C-SELECT セキュリティポリシー

Version 1.5
Date 2007 年 4 月 10 日
Author キヤノン(株) PF 技術開発センター

更新履歴

Version	日付	更新内容
1.0	2006/11/01	初版発行.
1.1	2006/12/08	対応アルゴリズムの見直し. 誤記修正.
1.2	2006/12/27	参考文献の更新.
1.3	2007/03/19	ソースコードレビュー時のコメントに対応.
1.4	2007/03/20	テスト環境の詳細化.
1.5	2007/04/10	公開版に対応.

目次

1 はじめに

本セキュリティポリシー（以下, SP と示す）は, キヤノンで開発した C-SELECT と呼ばれるソフトウエアライブラリのセキュリティポリシーであって, C-SELECT が JCMVP 暗号モジュールセキュリティ要件［MSR］のレベル 1 の要求を満たすことを示す. 本 SP は, Nonproprietary なドキュメントである.

1.1 セキュリティポリシーの参照情報

本節は, 本 SP の参照情報を示す.

SP タイトル	C-SELECT セキュリティポリシー
SP バージョン	1.5
SP 発行者	キヤノン(株) PF 技術開発センター
SP 発行日	2007 年 4 月 10 日

1.2 暗号モジュールの参照情報

本節は, 暗号モジュールである C-SELECT の参照情報を示す.

暗号モジュールのタイトル	C-SELECT
暗号モジュールのバージョン	1.0
暗号モジュールの開発者	キヤノン(株)

1.3 暗号モジュールの概要

C-SELECT は, 暗号化/復号, 電子署名生成/署名検証, メッセージダイジェスト, 擬似乱数生成, 鍵共有および鍵包装等のセキュリティ機能を提供するソフトウエアライブラリである. C-SELECT は, セキュリティプロトコル等のさまざまな利用を想定した一般用途向けのソフトウエアライブラリであり, RFC 2628［RFC2628］に基づいた Simple Cryptographic Program Interface (Crypto API) を拡張した拡張 Crypto API を提供する. さらに, さまざまなプラットフォームへの展開を容易にするために C 言語だけで書かれている.

1.4 セキュリティレベル

C-SELECT は, JCMVP 暗号モジュールセキュリティ要件［MSR］のレベル 1 の要求を満たすように設計，実装された. Table 1-1 に, C-SELECT が満たすセキュリティレベルを，セキュリティ分野ごとに示す.

Table 1-1 C-SELECT security levels

セキュリティ分野	レベル
暗号モジュールの仕様	1
暗号モジュールのポート及びインタフェース	1
役割，サービス，及び認証	1
有限状態モデル	1
物理的セキュリティ	1
動作環境	1
暗号鍵管理	1
電磁妨害/電磁両立性（EMI/EMC）	N/A
自己テスト	1
設計保証	1
その他の攻撃への対処	N/A

2 ドキュメントの構成

本 SP は, JCMVP 暗号モジュールセキュリティ要件 [MSR] に関連した暗号モジュールの特徴および機能性を示す. 1 章は序説として, SP の参照情報, 暗号モジュールの参照情報, 暗号モジュールの概要およびセキュリティ要求レベルを示し, 2 章は本 SP の構成を示す. 3 章から 10 章は各セキュリティ分野に対応して, C-SELECT がセキュリティ要求を満たしていることを説明する. 11 章は C-SELECT のセキュアな操作に関して説明し, 12 章は略語を示し, 13 章は参考文献を示す.

3 暗号モジュールの仕様

C-SELECT は「マルチチップスタンドアロン型暗号モジュール」に分類され，以下の汎用のコンピュータプラットフォーム（H/W および OS）上で動作する．

- コンピュータプラットフォーム その 1
  H/W　：PC/AT 互換機（x86 互換機）
  OS　：Microsoft Windows 2000 SP4

- コンピュータプラットフォーム その 2
  H/W　：PC/AT 互換機（x86 互換機）
  OS　：Microsoft Windows XP SP2

- コンピュータプラットフォーム その 3
  H/W　：PC/AT 互換機（x86 互換機）
  OS　：Microsoft Windows Vista

- コンピュータプラットフォーム その 4
  H/W　：PC/AT 互換機（x86 互換機）
  OS　：Linux (Fedora Core 5)

なお, C-SELECT の動作をテストしたコンピュータプラットフォームを表 3-1 に示す.

