Australian Government
Department of Defence

Australasian Information Security Evaluation Program

Certification Report

Certificate Number: 2004/33

August 2004

Version 1.2

Amendment Record

Version	Date	Description
1.2	20/8/2004	Public release.

Executive Summary

1 This report describes the findings of the evaluation of Rainbow Technologies (now SafeNet) iKey 2032 Universal Serial Bus token product, to the Common Criteria (CC) Evaluation Assurance Level EAL2. The product is now available from SafeNet. The iKey 2032 is the Target of Evaluation (TOE). The TOE has met the target Assurance Level of CC EAL2. The evaluation was performed by CSC Australia and completed on 12 December 2003.

2 The iKey 2032 is a product that is designed to act as a portable Public Key Infrastructure (PKI) authentication token that stores digital identification certificates in memory. The iKey 2032 token facilitates systems that allow users to interact securely on the Internet.

3 Ultimately, it is the responsibility of the user to ensure that the iKey 2032 meets their requirements. For this reason, it is recommended that a prospective user of the product refer to the Security Target at Attachment A and read this certification report thoroughly prior to deciding whether to purchase the product.

Table of Contents

Chapter 1 - Introduction

1.1 Overview

4 This chapter contains information about the purpose of this document and the identification of the Target of Evaluation (TOE), the iKey 2032.

1.2 Purpose

5 The purpose of this Certification Report is to:

a) report the certification of results of the IT security evaluation of the TOE against the requirements of the Common Criteria (CC) Evaluation Assurance Level EAL2, and

b) provide a source of detailed security information about the TOE.

6 The report should be read in conjunction with Attachment A, the TOE's Security Target, which provides a full description of the security requirements and specifications that were used as the basis of the evaluation.

1.3 Identification

7 Table 1 provides identification details for the evaluation. For details of all components included in the evaluated configuration refer to Section 2.6.1 Evaluated Configuration.

Item	Identifier
Evaluation Scheme	Australasian Information Security Evaluation Program
TOE	iKey 2032 (marked '2032-5')
Software Version	The end user applications must comply with PKCS#11 or MS CAPI protocols. The recommended host software is iKey 2032 Software Development Kit (SDK) v4.6.10 and iKey 2032 Cryptographic Interface Provider (CIP) v4.6.10.
Security Target	Security Target for iKey 2032, V8.0, May 2004.
Protection Profile Claims	N/A
Evaluation Level	CC EAL2
Conformance Result	CC Part 2 Conformant CC Part 3 Conformant
Evaluation Technical Report	iKey 2032 Evaluation Technical Report V3.0, June 2004.
Version of CC	CC Version 2.1, August 1999, Incorporated with Interpretations as of 14 May 2001.
CEM Version	CEM-99/045 Version 1.0, August 1999, Incorporated with Interpretations as of 14 May 2001.
Sponsor	Rainbow Technologies (now SafeNet)
Developer	Rainbow Technologies (now SafeNet)
Evaluation Facility	CSC Australia

Table 1 – Identification Information

Chapter 2 - Target of Evaluation

2.1 Overview

8 This chapter contains information about the TOE. More specifically, it includes a description of functionality provided, a summary of the TOE architecture, information on the scope of the evaluation, security policies and the use of the TOE.

2.2 Description of the TOE

9 The TOE is the iKey 2032 Universal Serial Bus (USB) Smart Token and its primary role is to act as a portable Public Key Infrastructure (PKI) hardware token that stores digital identification certificates in memory.

10 The iKey 2032 is a PKI authentication token capable of storing multiple digital IDs that can be used on any USB equipped workstation.

11 A digital ID is a set of electronic credentials that uniquely identify an individual. There are two parts to a digital ID:

a) A private key: this is the piece of information unique to the user within a PKI. Anyone who has access to the private key can impersonate the user.

b) A certificate: this is the public part of the user's digital ID. It binds the user's name and other identifying information with the user's public key, which is mathematically related to the private key.

12 The iKey 2032 is able to generate the private/public key pairs on the token, and also allows the import of an existing digital ID onto the token.

13 For further information on the specific hardware and software components included in the evaluated configuration refer to Section 2.6.1 Evaluated Configuration.

2.3 TOE Architecture

14 The iKey 2032 is a PKI authentication token that can be used on any USB equipped workstation and does not require a dedicated reader. The iKey 2032 consists of:

a) **USB Controller** – The USB controller provides the external interface for the TOE.

b) **Cryptographic Controller** – The cryptographic controller provides the cryptographic, data storage and operational functionality of the TOE.

c) **Epoxy** - This provides the physical protection of a tamper resistant layer over the cryptographic controller.

2.4 Clarification of Scope

15 The scope of the evaluation was limited to those claims made in the Security Target, provided at Attachment A.

2.4.1 Evaluated Functionality.

16 The TOE provides the following evaluated security functionality:

a) **Pass Phrase** – This function authenticates the user. Once authenticated the user can initiate protected iKey 2032 token operations. The Pass Phrase function intermediates in every requested encryption and signing operation.

b) **Lockout** – Locks out the iKey 2032 after 10 consecutive incorrect login attempts. Once locked out the iKey 2032 must be re-initialised. The Initialise function is described at sub-paragraph e) below.

c) **Private Key Protection** – Private keys are stored on the iKey 2032 token. These private keys are used for on-token cryptographic operations. The cryptographic use of an individual's private key(s) stored on the token is protected by their pass phrase. The token does not release private keys for external cryptographic operations.

d) **Cryptographic Algorithms** – Provides an array of cryptographic operations. The evaluated cryptographic functions are listed below.

i) Diffie-Hellman key agreement with a modulus 1024 bits in length.

ii) Digital signatures using DSA and RSA with a modulus 1024 bits in length.

iii) Digital signature verification using DSA and RSA with a modulus 1024 bits in length.

iv) Encryption and decryption using RSA with a modulus 1024 bits in length.

v) Encryption and decryption using Triple-DES with a key length of 112 or 168 bits.

vi) Encryption and decryption using DES with a key length of 56 bits.

vii) Hashing with SHA-1 and MD5.

viii) Key generation.

ix) Key destruction.

e) **Initialise** – Initialisation leaves the serial number and token label intact. All other information is removed from the token including pass phrase, key pairs, certificates, and data objects stored on the token. The token can be initialised by anyone running the appropriate utility software.

f) **Test** – Performs tests on cryptographic functions. The TOE conducts a Power On Self Test after reset and before executing the first command.

g) **Object** – Ability to view public/private objects on the iKey 2032. Authentication is required to view private objects. Only information about the private objects is available to be viewed and not information such as the value of the private key.

h) **Epoxy** – Provides resistance to tampering to the cryptographic controller part of the TOE.

i) **Electro** – The iKey 2032 will continue to function after exposure to 6kV electrostatic discharge.

j) **RAM** – The information in RAM is automatically erased when the token is removed from the USB port.

k) **Token Management** – Manage token information. The authenticated end user can configure various token settings including the pass phrase. Digital IDs can be imported onto the iKey 2032.

2.4.2 Non-evaluated Functionality

17 Potential users of the TOE are advised that a set of functions and services have not been evaluated as part of the evaluation. Potential users of the TOE should carefully consider their requirements for using functions and services outside of the evaluated configuration. The functions and services that have not been included as part of the evaluation are provided below.

a) **Token Authentication Timeout** – The TOE does not enforce a token authentication timeout function. The TOE can store timeout information and thus facilitate a token authentication timeout function enforced by host software. Token authentication timeout functionality is thus outside the scope of the TOE and has not been evaluated.

b) **User Payload Encryption** – In typical usage the TOE is not used to directly perform user payload symmetric encryption and decryption. For example, email content encryption performed by typical email applications will use the intended recipient's public key stored on the host computer for the encryption and will perform the encryption using the host computer CPU. In summary, the actual encryption is performed by the user's email application and is outside the scope of this evaluation.

c) **Unevaluated Cryptography** – The DSA, RSA and Diffie-Hellman algorithms utilising a modulus other than 1024 bits in length were not evaluated. The MD2 hashing algorithm was not evaluated.

2.5 Security Policy

18 The TOE Security Policy (TSP) is a set of rules that defines how the information within the TOE is managed and protected. The TSP is defined in the Security Target at Attachment A. A summary of the TSP is provided below:

a) **Cryptographic Support** – The TOE will provide for cryptographic key generation, key distribution and key destruction. The TOE also will provide digital signing, digital signature verification, key agreement, encryption and decryption services.

b) **Data Protection** – The TOE will allow the use of private key information once the user has been authenticated. The actual private key is not allowed to leave the TOE. The authenticated user can export PKI certificate information. The authenticated user can also import digital IDs. After the customer initialises the token previously stored digital IDs are not available for use. The TOE does not allow user access to sensitive system data.

c) **Identification and Authentication** – The TOE will provide a pass phrase based authentication mechanism. The TOE will enforce a lockout function after 10 unsuccessful authentication attempts. Minimum password length allowed is four characters.

d) **Protection of TOE Security Functions** – The TOE will perform a functional self-test at power on. TOE RAM will not retain information when power is removed. The TOE will resist physical tampering.

2.6 Usage

2.6.1 Evaluated Configuration

19 This section describes the configurations of the TOE that were included within scope of the evaluation. The assurance gained via evaluation applies specifically to the TOE in these defined evaluated configuration(s).

20 The TOE hardware is a compact moulded plastic USB token with the words “iKey 2032 by Rainbow” on one side. The other side is labelled with the text “2032-5”.

21 The iKey 2032 token hardware encapsulates the physical and logical boundaries of the TOE.

22 The table below identifies the operating systems, utility software and USB drivers that were used for testing the Rainbow iKey 2032 during the course of the evaluation. The evaluated security functions of the TOE did not depend on the host operating system version and driver.

Operating System	USB Driver Name and Version
Microsoft Windows 98 with high encryption (128-bit Internet Explorer)	iKey 2000 v4.6.10 (driver v3.1)
Microsoft Windows 2000 Professional SP 2	iKey 2000 v4.6.10 (driver v3.1)
Microsoft Windows NT 4.0 SP 6a	iKey 2000 v4.6.10 (driver v3.1)

2.6.2 Determining the Evaluated Configuration

23 The end-user can identify that they have the correct TOE by visual inspection of the hardware case. The TOE hardware is a compact moulded plastic USB token with the words “iKey 2032 by Rainbow” on one side. The other side is labelled with the text “2032-5”.

2.6.3 Delivery procedures

24 When placing an order for the iKey 2032 customers should make it clear to their supplier that they wish to receive the evaluated product. They should then receive a shipment specific Common Criteria Shipment Document that informs them about what to expect in the shipment so that they can identify the supplied product and thus verify the integrity of the shipment.

2.6.4 Documentation

25 It is important that the iKey 2032 is used in accordance with the guidance documentation in order to ensure the secure usage of the TOE. The iKey 2000 Series User’s Guide (Ref [1]) is provided with the product and provides instructions on how to use the functions of the iKey 2032 using Entrust, Netscape and Microsoft applications. It also contains information on how to administer the TOE securely and the necessary steps for customer initialisation of the TOE. During the course of the evaluation the developers also released two technical notes clarifying or correcting the

iKey 2000 Series User's Guide (Ref [1]). These were TN2140 (Ref [2]) and TN2156 (Ref [3]). These technical notes should be read in conjunction with the iKey 2000 Series User's Guide (Ref [1]).

