DBsign®
Data Security Suite

DBsign for HTML Applications version 3.0

Security Target

VERSION 1.2 FINAL

October 17, 2005

Developed for:

Gradkell Systems Inc.

4910 University Square
Suite 2
Huntsville, AL 35816
Main:
Toll Free: 1-866-GRADKELL
(866-472-3535)

Prepared by:

VERIDYN
Information Security for Regulated Industries

8310 N. Capital of Texas Highway, Ste 305
Austin, TX 78731
Main: (512) 310-2228
Toll Free: (877) 321-RISK
Fax: (512) 233-5924

Table of Contents

List of Figures

List of Tables

1.0 ST Introduction

1.1 ST Identification

Title:	DBsign for HTML Applications Version 3.0 Security Target
Version:	1.2
Status:	FINAL
Release Date:	October 17, 2005
Prepared By:	Veridyn, Inc., Gradkell Systems, Inc.
TOE Identifier(s):	DBsign for HTML Applications version 3.0
Assurance Level:	EAL 2
Common Criteria:	Common Criteria for Information Technology Security Evaluation (CC), Version 2.2, January 2004 (aligned with ISO/IEC 15408). Common Methodology for Information Technology Security Evaluation (CEM), Version 2.2, January 2004 (aligned with ISO/IEC 18045).
Interpretations:	Final National and International interpretations included within this ST that that have been released on or before the kick-off date, June 7, 2004, are identified within section 9.3 of this ST.
Keywords:	Digital Signature, Non-Repudiation, PKI, Database Integrity

1.2 CC Conformance

This TOE is:

CC Version 2.2 Part 1 – CONFORMANT

CC Version 2.2 Part 2 – EXTENDED

CC Version 2.2 Part 3 – CONFORMANT

EAL2 – CONFORMANT

1.3 Document Conventions

Assignment: An assignment allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]).

Iteration: An iteration allows for the use of a component more than once with varying operations. Iterations are indicated with a lowercase alphabetic character (e.g. FAU_GEN.1a).

Refinement: A refinement allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …"). Refinements resulting from an interpretation are additionally indicated with a red font.

Selection: A selection allows the specification of one or more elements from a list. Selections are indicated using italics and are surrounded by brackets (e.g., [***selection***]).

1.4 ST Overview

This Security Target (ST) defines the security environment, security requirements, and security functions of DBsign for HTML Applications Version 3.0, hereafter referred to as DBsign. DBsign consists of a digital signature system that provides provable cryptographic data integrity and non-repudiation for data stored in relational databases. DBsign supports digital signature operations for both statically stored data and application-constructed data stored within memory buffers or files. A co-existing application can interface to DBsign using DBsign's API or plug-in/control functions to perform digital signature operations for the given application.

The following sections are provided within this ST:

ST Introduction:	The ST introduction provides a unique identification and overview of this ST.
TOE Description:	The TOE description provides an overview of the TOE and describes the physical and logical boundaries of the TOE.
TOE Security Environment:	The security environment describes the assumptions, threats, and organizational security policies that pertain to both the TOE and TOE environment.
Security Objectives:	The security objectives describe the objectives necessary to counter the defined threats and satisfy the assumptions and organizational security policies.
IT Security Requirements:	The IT security requirements provide a set of security functional requirements to be met by the TOE and the TOE environment. The IT security requirements also provide a set of security assurance requirements that are to be satisfied by the TOE.
TOE Summary Specification:	The TOE Summary Specification describes the security functions of the TOE.
PP Claims:	The PP claims identify any PPs that the TOE claims compliance to.
Rationale:	The rationale provides mappings along with rationale for the security environment, security objectives, security requirements, and security functions to assess their completeness, consistency, and suitability.
Annex A:	Annex A lists the acronyms, terms, interpretations, and references used within this ST.

2.0 TOE Description

2.1 DBsign Overview

DBsign is a digital signature system designed specifically to perform digital signature generation and verification which provides provable methods to verify cryptographic data integrity and non-repudiation for data stored in relational databases. DBsign includes both a Software Development Kit (SDK) and a set of graphical administration tools that work together to make the integration of digital signatures into database driven applications a quick and easy process.

The DBsign SDK includes a simple, high-level application programming interface (API) that minimizes changes to existing application code. No specialized cryptographic or digital signature knowledge is required of developers or users. The DBsign SDK provides an interface to DBsign for a co-existing application so that the co-existing application may integrate the digital signature security functionalities of DBsign without the need of having to integrate the actual source code of DBsign into the co-existing application. Therefore, DBsign may be programmatically integrated into a co-existing application without the capability of modifying the security functionalities incorporated by DBsign.

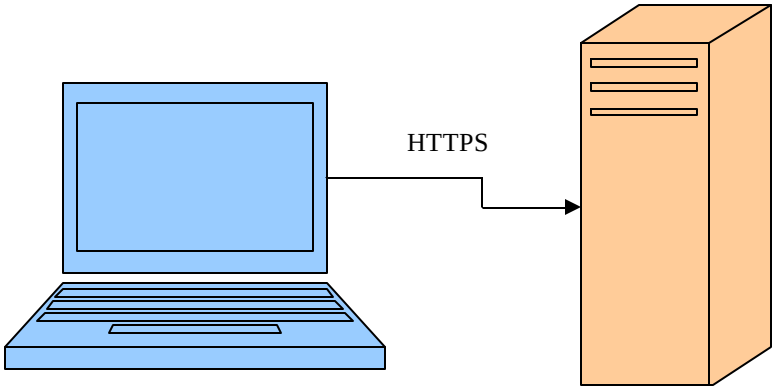
The DBsign Administration Tools is a Graphical User Interface (GUI) that allows for the DBsign Administrator to control the security and configuration parameters under which DBsign operates. The tools provide a means for the DBsign administrator to centrally configure and maintain the digital signature system. The DBsign Administration Tools may be used to configure and maintain multiple DBsign installations, however, the DBsign Administration Tools only allow for one installation at a time to be configured or maintained.

DBsign performs digital signature generation and verification using the DBsign Crypto Adaptor (DCA) which utilizes the RSA BSAFE Crypto-C Toolkit version 5.2.1 to perform the cryptographic operations. The RSA BASFE Crypto-C toolkit is FIPS 140-1 validated and provides protection of the signer's private key. All digital signature generation is performed on the client and all digital signature verification is performed on the server.

2.1.1 TOE Configuration

The client communicates with DBsign via a control/plugin within their web browser. Therefore the web browser is pointed to the web server hosting DBsign3.0 via HTTPS and the web server redirects the query to the application server in which DBsign resides. DBsign communicates with the database to retrieve data to be signed by the client. This configuration of DBsign supports most RDBMS.

Figure 1: DBsign Configuration



DBsign additionally provides an optional security feature called the User Policy feature. The User Policy feature provides access control enforcement to digital signatures using templates.

The User Policy feature is not include as part of the evaluated configuration of the TOE, therefore, this security feature cannot be guaranteed to perform its defined security functionality. If a third-party application developer wishes to implement this security feature, then this must be done at their own risk.

DBsign additionally provides an optional security feature called the Notary Signing. The Notary Signing feature provides server-side signing capability.

The Notary Signing feature is not included as part of the evaluated configuration of the TOE, therefore, this security feature cannot be guaranteed to perform its defined security functionality. If a third-party application developer wishes to implement this security feature, then this must be done at their own risk.

Entrust is not included as part of the evaluated configuration of the TOE.