表 3-1 テストしたコンピュータプラットフォームの一覧

	H/W	OS
テスト プラット フォーム 1	HP Compaq Business Desktop dc7100US ● Intel Cerelon D330 2.66GHz プロセッサ ● 768MB メモリ ● 40GB ハードディスク ● NIC, USB インタフェース	Microsoft Windows 2000 Professional 5.00.2195 SP4
テスト プラット フォーム 2	同上	Microsoft Windows XP Professional Version 2002 SP2

	H/W	OS
テスト プラット フォーム 3	IBM xSeries 100 ● Intel Cerelon D326 2.53GHz プロセッサ ● 512MB メモリ ● 74.5GB ハードディスク ● NIC, USB インタフェース	Microsoft Windows Vista Ultimate
テスト プラット フォーム 4	HP Compaq Business Desktop dc7100US ● Intel Cerelon D330 2.66GHz プロセッサ ● 768MB メモリ ● 40GB ハードディスク ● NIC, USB インタフェース	Fedora Core 5 (Linux)

C-SELECT の物理的な暗号境界は，暗号モジュールが実行する，上に示した特定のコンピュータプラットフォームであり，C-SELECT の論理的な暗号境界は，ダイナミックライブラリを構成するソフトウエアモジュールである．

コンピュータプラットフォーム，OS および C-SELECT を示したブロック図を図 3-1 に示す．

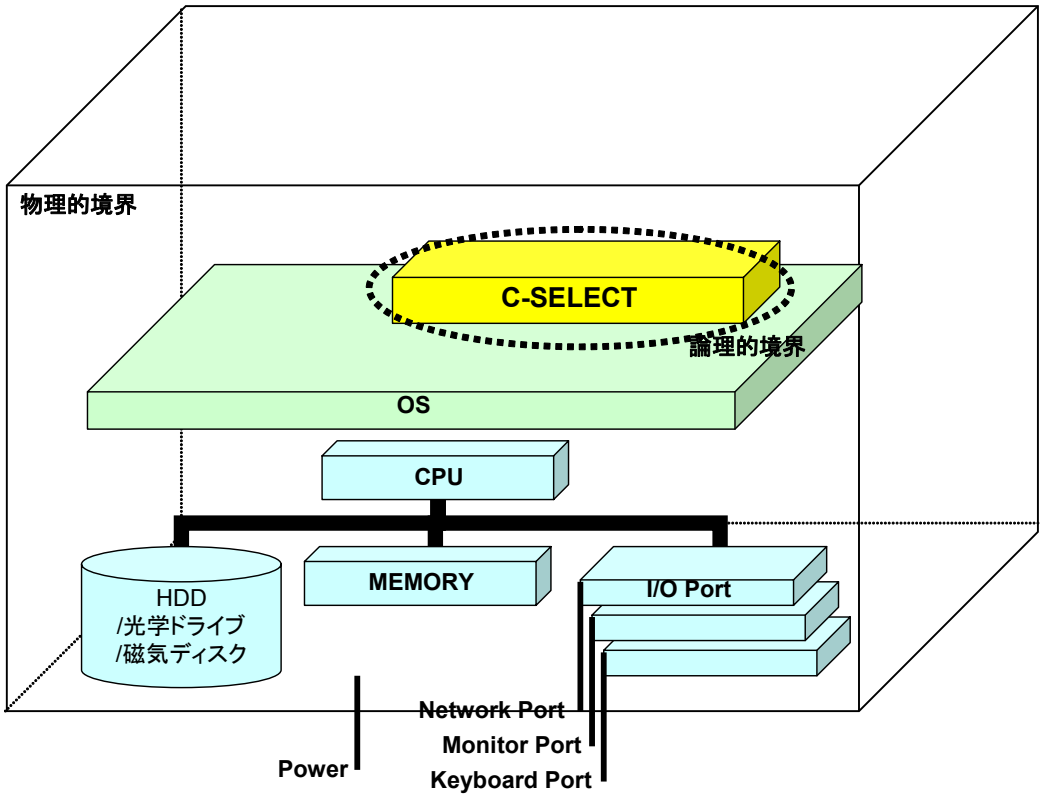

図 3-1C-SELECT のブロック図

4 暗号モジュールのポート及びインタフェース

C-SELECT は「マルチチップスタンドアロン型暗号モジュール」に分類される. C-SELECT の物理的ポートは, コンピュータプラットフォームのインタフェースである, キーボード, マウス, モニタ, CD/DVD ドライブ, フロッピードライブ, シリアルポート, パラレルポート, USB ポート, IEEE1394, LAN 等から構成される. しかし, C-SELECT はソフトウエアライブラリであるため, C-SELECT へのデータの送受信は論理的なインタフェースである API を通して行われる. C-SELECT の API は, RFC 2628 [RFC2628] に基づいた Simple Cryptographic Program Interface (Crypto API) を拡張した拡張 Crypto API である.