2.6.5 Secure Usage

26 The evaluation of the TOE took into account certain assumptions about its operational environment. These assumptions must be upheld in order to ensure the security objectives of the TOE are met.

a) Power and clock is supplied from the USB port.

b) The user of the TOE possesses the necessary privileges to access the information managed by the TOE.

c) For a user to access the functions and assets on the TOE, the host computer must be able to communicate correctly with the iKey 2032 token.

d) The host computer does not contain any unauthorised software that could intercept the user pass phrase or misuse the token functions once the user is authenticated.

e) When a given iKey 2032 token's security is breached then all relevant digital IDs should be revoked at the appropriate Certification Authority.

Chapter 3 - Evaluation

3.1 Overview

27 This chapter contains information about the procedures used in conducting the evaluation and the testing conducted as part of the evaluation.

3.2 Evaluation Procedures

28 The evaluation of the TOE was conducted using the Common Criteria for Information Technology Security Evaluation (Refs [4],[5],[6],[7]) under the procedures of the Australasian Information Security Evaluation Program (AISEP) (Refs [8],[9],[10],[11]). In addition, the conditions outlined in the Arrangement on the Recognition of Common Criteria Certificates in the field of Information Technology Security (Ref [12]) were also upheld during the evaluation and certification of the product.

3.3 Functional Testing

29 In this phase the evaluators analysed evidence of the developer's testing effort, including test coverage, test plans and procedures and expected and actual results, to gain confidence that the developer's testing was sufficient to ensure the correct operation of the TOE. In addition, the evaluators drew on this evidence to perform a sample of the developer tests in order to verify that the test results matched those recorded by the developers. The functional testing effort also included a selection of independent functional tests that expanded on the testing done by the developers.

3.4 Penetration Testing

30 The developer performed a vulnerability analysis of the iKey 2032, in order to identify any obvious vulnerability in the product and to show that the vulnerability is not exploitable in the intended environment of the TOE. This analysis included a search for possible vulnerability sources in the evaluation deliverables, the intended TOE environment, public domain sources and internal Rainbow Technologies (now SafeNet) sources. The developer identified a number of potential vulnerabilities, relevant to the product type and in each case the developers were able to show that the vulnerability was not exploitable in the TOE's intended operational environment.

31 Based on the information given in the developer's vulnerability analysis, the evaluators were able to devise a penetration test plan. After the completion of testing, the evaluators were able to determine that the TOE, in its intended environment, has no exploitable obvious vulnerabilities.

Chapter 4 - Certification

4.1 Overview

32 This chapter contains information about the result of the certification, an overview of the assurance provided by the Common Criteria level chosen and the certifier's recommendations.

4.2 Certification Result

33 After due consideration of the conduct of the evaluation, and of the Evaluation Technical Report (Ref [13]) the Australasian Certification Authority certifies the evaluation of the iKey 2032 performed by CSC Australia.

34 The evaluators found that the iKey 2032 upholds the claims made in the Security Target at Attachment A and has met the requirements of the Common Criteria EAL2 assurance level.

35 Evaluation is not a guarantee of freedom from security vulnerabilities; there is a small probability that exploitable vulnerabilities remain undiscovered.

4.3 Assurance Level Information

36 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 behaviour.

37 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).

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

Annex A. References and Abbreviations

A.1. References

[1] iKey 2000 Series User's Guide, Revision B, Part Number 700771-001, February 2002

[2] Rainbow Technical Note TN2140, Rev. A, iKey 2000 Login Attempt Counter is Preset at Factory, October 2002.

[3] Rainbow Technical Note TN2156, Rev. A, iKey 2000's use of the Inactivity Timer.

[4] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model (CC), Version 2.1, August 1999, CCIMB-99-031.

[5] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements (CC), Version 2.1, August 1999, CCIMB-99-032.

[6] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements (CC), Version 2.1, August 1999, CCIMB-99-033.

[7] Common Methodology for Information Technology Security Evaluation (CEM), Version 1.0, August 1999, CEM-99/045.

[8] AISEP Publication No. 1 – Description of the AISEP, AP 1, Version 2.0, February 2001, Defence Signals Directorate.

[9] AISEP Publication No. 2 – The Licensing of the AISEFs, AP 2, Version 2.1, February 2001, Defence Signals Directorate.

[10] Manual of Computer Security Evaluation Part I – Evaluation Procedures, EM 4, Issue 1.0, April 1995, Defence Signals Directorate. (EVALUATION-IN-CONFIDENCE)

[11] Manual of Computer Security Evaluation Part II – Evaluation Tools and Techniques, EM 5, Issue 1.0, April 1995, Defence Signals Directorate. (EVALUATION-IN-CONFIDENCE)

[12] Arrangement on the Recognition of Common Criteria Certificates in the field of Information Technology Security, May 2000.

[13] Rainbow iKey 2032 Evaluation Technical Report, Version 3.0, June 2004, CSC Australia. (EVALUATION-IN-CONFIDENCE)

A.2. Abbreviations

ACA	Australasian Certification Authority
AISEF	Australasian Information Security Evaluation Facility
AISEP	Australasian Information Security Evaluation Program
CC	Common Criteria
CEM	Common Evaluation Methodology
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
EAL	Evaluation Assurance Level
ETR	Evaluation Technical Report
MD2	Message Digest Algorithm #2
MD5	Message Digest Algorithm #5
MS CAPI	Microsoft Cryptographic Application Programming Interface
PKCS#11	Public-key Cryptography Standard #11
PKI	Public Key Infrastructure
RSA	Rivest, Shamir and Adleman
SDK	Software Development Kit
SFP	Security Function Policy
SFR	Security Functional Requirement
SHA-1	Secure Hash Algorithm (160 bit output)
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functions
TSP	TOE Security Policy
USB	Universal Serial Bus

Attachment A - Security Target

Security Target for the iKey 2032

for medium security applications

Version 8.0

May 2004

Prepared By: CSC Australia

Prepared For: Rainbow Technologies

Modification History

Version	Date	Description	Author
1.0	14/10/2000	Initial Security Target for acceptance into the AISEP.	Graeme Mahon
1.1	14/5/2001	Incorporating AISEP comments	Graeme Mahon
1.2	14/5/2001	Incorporating AISEP comments	Graeme Mahon
1.3	17/4/2002	Draft version for Rainbow review incorporating resolutions to EORs 001-004, 006-008.	Tony Hall / Graeme Mahon
2.0	17/4/2002	Release Version to DSD and CSC for Evaluation	Tony Hall / Julien Van Raalte
2.1	7/5/2002	Draft version for Rainbow review incorporating resolutions to EOR 005 and RFC 001.	Tony Hall / Graeme Mahon
3.0	13/5/2002	Release Version to DSD and CSC for Evaluation. This version resolved EOR 005 and RFC 001.	Tony Hall
3.1	August 2002	Addition of document number. Amend Physical Token Protection TSFs Addressing EORs 5, 9, 10, 11: EOR 5 – weak and semi-weak keys defined in 6.1.2 - TSF RAM expanded to list what may be in iKey memory - TSF Lockout mapped to FPT_FLS.1 (2) EOR 9 – Hardware encryption claims removed from TSF PKP - Electro TSF expanded to describe why ESD can affect security EOR 10 – Removed mapping between PKP and FPT_SEP - Fixed typos in section 8.3 EOR 11 – Removed SOF claim from group of TSFs in 6.1.4. Put explicit claims on Case and Electro - Removed rationale in section 8.3 relating to PKP	Aland Dent, Graeme Mahon
3.2	September 2002	Addressing Laszlo Elteto's comments: Updated A.HOST to have correct software versions, and to exclude malicious software from host. Updated FPT_AMT.1 and TEST to include POST Include Token Utility to TSF T_MGT Added P.PERMANENT, OE.PERMANENT, updated TE.INSECURE and associated rationales Addressing EOR 12 including key zeroisation in TSFs	Graeme Mahon
4.0	October 2002	Release for evaluation	Graeme Mahon
4.1	February 2003	1. Updated the document to reflect changes required for EOR 019 in sections 6.2.5 and 6.2.4.	Jenine McQuaid

		2. Changed reference from the Functional Spec to the Design Spec per request from Aleks at CSC. 3. Changed A.HOST to reflect only v4.6.10 software, not prior releases.	
4.2	April 2003	Updated assurance measures to reflect that Design Specification covers FSP, HLD and RCR (implied by EOR 019) EOR 21: Clarify logical scope and interfaces Power & Clock assumption and OE updated P.Crypt_Std references standards FCS class updated to reference standards EOR 22: Section 8.5.2 updated EOR 23: Environmental stress changed to electrical EOR 25: Removal of Inactivity Timer TSF and FIA_UAU.6.	Graeme Mahon
4.3	April 2003	Address comments	Graeme Mahon
5.0	April 2003	Inserting reference to TN 2156 and clarifying token usage when encrypting e-mails (EOR 024). Removed DESX algorithm.	Graeme Mahon
6.0	June 2003	Addressing EOR 19 (updating explicit assurance references in section 6.2 and table 8-9) and 21 (removing P.Crypt_Std)	Graeme Mahon
7.0	May 2004	Addressing EOR 31, 32, 33 and 34. Removed Strength of Function for Electro and Epoxy. Updated “2032-5”s and further consistency check in Table 6.1	Laszlo Elteto
8.0	May 2004	Address comments; new version.	Laurie Smith

Table of Contents

Terminology & Acronyms

CA	Certificate Authority
CC	Common Criteria
CIP	Cryptographic Interface Provider
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
DH	Diffie-Hellman Key Exchange algorithm
Digital ID	A set of electronic credentials that uniquely identify an individual
EAL	Evaluation Assurance Level
ES	Embedded Software (on the token)
MS CAPI	Microsoft Cryptographic Application Programming Interface
PCB	Printed Circuit Board
PKCS	Public Key Cryptography Standard
PKI	Public Key Infrastructure
RAM	Random Access Memory
RSA	Encryption algorithm created by Rivest, Shamir and Adleman
SFP	Security Function Policy
ST	Security Target
Token	The portable physical device that stores Digital ID
TOE	Target of Evaluation
USB	Universal Serial Bus
VPN	Virtual Private Network

Document Organisation

An acronym list is provided to define frequently used acronyms.

Section 1 provides the introductory material for the security target

Section 2 provides general purpose and TOE description

Section 3 provides a discussion of the expected environment for the TOE. This section also defines the set of threats that are to be addressed by either the technical countermeasures implemented in the TOE hardware or software or through the environmental controls.

Section 4 defines the security objectives for both the TOE and the TOE environment.

Section 5 contains the functional and assurance requirements derived from the Common Criteria, Part 2 and 3, respectively, that must be satisfied by the TOE.

Section 6 provides the functions and measures to meet the requirements specified in section 5.

Section 7 provides the Protection Profile claims of this document

Section 8 provides a rationale to explicitly demonstrate that the information technology security objectives satisfy the policies and threats. Arguments are provided for the coverage of each policy and threat. The section then explains how the set of requirements are complete relative to the objectives, and that each security objective is addressed by one or more component requirements. Arguments are provided for the coverage of each objective. Next Section 8 provides a set of arguments that address dependency analysis, strength of function issues, and the internal consistency and mutual supportiveness of the security target requirements

1 Introduction

This introductory section presents the Security Target (ST) identification and an overview of the ST structure.

1.1 Security Target and TOE Identification

Title: Security Target for the iKey 2032 for medium security applications

CC Version: 2.1 Final

TOE Identification: iKey 2032 with marking “2032-5” printed on the back of the token

NOTE: Will use the term “iKey” to refer to the Rainbow Technologies product family encompassing model 2032-5.