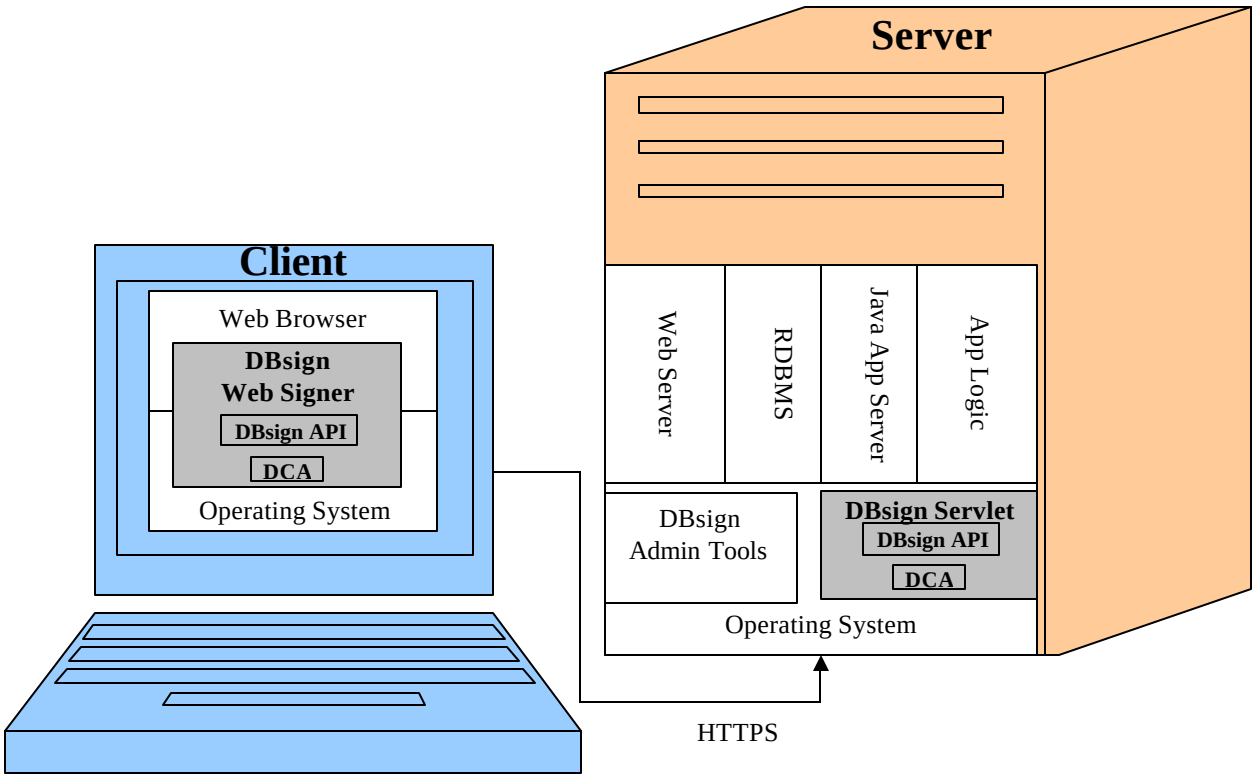
2.2 TOE Boundaries

2.2.1 Physical Boundaries

At a minimum, DBsign consists of two physical computers. DBsign supports multiple clients to a server, however, at least one client is required to support the full functionality of DBsign. The first computer is the client, which includes an operating system, a web browser client, and the DBsign Web Signer control/plugin installed. The second computer is the server which includes an operating system, web server, Java application server, App Logic, RDBMS, the DBsign Administration Tools, and the DBsign Web Servlet. The TOE also requires connectivity between the client and server to support the digital signature operations performed by DBsign.

The following figure depicts the physical architecture of DBsign. The grayed rectangles labeled DBsign Web Signer and DBsign Servlet represent the TOE components and boundaries in a physical aspect in relation to the non-TOE components. The non-TOE components of the client include the operating system, web browser, and the underlying hardware The non-TOE components of the server include the operating system, web server, Java application server, App Logic, the RDBMS[1], the DBsign Administration Tools, and the underlying hardware. In addition, the HTTPS protocol used to communicate between the client and server is also a non-TOE component.

Figure 2: DBsign Physical Boundaries



[1] The audit data and DBS tables reside in the RDBMS, which is in the TOE environment.


Gradkell
Systems Inc.


VERIDYN
Information Security for Regulated Industry

2.2.2 Logical Boundaries

This section identifies the logical boundaries of the TOE in terms of the IT security features provided by the TOE. The IT security features include auditing and digital signature operations. A description of each IT security feature identified is provided in the following subsections.

2.2.2.1 Auditing

The TOE provides auditing record generation capabilities for digitally signing data and verifying the digital signature of data. The auditing record generation capabilities of the TOE also report any integrity violations for verifications that are performed. It also identifies the specific data that has been modified.

2.2.2.2 Digital Signature

The TOE provides the capability to perform digital signature operations which include digitally signing data and verifying digitally signed data. The TOE supports the defined digital signature operations specified in FCS_COP.1 and FCO_NRO.1 on statically stored data within a database. DBsign additionally provides the capability to perform the defined digital signature operations against application-constructed data stored in memory buffers or files. The TOE utilizes the defined digital signature operations to integrate with third-party applications that require the use of the digital signature operations that the TOE provides.

2.3 System Requirements

This section identifies the minimum software and hardware requirements applicable to DBsign. The hardware requirements for all DBsign components are dependent upon the minimum requirements stated for the selected operating system. Therefore, this section will only identify the minimum software requirements required for DBsign and assume that an administrator will install DBsign using hardware that meets the minimum hardware requirements specified for the selected operating system.

Client:

- 1 Network interface card
- Microsoft Windows 98, Me, NT, 2000, XP, 2003
- Database client that supports DB2-CLI, JDBC, ODBC, OCI 7.0, OCI 8.0, or OCI 8i

For DBsign Web Signer Plugin:

- Netscape Navigator 4.x, Microsoft Internet Explorer 4.x-5.5 SP1[2]

For DBsign Web Signer Control:

- Microsoft Internet Explorer 4.x and higher

For DBsign Administration Tools:

- Java 1.3 (or higher) Java Runtime Environment (JRE)

Server:

- 1 Network interface card
- Java Virtual Machine version 1.3 or higher
- J2EE compliant Java application server supporting the Java Servlet API version 2.2 or higher
- Operating system that is supported by the Java application server

[2] Netscape-style plug-ins are not supported by Internet Explorer versions 5.5 SP2 and higher. Further information regarding this issue can be found in Microsoft's Knowledge Base Article #303401.

3.0 TOE Security Environment

Table 1: TOE Security Environment

Assumptions
A.ADMIN
A.LOCATE
A.INSTALLER
A.USER_ID
Threats
T.AUDIT_SEQUENCE
T.KEY_COMPROMISE
T.MODIFY
T.NO_LOG
T.USER_DENY

3.1 Assumptions

3.1.1 Administrator Assumptions

A.ADMIN It is assumed that one or more authorized administrators are assigned who are competent to manage the TOE, the IT environment supporting the TOE, the security of the information the TOE contains, and who can be trusted not to deliberately abuse their privileges so as to undermine security.

A.INSTALLER It is assumed that the installer of the TOE is provided by Gradkell and has sufficient expertise and knowledge to properly install the TOE within its evaluated configuration.

3.1.2 Physical Assumptions

A.LOCATE The processing resources of the TOE are assumed to be located within controlled access facilities that will restrict unauthorized physical access.

3.1.3 User Assumptions

A.USER_ID It is assumed that the certificate user or certificate user's certificate authority has correctly associated the certificate user's user identity and certificate issuer with their certificate.

3.2 Threats

3.2.1 Auditing Threats

T.NO_LOG A user may receive an integrity violation while verifying a digital signature and the integrity violation does not get recorded.

3.2.2 Cryptography Threats

T.KEY_COMPROMISE A user utilizes a non-FIPS 140-1 or non-FIPS 140-2 conformant cryptographic mechanism for generating a cryptographic key to be used with DBsign and the cryptographic key is compromised by an attacker.

3.2.3 Integrity Threats

T.MODIFY The integrity of data stored, processed, or transmitted may be compromised due to the unauthorized modification or destruction of the data or stored digital signatures by an attacker.

3.2.4 Non-repudiation Threats

T.USER_DENY A user denies having modified or inserted a database record that is digitally signed by that user.

3.2.5 Time Stamping Threats

T.AUDIT_SEQUENCE An administrator is unable to distinguish the sequence of audit events and therefore cannot detect recent integrity violations.

4.0 Security Objectives

Table 2: Security Objectives

Security Objectives for the TOE
O.AUDIT
O.CRYPTO_OPERATION
O.INTEGRITY
Security Objectives for the IT Environment
OE.AUDIT_REVIEW
OE.CRYPTO_OPERATION
OE.TIMESTAMP
Security Objectives for the Non-IT Environment
OE.ADMIN_GUIDANCE
OE.CERTIFICATE_USERS
OE.PHYSICAL_CONTROL
OE.TOE_INSTALLATION

4.1 Security Objectives for the TOE

O.AUDIT The TOE will provide the means of generating any security relevant events, so as to assist an administrator in the detection of potential attacks or misconfiguration of the TOE security features.

O.CRYPTO_OPERATION The TOE shall provide cryptographic operations necessary for digitally signing data and verifying the digital signature applied to data.

O.INTEGRITY The TOE will provide the means to verify the integrity of data that has been digitally signed by the TOE.

4.2 Security Objectives for the Environment

4.2.1 Security Objectives for the IT Environment

OE.AUDIT_REVIEW	Provide a means to review, search, and sort audit records generated by the TOE.
OE.CRYPTO_OPERATION	Provide FIPS 140-1[3] or FIPS 140-2 compliant cryptographic key operations necessary to enable a user to utilize their digital signature.
OE.TIMESTAMP	Provide a time stamping mechanism that can be relied upon to provide an accurate date and time.