「データ入力インタフェース」は, 拡張 Crypto API へ入力されるデータで実現され, 「データ出力インタフェース」は, 拡張 Crypto API から出力されるデータで実現される.

「制御入力インタフェース」は, 拡張 CryptoAPI で実現され, 「状態出力インタフェース」は, 拡張 Crypto API の戻り値および `CryptoFIPSControl()` から出力される状態情報によって実現される.

なお, C-SELECT は特定のコンピュータプラットフォーム上で動作するため, 「電源ポート」はコンピュータプラットフォームへの電源である.

インタフェースの種別, 物理的ポートおよび論理的インタフェースの対応を表 4-1 にまとめる.

表 4-1 インタフェースの対応

I/F の種別	物理的ポート	論理的インタフェース
データ入力 I/F	キーボード等の標準入力ポート	拡張 CryptoAPI に入力されるデータ
データ出力 I/F	モニタ等の標準出力ポート	拡張 CryptoAPI から出力されるデータ
制御入力 I/F	キーボード等の標準入力ポート	拡張 CryptoAPI
状態出力 I/F	モニタ等の標準出力ポート	拡張 CryptoAPI の戻り値, および `CryptoFIPSContorl()` API から出力される状態情報
電源ポート	PC への電源ポート	N/A

5 役割，サービス，及び認証

5.1 役割

C-SELECT は，「ユーザ役割（以下, USER と示す)」と「クリプトオフィサ役割（以下, CO と示す)」の役割をサポートする．なお, C-SELECT は, JCMVP 暗号モジュールセキュリティ要件［MSR］のレベル 1 のセキュリティ要求だけを満たすため，ユーザの識別および認証機能をサポートしない．Table 5-1 に役割とサービスの対応を示す．なお, C-SELECT は, メンテナンスサービスを提供しないため，メンテナンス役割をサポートしない．

Table 5-1 役割とサービスの対応

役割	サービス
USER	USER は，C-SELECT が提供する全てのサービスに，API を介してアクセスできる．
CO	CO は, USER 同様に, C-SELECT が提供する全てのサービスに, API を介してアクセスできるため，暗号サービスの利用時において USER とみなす. CO はさらに, C-SELECT をコンピュータプラットフォーム上にインストールできる．なお，CO は，C-SELECT の鍵等の CSP への特別な権限を有さない．

5.2 サービス

C-SELECT の提供する暗号サービスを示す．C-SELECT は，異なる暗号サービスを提供する暗号モジュールの動作モード（以下，モジュールのモードと示す）を複数有する．Table 5-2 に C-SELECT が有するモジュールのモードを示す．