1.2 Security Target Overview

Rainbow Technologies’ iKey is a portable Public Key Infrastructure (PKI) authentication token that stores digital identification certificates in memory. iKey is able to generate the private/public key pair on the physical device, and also allows the import of an existing certificate onto the token. Once on the token, the private key can never leave the token. Access to the information stored on the iKey is protected with a pass phrase. Wherever smart cards and digital certificates are used, iKey can provide a plug and play PKI solution. The iKey is also portable and small enough to fit on a person’s key ring, allowing the token to be used remotely.

Unlike smart cards, the iKey does not require the installation of a dedicated reader. The iKey plugs into any standard Universal Serial Bus (USB) port, making it a smart card and reader encompassed in a USB form. The Universal Serial Bus is a low cost, plug and play, standardised interface included on virtually all personal computers built since early 1997. USB ports have moved to the front of workstations, and to the side of laptops in more recent models, making them convenient as well as ubiquitous. All future PCs are certain to have USB ports.

iKey can be integrated into systems to provide user authentication for multiple applications and network services. In addition iKey can be used in protection of Virtual Private Network (VPN) and control intranet, extranet, and Internet access. Also, iKey can be used for secure e-mail services by digitally signing messages for non-repudiation, decrypting messages with the private key stored on the token. This allows transactions or e-commerce conducted over the Internet to be handled securely.

This ST describes the IT security requirements to protect the information stored on the iKey. The security requirements in this ST cover the iKey's integrated circuit and operating software, but do not include specific applications to which the iKey interfaces to. This ST does not cover security requirements for terminals or networks interfacing with the iKey.

The security needs for the iKey can be summarised as being able to counter those who want to gain unauthorised access to data and information stored on an iKey token.

Therefore it is mandatory to:

- maintain the integrity and the confidentiality of the contents of the iKey token non-volatile memory (program and data memories), and

- maintain the integrity and the confidentiality of the security enforcing and security relevant architectural components (security mechanisms and associated functions) embedded into the iKey token.

1.3 CC Conformance Claim

The TOE is conformant with Parts 2 and 3 of the CC, version 2.1.

1.4 Assurance Claim

This Security Target claims an Evaluation Assurance Level (EAL) 2 as specified in the CC.

2 TOE Description

The Target of Evaluation (TOE) is an iKey 2032-5, a PKI authentication token capable of storing multiple Digital IDs that can be used on any Universal Serial Bus (USB) equipped workstation. The TOE is like a smart card that does not require a designated reader as most new computers will have a USB port. The TOE can be utilised wherever smart cards and digital certificates are used.

A Digital ID is a set of electronic credentials that uniquely identify an individual. There are two parts to a Digital ID:

- a private key
- a certificate.

The private key is the piece of information unique to the user within the Public Key Infrastructure. Anyone who has access to the private key can impersonate the user without detection. An impersonator can read eyes-only messages or sign documents as the user. It is important to keep the private key secure. iKey serves as an impenetrable safe for the private key, ensuring that only the intended user has access to it.

The certificate is the public part of the user's Digital ID. It contains the user's name and other identifying information. It also contains the public key, which is mathematically related to the private key. Using your certificate, other people out in the world can verify that you hold your private key, and therefore, must really be who you say you are.

In generating a Digital ID, the user generates a public and private key pair. This can be done directly on the iKey token and the private key is permanently stored there. It never leaves. The public key is sent off to a trusted third party, called a Certification Authority or CA.

1. The CA verifies that the public key sent really belongs to the user. If the verification succeeds, the CA creates a certificate for the user and sends instructions on how to obtain the certificate.

2. The user then downloads the certificate, completing the Digital ID.

The iKey also has the capability to import an existing Digital ID onto the token. Once the private key of the existing Digital ID is imported onto the token, it may never leave.

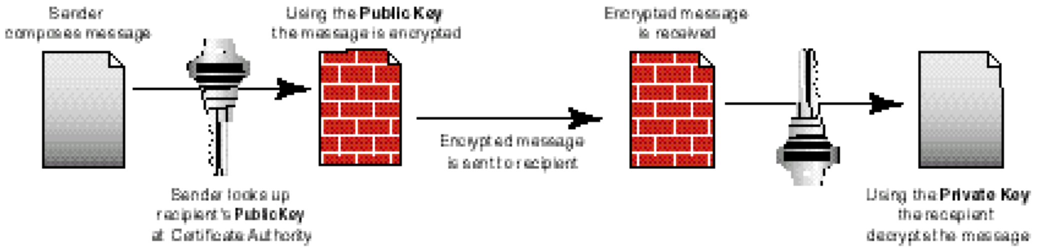
Using the iKey token enables users to send and receive secure e-mail and to interact securely on the World Wide Web. The iKey provides protection against many undesirable occurrences, such as data disclosure to unauthorised recipients, unauthorised content changes, message spoofing, and message repudiation. This protection comes from encryption and a digital signature. Encryption scrambles data so that only the intended recipients (who have the correct "key") may view it. A digital signature is an electronic mark attached to a message that creates a strong binding between the signer and the contents of the document. No unauthorised changes to a message can be made. A digital signature proves whom the author of the message was and the author can't deny sending the message. This provides non-repudiation for commercial transactions that are conducted over the Internet.

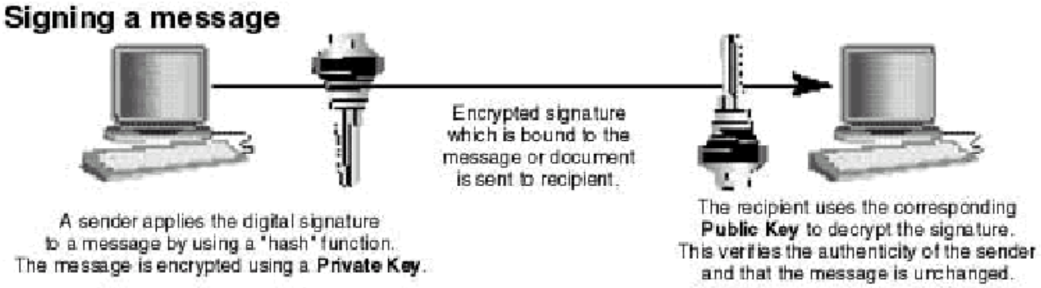
Access to the information stored on the TOE is pass phrase protected, meaning only the users who know the pass phrase are allowed to access the Digital ID.

A typical sequence of events for use of the TOE would be:

1. A travelling company employee needs access to company network, which is protected by a VPN. The employee gets access to a computer with a USB port and Internet connection.
2. The employee accesses the VPN and is prompted for his Digital ID.
3. The iKey token is inserted into the USB port and employee is prompted for pass phrase to iKey.
4. Employee enters pass phrase, allowing the VPN to authenticate with the Digital ID on the TOE.
5. The employee can receive an e-mail encrypted with his public key and can decrypt it with his private key stored on the TOE. Any outgoing e-mails can be encrypted with the recipient's public key (not on the token) and/or digitally signed with the private key (stored on token).
6. Any e-commerce transactions the employee has to conduct can be encrypted for confidentiality or digitally signed for integrity and non-repudiation of the transaction.

Sending an Encrypted Document or message


Sender composes message
Using the Public Key the message is encrypted
Encrypted message is received
Encrypted message is sent to recipient
Sender looks up recipient's Public Key at Certificate Authority
Using the Private Key the recepient decrypts the message


Signing a message
Encrypted signature which is bound to the message or document is sent to recipient.
A sender applies the digital signature to a message by using a "hash" function. The message is encrypted using a Private Key.
The recipient uses the corresponding Public Key to decrypt the signature. This verifies the authenticity of the sender and that the message is unchanged.

The TOE is composed of the iKey token, which contains a processing unit, security components, I/O port, LED indicating power or activity, volatile and non-volatile memories including the embedded software.

Physical scope and boundaries of TOE is the iKey token. The USB port that the TOE may be connected to is outside of scope.

Logical scope and boundaries of TOE is the cryptographic functions and protection that the iKey token provides.

The TOE can be interfaced with PKCS#11 Version 2.01 or MS CAPI enabled applications. The iKey is supplied with interfacing software, which is not included in the scope of the TOE. However, this software *may* be used in the configuration of this evaluation. The software is iKey 2032 Software Development Kit (SDK) v4.6.10 and iKey 2032 Cryptographic Interface Provider (CIP) v4.6.10.

The security features provided by the TOE are:

Authentication – A user of the TOE is identified and authenticated by their Pass Phrase. The TOE has a Lockout feature that disables the TOE when the incorrect Pass Phrase has been entered too many times.

Access Control – Access to information and operations of the TOE are controlled.

Management – The TOE is PKCS #11 compliant and provides interfaces for initialisation, testing and management of the TOE.

Cryptographic – The TOE performs cryptographic operations.

Physical Protection – The TOE has the physical protection of a tamper resistant epoxy layer and can withstand electrostatic discharges, as well as removing any information residual in RAM when power is lost.

3 TOE Security Environment

This section describes the security aspects of the environment in which the TOE is intended to be used and addresses the description of the assets to be protected, the assumptions, the threats and the organisational security policies.

3.1 Assets

The primary asset of the TOE is the private user information stored on the token that is to be protected.

The use of the iKey tokens require that they are in the user's possession, which can be a hostile environment. Therefore it is necessary to consider the protection of the TOE's characteristics which maintain the security of the primary asset. Therefore, this ST considers the secondary assets to be:

- The cryptographic keys which are used in the security processes of the TOE;
- The iKey's embedded software which access and manipulates the data stored on the token.

The assets are to be protected in terms of confidentiality and integrity.

3.2 Assumptions

Assumptions regarding the TOE can be divided into assumptions about the intended usage of the TOE, and assumptions about the environment of use of the TOE.

3.2.1 Assumptions for intended usage of the TOE

Usage of the TOE is to store Digital IDs securely and to perform authorised cryptographic functions relating to the Digital ID.

The token is used for PKI authentication, non-repudiation, confidentiality and integrity. The token can be used in PKI applications e.g. identify and authenticate users in network applications, or for non-repudiation of transactions payments over the Internet, or for confidentiality and integrity of messages with encryption and digital signatures.

For the information stored on the TOE to be accessed and managed by the authorised user, there needs to be client applications like Netscape and Internet Explorer which interface to the TOE via the USB port.

The TOE is intended to protect the confidentiality and integrity of the PKI information that it stores.

The token is assumed to be in the uncontrolled possession of the token holder. This environment is considered hostile and the token must therefore protect its assets against unauthorised alteration or disclosure that may be accomplished with standard personal computers and with laboratory equipment used without any supervision.

3.2.2 Assumptions about the environment of use of TOE

The following assumptions regarding the operation of the TOE are made:

A.PWR_CLOCK: Power and Clock
Power and clock to be supplied from the USB port.

A.USER User Privilege

User of the TOE is assumed to possess the necessary privileges to access the information managed by the TOE.

Only the user/owner of the TOE knows the pass phrase required to access the personal information stored on the iKey.

A.Host Host Computer Requirements

For a user to access the functions and assets on the TOE, the host computer must be able to communicate correctly with the iKey 2032-5 token. The host computer does not contain any malicious code (rogue or Trojan software) while the valid user is accessing the TOE.