4.2.2 Security Objectives for the Non-IT Environment

OE.ADMIN_GUIDANCE	Deter administrator errors by providing adequate administrator guidance.
OE.CERTIFICATE_USERS	Certificate users of the TOE shall associate an accurate user identity with their certificate.
OE.PHYSICAL_CONTROL	TOE data shall be physically protected to prevent unauthorized disclosure, destruction, or modification.
OE.TOE_INSTALLATION	The TOE shall be properly installed by a competent individual in accordance with its evaluated configuration.

[3] The RSA BASFE Crypto-C toolkit is FIPS 140-1 validated and provides protection of the signer's private key.

5.0 IT Security Requirements

Table 3: IT Security Requirements

Security Functional Requirements for the TOE	CC Conformance:
FAU_REC.1: DBsign audit record generation	Explicitly Stated
FCO_NRO.1: Selective proof of origin	Drawn from CC Part 2
FCS_COP.1: Cryptographic operation	Drawn from CC Part 2
FIA_CID.2: Certificate identification before any action	Explicitly Stated
Security Functional Requirements for the IT Environment	**CC Conformance:**
FAU_SAR.1: Audit review	Drawn from CC Part 2
FAU_SAR.3: Selectable audit review	Drawn from CC Part 2
FMT_MSA.2: Secure security attributes	Drawn from CC Part 2
FCS_CKM.1: Cryptographic key generation	Drawn from CC Part 2
The **IT Environment** shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**FIPS 140-1 approved algorithms (rDSA, ECDSA, or DSS) or FIPS 140-2 approved algorithms (DSA, ECDSA, or RSA)**] and specified cryptographic key sizes [**(512 or 1024 bit for DSS & DSA), (80, 112, 128, 192, or 256 bit for ECDSA), or (multiples of 256 bit for RSA)**] that meet the following: [**ANSI X9.31-1998 (rDSA), ANSI X9.62-1998 (ECDSA), FIPS 186-2 (DSS & DSA), or PKCS #1 v2.1 (RSA)**].	Drawn from CC Part 2
FCS_CKM.4: Cryptographic key destruction	
FPT_STM.1: Reliable time stamps	Drawn from CC Part 2
Security Assurance Requirements for the TOE	**CC Conformance:**
ACM_CAP.2: Configuration items	Drawn from CC Part 3
ADO_DEL.1: Delivery procedures	Drawn from CC Part 3
ADO_IGS.1: Installation generation and start-up procedures	Drawn from CC Part 3
ADV_FSP.1: Informal functional specification	Drawn from CC Part 3
ADV_HLD.1: Descriptive high-level design	Drawn from CC Part 3
ADV_RCR.1: Informal correspondence demonstration	Drawn from CC Part 3
AGD_ADM.1: administrator guidance	Drawn from CC Part 3
AGD_USR.1: User guidance	Drawn from CC Part 3
ATE_COV.1: Evidence of coverage	Drawn from CC Part 3
ATE_FUN.1: Functional testing	Drawn from CC Part 3
ATE_IND.2: Independent testing – sample	Drawn from CC Part 3
AVA_SOF.1: Strength of TOE security function evaluation	Drawn from CC Part 3
AVA_VLA.1: Developer vulnerability analysis	Drawn from CC Part 3

5.1 Security Functional Requirements for the TOE

5.1.1 FCO (Communication)

5.1.1.1 FCO_NRO: Non-repudiation of origin

5.1.1.1.1 FCO_NRO.1: Selective proof of origin

FCO_NRO.1.1

The TSF shall be able to generate evidence of origin for transmitted [**data stored within a database, memory buffer, or file**] at the request of the [***originator***].

FCO_NRO.1.2

The TSF shall be able to relate the [certificate] of the originator of the information, and the [**data stored within a database, memory buffer, or file**] of the information to which the evidence applies.

FCO_NRO.1.3

The TSF shall provide a capability to verify the evidence of origin of information to [***originator,*** [**or recipients**]] given [**the digital signature, the originator's certificate and the data stored within a database, memory buffer, or file**].

5.1.2 FCS (Cryptographic Support)

5.1.2.1 FCS_COP: Cryptographic operation

5.1.2.1.1 FCS_COP.1: Cryptographic operation

FCS_COP.1.1

The TSF shall perfo rm [**digitally signing data and verification of digitally signed data for data stored within a database, memory buffer, or file**] in accordance with a specified cryptographic algorithm [**RSA or DSA**] and cryptographic key sizes [**256-2048**] that meet the following: [**ANSI X9.31 (RSA) or FIPS 186-2 (DSA)**].

5.2 Explicitly Stated Security Functional Requirements for the TOE

5.2.1 FAU (Security Audit)

5.2.1.1 FAU_REC: DBsign audit record generation

5.2.1.1.1 FAU_REC.1: DBsign audit record generation

FAU_REC.1.1

The TSF shall be able to generate an audit record of the following auditable events:

a) The following log types:

DBsign Sign (4)	logged attempts to sign data
DBsign Verify (8)	logged attempts to verify the signatures on data

FAU_REC.1.2

The TSF shall include for signing and verification operations related to data stored in a database within each audit record at least the following information:

a) Date and time of the event (LOG_DATE), type of event (LOG_TYPE), and the outcome (success or failure) of the event; and

b) For each audit event type, log number (LOG_NO), status code (LOG_STATUS), log message (LOG_MESG), and data (LOG_DATA); and

c) For each audit event type which results in success, the following additional fields: template id (TEMPLATE_ID), sign date (SIGN_DATE), signer certificate id (SIGNER_CERT_ID) and signature (SIGNATURE); and

d) For each audit event type which results in success and when templates contain primary Keys, template primary keys (PRIMARY_KEYS).

5.2.2 FIA (Identification and Authentication)

5.2.2.1 FIA_CID: Certificate identification

5.2.2.1.1 FIA_CID.2: Certificate identification before any action

FIA_CID.2.1

The TSF shall require each originator to present a certificate before allowing any other TSF-mediated actions on behalf of that originator.[4]

[4] The binding between the originator's private key and the certificate provides the support for nonrepudiation (FCO_NRO.1).

5.3 Security Functional Requirements for the IT Environment

5.3.1 FAU (Security Audit)

5.3.1.1 FAU_SAR: Security audit review

5.3.1.1.1 FAU_SAR.1: Audit review

FAU_SAR.1.1

The **IT Environment** shall provide [**administrator**] with the capability to read [**all DBsign logged events**] from the audit records.

FAU_SAR.1.2

The **IT Environment** shall provide the audit records in a manner suitable for the user to interpret the information.

5.3.1.1.2 FAU_SAR.3: Selectable audit review

FAU_SAR.3.1

The **IT Environment** shall provide the ability to perform [*sorting*] of audit data based on [**log number, date after, date before, log type, log status, template, signer dbs certs id, verifier dbs certs id, sign date after, sign date before, or primary key custom values**].

5.3.2 FMT (Secure security attributes)

5.3.2.1 FMT_MSA.2 Secure Security Attributes

FMT_MSA.2.1

The IT Environment shall ensure that only secure values are accepted for security attributes.[5]

5.3.3 FCS (Cryptographic Support)

5.3.3.1 FCS_CKM: Cryptographic key management

5.3.3.1.1 FCS_CKM.1: Cryptographic key generation

FCS_CKM.1.1

The **IT Environment** shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**FIPS 140-1 approved algorithms (rDSA, ECDSA, or DSS) or FIPS 140-2 approved algorithms (DSA, ECDSA, or RSA)**] and specified cryptographic key sizes [**(512 or 1024 bit for DSS & DSA), (80, 112, 128, 192, or 256 bit for ECDSA), or (multiples of 256 bit for RSA)**] that meet the following: [**ANSI X9.31-1998 (rDSA), ANSI X9.62-1998 (ECDSA), FIPS 186-2 (DSS & DSA), or PKCS #1 v2.1 (RSA)**].

[5] The security attributes are the cryptographic key attributes (e.g., key size, key use, etc.) and that this SFR supports the FCS SFRs.

5.3.3.1.2 FCS_CKM.4: Cryptographic key destruction

FCS_CKM.4.1

The **IT Environment** shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [**that provides a zeroization method that is sufficient not to compromise plaintext secret and private keys**] that meets the following: [**FIPS 140-1 or FIPS 140-2 standard with a minimum of a Level 1 of assurance**].

5.3.4 FPT (Protection of the TSF)

5.3.4.1 FPT_STM: Time stamps

5.3.4.1.1 FPT_STM.1: Reliable time stamps

FPT_STM.1.1

The **IT Environment** shall be able to provide reliable time stamps for its own use.