Table 5-2 「モジュールのモード」一覧

モード名称	概要
JCMVP モード	JCMVP で承認されたセキュリティ機能のみを採用したモード
CMVP モード	CMVP[1]で承認されたセキュリティ機能のみを採用したモード
NonCMVP モード	JCMVP または CMVP で非承認のセキュリティ機能を含み, JCMVP モードでも CMVP モードでもないモード

[1] 北米で実施されている「暗号モジュール試験及び認証制度」．暗号アルゴリズムが，ソフトウエアやハードウエアへ適切に実装されているか否かの試験等を行う制度．

さらに，サービスの詳細として，モジュールのモードごとに C-SELECT が提供する暗号サービスを示す. JCMVP モードが提供する暗号サービスを Table 5-3 に，CMVP モードが提供する暗号サービスを Table 5-4 に, NonCMVP モードが提供する暗号サービスを Table 5-5 にそれぞれ示す.

Table 5-3 JCMVP モードの提供する暗号サービスの一覧

サービスのタイプ	アルゴリズム	仕様	備考
秘密鍵暗号	TDES (3 keys)	[SP800-67]	共通鍵暗号の動作モード（以下，BC モードと示す）：ECB, CBC, CFB(1), CFB(8), CFB(64), OFB TDES の動作モード：3 keys EDE
	AES	[FIPS197]	BC モード：ECB, CBC, CTR 鍵長：128, 192, 256 bits
	Camellia	[Camellia]	BC モード：ECB, CBC 鍵長：128, 192, 256 bits
デジタル署名	DSA	[ANSI X9.30]	-
	ECDSA	[SEC1]	-
	RSASSA-PKCS1-v1_5	[PKCS#1]	-
ハッシュ関数	SHA	[FIPS180-2]	サポートしている Hash アルゴリズム：SHA-1, SHA-256, SHA-384, SHA-512
メッセージ認証	HMAC	[FIPS198]	サポートしている Hash アルゴリズム：SHA-1, SHA-256, SHA-384, SHA-512
乱数生成	PRNG based on SHA-1 for general purpose in FIPS 186-2 (+ change notice 1) Appendix 3.1	[FIPS186-2]	-
鍵共有	DH	[ANSI X9.42]	-
	ECDH	[SEC1]	-
鍵包装	RSA-OAEP	[PKCS#1]	-

サービスのタイプ	アルゴリズム	仕様	備考
	RSAES-PKCS1-v1_5	[PKCS#1]	-
自己テスト	-	-	-
状態取得	-	-	-

なお, BC モードは, [SP800-38A] に準ずる. また, 鍵共有 DH および ECDH に関しては, SSL Ver.3.1, TLS 及び EAP-TLS, IPSEC, SSH Ver.2 等の鍵確立プロトコルとして使用する場合に限り, 認証の対象となる.

Table 5-4 CMVP モードの提供する暗号サービスの一覧

サービスのタイプ	アルゴリズム	仕様	備考
秘密鍵暗号	TDES	[SP800-67]	BC モード : ECB, CBC, CFB(1), CFB(8), CFB(64), OFB TDES の動作モード : 2 keys EDE, 3 keys EDE
	AES	[FIPS197]	BC モード : ECB, CBC, CTR, CCM 鍵長 : 128, 192, 256 bits
デジタル署名	DSA	[FIPS186-2]	-
	ECDSA	[FIPS186-2]	-
	RSASSA-PKCS1-v1_5	[PKCS#1]	-
ハッシュ関数	SHA	[FIPS180-2]	サポートしている Hash アルゴリズム : SHA-1, SHA-224, SHA-256, SHA-384, SHA-512
メッセージ認証	HMAC	[FIPS198]	サポートしている Hash アルゴリズム : SHA-1, SHA-224, SHA-256, SHA-384, SHA-512
	CMAC	[SP800-38B]	サポートしている暗号アルゴリズム : TDES, AES