The host computer must have a USB port and the appropriate USB driver, and end-user applications that comply with PKCS #11 or MS CAPI protocols. Examples of end-user applications include but are not limited to web browsers like Internet Explorer and Netscape Navigator. Recommended host software is iKey 2032 Software Development Kit (SDK) v4.6.10 and iKey 2032 Cryptographic Interface Provider (CIP) v4.6.10. Below is a list of version numbers of the USB driver required for each supported operating system:

Operating System	USB Driver Name and Version
Microsoft Windows 98 with high encryption (128-bit Internet Explorer)	iKey 2000 v4.6.10 (driver v3.1)
Microsoft Windows 2000 Professional SP 1 and SP 2	iKey 2000 v4.6.10 (driver v3.1)
Microsoft Windows NT 4.0 SP 6a	iKey 2000 v4.6.10 (driver v3.1)

3.3 Threats to Security

Threats may be addressed by either the TOE or its intended environment.

The assumed threats to be addressed by the TOE could be described in three types:

- unauthorised disclosure of assets,
- unauthorised use of assets,
- unauthorised modification of assets.

3.3.1 Unauthorised disclosure of assets

This type of threats covers unauthorised disclosure of assets by attackers who may possess a wide range of technical skills, resources and motivation. Such attackers may also have technical awareness of the product.

T.DISCLOSURE Disclosure of assets

Unauthorised disclosure of the protected information stored on the TOE i.e. private cryptographic keys.

3.3.2 Unauthorised use of assets

Potential attackers may gain access to the TOE and perform operations for which they are not allowed. For example, such attackers may personalise the product in an unauthorised manner, or try to gain fraudulently access to the protected information on the TOE.

T.IMPERSON Impersonation of authorised user

An attacker may gain unauthorised access to information or resources by impersonating an authorised user of the TOE.

3.3.3 Unauthorised modification of assets

The TOE may be subjected to different types of logical or physical attacks which may compromise security. Due to the intended usage of the TOE (the TOE environment may be hostile), the TOE security parts may be bypassed or compromised reducing the integrity of the TOE security mechanisms and disabling their ability to manage the TOE security. This type of threat includes the implementation of malicious Trojan horses, Trapdoors, downloading of viruses or unauthorised programs.

T.MODIFY Modify TOE

Unauthorised modification of TOE memory to compromise the confidentiality or integrity of the assets on the TOE. This may include the unauthorised loading of software onto the TOE.

T.PHYSICAL Physical Attack

Security-critical parts of the TOE may be subject to physical attack which may compromise security.

T.CORRUPT Corruption of token operations

Token may become corrupted making the token unreliable in its operations.

The assumed threats to be addressed by the TOE's intended environment are:

TE.INSECURE Insecure token usage

The token may be used in such a manner that undermines security i.e. the default pass phrase may not be changed or the token pass phrase may be disclosed to an untrusted person or assets may be permanently placed onto the token.

3.4 Organisational Security Policies

P.INITIALISE: Initialisation of TOE

Any organisational policies regarding pass phrase complexity and re-authentication timeouts that are within the TOE constraints must be implemented into the TOE at initialisation.

P.BREACH Token security breached

Any suspicion of the token security being breached should result in all relevant Digital Ids being revoked at the appropriate CA.

P.PERMANENT Permanent files on TOE

The TOE has the ability keep files during initialisation of the TOE eg. Serial Number and label. As this is a security risk, the TOE user should not create assets upon the TOE that will remain permanent through a TOE initialisation.

4 Security Objectives

4.1 Security Objectives for the TOE

Security Objectives for the TOE are prefixed by "O".

O.AUTHENTICATE Authentication of User

The token will authenticate before granting access to the token protected assets and operations.

O.TAMPER Tamper of Token

The token must prevent tampering that will result in protected assets being revealed. The token shall resist "brute force" type impersonation tampering.

O.OPERATE Operation of Token

The token must provide a means of verifying the correctness of its cryptographic operations.

O.DISCLOSURE Disclosure of Assets

The token shall ensure that protected assets stored in memories is protected against unauthorised disclosure.

O.MODIFY Modification of Assets

The token shall ensure that protected assets stored in memories is protected against unauthorised modification.

O.CRYPT: Cryptography

The token must support cryptographic functions in a secure manner. All token cryptographic operations are conducted upon the token, meaning the private key does not leave the token.

O.ELEC_STRESS: Electrical Stress

The TOE must protect itself against compromise by having a structure which does not reveal security information nor which operates in an insecure fashion when exposed to electrostatic discharge conditions and removal of the token from the USB port.

O.SECURE: Secure State

The TOE shall have the ability to be returned to a secure state without revealing protected assets.

4.2 Security Objectives for the Environment

Security Objectives for the Environment are prefixed by "OE".

OE.INITIALISE: Initialisation of Token

Those responsible for the token must ensure that it is initialised and operated in a manner which maintains IT security.

OE.AUTHDATA Authentication Data

Users of the token must not disclose their pass phrase that protects the assets on the token.

OE.BREACH Token security breached

Those responsible for the token should inform the CA to revoke Digital Ids that have been stored upon a token where it is suspected a security breach has occurred. The suspected token should be re-initialised.

OE.PWR_CLOCK External Power and Clock

The TOE is internally unpowered, so power and clock must be delivered to it from the USB port.

OE.HOST_ENV Host Computer Environment

For the user to access the functions and assets on the TOE, the user's host computer requires certain hardware and software to be able to communicate correctly with the TOE.

OE.PERMANENT Permanent files on TOE

The TOE has the ability to keep files during initialisation of the TOE eg. Serial Number and label. As this is a security risk, the TOE user should not create assets upon the TOE that will remain permanent through a TOE initialisation.

5 IT Security Requirements

5.1 TOE Security Functional Requirements

All the Security Functional Requirements (SFRs) in this section were drawn from CC Part 2 functional requirements components with the operations completed. All standard CC text is in regular font, whereas the text inserted for this ST is enclosed by square braces "[ ]" and provided in *italics*. Refinements to the SFRs are indicated by **bold characters** when new text added and when text is removed. Table 5-1 presents a summary of the SFRs used for this ST.

Table 5-1 Security Functional Requirements

Functional Class	Functional Components
FCS - Cryptographic	CKM.1, CKM.2, CKM.4, COP.1
FDP – Data Protection	ACC.1, ACF.1, ETC.1, ITC.1
FIA – Identification and Authentication	ATD.1, AFL.1, UAU.2, UAU.6
FMT – Security Management	MSA.1, MSA.2, MSA.3
FPT – Protection of TSF	AMT.1, FLS.1, PHP.3, SEP.1

5.1.1 Cryptographic support

Cryptographic key generation (FCS_CKM.1)

The TSF shall generate cryptographic keys in accordance with ~~a~~ specified cryptographic key generation algorithm**s** [*RSA, DSA, Diffie-Hellman*] and specified cryptographic key sizes [*512, 768, 1024 bits*] that meet the following: [*IEEE Standard 1363-2000 (http://grouper.ieee.org/groups/1363), RSA standard PKCS #1 (http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/index.html), PKCS# 3 (http://www.rsasecurity.com/rsalabs/pkcs/pkcs-3/index.html), Key management FIPS PUB 171 (http://csrc.nist.gov/publications/fips/fips171/fips171.txt), FIPS Pub 186*].[FCS_CKM.1.1]

Cryptographic key distribution (FCS_CKM.2)

The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method**s** [*Diffie-Hellman*] that meets the following: [*PKCS# 3 (http://www.rsasecurity.com/rsalabs/pkcs/pkcs-3/index.html), Key management FIPS PUB 171 (http://csrc.nist.gov/publications/fips/fips171/fips171.txt)*].[FCS_CKM.2.1]

Cryptographic key destruction (FCS_CKM.4)

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method, [*zeroisation*] that meets the following : [*no standard*]. $^{\text{FCS\_CKM.4.1}}$

Cryptographic operation (FCS_COP.1)

The TSF shall perform [

a) Digital signing

b) Digital Signature Verification

c) Cryptographic Key agreement

d) Encryption / Decryption

] in accordance with ~~a~~ specified cryptographic algorithm**s** [

a) RSA, DSA

b) RSA

c) Diffie-Hellman

d) RSA, DES, 3DES

] and cryptographic key sizes [

a) 512, 768 and 1024 bits for DSA and RSA

b) 512, 768 and 1024 bits

c) Diffie-Hellman: Primes from 512 – 1024 bits and Exponents from 128 – 256 bits

d) RSA (512, 768 and 1024)

DES (40 and 56 bit)

3DES (112 and 168 bits)

] that meet the following: [

a) RSA signing (PKCS #1), DSA (FIPS PUB 186)

b) RSA signature verification (PKCS #1)

c) Diffie Hellman (PKCS #3)

*d) Encryption/Decryption RSA (PKCS #1), DES (FIPS 46-3), 3DES (*ANSI X9.52-1998)
]. $^{\text{FCS\_COP.1.1}}$

5.1.2 User Data Protection

Subset access control (FDP_ACC.1)

The TSF shall enforce the [*Access Control SFP*] on [

a) Subject: *Client Application*

b) Object: *Protected user information on TOE (i.e. private key, Digital Id)*

c) Operation: *access, modification, addition and deletion*] FDP_ACC.1.1

Security attribute based access control (FDP_ACF.1)

The TSF shall enforce the [*Access Control SFP*] to objects based on [*user authentication*].FDP_ACF.1.1
The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [*allow access to user information once the user has been authenticated by supplying the correct pass phrase*].FDP_ACF.1.2
The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [*view only access is permitted to Inactivity Timer Setting, Token Status, Reader Status and Token Information*].FDP_ACF.1.3
The TSF shall explicitly deny access of subjects to objects based on the [

a) Memory	*No access to memory shall be allowed except as mediated through the TOE embedded software.*
b) Embedded Software	*No access to the embedded software shall be allowed.*
c) Token Lockout	*The token locking itself shall deny all access to the token, except for TOE initialisation.*

].FDP_ACF.1.4

Export of user data without security attributes (FDP_ETC.1)

The TSF shall enforce the [*Access Control SFP*] when exporting user data, controlled under the SFP(s), outside of the TSC.FDP_ETC.1.1
The TSF shall export the user data without the user data's associated security attributes.FDP_ETC.1.2

Import of user data without security attributes (FDP_ITC.1)

The TSF shall enforce the [*Access Control SFP*] when importing user data, controlled under the SFP, from outside of the TSC. FDP_ITC.1.1
The TSF shall ignore any security attributes associated with the user data when imported from outside the TSC.FDP_ITC.1.2
The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [*no other rules*].FDP_ITC.1.3

5.1.3 Identification and Authentication

Authentication failure handling (FIA_AFL.1)

The TSF shall detect when [*10*] unsuccessful authentication attempts occur related to [*the user*

entering their pass phrase].$^{FIA\_AFL.1.1}$
When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [*lock itself out. The TOE will need re-initialisation*].$^{FIA\_AFL.1.2}$

User attribute definition (FIA_ATD.1)

a) The TSF shall maintain the following list of security attributes belonging to individual users: [*User access pass phrase*]. $^{FIA\_ATD.1.1}$

User authentication before any action (FIA_UAU.2)

The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.$^{FIA\_UAU.2.1}$

5.1.4 Security Management

Management of security attributes (FMT_MSA.1)

The TSF shall enforce the [*Access Control SFP*] to restrict the ability to [*modify*] the security attributes [*User access pass phrase*] to [*authenticated users of the TOE*].$^{FMT\_MSA.1.1}$

Secure security attributes (FMT_MSA.2)

The TSF shall ensure that only secure values are accepted for security attributes.$^{FMT\_MSA.2.1}$

Static attribute initialisation (FMT_MSA.3)

The TSF shall enforce the [*Access Control SFP*] to provide [*restrictive*] default values for security attributes that are used to enforce the SFP.$^{FMT\_MSA.3.1}$
The TSF shall allow the [*authenticated users*] to specify alternative initial values to override the default values when an object or information is created.$^{FMT\_MSA.3.2}$

5.1.5 Protection of the TOE Security Functions

Abstract machine testing (FPT_AMT.1)

The TSF shall run a suite of self-tests [*during initial start-up and at the request of the authenticated user*] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.$^{FPT\_AMT.1.1}$

Failure with preservation of secure state (FPT_FLS.1)

The TSF shall preserve a secure state when the following types of failures occur: [*iKey token removed from USB port*].$^{FPT\_FLS.1.1}$

Resistance to physical attack (FPT_PHP.3)

The TSF shall resist [*electrostatic discharge and physical tampering*] to the [*iKey token*] by responding automatically such that the TSP is not violated.$^{FPT\_PHP.3.1}$

TSF domain separation (FPT_SEP.1)

The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.FPT_SEP.1.1
The TSF shall enforce separation between the security domains of subjects in the TSC.FPT_SEP.1.2

5.2 TOE Security Assurance Requirements

The TOE Security Assurance Requirements (SARs) satisfy the assurance requirements for a target of EAL 2 as defined in CC part 3. Table 5-2 lists the assurance components required for EAL 2.

