utilized for things such as billing for the time a user is logged in to the network.

In challenge-response mode, the user is given a random (unpredictable) number, and the user must respond with a result. The user generates the response by entering the number into a physical device such as a token card or by using software that calculates the required response. The challenge also can be passed to a smart card reader at the client, allowing the smart card to calculate the response. Because

unauthorized users have no way to calculate the appropriate response, they cannot gain access to the network. Challenge-response mode operates in the following way:

- The application server receives an access challenge request from the RADIUS server
- The challenge is passed to the client
- The user is presented a challenge (e.g., a random number) in a client GUI
- The user provides a response to the challenge
- The response is passed via the application server to the RADIUS server
- The RADIUS server validates the response and sends a message accepting access

Enabling RADIUS accounting results in the transmission of a session start and stop packet with each session. RADIUS accounting provides the capability for using audit trails. Accounting information logged by RADIUS can be used for analyzing usage, for security purposes, or for billing purposes. Currently many Internet Service Providers (ISP) use RADIUS for remote access specifically to take advantage of the elaborate billing features enabled by the accounting capabilities built into RADIUS.

While the IETF standard for the RADIUS protocol supports the storage of user authorization information, support has not been incorporated in Oracle Advanced Security for use by the Oracle8*i* data server.

AUTHENTICATION VENDORS: INTEGRATING AUTHENTICATION DEVICES USING RADIUS

Oracle provides the tools for virtually any RADIUS authentication provider to integrate into Oracle Advanced Security, allowing such devices to authenticate Oracle users. This section explains how RADIUS authentication device vendors customize the RADIUS challenge-response user interface to fit their particular devices.

RADIUS Challenge-Response User Interface for Smart Cards

You can set up any authentication device that supports the RADIUS standard to authenticate Oracle users. When the authentication device uses the challenge-response mode, a graphical user interface prompts the user first for a password, then for additional information--for example, a dynamic password that the user obtains from a token card. The interface is Java-based to provide optimal platform independence.

Authentication devices vendors must customize the GUI to fit their particular devices. For example, a smart card vendor can customize the Oracle client to issue the challenge to the smart card reader. Then, when the smart card receives a challenge, it responds by prompting the user for more information, such as a PIN.

Oracle has developed a Java interface class for this interface. It is a set of methods written in C code using the Java Native Interface as specified in release 1.1 of the Java Development Kit™ from JavaSoft. This code is system specific. You can find it in the file Oracle Radius Interface in the appropriate directory specified in the *Oracle Advanced Security Administrator's Guide*.

VIRTUAL PRIVATE DATABASE

The Virtual Private Database— server-enforced, fine-grained access control, together with a secure application context—in the Oracle8*i* server provides a more flexible mechanism for building applications that enforce the security policies customers want enforced, only where such control is necessary. The Virtual Private Database offers the following benefits:

- *Lower cost of ownership.* Organizations can reap huge cost savings by building security once, in the data server, instead of implementing the same security in each application that accesses data.
- *Elimination of the "application security problem."* Users no longer bypass security policies embedded in applications because the security policy is attached to the data. The same security policy is automatically enforced by the data server, no matter how a user accesses data, whether through a report-writing tool, a query, or through an application.
- *New business opportunities.* In the past, organizations couldn't give customers and partners direct access to their production systems, because there was no way to secure the data. Hosting companies couldn't have data for multiple companies reside in the same data server, because they could not separate each company's data. Now, all these scenarios are possible, because fine-grained access control gives you server-enforced data security with the assurance of physical data separation

The following sections describe the functionality of the Virtual Private Database—fine-grained access control and a related feature, secure application context—provided in Oracle8*i*.

Dynamically Modified Queries

Fine-grained access control relies upon "dynamic query modification" to enforce security policies on the objects with which the policies are associated. Here, "query" refers to any selection from a table or view, including data access through a query-for-update, insert or delete statements, or a subquery, not just statements which begin with SELECT.

A user directly or indirectly accessing a table or view having a security policy associated with it causes the server to dynamically modify the statement based on a "WHERE" condition (known as a predicate) returned by a function which implements the security policy. The user's SQL statement is modified dynamically, transparently to the user, using any condition which can be expressed in, or returned by a function. Functions which return predicates can also include callouts to other functions; you could embed a C or Java callout within your PL/SQL package that could either access operating system information or return WHERE clauses from an operating system file or central policy store. You have great flexibility within a policy function, which can return different predicates for each user, each group of users, or each application.