サービスのタイプ	アルゴリズム	仕様	備考
乱数生成	PRNG based on SHA-1 for general purpose in FIPS 186-2 (+ change notice 1) Appendix 3.1	[FIPS186-2]	-
鍵共有	DH	[ANSI X9.42]	-
	ECDH	[SEC1]	-
鍵包装	RSA-OAEP	[PKCS#1]	
	RSAES-PKCS1-v1_5	[PKCS#1]	
	AES Key wrap	[AESKW]	
自己テスト	-	-	-
状態取得	-	-	-

Table 5-5 NonCMVP モードの提供する暗号サービスの一覧

サービスのタイプ	アルゴリズム	仕様	備考
秘密鍵暗号	TDES	[SP800-67]	BC モード：ECB, CBC, CFB(1), CFB(8), CFB(64), OFB TDES の動作モード：2 keys EDE, 3 keys EDE
	DES	[FIPS46-3]	BC モード：ECB, CBC, CFB(1), CFB(8), CFB(64), OFB
	RC2	[RFC2268]	BC モード：ECB, CBC, CFB(1), CFB(8), CFB(64), OFB
	AES	[FIPS197]	BC モード：ECB, CBC, CTR, CCM 鍵長：128, 192, 256 bits
	Camellia	[Camellia]	BC モード：ECB, CBC 鍵長：128, 192, 256 bits
公開鍵暗号	RSA-OAEP	[PKCS#1]	-
	RSAES-PKCS1-v1_5	[PKCS#1]	-

サービスのタイプ	アルゴリズム	仕様	備考
	Elgamal	[Elgamal]	-
デジタル署名	DSA	[FIPS186-2]	-
	ECDSA	[SEC1]	-
	RSASSA-PKCS1-v1_5	[PKCS#1]	-
ハッシュ関数	SHA	[FIPS180-2]	サポートしている Hash アルゴリズム：SHA-1, SHA-224, SHA-256, SHA-384, SHA-512
	MD2	[RFC1319]	-
	MD4	[RFC1320]	-
	MD5	[RFC1321]	-
メッセージ認証	HMAC	[FIPS198]	サポートしている Hash アルゴリズム：SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, MD5
	CMAC	[SP800-67]	サポートしている秘密鍵暗号アルゴリズム：TDES, AES
	DESMAC	[FIPS113]	-
乱数生成	PRNG based on SHA-1 for general purpose in FIPS 186-2 (+ change notice 1) Appendix 3.1	[FIPS186-2]	-
	PRNG based on DES for general purpose in FIPS 186-2 (+ change notice 1) Appendix 3.1	[FIPS186-2]	-
鍵共有	DH	[ANSI X9.42]	-
	ECDH	[SEC1]	-
鍵包装	RSA-OAEP	[PKCS#1]	-
	RSAES-PKCS1-v1_5	[PKCS#1]	-
	TDES Key wrap	[RFC3217]	-
	AES Key wrap	[AESKW]	-

なお, BC モードは, [SP800-38A] および [SP800-38C] に準ずる.

5.3 暗号モジュールの動作モード

C-SELECT のモジュールのモードは，`CryptoFIPSControl()` を用いて制御する．モジュールのモードを指定可能なタイミングは，C-SELECT の起動時，つまり C-SELECT のロード時だけである．一旦モジュールのモードを設定した後は，異なるモジュールのモードへ変更することはできない．異なるモジュールのモードへ変更するためには，C-SELECT をアンロードし，C-SELECT をロードして，`CryptoFIPSControl()` を用いてモジュールのモードを設定しなければならない．

6 物理的セキュリティ

C-SELECT は，ソフトウエアライブラリであって，PC/AT 互換機上で動作させて，テストした．PC/AT 互換機であるコンピュータプラットフォームと C-SELECT は，製品グレードコンポーネントであって，製品グレードの囲いを実現した「マルチチップスタンドアロン型暗号モジュール」を構成する．