Table 5-2 EAL 2 Assurance Requirements

Assurance Class	Assurance Components
ACM	ACM_CAP.2
ADO	ADO_DEL.1 ADO_IGS.1
ADV	ADV_FSP.1 ADV_HLD.1 ADV_RCR.1
AGD	AGD_ADM.1 AGD_USR.1
ATE	ATE_COV.1 ATE_FUN.1 ATE_IND.2
AVA	AVA_SOF.1 AVA_VLA.1

6 TOE Summary Specification

This section presents the Security Functions and the Assurance Measures implemented by the TOE Security Requirements.

6.1 IT Security Functions

6.1.1 Authentication Functions

This section provides information regarding the authentication functions of the iKey.

Lockout

The iKey token comes with a preset number of authentication attempts allowed. This preset number is 10 attempts. iKey locks out after the preset number of consecutive incorrect login attempts. Once locked, the token requires re-initialisation.

Passphrase

The iKey token has one Pass Phrase that is required to authenticate the token user. A user that is authenticated by entering the Pass Phrase can initiate protected iKey token operations.

Pass Phrases are case-sensitive and the minimum length is four alphanumeric characters, and the maximum length is 20 alphanumeric characters.

The Pass Phrase is required for every requested encryption and signing operation.

Because of the Lockout feature the strength of function claim for iKey pass phrase authentication is SOF_High.

6.1.2 Cryptographic Functions

This section provides information regarding the cryptographic functions of the iKey. It is not appropriate to make a strength of function claim for the iKey cryptographic functions.

PKP - Private Key Protection

Private keys are encrypted (software) on the iKey token. All private keys are protected from access by a user-defined password. The private key never leaves the token.

Crypto Algs

The following table provides the cryptographic algorithm specifications for iKey. The cryptographic operations are provided by the Datakey smart card chip within the iKey.

Table 6-1 Cryptographic Algorithm Specification

Support for Digital Signature:	**Performed on the Token**	**Algorithm / Function**	**Key Size**	**Additional Information**
Key Pair Generation on the Token	Yes	N/A	N/A	Extended functions available from Datakey.
Algorithms and key size(s) supported	Yes	RSA / DSA	Up to 1024-Bit	N/A
Private Key Storage on the Token	Yes	N/A	N/A	N/A
Verification Certificate Storage on the Token	Yes	N/A	N/A	N/A
Hashing Operation on the Token	No	N/A	N/A	Hashing operations are performed at the library level.
Hashing Functions supported.	Yes	MD2, MD5, SHA-1	N/A	N/A
Signing Operation on the Token	Yes	N/A	N/A	N/A
Signature Verification on the Token	Yes	N/A	N/A	Extended functions available from Datakey.
Support for encrypt/decrypt				
Private Key Unwrapping on the Token	No	N/A	N/A	N/A
Private Key(s) storage on the Token	Yes	N/A	N/A	N/A
Algorithms and key size(s) supported (e.g. RSA, ECC)	Yes	RSA	Up to 1024 bits	N/A
Algorithms and key size(s)	Yes	DES	40 and 56	N/A

supported (e.g. RSA, ECC)			bits	
Algorithms and key size(s) supported (e.g. RSA, ECC)	Yes	3DES	112 and 168 bits	N/A
Encryption Certificate storage on the Token	Yes	N/A	N/A	N/A
Encryption on the Token	Yes	N/A	N/A	Extended functions available from Datakey.
Decryption on the Token	Yes	N/A	N/A	N/A
Support for Key Agreement				
Private Key Unwrapping on the Token	No	N/A	N/A	N/A
Private Key(s) storage on the Token	Yes	N/A	N/A	N/A
Algorithms and key size(s) supported (e.g. Diffie-Hellman, ECDH)	Yes	Diffie-Hellman	Primes from 512-1024 bits and Exponents from 128-256 bits	N/A
Key Agreement Certificate storage on the Token	Yes	N/A	N/A	N/A
Key Generation on the Token	Yes	N/A	N/A	Extended functions available from Datakey.
Key Derivation on the Token	No	N/A	N/A	N/A
Key Wrapping on the Token	No	N/A	N/A	N/A
Key Unwrapping on the Token	No	N/A	N/A	N/A

When cryptographic keys are generated the token identifies if keys are considered to be weak or semi-weak and the keys are discarded as appropriate.

Weak keys are those that auto-inverse. So if the plain text is encrypted twice using the same weak key then the original plain text would result.

Semi-weak key pairs are inverting pairs. If one key is used to encrypt the plain text and the result is encrypted using the second key, then the original plain text would result.

When cryptographic keys are no longer required they are removed by having zeroes written over their memory location.

The above table lists that encryption on the token is supported, but in a public key infrastructure the user is encrypting with the recipient's public key, which is not stored on the token. An example of this situation is encrypting e-mails via Outlook. The token does not play any part in encrypting Outlook e-mails.

6.1.3 Token Operations

This section provides information on initialising the iKey token, running a diagnostic on it, and displaying the information stored in it. It is not appropriate to make a strength of function claim for these iKey functions.

Initialise

Initialising the token leaves only the serial number and token label intact. All other information is removed from the token. This includes all key pairs, certificates, and data objects stored on the token. Any pass phrase entered is replaced with the default pass phrase.

Test

The user can initiate the token into performing a test of its cryptographic functions. A pair of keys are generated on the iKey token if there is enough space on it. The keys are used to encrypt and decrypt test data. The keys are then signed and verified using the same key. If the token is good, these cryptographic operations are performed successfully. If the test fails, the token may be corrupted.

The TOE also conducts a Power On Self Test (POST) after reset and before executing the first command. The areas tested are:

- RAM;
- FAME co-processor;
- file system integrity;
- checksum of the Read Only Memory (ROM) contents; and
- Known Answer Tests for DES, 3DES, SHA-1, DSA sign / verify, RSA sign / verify and RSA encrypt / decrypt.

The POST ensures the TOE is functioning correctly before any user operations take place.

Object

Information about the objects stored on the iKey token can be displayed. Public objects like certificates and public keys can be viewed without authentication, while supplying the Pass Phrase will also show information about the private objects, like the Private Key. NOTE this information is only about the objects, like names and size in bytes, and not sensitive information like the private key value.

6.1.4 Physical Token Protection

This section provides information regarding the physical protection of the iKey.

Epoxy

Critical components of the iKey are covered with a layer of Epoxy. The Epoxy provides resistance against tampering which if tried to be removed will damage the token and make it non-functional.

Electro

The iKey is rated to withstand 6kV Electrostatic Discharge (ESD). ESD can cause irreparable damage to electronic components and chips. Such damage caused by ESD on any of the iKey components and chips may result in protected assets being disclosed or security functions being bypassed.

RAM

The RAM temporarily stores information regarding the iKey's current operations. Information within the iKey's RAM is automatically erased when the token is removed from the USB port. The sort of security related information that may be in RAM includes any current cryptographic operations and temporary keys, and a flag indicating whether the token is logged onto. Protected information like private keys, digital IDs and pass phrase are stored in non-volatile memory and is not affected by the clearing of RAM.

6.1.5 Token Management (T_MGT)

This section provides information regarding the management of the token. It is not appropriate to make a strength of function claim for these iKey functions.

The iKey has the following management functions:

- initialise and personalise the iKey by modifying the Pass Phrase and Token Label stored on the token;
- A user with the token Pass Phrase can configure the token settings for the Token Label, Pass Phrase and Inactivity Timer; and
- An existing Digital ID can be imported onto the iKey token. This action requires Netscape Communicator 4.x or the Token Utility software on a trusted computer.

The Inactivity Timer is used by the MS CAPI layer to determine when a user should re-authenticate. The TOE does not enforce re-authentication when the Inactivity Timer has expired.

6.2 Assurance Measures

This section describes the assurances measures taken so that iKey satisfies the Security Assurance Requirements for an Evaluation Assurance Level (EAL) 2.

6.2.1 Configuration Management for iKey

The Configuration Management (CM) system used in developing the TOE uniquely identifies the configuration items that compose the TOE. The CM plan (document number 011-0001-002 and Bill of Materials for item 107650-001, revision E) lists the configuration items of the TOE and describes how each item is uniquely identified.

Each iKey token contains model and serial numbers on the external casing.

6.2.2 iKey Distribution & Delivery Procedures

When distributing or delivering the TOE to users the developers follow the iKey Distribution and Delivery Procedures (document number 011-0002-001) to ensure that the users get an authentic product.

6.2.3 iKey Design Specification

Documentation generated throughout the development of iKey is contained in the iKey Design Specification (document number 150-0002-002), which:

- describes the features and interfaces of the iKey and relevant software. Includes a technical specification of the dimensions of the iKey token.
- provides information of the major components that make up the TOE. It provides the functionality of each component, emphasising the security functions and the related interfaces.

- shows that TSF representations are complete and consistent throughout the design documentation.

The Design Specification is an umbrella document and refers to other sources of design information. Two Rainbow controlled sources are document numbers 005-0003-001 and 107227.

6.2.4 iKey Guidance Documentation

The iKey User's Guide (document number 700771-001) and Technical Notes TN2140 and TN2156 provides all the information needed to initialise, personalise and use the iKey token. This guide also provides information about installing and integrating the TOE with applications. This manual comes in electronic format with the software provided with the iKey.

6.2.5 iKey Testing

Development of the iKey includes the validation and verification of the product. Test plans and procedures have been developed to prove the functionality of the product as defined in the Design Specification. Test results have been fed back into the development cycle. The test plans and results are contained in document Software Test Report for iKey 2000 Comet Version 4.6.10 (document number 108601), which is supplemented by TestEngine software (document number 155-0001-011). Analysis that the documented tests sufficiently cover the TSFs is provided in a spreadsheet (document number 150-0004-001).

6.2.6 iKey EAL 2 Vulnerability Analysis

An evaluation specific analysis (document number 150-0003-002) of the TOE will be conducted to show that each claim of strength is exceeded and that all identified vulnerabilities cannot be exploited.

7 PP Claims

This ST does not make any compliance claim with any Protection Profile.