5.4 Security Assurance Requirements for the TOE

EAL 2 – Structurally tested

EAL2 requires the co-operation of the developer in terms of the delivery of design information and test results, but should not demand more effort on the part of the developer than is consistent with good commercial practice. As such it should not require a substantially increased investment of cost or time.

EAL2 is therefore applicable in those circumstances where developers or users require a low to moderate level of independently assured security in the absence of ready availability of the complete development record. Such a situation may arise when securing legacy systems, or where access to the developer may be limited.

EAL2 provides assurance by an analysis of the security functions, using a functional and interface specification, guidance documentation and the high-level design of the TOE, to understand the security behavior.

The analysis is supported by independent testing of the TOE security functions, evidence of developer testing based on the functional specification, selective independent confirmation of the developer test results, strength of function analysis, and evidence of a developer search for obvious vulnerabilities (e.g. those in the public domain).

EAL2 also provides assurance through a configuration list for the TOE, and evidence of secure delivery procedures.

This EAL represents a meaningful increase in assurance from EAL1 by requiring developer testing, a vulnerability analysis, and independent testing based upon more detailed TOE specifications.

5.4.1 ACM: Configuration Management

Configuration management (CM) is one method or means for establishing that the functional requirements and specifications are realized in the implementation of the TOE. CM meets these objectives by requiring discipline and control in the processes of refinement and modification of the TOE and the related information. CM systems are put in place to ensure the integrity of the portions of the TOE that they control, by providing a method of tracking any changes, and by ensuring that all changes are authorized.

5.4.1.1 ACM_CAP.2: Configuration items

A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using.

Unique identification of the configuration items leads to a clearer understanding of the composition of the TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the TOE.

Developer action elements:

ACM_CAP.2.1D The developer shall provide a reference for the TOE.

ACM_CAP.2.2D The developer shall use a CM system.

ACM_CAP.2.3D The developer shall provide CM documentation.

Content and presentation of evidence elements:

ACM_CAP.2.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.2.2C The TOE shall be labelled with its reference.

ACM_CAP.2.3C The CM documentation shall include a configuration list.

ACM_CAP.2.4C The configuration list shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.2.5C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.2.6C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.2.7C The CM system shall uniquely identify all configuration items.

Evaluator action elements:

ACM_CAP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.4.2 ADO: Delivery and Operation

Delivery and operation provides requirements for correct delivery, installation, generation, and start-up of the TOE.

5.4.2.1 ADO_DEL.1: Delivery procedures

The requirements for delivery call for system control and distribution facilities and procedures that detail the measures necessary to provide assurance that the security of the TOE is maintained during distribution of the TOE. For a valid distribution of the TOE, the procedures used for the distribution of the TOE address the threats identified in the PP/ST relating to the security of the TOE during delivery.

Developer action elements:

ADO_DEL.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.1.2D The developer shall use the delivery procedures.

Content and presentation of evidence elements:

ADO_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

Evaluator action elements:

ADO_DEL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.4.2.2 ADO_IGS.1: Installation generation and start-up procedures

Installation, generation, and start-up procedures are useful for ensuring that the TOE has been installed, generated, and started up in a secure manner as intended by the developer. The requirements for installation, generation and start-up call for a secure transition from the TOE's implementation representation being under configuration control to its initial operation in the user environment.

Developer action elements:

ADO_IGS.1.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

ADO_IGS.1.1C The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.

Evaluator action elements:

ADO_IGS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.4.3 ADV: Development

The development class encompasses four families of requirements for representing the TSF at various levels of abstraction from the functional interface to the implementation representation. The development class also includes a family of requirements for a correspondence mapping between the various TSF representations, ultimately requiring a demonstration of correspondence from the least abstract representation through all intervening representations to the TOE summary specification provided in the ST. In addition, there is a family of requirements for a TSP model, and for correspondence mappings between the TSP, the TSP model, and the functional specification. Finally, there is a family of requirements on the internal structure of the TSF, which covers aspects such as modularity, layering, and minimization of complexity.

5.4.3.1 ADV_FSP.1: Informal functional specification

The functional specification is a high-level description of the user-visible interface and behavior of the TSF. It is an instantiation of the TOE security functional requirements. The functional specification has to show that all the TOE security functional requirements are addressed.

Developer action elements:

ADV_FSP.1.1D The developer shall provide a functional specification.

Content and presentation of evidence elements:

ADV_FSP.1.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.1.2C The functional specification shall be internally consistent.

ADV_FSP.1.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

ADV_FSP.1.4C The functional specification shall completely represent the TSF.

Evaluator action elements:

ADV_FSP.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

5.4.3.2 ADV_HLD.1: Descriptive high-level design

The high-level design of a TOE provides a description of the TSF in terms of major structural units (i.e. subsystems) and relates these units to the functions that they provide. The high-level design requirements are intended to provide assurance that the TOE provides an architecture appropriate to implement the TOE security functional requirements.

The high-level design refines the functional specification into subsystems. For each subsystem of the TSF, the high-level design describes its purpose and function, and identifies the security functions contained in the subsystem. The interrelationships of all subsystems are also defined in the high-level design. These interrelationships will be represented as external interfaces for data flow, control flow, etc., as appropriate.

Developer action elements:

ADV_HLD.1.1D The developer shall provide the high-level design of the TSF.

Content and presentation of evidence elements:

ADV_HLD.1.1C The presentation of the high-level design shall be informal.

ADV_HLD.1.2C The high-level design shall be internally consistent.

ADV_HLD.1.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.1.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.1.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.1.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.1.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

Evaluator action elements:

ADV_HLD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_HLD.1.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.4.3.3 ADV_RCR.1: Informal correspondence demonstration

The correspondence between the various TSF representations (i.e. TOE summary specification, functional specification, high-level design, low-level design, and implementation representation) addresses the correct and complete instantiation of the requirements to the least abstract TSF representation provided. This conclusion is achieved by step-wise refinement and the cumulative results of correspondence determinations between all adjacent abstractions of representation.

Developer action elements:

ADV_RCR.1.1D The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

Content and presentation of evidence elements:

ADV_RCR.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Evaluator action elements:

ADV_RCR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.4.4 AGD: Guidance Documents

The guidance documents class provides the requirements for user and administrator guidance documentation. For the secure administration and use of the TOE it is necessary to describe all relevant aspects for the secure application of the TOE.

5.4.4.1 AGD_ADM.1: administrator guidance

Administrator guidance refers to written material that is intended to be used by those persons responsible for configuring, maintaining, and administering the TOE in a correct manner for maximum security. Because the secure operation of the TOE is dependent upon the correct performance of the TSF, persons responsible for performing these functions are trusted by the TSF. Administrator guidance is intended to help administrators understand the security functions provided by the TOE, including both those functions that require the administrator to perform security-critical actions and those functions that provide security-critical information.

Developer action elements:

AGD_ADM.1.1D The developer shall provide administrator guidance addressed to system administrative personnel.

Content and presentation of evidence elements:

AGD_ADM.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.

AGD_ADM.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

AGD_ADM.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.4.4.2 AGD_USR.1: User guidance

User guidance refers to material that is intended to be used by non-administrative human users of the TOE, and by others (e.g. programmers) using the TOE's external interfaces. User guidance describes the security

functions provided by the TSF and provides instructions and guidelines, including warnings, for its secure use.

The user guidance provides a basis for assumptions about the use of the TOE and a measure of confidence that non-malicious users, application providers and others exercising the external interfaces of the TOE will understand the secure operation of the TOE and will use it as intended.

Developer action elements:

AGD_USR.1.1D The developer shall provide user guidance.

Content and presentation of evidence elements:

AGD_USR.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.

AGD_USR.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

AGD_USR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.4.5 ATE: Tests

The class "Tests" encompasses four families: coverage (ATE_COV), independent testing (e.g. functional testing performed by evaluators) (ATE_IND), and functional tests (ATE_FUN). Testing helps to establish that the TOE security functional requirements are met. Testing provides assurance that the TOE satisfies at least the TOE security functional requirements, although it cannot establish that the TOE does no more than what was specified. Testing may also be directed toward the internal structure of the TSF, such as the testing of subsystems and modules against their specifications.

The aspects of coverage and depth have been separated from functional tests for reasons of increased flexibility in applying the components of the families. However, the requirements in these three families are intended to be applied together.

The independent testing family has dependencies on the other families to provide the necessary information to support the requirements, but is primarily concerned with independent evaluator actions.