Consider an HR clerk who is only allowed to see employee records in the Aircraft Division. When the user initiates the query "SELECT * FROM emp," the function implementing the security policy returns the predicate "division = 'AIRCRAFT'", and the database transparently rewrites the query, so that the query actually executed becomes "SELECT * FROM emp WHERE division = 'AIRCRAFT'".

As will be shown below, the use of dynamically modified queries is enhanced by use of application context.

Secure Application Contexts

Many organizations want to make access control decisions based on something about the user, such as the user's role in the organization, his organizational unit, whether he is a customer or partner. Application contexts give application developers an easy mechanism to define, set, and validate the security attributes on which to base fine-grained access control and thus enhance the ability of developers to implement the Virtual Private Database within Oracle8*i*. Application contexts offer the benefit of extensibility, ease-of-use, and security.

Extensibility

Application contexts are completely user-definable, as are their attributes, and each application can have its own context, with different attributes. For example, an e-commerce application can base access control on customer number and order number, and a human resources application can base access control on position, organizational unit, and management hierarchy attributes.

Ease of use

Application contexts make fine-grained access control easy to implement because they can be accessed within a policy function to determine the correct predicate to return. For example, HR access control may be based on "position" and "organizational unit." Customers can use application contexts to ensure that a user with the "manager" position is able to see employee records of all employees in his "organizational unit," but a user with an "employee" position is only able to see his own record. Or, you can use a context attribute within an access control condition, for example, to limit customers to just the records matching their "customer number" attribute, which will be different for every user. Applications can set, verify, and retrieve any context attribute conveniently and unambiguously

Security

Application contexts may be safely used to enforce fine-grained access control because the contexts themselves are secure. Oracle8i enforces that context names are unique across an entire database, so that contexts can't be duplicated or spoofed. Also, the ability to create a security context is a separate system privilege. Lastly, the database ensures that whenever a context attribute is set, it is only the trusted package implementing the context that sets the context attribute. As a result, system security officers can comfortably base security decisions on application contexts, because they can be assured that a context is set correctly, by a trusted and known package and not a malicious user or process.

Scalable Security

The Virtual Private Database's fine-grained access control has been designed to be highly scalable, and to use the underlying optimization features of Oracle8i. Under most circumstances, the addition of a security policy to a table should not adversely impact performance. The addition of a WHERE clause, appended dynamically to a statement, occurs before a statement is optimized. This means that the full statement (including the appended WHERE condition) participates in optimization, so that it is parsed and executed efficiently. And of course, the full statement can participate in shared memory, so that any user executing the same statement (including the WHERE condition) can reexecute the statement without reparsing it.

The use of application context with fine-grained access control can deliver even greater performance benefits. because application context can function as a secure data cache. For example, to implement the policy "customers can see their own orders," one could have the actual policy function determine the

customer number for the logged-in user, by querying the CUSTOMERS table. Or, a developer can create an application context having a "cust_num" attribute; the policy function (or functions) can then access the "cust_num" attribute when needed instead of querying the CUSTOMERS table repeatedly. It's the difference between writing an often-used phone number on a Post-It and sticking it on your telephone where you can access it readily, and looking the phone number up each time you need to use it.

While the value of using an application context may not seem evident in such a simple example, consider that many applications have a variety of access control attributes; your policy might be "customers can see their own orders, order entry clerks can update all orders for customers in their region only, sales reps can query orders for only their customers." In this case, your context attributes could include "customer_number," "position" (clerk, customer, rep, manager of rep), and "sales_region." Now you can clearly see the benefit of caching the attribute values for the logged-in user (once), instead of doing multiple queries to retrieve multiple attribute values within a policy function.

Summary

The Virtual Private Database is key enabling technology opening mission-critical systems to partners and customers over the Internet. Fine-grained access control, with secure application contexts, enable organizations to secure data in the Oracle8*i* server, and ensure that, no matter how a user accesses the data (through an application, a report writing tool or SQL*Plus) the same access control policy will be enforced. The Virtual Private Database can help banks ensure that customers see their own accounts and nobody else's, that telecommunications firms can keep customer records safely segregated, and that human resources applications can support their complex rules of data access to employee records. The Virtual Private Database also enables you to lower your cost of development, by building security once, in the data server, instead of in every application that accesses the data.