8 Rationale

The purpose of the ST rationale is to demonstrate that a conformant TOE would provide an effective set of IT security countermeasures within the TOE security environment. In particular, it shows that the IT security requirements are suitable to meet the security objectives, which in turn are shown to be suitable to cover all aspects of the TOE security environment (which defines the security needs).

8.1 Security Objectives Rationale

This section shows that the security objectives are *sufficient* and *necessary* to address the security needs. Table 8-1 shows that each threat, OSP and assumption is covered by at least one security objective.

Table 8-1 Mapping the TOE Security Environment to Security Objectives

Policy/Threat/Assumptions	Objectives
Security Objectives for the TOE	
A.USER	O.AUTHENTICATE
T.DISCLOSURE	O.AUTHENTICATE, O.DISCLOSURE, O.CRYPT
T.IMPERSON	O.AUTHENTICATE, O.SECURE
T.CORRUPT	O.OPERATE
T.MODIFY	O.TAMPER, O.MODIFY
T.PHYSICAL	O.TAMPER, O.ELEC_STRESS
Security Objectives for the Environment	
A.PWR_CLOCK	OE.PWR_CLOCK
A.USER	OE.AUTHDATA
A.HOST	OE.HOST_ENV
T.IMPERSON	OE.AUTHDATA

Policy/Threat/Assumptions	Objectives
Security Objectives for the TOE	
TE.INSECURE	OE.INITIALISE, OE.BREACH, OE.PERMANENT, O.SECURE
P.INITIALISE	OE.INITIALISE
P.BREACH	OE.BREACH, O.SECURE
P.PERMANENT	OE.PERMANENT, O.SECURE

Table 8-2 shows that each security objective covers at least one threat, OSP or assumption and therefore all security objectives are necessary.

Table 8-2 Tracing of Security Objectives to the TOE Security Environment

Objectives	Policy/Threat/Assumptions
Security Objectives for the TOE	
O.AUTHENTICATE	T.DISCLOSURE, T.IMPERSON, A.USER
O.TAMPER	T.MODIFY, T.PHYSICAL
O.OPERATE	T.CORRUPT
O.DISCLOSURE	T.DISCLOSURE
O.MODIFY	T.MODIFY
O.CRYPT	T.DISCLOSURE
O.ELEC_STRESS	T.PHYSICAL
O.SECURE	T.IMPERSON, TE.INSECURE, P.BREACH, P.PERMANENT
Security Objectives for the Environment	
OE.INITIALISE	TE.INSECURE, P.INITIALISE
OE.AUTHDATA	T.IMPERSON, A.USER
OE.BREACH	TE.INSECURE, P.BREACH

OE.PWR_CLOCK	A.PWR_CLOCK
OE.HOST_ENV	A.HOST
OE.PERMANENT	TE.INSECURE, P.PERMANENT

To show that the security objectives are sufficient to meet the security needs, an informal argument for each aspect of the TOE security environment is provided below.

8.1.1 Assumptions

Table 8-3 provides informal arguments as to why the identified security objectives are sufficient to uphold the assumptions.

Table 8-3 Sufficiency of Security Objectives for Assumptions

Assumption	Argument to show Sufficiency of Security Objectives
A.PWR_CLOCK Power and clock	The objective OE.PWR_CLOCK will uphold this assumption because: ▪ The token uses the power and clock sources from the USB port.
A.USER User Privilege	The objectives O.AUTHENTICATE, OE.AUTHDATA will uphold this assumption because: ▪ Proper owner/user of token will require the pass phrase to access information from the token; and ▪ The user will know the pass phrase but not disclose it to anyone else.
A.HOST Host Computer Requirements	The objective OE.HOST_ENV will uphold this assumption because: ▪ Installation of the appropriate hardware and software will enable users to access the data stored on the token and use the functions provided by the TOE.

8.1.2 Policies

Table 8-4 provides informal arguments as to why the identified security objectives are sufficient to provide complete coverage of the OSP.

Table 8-4 Sufficiency of Security Objectives for Policies

Policy	Argument to show Sufficiency of Security Objectives
P.INITIALISE	The objective OE.INITIALISE will completely cover this policy because: ▪ Those responsible for the token (owner/user) will ensure the organisational policies will be followed at initialisation.
P.BREACH	The objectives OE.BREACH and O.SECURE will completely cover this policy because: ▪ The TOE provides a facility to initialise any suspected breached token; and ▪ The CA controlling the public certificates of the Digital ID will revoke the relevant Digital Ids.
P.PERMANENT	The objectives OE. PERMANENT and O.SECURE will completely cover this policy because: ▪ Those responsible for the token (owner/user) will ensure there will not be any permanent assets upon the token; and ▪ When the token is initialised there will not be assets remaining upon the token.

8.1.3 Threats

Table 8-5 provides informal arguments as to why the identified security objectives are sufficient to counter the threats.

Table 8-5 Sufficiency of Security Objectives for Threats

Threat	Argument to show Sufficiency of Security Objectives
T.DISCLOSURE Unauthorised disclosure of assets	The objectives O.AUTHENTICATE, O.DISCLOSURE, O.CRYPT will address this threat because: ▪ Any attempted access to the assets will require authentication via the pass phrase reducing likelihood of unauthorised disclosure; and ▪ Information contained on the token will be protected and not disclosed unless correct pass phrase is presented; and ▪ Crypto operations are performed upon the token meaning the private key has no need to leave the token. This reduces possibility of private key being disclosed.
T.IMPERSON	The objectives O.AUTHENTICATE, O.SECURE OE.AUTHDATA will address this threat because: ▪ The likelihood of a successful impersonation is reduced by the authentication measures (pass phrase) required to access protected data and operations on the token; and ▪ Impersonations are further reduced by the token restricting the number invalid authentication attempts before lockout. The only way to re-use token again is via initialisation, which removes existing assets from token. ▪ Proper owners of the token shall not reveal the authenticating pass phrase, reducing likelihood of impersonation.
T.CORRUPT	The objective O.OPERATE will address this threat because: ▪ The token will allow the user to test token operations to determine whether token is corrupted or not.
T.MODIFY	The objectives O.TAMPER, O.MODIFY will address this threat because: ▪ The token will protect its security parts from modification preventing a compromise in security of assets by not allowing any alien software to be loaded on or against the token; and ▪ The token will protect the assets stored in memory against unauthorised modification.

Threat	Argument to show Sufficiency of Security Objectives
T.PHYSICAL	The objectives O.TAMPER and O.ELEC_STRESS will address this threat because: ▪ The token will resist physical attacks from revealing protected assets; and ▪ The token has a structure that will not allow a compromise of security when exposed to stressful electrical conditions.
TE.INSECURE	The objectives OE.INITIALISE, OE.BREACH, OE.PERMANENT and O.Secure will address this threat because: ▪ Initialisation of the token will be done in such a manner to ensure that the security of the token is at an acceptable level to the owner. ▪ Any suspicion of a token's security being breached will result in the token being initialised and the Digital Ids revoked at the CA. ▪ There will not be any permanent assets put onto the token by the owner/user. ▪ When the token is initialised assets will be returned to a secure state.

8.2 Security Requirements Rationale

This section shows that the identified IT security requirements (and the SFRs in particular) are suitable to meet the identified security objectives, and thereby address the security needs.

8.2.1 Security Functional Requirements Rationale

Table 8-6 and Table 8-7 show that each security objective is addressed by at least one Security Functional Requirement (SFR), and vice versa, therefore each SFR is necessary.

Table 8-6 Mapping of Security Objectives to SFRs

Objectives	Requirements
O.AUTHENTICATE	FIA_AFL.1, FIA_ATD.1, FIA_UAU.2
O.TAMPER	FDP_ACC.1, FDP_ACF.1, FPT_FLS.1, FPT_PHP.3, FPT_SEP.1,

	FIA_AFL.1
O.OPERATE	FPT_AMT.1
O.DISCLOSURE	FDP_ACC.1, FDP_ACF.1, FDP_ETC.1
O.MODIFY	FDP_ACC.1, FDP_ACF.1, FDP_ITC.1, FMT_MSA.1, FMT_MSA.2, FMT_MSA.3
O.CRYPT	FCS_CKM.1, FCS_CKM.2, FCS_CKM.4, FCS_COP.1
O.ELEC_STRESS	FPT_FLS.1, FPT_PHP.3
O.SECURE	FDP_ACC.1, FDP_ACF.1, FIA_AFL.1, FMT_MSA.3

Table 8-7 Mapping of SFRs to Security Objectives

Requirements	Objectives
FCS_CKM.1	O.CRYPT
FCS_CKM.2	O.CRYPT
FCS_CKM.4	O.CRYPT
FCS_COP.1	O.CRYPT
FDP_ACC.1	O.TAMPER, O.DISCLOSURE, O.MODIFY, O.SECURE
FDP_ACF.1	O.TAMPER, O.DISCLOSURE, O.MODIFY, O.SECURE
FDP_ETC.1	O.DISCLOSURE
FDP_ITC.1	O.MODIFY
FIA_AFL.1	O.AUTHENTICATE, O.TAMPER, O.SECURE
FIA_ATD.1	O.AUTHENTICATE
FIA_UAU.2	O.AUTHENTICATE
FMT_MSA.1	O.MODIFY
FMT_MSA.2	O.MODIFY
FMT_MSA.3	O.MODIFY, O.SECURE

Requirements	Objectives
FPT_AMT.1	O.OPERATE
FPT_FLS.1	O.TAMPER, O.ELEC_STRESS
FPT_PHP.3	O.TAMPER, O.ELEC_STRESS
FPT_SEP.1	O.TAMPER

Table 8-8 provides the informal argument for the sufficiency of the SFRs to satisfy each security objective.

Table 8-8 Sufficiency of SFRs to meet Security Objectives

Security Objective	Argument to show Sufficiency of Security Functional Requirements
O.AUTHENTICATE	This objective is satisfied by the following SFRs: FIA_ATD.1 provides the list of security attributes required to enforce the TSP FIA_UAU.2 denies protected actions requested from the user before authenticating FIA_AFL.1 restricts the number of unsuccessful authorisation attempts so an attacker cannot use a "brute-force" or dictionary pass phrase attack
O.TAMPER	This objective is satisfied by the following SFRs: FDP_ACC.1 defines the Access Control SFP for the TOE and, FDP_ACF.1 provides the rules that apply for the Access Control SFP which restrict attackers from tampering with the security critical parts of the TOE FIA_AFL.1 restricts the number of unsuccessful authorisation attempts so an attacker cannot use a "brute-force" or dictionary pass phrase attack FPT_FLS.1 preserves the assets to a secure state when failures occur due to tampering FPT_PHP.3 provides resistance to physical tampering FPT_SEP.1 allows execution in own domain to protect from tampering of untrusted subjects
O.OPERATE	This objective is satisfied by the following SFRs: FPT_AMT.1 provides self-testing to ensure the correct operation of the token.
O.DISCLOSURE	This objective is satisfied by the following SFRs: FDP_ACC.1 and FDP_ACF.1 define and enforce the Access Control SFP to stop unauthorised disclosure of assets FDP_ETC.1 enforces the Access Control SFP when exporting data so no protected assets are disclosed to users who haven't been authorised
O.MODIFY	This objective is satisfied by the following SFRs: FDP_ACC.1 and FDP_ACF.1 define and enforce the Access Control SFP to stop unauthorised modification of assets FDP_ITC.1 enforces the Access Control SFP when importing data so no

Security Objective	Argument to show Sufficiency of Security Functional Requirements
	assets are modified by users who haven't been authorised FMT_MSA.1, FMT_MSA.2, FMT_MSA.3 allows authorised users to manage the security attributes of the TOE
O.CRYPT	This objective is satisfied by the following SFRs: FCS_COP.1 providing cryptographic key operations, which is supported by FCS_CKM.1 for key generation, FCS_CKM.2 for key distribution, and FCS_CKM.4 for key destruction.
O.ELEC_STRESS	This objective is satisfied by the following SFRs: FPT_PHP.3 provides physical protection, while FPT_FLS.1 provides security of assets when failures occur.
O.SECURE	This objective is satisfied by the following SFRs: FDP_ACC.1 and FDP_ACF.1 provide that access to the protected information and operations is denied if the token is locked. In addition, FDP_ACF.1.4 c) provides the ability for a locked token to be re-initialised, which destroys all protected assets. FIA_AFL.1 provides lockout against unauthorized access attempts. FMT_MSA.3 ensures that security attributes are restrictive when the token is initialised meaning no previous user information can be revealed.