The emphasis in this class is on confirmation that the TSF operates according to its specification. This will include both positive testing based on functional requirements, and negative testing to check that undesirable behavior is absent. This class does not address penetration testing, which is directed toward finding vulnerabilities that enable a user to violate the security policy. Penetration testing is based upon an analysis of the TOE that specifically seeks to identify vulnerabilities in the design and implementation of the TSF, and is addressed separately as an aspect of vulnerability assessment in the class AVA.

5.4.5.1 ATE_COV.1: Evidence of coverage

In this component, the objective is to establish that the TSF has been tested against its functional specification. This is to be achieved through an examination of developer evidence of correspondence.

Developer action elements:

ATE_COV.1.1D The developer shall provide evidence of the test coverage.

Content and presentation of evidence elements:

ATE_COV.1.1C The evidence of the test coverage shall show the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

Evaluator action elements:

ATE_COV.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.4.5.2 ATE_FUN.1: Functional testing

Functional testing performed by the developer establishes that the TSF exhibits the properties necessary to satisfy the functional requirements of its PP/ST. Such functional testing provides assurance that the TSF satisfies at least the security functional requirements, although it cannot establish that the TSF does no more than what was specified. The family "Functional tests" is focused on the type and amount of documentation or support tools required, and what is to be demonstrated through developer testing. Functional testing is not limited to positive confirmation that the required security functions are provided, but may also include negative testing to check for the absence of particular undesired behavior (often based on the inversion of functional requirements).

The objective is for the developer to demonstrate that all security functions perform as specified. The developer is required to perform testing and to provide test documentation.

Developer action elements:

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation of evidence elements:

ATE_FUN.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

Evaluator action elements:

ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.4.5.3 ATE_IND.2: Independent testing – sample

The objective is to demonstrate that the security functions perform as specified. Evaluator testing includes selecting and repeating a sample of the developer tests.

Developer action elements:

ATE_IND.2.1D The developer shall provide the TOE for testing.

Content and presentation of evidence elements:

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Evaluator action elements:

ATE_IND.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.4.6 AVA: Vulnerability Assessment

The class addresses the existence of exploitable covert channels, the possibility of misuse or incorrect configuration of the TOE, the possibility to defeat probabilistic or permutational mechanisms, and the possibility of exploitable vulnerabilities introduced in the development or the operation of the TOE.

5.4.6.1 AVA_SOF.1: Strength of TOE security function evaluation

Even if a TOE security function cannot be bypassed, deactivated, or corrupted, it may still be possible to defeat it because there is a vulnerability in the concept of its underlying security mechanisms. For those functions a qualification of their security behavior can be made using the results of a quantitative or statistical analysis of the security behavior of these mechanisms and the effort required to overcome them. The qualification is made in the form of a strength of TOE security function claim.

Developer action elements:

AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

Content and presentation of evidence elements:

AVA_SOF.1.1C For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2C For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

Evaluator action elements:

AVA_SOF.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_SOF.1.2E The evaluator shall confirm that the strength claims are correct.

5.4.6.2 AVA_VLA.1: Developer vulnerability analysis

Vulnerability analysis is an assessment to determine whether vulnerabilities identified, during the evaluation of the construction and anticipated operation of the TOE or by other methods (e.g. by flaw hypotheses), could allow users to violate the TSP.

Vulnerability analysis deals with the threats that a user will be able to discover flaws that will allow unauthorized access to resources (e.g. data), allow the ability to interfere with or alter the TSF, or interfere with the authorized capabilities of other users.

A vulnerability analysis is performed by the developer to ascertain the presence of obvious security vulnerabilities, and to confirm that they cannot be exploited in the intended environment for the TOE.

Developer action elements:

AVA_VLA.1.1D The developer shall perform a vulnerability analysis.

AVA_VLA.1.2D The developer shall provide vulnerability analysis documentation.

Content and presentation of evidence elements:

AVA_VLA.1.1C The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP.

AVA_VLA.1.2C The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities.

AVA_VLA.1.3C The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

Evaluator action elements:

AVA_VLA.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.1.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

5.5 Strength of Function Claim

There is no strength of function claim specified for this security target.

6.0 TOE Summary Specification

6.1 TOE Security Functions

Table 4: TOE Security Functions

TOE SECURITY FUNCTIONS
Auditing
Digital Signature

6.1.1 Auditing

The TOE provides the capability to generate audit events as a result of successful and failing requests for DBsign to digitally sign data or verify digitally signed data. In the event that a digital signature generation or verification process has failed or has been prevented from processing, an audit event is generated by DBsign and recorded into the DBsign audit log.. The DBsign audit log is stored in the database as database tables using the following format "DBS_LOG_xxx". These tables are linked by a log entry number (LOG_NO) that is generated internally by DBsign.

The audit logging system requires its own database connection. This is to ensure that the logged data can be committed to the database without effecting the application's current transaction. It also ensures that the logged data will not get rolled back by the application should the application abort a transaction.

The DBsign audit logging system can be used to research signature failures. When a signature fails to verify because data was changed, it is important to be able to determine which data items were changed. DBsign accomplishes this by logging a copy of the data (in a highly compressed form) whenever data is signed. This allows DBsign to present a "before and after" picture of the data and to identify the offending data elements.

The audit logging feature may be enabled or disabled by the administrator. However, the evaluated configuration of the TOE requires, at a minimum, for the audit logging feature to be enabled to audit the successful and failed signature generation and signature verification processes. This is required to support the determination of which data elements were changed.

FAU_REC.1: DBsign audit record generation

There are two types of DBsign log entries which include sign (4), and verify (8). The sign (4) log entry indicates successful and failed attempts to perform digital signature generation. The verify (8) log entry indicates successful and failed attempts to perform digital signature verification. Each audit event recorded includes the date and time of the event, type of event, user's authenticated identity, the outcome (success or failure) of the event, log number, status code, status error, and message.

Note that the audit log information indicates success whenever the LOG_STATUS equals zero and failure when LOG_STATUS does not equal zero.

Also note that in the case that a template does not contain primary keys, there will be no primary key items in the audit log information for events that reference that template.

Also note that failed audit log entries may not include some of the audit log fields because the event failed because they could not be determined. For example, if a verification event fails because data was not

signed, that event will be missing SIGN_DATE, SIGNER_CERT_ID and SIGNATURE. If a numeric field is missing, it may have a value that is < 0 (e.g., "-1" means that no value for this field exists).

Also note that only signing and verification operations related to data stored in a database generate log records. No log records are generated for file or buffer signing and verification.

6.1.2 Digital Signature

The TOE provides a digital signature function which, in general, enables a user to generate and verify a digital signature applied to data. This allows for the author of the signed data to be uniquely identified and for the authenticity of the signed data to be verified. In addition, the digital signature function enforces personal accountability for approved changes made by an administrator to the security sensitive configuration data contained in the DBsign system tables. The TOE digitally signs data and verifies digitally signed data and data integrity using the RSA BSAFE Crypto-C Toolkit version 5.2.1.

The TOE provides data integrity verification by enabling applications to verify the data integrity of previous transactions from unauthorized modification, based on the originator's digital signature. The data integrity verification process executes in real-time, before proceeding with the transaction currently being processed. Since DBsign is tightly integrated into the application, this verification happens automatically with no user intervention. The data integrity verification function is performed whenever the digital signature function verifies digitally signed data using the DBS_CheckSig() API function or plug-in/control method.

FCS_COP.1: Cryptographic operation

The Digital Signature security function provides DBsign the capability to digitally sign and verify digitally signed data stored within a database, memory buffer, or file.

To digitally sign data stored within a database, a user must initiate a DBsign session and then make a call to the DBS_MakeSig() plug-in/control method. The DBS_MakeSig() plug-in/control method is a part of the DBsign plug-in/control which provides developers a way to integrate the DBsign digital signature functionality into their product. When DBS_MakeSig() is called upon, DBsign checks the primary key values as defined by the signature template. When the digital signing operation has completed, DBS_MakeSig() logs the action to the DBsign audit log and records whether the event was a success or failure.

To digitally sign application-constructed data stored in a memory buffer or a file, a user must initiate a DBsign session and then make a call to the DBS_AppSign() plug-in/control method.

To verify digitally signed data stored within a database, a user must initiate a DBsign session and then make a call to the DBS_CheckSig()plug-in/control method. The DBS_CheckSig()plug-in/control method is a part of the DBsign plug-in/control which provides developers a way to integrate the DBsign digital signature verification functionality into their product. When DBS_CheckSig() is called upon, DBsign checks the primary key values as defined by the signature template. When the digital signing operation has completed, DBS_CheckSig() logs the action to the DBsign audit log and records whether the event was a success or failure.