8.2.2 Security Assurance Requirements Rationale

This section shows that the identified Security Assurance Measures are appropriate to meet the Security Assurance Requirements (SARs) for an EAL 2. Table 8-9 shows the SARs and the corresponding measure to which it is mapped. Also provided is an explanation of how the requirements will be met by the measure.

Table 8-9 Assurance Measures meet Assurance Requirements

Assurance Requirement	Assurance Measure	How measure meets requirement
ACM_CAP.2	Configuration Management for iKey	Provides all documentation related to the Configuration Management system used in developing the TOE, including information about unique labelling and referencing of the TOE.
ADO_DEL.1	iKey Distribution & Delivery Procedures	Describes the delivery procedures necessary when distributing or delivering the TOE to users.
ADO_IGS.1	The iKey Guidance Documentation	Describes the necessary steps for secure installation, generation and start-up of the TOE.

Assurance Requirement	Assurance Measure	How measure meets requirement
ADV_FSP.1	iKey Design Specification	Describes the purpose and method of use of the TSF and external interfaces.
ADV_HLD.1	iKey Design Specification	Presents the structure in terms of sub-systems of the TSF. These sub-systems shall have their security features and structure described and interfaces identified.
ADV_RCR.1	iKey Design Specification	The Design Specification will demonstrate that as security functionality is more refined in documentation that no details are missing or inconsistent. This analysis will show the required mappings and justifications.
AGD_ADM.1	The iKey Guidance Documentation	Shall describe administrative functions and how to administer the TOE securely. The guidance is sufficient for the AGD_ADM.1 requirement because the administrator and user are considered to be the same.
AGD_USR.1	The iKey Guidance Documentation	The guidance provides user function guidance and warnings about using the TOE.
ATE_COV.1	iKey Testing	The analysis (document number 150-0004-001) will show the correspondence between the TSF as described in the Design Specification and the tests identified in the Test Documentation.
ATE_FUN.1	iKey Testing	Test report (document number 108601) contains the test plans, procedures, expected outcomes and actual results
ATE_IND.2	iKey Testing	Resources equivalent to the developer's testing shall be provided to the evaluators so they can conduct independent testing. These resources include the TOE, interfaces, test software and test documentation.
AVA_SOF.1	iKey EAL 2 Vulnerability Analysis	Each claim of strength of security mechanism shall be assessed and shown to exceed that claim.
AVA_VLA.1	iKey EAL 2 Vulnerability Analysis	This analysis will show that all identified vulnerabilities cannot be exploited.

As stated in section 3.2.1 of this ST, the TOE must protect against attacks accomplished with standard personal computers and with laboratory equipment used without any supervision. The developers have chosen EAL 2 because it provides a low to moderate level of independently assured security and ensures the token is structurally tested. EAL 2 requires the high-level design is independently analysed to provide assurance is the security functions of the token. The developers have determined this level of design information is suitable, sufficient and attainable. EAL 2 also requires independent testing, confirmation of developer testing, strength of function analysis and evidence of a developer search for obvious vulnerabilities. This testing and analyses is sufficient for the token to be securely used in its intended environment. The developers conclude that the assurance provided by EAL 2 is suitable to meet the needs of the TOE and its environment.

8.3 Strength of Function Claim

This section shows how the minimum strength of function level for the ST is consistent with the security objectives for the TOE. This ST claims SOF_High for the strength of function level of the TOE, as the TOE is expected to be in a hostile, uncontrolled environment.

The SOF_High claim and EAL 2 may seem mismatched. However, if a token owner suspects that the token security is breached (stolen and/or the pass phrase is disclosed and/or token physically tampered with) then the Digital ID on the token can be revoked at the CA, which will render the private key on the token useless. The policy P.BREACH and Environmental Objective OE.BREACH are in place to ensure that this action occurs. Therefore, the SOF_High claim and EAL 2 are appropriate for this ST.

A specific strength of function claim of SOF_High is made for FIA_UAU.2 as this SFR relates to authentication of a user to the token. This SOF claim has been determined based on the following:

Assuming the worst-case scenario of only 4-character pass phrase and digit-only or common English words used there are 25210 pass phrase combinations. The attacker will guess the pass phrase within half of the total possible pass phrase which results 12605 guesses. As the iKey will lockout the token after 10 unsuccessful pass phrase entries, the attacker will have 8 attempts between the owner's successful logons in order the attempts not to be detected. We assume the valid owner logs in 6 times a day, giving the attacker 48 guesses per day which results 262 days for the attack to succeed. If attacker is a co-worker and there is 20 working days a month the attacker would have to keep up this style of attack for 13.1 months.

It is not appropriate for cryptographic claims to be made in this ST, as evaluation of strength of cryptographic functions is the responsibility of the National Comsec Authority. This claim relates to the FCS Class of SFR, and to the TSFs PKP, Crypto Algs and Test.

The TSF Passphrase claims a strength of function claim of SOF_High as this TSF implements FIA_UAU.2 to provide user authentication. Further details are provided in Section 6.1.1.

8.4 Dependency and Mutual Support Rationale

The purpose of this rationale is to show that the IT security requirements (and the SFRs in particular) are complete and internally consistent by demonstrating that they are mutually supportive and provide an 'integrated and effective whole'.

Dependency helps in showing mutual support because if SFR-A is dependent on SFR-B then by definition, SFR-B is supportive of SFR-A. Table 8-10 shows the required dependencies of the Security Functional Requirements. Where an SFR has no required dependencies the corresponding dependency field is left blank.

This ST is targeting a standard EAL 2 assurance package and so the dependency and mutual support of the assurance requirements is self-evident as the EAL is taken from the CC.

Table 8-10 Functional and Assurance Requirements Dependencies

Requirement	Dependencies
Functional Requirements	
FCS_CKM.1	FCS_COP.1, FCS_CKM.4, FMT_MSA.2
FCS_CKM.2	FCS_CKM.1, FCS_CKM.4, FMT_MSA.2
FCS_CKM.4	FCS_CKM.1, FMT_MSA.2
FCS_COP.1	FCS_CKM.1, FCS_CKM.4, FMT_MSA.2
FDP_ACC.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1, FMT_MSA.3
FDP_ETC.1	FDP_ACC.1
FDP_ITC.1	FDP_ACC.1, FMT_MSA.3
FIA_AFL.1	FIA_UAU.2
FIA_ATD.1	
FIA_UAU.2	
FMT_MSA.1	FDP_ACC.1
FMT_MSA.2	FDP_ACC.1, FMT_MSA.1
FMT_MSA.3	FMT_MSA.1
FPT_AMT.1	
FPT_FLS.1	
FPT_PHP.3	
FPT_SEP.1	

8.4.1 Justification of Unsupported Dependencies

The following is a justification of required dependencies that are not met by this ST.

FIA_UAU.2 depends on FIA_UID.1, however the token does not have the concept of multiple users and user identification. The possessor of the token verifies their privilege to protected resources via the token pass phrase. Therefore, FIA_UID.1 is not applicable for the TOE, which is why it is justifiably not included in this ST.

FMT_MSA.1, .2 and .3 all depend upon FMT_SMR.1. Again the token does not recognise roles as access is granted on supply of the correct pass phrase. Therefore, FMT_SMR.1 is not applicable for the TOE, which is why it is justifiably not included in this ST.

The Common Criteria recommends that FMT_MSA.2 and FPT_FLS.1 depend upon the assurance component ADV_SPM.1.

FMT_MSA.2 states that only secure attributes are to be used. This SFR has a dependency upon ADV_SPM.1 to define the secure attributes. The Cryptographic Support (FCS) class SFRs have dependencies on FMT_MSA.2 regarding cryptographic keys. The TOE ensures that only secure attributes (cryptographic keys) are used by discarding weak and semi-weak keys as appropriate. Since the secure attributes has been defined in the TSF Crypto Algs i.e. weak or semi-weak keys are rejected then there is no need to provide the Security Policy model for this case.

FMT_MSA.2 is also used for T_MGT. The user can edit the Passphrase values. However, as the ranges defined for these values are considered to be secure (see section **6.1.1** Authentication Functions) there is no need to provide the Security Policy model.

FPT_FLS.1 states that a secure state is preserved. This SFR has a dependency upon ADV_SPM.1 to define the secure state. The secure state of this SFR is defined in the TSF. The TSF RAM defines what the state of the TOE is once removed from the USB port. This state as defined in the TSFs is considered to be secure. With all secure states defined in the TSF, there is no need to provide the Security Policy model for this SFR.

Because all pertinent security attribute and state information is defined in this ST there is no need to provide the ADV_SPM.1 assurance measure.

8.4.2 Mutual Support of Security Functional Requirements

For those SFRs not directly related by dependency, mutual support can be provided by SFRs which address the following:

Help prevent bypassing of other SFRs

FIA_UAU.2 support other functions which allow the user access to the assets by restricting the actions the user can take before being authorised.

The management function FMT_MSA.1 supports all other SFRs by restricting the ability to change certain management functions to authorised users, ensuring other users cannot circumvent these SFRs.

FMT_MSA.2 and FMT_MSA.3 limit the acceptable values for secure data, protecting the SFRs dependent on those values from being bypassed.

FPT_PHP.3 provides protection against bypass to all other SFRs by maintaining acceptable security in the event of environmental stress or tampering.

FPT_FLS.1 provides for recovery of a secure state after failure or service discontinuity, preventing bypass of other SFRs.

FPT_AMT.1 provides for user initiated testing to ensure the cryptographic security functions are operational, thus preventing their bypass.

Help prevent tampering of other SFRs

The cryptographic functions FCS_CKM.1, FCS_CKM.2, FCS_CKM.4 and FCS_COP.1 provide for the secure generation, handling, destruction and operation of keys, and therefore support those SFRs which may rely on the use of those keys.

FIA_AFL.1 supports all SFRs dealing with authentication by limiting the number of authentication attempts and taking appropriate action to protect the token if the limit has been reached.

FIA_UAU.2 supports other functions that allow the user access to the assets by restricting the actions the user can take before being authorised.

FMT_MSA.1 supports all other SFRs by restricting the ability to change certain management functions to authorised users, ensuring other users cannot tamper with these SFRs.

FMT_MSA.2 and FMT_MSA.3 limit the acceptable values for secure data, protecting the SFRs dependent on those values from being tampered with.

FPT_FLS.1 provides for recovery of a secure state after failure or service discontinuity, preventing tampering of other SFRs

FPT_PHP.3 provides protection from tampering of all other SFRs by resisting physical tampering or electro static stress.