To verify digitally signed application-constructed data stored within a memory buffer or a file, a user must initiate a DBsign session and then make a call to the DBS_AppVerify plug-in/control method.

FCO_NRO.1: Selective proof of origin

The TOE provides the capability to generate evidence of origin for transmitted application-constructed data (stored within memory buffers or files) or stored database records at the request of the originator through the use of digital signature. When a user digitally signs data, the certificate associated with the user and the digital signature is applied to the data.

The TOE also provides the capability to verify the evidence of origin of information that was generated.

FIA_CID.2: Certificate identification before any action

To support the non-repudiation capabilities of the TOE, the TOE requires each originator to present a certificate before allowing any other TSF-mediated actions on behalf of that originator.

The originator's private key and the certificate provide the support that's needed for nonrepudiation. The originator does not have to provide the certificate when verifying.

6.2 Non-Cryptographic Probabilistic and Permutational Mechanisms

There are no non-cryptographic permutational or probabilistic mechanisms identified for the security functions of the TOE.

6.3 Assurance Measures

Table 5: TOE Assurance Measures

	ACM_CAP.2: Configuration items	ADO_DEL.1: Delivery procedures	ADO_IGS.1: Installation generation and start-up procedures	ADV_FSP.1: Informal functional specification	ADV_HLD.1: Descriptive high-level design	ADV_RCR.1: Informal correspondence demonstration	AGD_ADM.1: administrator guidance	AGD_USR.1: User guidance	ATE_COV.1: Evidence of coverage	ATE_FUN.1: Functional testing	ATE_IND.2: Independent testing – sample	AVA_SOF.1: Strength of TOE security function evaluation	AVA_VLA.1: Developer vulnerability analysis
Configuration Management for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0, Version 1.3	X												
DBsign Concepts Manual, Version 3.0, 8 September 2005							X						
DBsign for HTML Applications: Integration Guide, Version 3.0, 8 September 2005				X	X		X						
DBsign for HTML Applications Installation Manual, Version 3.0, 8 September 2005			X				X						
DBsign Administration Tools Manual, Version 3.0, 8 September 2005							X						
Delivery Procedures for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0, Version 0.2		X											
Functional Specification and Correspondence for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0, Version 0.6				X		X							

	ACM_CAP.2: Configuration items	ADO_DEL.1: Delivery procedures	ADO_IGS.1: Installation generation and start-up procedures	ADV_FSP.1: Informal functional specification	ADV_HLD.1: Descriptive high-level design	ADV_RCR.1: Informal correspondence demonstration	AGD_ADM.1: administrator guidance	AGD_USR.1: User guidance	ATE_COV.1: Evidence of coverage	ATE_FUN.1: Functional testing	ATE_IND.2: Independent testing – sample	AVA_SOF.1: Strength of TOE security function evaluation	AVA_VLA.1: Developer vulnerability analysis
High-Level Design for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0, Version 0.6					X								
Testing Procedures for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0, Version 1.5									X	X	X		
Vulnerability Analysis for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0, Version 0.6													X

7.0 PP Claims

There are no protection profile claims specified for this security target.

7.1 PP Reference

None

7.2 PP Tailoring

None

7.3 PP Additions

None

8.0 Rationale

8.1 Security Objectives Rationale

Table 6: Mapping of Objectives to Security Environment

	A.ADMIN	A.LOCATE	A.INSTALLER	A.USER_ID	T.KEY_COMPROMISE	T.MODIFY	T.NO_LOG	T.AUDIT_SEQUENCE	T.USER_DENY
O.AUDIT							X		
O.CRYPTO_OPERATION						X			X
O.INTEGRITY						X			
OE.AUDIT_REVIEW							X		
OE.CRYPTO_OPERATION					X				
OE.TIMESTAMP								X	
OE.ADMIN_GUIDANCE	X								
OE.CERTIFICATE_USERS				X					
OE.PHYSICAL_CONTROL		X							
OE.TOE_INSTALLATION			X						

8.1.1 Assumptions

Table 7: Justification for Assumptions Meeting Security Objectives

A.ADMIN: One or more authorized administrators should be assigned who are competent to manage the TOE, the IT environment supporting the TOE, the security of the information the TOE contains, and who can be trusted not to deliberately abuse their privileges so as to undermine security. These administrators must read and be familiar with all TOE related documents. These administrators should know which features are part of the TOE and which ones are not. These administrators should use the admin guidance to keep the TOE in compliance with evaluated configuration at all times.	**OE.ADMIN_GUIDANCE** is suitable to cover this assumption by providing adequate administrator guidance documentation.
A.LOCATE: The processing resources of the TOE must be located within controlled access facilities that will restrict unauthorized physical access. This is necessary in order to keep TOE in compliance with evaluated configuration.	**OE.PHYSICAL_CONTROL** is suitable to cover this assumption by ensuring that information is physically protected.
A.INSTALLER: The installer of the TOE is provided by Gradkell and has sufficient expertise and knowledge to properly install the TOE within its evaluated configuration. The Installer is aware of the TOE delivery procedures and follows it each time. Furthermore, the Installer follows the guidance provided for the TOE to make sure TOE is in compliance with evaluated configuration.	**OE.TOE_INSTALLATION** is suitable to cover this assumption by ensuring that the TOE is adequately installed in accordance with its evaluated configuration.
A.USER_ID: It is assumed that the certificate user or certificate user's certificate authority has correctly associated the certificate user's user identity and certificate issuer with their certificate.	**OE.CERTIFICATE_USERS** is suitable to cover this assumption by ensuring that users or their certificate authority supply a user id that accurately identifies the certificate users.

8.1.2 Threats

Table 8: Justification for Threats Countered By Security Objectives

T.AUDIT_SEQUENCE: An administ rator is unable to distinguish the sequence of audit events and therefore cannot detect recent integrity violations.	**OE.TIMESTAMP** is suitable to counter this threat by providing a reliable time stamp so that an accurate time may be associated to audit events generated by the TOE.
T.KEY_COMPROMISE: A user utilizes a non-FIPS 140 conformant cryptographic mechanism for generating a cryptographic key to be used with DBsign and the cryptographic key is compromised by an attacker.	**OE.CRYPTO_OPERATION** is suitable to counter this threat by providing a FIPS 140 conformant cryptographic mechanism for generating and destroying keys to be used with the TOE.

T.MODIFY: The integrity of data stored, processed, or transmitted may be compromised due to the unauthorized modification or destruction of the data or stored digital signatures by an attacker.	**O.INTEGRITY** is suitable to counter this threat by detecting a loss of integrity of digitally signed database records. **O.CRYPTO_OPERATION** is suitable to counter this threat by defining cryptographic key operations necessary to digitally sign data so that the integrity of data may be verified.
T.NO_LOG: Integrity violations and digital signature verification failures may take place and not get recorded.	**O.AUDIT** is suitable to counter this threat by generating auditable events for any security-relevant events pertaining to the TOE. **OE.AUDIT_REVIEW** is suitable to counter this threat by providing a means for the administrator to view audit data generated by the TOE.
T.USER_DENY: A user denies modifying a database record that is digitally signed by that user.	**O.CRYPTO_OPERATION** is suitable to counter this threat by defining cryptographic key operations necessary to verify a digital signature.

8.2 Security Requirements Rationale

8.2.1 Security Functional Requirements Coverage

8.2.1.1 Security Functional Requirements for the TOE

Table 9: Mapping of TOE SFRs to TOE Security Objectives

	O.AUDIT	O.CRYPTO_OPERATION	O.INTEGRITY
FAU_REC.1: DBsign audit record generation	X		
FCO_NRO.1: Selective proof of origin		X	
FCS_COP.1: Cryptographic operation		X	X
FIA_CID.2: Certificate identification before any action	X	X	

Table 10: Justification for Security Objectives to be met by the TOE SFRs

O.AUDIT: The TOE will provide the means of generating any security relevant events, so as to assist an administrator in the detection of potential attacks or misconfiguration of the TOE security features.	**FAU_REC.1** is suitable to meet this objective by requiring audit events to be generated for security-relevant events related digital signature generation and verification. **FIA_CID.2** is suitable to meet this objective by requiring audit events to include the signer certificate for signature generations and the verifier certificate for signature verifications.
O.CRYPTO_OPERATION: The TOE shall provide cryptographic operations necessary for digitally signing data and verifying the digital signature applied to data.	**FCO_NRO.1** is suitable to meet this objective by requiring cryptographic operations necessary to support non-repudiation of a signed database record. **FIA_CID.2** is suitable to meet this objective by requiring a certificate identity to be provided before signing data or verifying signed data. **FCS_COP.1** is suitable to meet this objective by requiring DBsign to provide cryptographic operations necessary for digitally signing data and verifying digitally signed data that is stored within the database, memory buffer, or file.
O.INTEGRITY: The TOE will provide the means to verify the integrity of data that has been digitally signed by the TOE.	**FCS_COP.1** is suitable to meet this objective by requiring DBsign to provide cryptographic operations necessary to verify the integrity of digitally signed data that is stored within a database, memory buffer, or file.

8.2.1.2 Security Functional Requirements for the TOE Environment

Table 11: Mapping of Environmental Requirements to Security Objectives for the TOE Environment

	OE.AUDIT_REVIEW	OE.CRYPTO_OPERATION	OE.TIMESTAMP	OE.ADMIN_GUIDANCE	OE_CERTIFICATE_USERS	OE.PHYSICAL_CONTROL	OE.TOE_INSTALLATION
AGD_ADM.1: administrator guidance				**X**	**X**		
ADO_IGS.1: Installation generation and start-up procedures						**X**	**X**
FAU_SAR.1: Audit review	**X**						
FAU_SAR.3: Selectable audit review	**X**						
FMT_MSA.2: Secure security attributes		**X**					
FCS_CKM.1: Cryptographic key generation		**X**					
The IT Environment shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [FIPS **140-1 approved algorithms (rDSA, ECDSA, or DSS) or FIPS 140-2 approved algorithms (DSA, ECDSA, or RSA)]** and specified cryptographic key sizes [(512 **or 1024 bit for DSS & DSA), (80, 112, 128, 192, or 256 bit for ECDSA), or (multiples of 256 bit for RSA)]** that meet the following: [ANSI **X9.31-1998 (rDSA), ANSI X9.62-1998 (ECDSA), FIPS 186-2 (DSS & DSA), or PKCS #1 v2.1 (RSA)].** FCS_CKM.4: Cryptographic key destruction		**X**						
FPT_STM.1: Reliable time stamps			**X**				

Table 12: Justification for Security Objectives to be met by the SFRs of the TOE Environment

SECURITY OBJECTIVES FOR THE IT ENVIRONMENT	
OE.AUDIT_REVIEW: Provide a means to review, search, and sort audit records generated by the TOE.	**FAU_SAR.1** is suitable to meet this objective by requiring the TOE environment to provide the administrator with the capability to access and interpret the audit records. **FAU_SAR.3** is suitable to meet this objective by requiring the TOE environment to provide the capability for the administrator to sort audit data.
OE.CRYPTO_OPERATION: Provide cryptographic key operations necessary to enable a user to utilize their digital signature.	**FCS_CKM.1** is suitable to meet this objective by requiring the TOE environment to ensure that a user can perform cryptographic key generation to support the use of a digital signature. **FCS_CKM.4** is suitable to meet this objective by requiring the TOE environment to ensure that a user can perform cryptographic key destruction to support the proper destruction of a digital signature. **FMT_MSA.2** is suitable to meet this objective by requiring the TOE environment to ensure that only secure values are accepted for security attributes to support proper generation of digital signatures.
OE.TIMESTAMP: Provide a time stamping mechanism that can be relied upon to provide an accurate date and time.	**FPT_STM.1** is suitable to meet this objective by requiring the TOE environment to provide a reliable time stamping mechanism.
SECURITY OBJECTIVES FOR THE NON-IT ENVIRONMENT	
OE.ADMIN_GUIDANCE: Deter administrator errors by providing adequate administrator guidance.	**AGD_ADM.1** is suitable to meet this objective by ensuring the administrator guidance describes: • The administrative functions and interfaces available to the administrator of the TOE, • How to administer the TOE in a secure manner, • Warnings about functions and privileges that should be controlled in a secure processing environment, • All assumptions regarding user behavior that are relevant to the secure operation of the TOE, • All security parameters under the control of the administrator, • Each type of security-relevant event relative to the administrative functions that need to be performed, and • All IT security requirements for the IT environment of the TOE that are relevant to the administrator.
OE.CERTIFICATE_US ERS	**AGD_ADM.1** is suitable to meet this objective by ensuring the administrator guidance provides a warning that users of the TOE are relied upon to associate an accurate user identity with their certificate.

OE.PHYSICAL_CONTROL: TOE data shall be physically protected to prevent unauthorized disclosure, destruction, or modification.	**ADO_IGS.1** is suitable to meet this objective by ensuring the installation or administrator guidance: • Provides procedures necessary for the secure installation, generation and s tart-up of the TOE, and • Describes the steps necessary for secure installation, generation, and start-up of the TOE.
OE.TOE_INSTALLATION The TOE shall be properly installed by a competent individual in accordance with its evaluated configuration.	**ADO_IGS.1** is suitable to meet this objective by ensuring the installation or administrator guidance: • Provides procedures necessary for the secure installation, generation and start-up of the TOE, and • Describes the steps necessary for secure installation, generation, and start-up of the TOE.

8.2.1.3 Justification of Explicitly Stated SFRs

FAU_REC.1	FAU_REC.1 was explicitly stated in this ST because the functionality is not intended to meet FAU_GEN.1, in that it does not audit the startup and shutdown of the audit mechanism. In addition, the TOE does not perform the actual recording of audit data which is required within FAU_GEN.1.2. However FAU_REC.1 does satisfy the rest of the requirement's functionalities as defined within the CC context. Therefore, an explicit require ment was stated to provide appropriate definition to the intended functionality for audit data generation.
FIA_CID.2:	FIA_CID.2 was explicitly stated in this ST because the functionality is not intended to meet FIA_UID.2, in that it presents a certificate, rather than verifying the identity of a user. Therefore, an explicit requirement was stated to provide appropriate definition to the intended functionality for presentation of a certificate.

8.2.1.4 Security Functional Requirements Dependencies

The following table identifies the dependencies on the security functional requirements for the TOE and the TOE environment.

Table 13: Security Functional Requirements Dependencies

Requirements:	Dependencies:	Satisfied:
FAU_REC.1: DBsign audit record generation	FPT_STM.1	*Yes*
FAU_SAR.1: Audit review	FAU_GEN.1[6]	*Yes*
FAU_SAR.3: Selectable audit review	FAU_GEN.1[7]	*Yes*
FCO_NRO.1: Selective proof of origin	FIA_UID.1[8]	*Yes*
FCS_CKM.1: Cryptographic key generation	FCS_CKM.4, FCS_COP.1, FMT_MSA.2	*Yes*
The **IT Environment** shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**FIPS 140-1 approved algorithms (rDSA, ECDSA, or DSS) or FIPS 140-2 approved algorithms (DSA, ECDSA, or RSA)**] and specified cryptographic key sizes [**(512 or 1024 bit for DSS & DSA), (80, 112, 128, 192, or 256 bit for ECDSA), or (multiples of 256 bit for RSA)**] that meet the following: [**ANSI X9.31-1998 (rDSA), ANSI X9.62-1998 (ECDSA), FIPS 186-2 (DSS & DSA), or PKCS #1 v2.1 (RSA)**]. FCS_CKM.4: Cryptographic key destruction	FCS_CKM.1, FMT_MSA.2	*Yes*
FCS_COP.1: Cryptographic operation	FCS_CKM.1, FCS_CKM.4, FMT_MSA.2	*Yes*
FIA_CID.2: Certificate identification before any action	None	*Yes*
FPT_STM.1: Reliable time stamps	None	*Yes*

[6] This dependency is implicitly satisfied by FAU_REC.1.

[7] This dependency is implicitly satisfied by FAU_REC.1.

[8] This dependency is implicitly satisfied by FIA_CID.2.

8.2.1.5 Justification of Unsatisfied Dependencies

FAU_GEN.1	This security functional requirement is a dependency for FAU_SAR.1 and FAU_SAR.3 to support the generation of audit events to be reviewed. FAU_GEN.1 is implicitly included in this ST through the explicitly stated requirement, FAU_REC.1. Therefore the dependency for FAU_GEN.1 is implicitly satisfied.
FIA_UID.1	This security functional requirement is a dependency for FCO_NRO.1 to provide a user identity in which the TOE supports non-repudiation of. FIA_UID.2 is implicitly included in this ST through the explicitly stated requirement, FIA_CID.2. FIA_UID.2 is hierarchal to FIA_UID.1. Therefore the dependency for FIA_UID.1 is implicitly satisfied.

8.2.1.6 Internal Consistency of SFRs

The IT security requirements defined for the TOE are stated in a manner in which they do not conflict with each other. Therefore, no justification is needed for conflicting IT security requirements.

8.2.2 EAL Justification

Gradkell has chosen to pursue a Common Criteria evaluation because of the government customer requirements that are mandated by NSTISS Policy 11. This policy requires a Common Criteria certification for all products to be used within systems used for entering, processing, storing, displaying, or transmitting national security information.