FPT_SEP.1 prevents tampering from untrusted subjects, thus preventing tampering with the correct operation of other SFRs.

FPT_AMT.1 provides for user initiated testing to ensure the cryptographic security functions are operational, thus checking for tampering.

Help prevent de-activation of other SFRs

The Access Control policy detailed in FDP_ACF.1 and FDP_ACC.1 along with the other SFRs involved in access control, provide for rigorous control of allowed data manipulation, preventing unauthorised deactivation of SFRs.

FMT_MSA.1 supports all other SFRs by restricting the ability to change certain management functions to authorised users, ensuring other users cannot de-activate these SFRs.

FMT_MSA.2 and FMT_MSA.3 limit the acceptable values for secure data, protecting the SFRs dependent on those values from being de-activated.

FPT_FLS.1 provides for recovery of a secure state after failure or service discontinuity, preventing de-activation of other SFRs

FPT_PHP.3 provides protection from de-activation of all other SFRs by maintaining acceptable security in the event of physical tampering or electro static stress.

FPT_AMT.1 provides for user initiated testing to ensure the cryptographic security functions are operational, thus checking for de-activation. FPT_AMT.1 also provides self-testing to ensure the token is operating correctly thus checking for de-activation.

Enable detection of misconfiguration or attack of other SFRs

FPT_AMT.1 provides for user initiated testing to ensure the cryptographic security functions are operational, thus checking for attacks on the FCS class of SFRs. FPT_AMT.1 also provides self testing to ensure the token is operating correctly thus checking for attacks on other SFRs.

8.5 IT Security Function Rationale

This section shows that the iKey IT security functions (ITSF) are *sufficient* and *necessary* to address the security requirements.

8.5.1 Tracing of Security Functional Requirements to IT Security Functions

Table 8-11 shows that each SFR maps to at least one ITSF, and vice versa. The mappings are indicated by the • symbol. Therefore all SFRs and ITSFs are necessary for this ST.

Table 8-11 SFRs mapped to IT Security Functions

TSFs Functional Requirements	Passphrase	Lockout	PKP	Crypto Algs	Initialise	Test	Objects	Epoxy	Electro	RAM	TMGT Mgr
FCS_CKM.1				•		•					
FCS_CKM.2				•		•					
FCS_CKM.4				•		•					
FCS_COP.1			•	•		•					
FDP_ACC.1	•	•	•		•						•
FDP_ACF.1	•	•	•		•						•
FDP_ETC.1							•				•

FDP_ITC.1											•
FIA_AFL.1		•									
FIA_ATD.1	•										
FIA_UAU.2	•										
FMT_MSA.1											•
FMT_MSA.2				•							•
FMT_MSA.3					•						•
FPT_AMT.1						•					
FPT_FLS.1										•	
FPT_PHP.3								•	•		
FPT_SEP.1										•	

8.5.2 Justification that IT Security Functions meet requirements

The rationale provided below justifies why each ITSF is suitable to meet the requirement(s) it is mapped to. Therefore the ITSFs are sufficient to meet the SFRs.

Passphrase

FIA_ATD.1 - The token pass phrase is the attribute that governs access to protected information and operations of the token.

FDP_ACF.1 FDP_ACC.1 - The protected information and operations of the token require the pass phrase to be supplied before access is granted.

FIA_UAU.2 – Successful authentication will grant user access to *protected* TSF- mediated actions, like cryptographic and token operations. This does not include the view only operations and status feedback, which is granted without authenticated. These unauthenticated actions are considered not to be TOE Security Function actions.

Lockout

FIA_AFL.1 – Once the preset limit of failed authentication attempts has been exceeded the token locks itself.

FDP_ACF.1 FDP_ACC.1 – Access to the protected information and operations is denied if the token is locked.

MT_MSA.3 – The lockout counter is always set to a default, secure value at

initialisation.PKP - Private Key Protection

FDP_ACF.1 FDP_ACC.1 – access to usage of private keys on the token is only granted if the correct pass phrase is entered.

FCS_COP.1 – Private keys stored on the token are encrypted. All crypto operations occur upon the token, meaning the private keys never have to leave the token.

Crypto Algs

FCS_CKM.1 – Key generation is supported for the digital signing, key transport and key agreement requirements.

FCS_CKM.2 – The Key agreement requirement uses Diffie-Hellman for key distribution.

FCS_CKM.4 – Once the encryption keys are no longer required, the memory areas used for the keys are filled with null values or “zeroed”.

FCS_COP.1 – The specifications provided in section 6 satisfy the FCS_COP.1 requirements of section 5.

FMT_MSA.2 is indirectly met as only “secure” keys (not weak or semi-weak) are generated.

Initialise

FDP_ACF.1 FDP_ACC.1 – The “Initialise” operation removes all protected information from the token, resulting in security of the pre-existing information being maintained and contributing to the Access Control SFP defined by FDP_ACF.1 and FDP_ACC.1.

FMT_MSA.3 – Any previously stored security attributes and user information, such as private keys, digital certificates, and user pass phrases are always set to a default, secure value at initialisation.

Test

FPT_AMT.1 – The authenticated user can initiate the a correctness test of the underlying cryptographic features (FCS_CKM.1, FCS_CKM.2, FCS_CKM.4 and FCS_COP.1). The TOE conducts self-tests to ensure the correctness of the underlying abstract machine operations like the RAM, processor, ensuring the underlying operating system maintains its integrity, and that the ROM contents have not changed.

Object

FDP_ETC.1 - Information about the objects stored on the iKey token can be displayed and saved to file. The private key(s) and pass phrase is never revealed.

Epoxy

FPT_PHP.3 - The iKey epoxy layer protects the token’s internal devices by resisting against physical tampering.

Electro

FPT_PHP.3 - The iKey is protected against 6kV Electrostatic Discharge (ESD).

RAM

FPT_SEP.1 – The iKey token contains its own RAM, processor and non-volatile memory resulting in the token having its own domain to perform its functions.

FPT_FLS.1 – Extraction of the token from the USB port (its power supply) will result in the token's RAM being cleared. The token's non-volatile memory would not be affected so the token will still be in the secure state it was before power loss.

Token Management (T_MGT)

Behaviour displayed by the Token Management functions are consistent with the token Access Control SFP (FDP_ACC.1, FDP_ACF.1).

FDP_ETC.1 – T_MGT provides a way for a user to obtain information regarding the token. The private key or pass phrase cannot be obtained this way.

FDP_ITC.1 – non-security related information can be imported into the token via T_MGT e.g. the token label. Using Netscape Communicator 4.0 or greater or the Token Utility allows a user to import an existing Digital ID onto the token.

T_MGT allows changes to security settings of the token like the pass phrase. This function satisfies FMT_MSA.1, FMT_MSA.2 and FMT_MSA.3.

The above justification is sufficient to show when a single TSF fully implements a SFR. As some SFRs are met by multiple TSFs, it is necessary to demonstrate that the combined TSFs are suitable to fully meet each SFR. The following shows how SFRs that are mapped to multiple TSFs (in **bold**) are met:

FCS_CKM.1

Crypto Algs implements the cryptographic key generation, while **Test** allows the authenticated user to test that key generation functions properly. Therefore this SFR is fully met by these TSFs.

FCS_CKM.2

Crypto Algs implements the Diffie-Hellman algorithm, while **Test** allows the authenticated user to test that the Diffie-Hellman algorithm functions properly. Therefore this SFR is fully met by these TSFs.

FCS_CKM.4

Crypto Algs zeroises cryptographic keys, while **Test** allows the authenticated user to test that key deletion functions properly. Therefore this SFR is fully met by these TSFs.

FCS_COP.1

Crypto Algs implements the specified cryptographic operations, while **Test** allows the authenticated user to test that these operations function properly. **PKP** ensures that all operations are performed upon the token meaning the private keys never has to leave the token. Therefore this SFR is fully met by these TSFs.

FDP_ACC.1 and FDP_ACF.1

Since these two SFRs specify the Access Control SFP it is appropriate to address them together.

The Access Control SFP specifies that view access is provided to Inactivity Timer Setting, Token Status, Reader Status and Token Information before the user is required to be authenticated. The **Token Management** TSF allows this. For access to other user information the correct pass phrase has to be supplied and verified by the **Passphrase** TSF. This includes access to cryptographic functions requiring the private key (**PKP**). If the token has been locked due to the incorrect pass phrase limit being exceeded, then access to protected information and operations is denied (**Lockout**). Once locked, the only operation allowed is to initialise the token back to default settings (**Initialise**). These TSFs fully implement all aspects of the Access Control SFP and therefore meet FDP_ACC.1 and FDP_ACF.1.

FDP_ETC.1

Information about the objects stored on the iKey token can be displayed (**Object**) and saved to file (**Token Management**). The private key(s) and pass phrase is never revealed by these TSFs. Therefore this SFR is fully met by these TSFs.

FMT_MSA.2

Crypto Algs contributes to this SFR as only “secure” cryptographic keys (not weak or semi-weak) are generated. Also **Token Management** allows secure attributes like the user pass phrase to be changed and allows user certificates to be loaded onto the token. As these are the only secure attributes that can be accepted this SFR is fully met by these TSFs.

FMT_MSA.3

Initialise will remove any previously stored security attributes and user information, such as private keys, digital certificates, and set the user pass phrase to a default value. **Token Management** allows secure attributes like the user pass phrase to be changed and allows user information such as public/private key pairs to be created on the token. Therefore this SFR is fully met by these TSFs.

FPT_PHP.3

This SFR resists two kinds of physical attacks: electrostatic discharge, which is implemented by **Electro;** and physical tampering, which is implemented by **Epoxy.** Therefore this SFR is fully met by these TSFs.

8.5.3 IT Security Function Mutual Support Rationale

The mutual support rationale provided for the SFRs applies to the IT Security Functions. The descriptions of the IT Security Functions provide details that are extra to the SFR definitions. The following analysis addresses how this extra information may effect the mutual support rationale for the SFRs.

Pass Phrase – Details regarding the valid range of allowable pass phrases is provided. The range of 4 – 20 alphanumeric characters does not bypass, deactivate or tamper with any other security function.

Lockout – No extra information provided in description.

Private Key Protection – The private key is stored in encrypted format on the token. This extra information shows the private key has an extra layer of protection providing support for other functions that rely on the private key. This function does not interfere with other security functions.

Crypto Algs – No extra information is provided.

Initialise – The extra information lists the secure values the token will have once it has been initialised. These values do not affect the TOE's mutual support of functionality.

Test – The specific cryptographic tests are listed in the extra information. The areas covered by the POST are listed in the extra information. These tests do not bypass, deactivate or tamper with any other security function.

Object – The description provides an example of the information available, which are not security related and do not affect other security functionality.

Epoxy – The description provides specific details regarding the epoxy layer. This extra information does not undermine any of the other security functions.

Electro – The ESD rating specifies the level the token can withstand. Specifying the rating does not undermine the security of the token.

RAM – Describes what is likely to be in the token's RAM when cleared. This information does not detract from other security functions.

TOE Management – An existing digital ID and private key can be imported onto the token via Netscape or the Token Utility, but once the private key is on board it is protected as same as a private key generated on board the token. This will not bypass or deactivate or tamper with any other security functions.

From the above analysis the developers have determined that the extra information provided in section 6.1 IT Security Functions does not impact on the SFR mutual support rationale in any way. Therefore, the iKey IT Security Functions are mutually supportive of each other.