Gradkell has specifically chosen an EAL2 evaluation assurance level to meet the requirements mandated by the DoD and Air Force divisions of the government in accordance with the USDoD NSTISSP #11 Interpretation and the USAF CIO Memorandum.

8.2.3 Validation of Strength-Of-Function Claims

The TOE does not provide any non-cryptographic probabilistic or permutational mechanisms. Therefore, no strength of function claim is specified for this security target.

8.3 TOE Summary Specification Rationale

8.3.1 Security Functions Meet SFRs

Table 14: Mapping of TOE SFRs to TOE Security Functions

	Auditing	Digital Signature
FAU_REC.1: DBsign audit record generation	X	
FCO_NRO.1: Selective proof of origin		X
FCS_COP.1: Cryptographic operation		X
FIA_CID.2: Certificate identification before any action		X

Table 15: Rationale for Security Functions Satisfying SFRs

Security Functions	SFRs	Rationale
Auditing	FAU_REC.1	The TOE implements audit data generation for digital signature generation and verification events related to operations performed by DBsign. These events include digital signature generation and verification events, and data integrity verification events.
Digital Signature	FCO_NRO.1, FCS_COP.1, FIA_CID.2	The TOE implements the ability to digitally sign data and verify the validity of digitally signed data.

8.3.2 Assurance Measures Meet Assurance Requirements

Table 16: Rationale for Assurance Measures Satisfying SARs

Assurance Requirements	Assurance Measures	Rationale
ACM_CAP.2.1D-NIAP-0412	Configuration Management for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0, Version 1.3	The configuration items that comprise the TOE are specified in the document listed here.
ADO_DEL.1.1D	Delivery Procedures for DBsign for Client/Server Applications version 3.0, DBsign for Client/Server Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0, Version 0.2	Procedures defining the delivery method of the TOE to the consumer are provided in the document listed here.
ADO_IGS.1.1D	DBsign for HTML Applications Installation Manual, Version 3.0, 8 September 2005	The steps necessary for secure installation, generation, and start-up of the TOE are described within the documents listed here.
ADV_FSP.1.1D	Functional Specification and Correspondence for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0, Sections 2 and 3, Version 0.6, DBsign for HTML Applications: Integration Guide, Version 3.0, 8 September 2005	The functional specification describes the TSF and the external interface to the TOE. The functional specification is listed here along with other corresponding documents that provide additional details to the TOE's interfaces.
ADV_HLD.1.1D	High-Level Design for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0, Version 0.6, DBsign for HTML Applications: Integration Guide, Version 3.0, 8 September 2005	The high-level design describes the TOE subsystems and their interfaces. The High-Level Design is listed here.
ADV_RCR.1.1D	Functional Specification and Correspondence for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0, Version 0.6, Section 4	The correspondence representation is a demonstration of mappings between all adjacent pairs of available TSF representations, from the TOE summary specification through to the least abstract TSF representation that is provided. The correspondence representation is provided within the functional specification as it is listed here.

Assurance Requirements	Assurance Measures	Rationale
AGD_ADM.1.1D	DBsign Concepts Manual, Version 3.0, 8 September 2005, DBsign for HTML Applications: Integration Guide, Version 3.0, 8 September 2005, DBsign Administration Tools Manual, Version 3.0	Administrative guidance provides the TOE administrators with detailed, accurate information of how to administer the TOE in a secure manner. Documents listed here satisfy these requirements.
AGD_USR.1.1D	NOT APPLICABLE	Since the TOE does not perform "I&A" functions, and therefore does not make distinctions between administrators and non-administrative users, AGD_USR.1 does not apply and the requirement is vacuously satisfied.
ATE_COV.1.1D	Testing Procedures for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0, Version 1.5	Testing coverage shows the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification. The testing coverage is provided within the testing procedures document as it is listed here.
ATE_FUN.1.1D	Testing Procedures for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0, Version 1.5	Functional testing of the TOE involves providing a test plan, test procedure descriptions, expected test results and actual test results.
ATE_IND.2.1D	Testing Procedures for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0, Version 1.5	Independent testing requires Gradkell to provide the TOE suitable for testing and Gradkell has fulfilled this requirement.
AVA_SOF.1.1D	NOT APPLICABLE	Strength of function analysis requires the developer to provide an analysis of the strength of function claimed in this ST. However, no strength of function claim has been made and is therefore, not applicable.
AVA_VLA.1.1D	Vulnerability Analysis for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0, Version 0.3	A vulnerability analysis of the TOE involves describing the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP as to ensure that all obvious vulnerabilities have been addressed. The document listed here satisfies these requirements.

8.4 PP Claims Rationale

There is no protection profile claim specified for this security target.

9.0 Annex A

Annex A provides a list of acronyms, terms, and references used throughout this document.

9.1 Acronyms

API	Application Programming Interface
CC	Common Criteria
CEM	Common Evaluation Methodology
DB	Database
DBSAPI	DBsign API
DCA	DBsign Crypto Adaptor
DLL	Dynamically Linking Library
FIPS	Federal Information Processing Standard
GUI	Graphical User Interface
HTML	Hyper-Text Machine Language
HTTPS	Secure Hyper-Text Transfer Protocol
IT	Information Technology
JDBC	Java DataBase Connection
JRE	Java Runtime Environment
LDAP	Lightweight Directory Access Protocol
PKI	Public Key Infrastructure
PP	Protection Profile
QM	Query Module
RDBMS	Relational Database Management Systems
SF	Security Function
SFR	Security Functional Requirement
ST	Security Target
TOE	Target Of Evaluation
TSF	TOE Security Function
TSP	TOE Security Policy
VM	Virtual Machine

9.2 Terms

Security Function	A part or parts of the TOE that have to be relied upon for enforcing a closely related subset of the rules from the TSP.
Security Functional Requirement	A statement of security functionality that is to be required by a product claiming to meet the stated requirement.
TOE Security Policy	A set of rules that regulate how assets are managed, protected and distributed within a TOE.

9.3 Interpretations

9.3.1 International Interpretations

No international (CCIMB) interpretations are included within this ST

9.3.2 National Interpretations

The following national (NIAP) interpretations are included within this ST:

#	Interp Name	Affected Requirements	Description
0407	Empty Selections Or Assignments	FDP_ACF.1.3 FDP_ACF.1.4	CC v2.1 is ambiguous as to whether assignments could be completed by selecting none, i.e., providing no list. Similarly, it is unclear whether "none" is available as a selection. In some cases, "none" is given as an option in the Annex, but not indicated in the normative portion of Part 2.
0410	Auditing Of Subject Identity For Unsuccessful Logins	FAU_REC.1.2a	Both the FIA_UAU and FIA_UID components call for auditing of unsuccessful logins. However, if the login is unsuccessful, there is no subject identity to put in the audit record (as there is no subject in place). This is an inconsistency. In a similar fashion, FAU_REC.2.1 cannot be satisfied in the face of an invalid login, for there is no identity of the user that caused the event.

#	Interp Name	Affected Requirements	Description
0422	Clarification Of “Audit Records”	FAU_STG.1.2	There is a confusion introduced with the Part 2 usage of the term “Audit Records”, as opposed to the term “Audit Trail”. The Part 2 Annex, Section C.6, clarifies by implication that the term “Audit Records” refers to the records in the audit trail, as the application notes refer almost exclusively to the “audit trail” or the records in the trail. The problem with the use of the term “audit records” is that audit records may appear outside the audit trail, for example, after they have been retrieved through a selection.

9.4 Document References

Title	Version	Date	Author
DBsign for HTML Applications Integration Guide	3.0	2005-09-13	Gradkell Systems, Inc.
DBsign for HTML Applications Installation Manual	3.0	2005-09-13	Gradkell Systems, Inc.
DBsign Concepts Manual	3.0	2005-09-13	Gradkell Systems, Inc.
DBsign Administration Tools Manual	3.0	2005-09-13	Gradkell Systems, Inc.
Functional Specification and Correspondence for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0	0.6	2005-09-13	Veridyn, Inc., Gradkell Systems, Inc.
Testing Procedures for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0	1.5	2005-09-13	Gradkell Systems, Inc.
Configuration Management for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0	1.3	2005-09-13	Gradkell Systems, Inc.
Delivery Procedures for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0	0.2	2005-09-13	Veridyn, Inc.
Vulnerability Analysis for DBsign for Client/Server Applications version 3.0, DBsign for HTML Applications version 3.0, and DBsign for Oracle Web Forms Applications version 3.0	0.3	2005-09-13	Veridyn, Inc., Gradkell Systems, Inc.