7 動作環境

C-SELECT はダイナミックライブラリであるため，プログラムファイルが仮想アドレス空間へロードされるタイミングで，仮想アドレス空間へロードされ，コンストラクタ関数が実行される．

OS は，プログラム（プロセス）ごとに異なる仮想アドレス空間を用意し，プログラムファイルを仮想アドレス空間へロードし，実行する．同時に，ダイナミックライブラリである C-SELECT も仮想アドレス空間へロードする．よって，C-SELECT は，OS によって用意された独立した仮想アドレス空間で実行する．

C-SELECT の動作モードは，JCMVP 暗号モジュールセキュリティ要件［MSR］で規定された，単一オペレータ動作モードに限定されるものとする．また，仮想アドレス空間へロードされた C-SELECT は，プロセス間通信によって，他のプロセスと情報を交換しないものとする．

8 暗号鍵管理

Table 8-1 に C-SELECT の JCMVP モードで取り扱う暗号鍵等の CSP, および CSP に対する操作種別を, 暗号アルゴリズムごとに示す.

Table 8-1 JCMVP モードにおける CSP の種別と操作種別

サービスのタイプ	アルゴリズム	CSP 識別	操作種別
秘密鍵暗号	TDES (3 keys)	秘密鍵	R, W
	AES	秘密鍵	R, W
	Camellia	秘密鍵	R, W
デジタル署名	DSA	プライベート鍵	R, W
	ECDSA	プライベート鍵	R, W
	RSASSA-PKCS1-v1_5	プライベート鍵	R, W
ハッシュ関数	SHA	-	-
メッセージ認証	HMAC	秘密鍵	R, W
乱数生成	PRNG based on SHA-1 for general purpose in FIPS 186-2 (+ change notice 1) Appendix 3.1	シード鍵	R, W
鍵共有	DH	プライベート鍵	R, W
	ECDH	プライベート鍵	R, W
鍵包装	RSA-OAEP	プライベート鍵	R, W
	RSAES-PKCS1-v1_5	プライベート鍵	R, W
自己テスト	-	-	-
状態取得	-	-	-

8.1 鍵の生成

C-SELECT は, Table 8-1 の CSP 識別を有する暗号アルゴリズムに対して, 以下に従って鍵を生成する.

- 共通鍵暗号（TDES, AES, Camellia）およびメッセージ認証は, C-SELECT が有する擬似乱数生成によって鍵を生成するものとする.
- DSA は, C-SELECT で生成した擬似乱数生成を利用して, [ANSI X9.30] にしたがって鍵を生成するものとする.
- ECDSA は, C-SELECT で生成した擬似乱数生成を利用して, [SEC1] にしたがって

鍵を生成するものとする.

- RSAES-PKCS1-v1_5, RSASSA-PKCS1-v1_5 および RSA-OAEP は, C-SELECT で生成した擬似乱数生成を利用して, [PKCS#1] にしたがって鍵を生成するものとする.
- DH は, C-SELECT で生成した擬似乱数生成を利用して, [ANSI X9.42] にしたがって鍵を生成するものとする.
- ECDH は, C-SELECT で生成した擬似乱数生成を利用して, [SEC1] にしたがって鍵を生成するものとする.

なお, C-SELECT は, PRNG のシード鍵を内部で生成しない.

8.2 鍵の入力および出力

C-SELECT の物理的境界に対する CSP の入出力は, キーボード等の標準入力ポートまたはモニタ等の標準出力ポートを通して, 電子的に行われるものとする. また, C-SELECT の論理的境界に対する CSP の入出力は, API を通して, 電子的に行われるものとする.

8.3 鍵の破棄

C-SELECT は, 論理的境界内で不要になった CSP をゼロ化して破棄する. C-SELECT が論理的境界内で CSP を破棄するタイミングを下記に示す.

1. `CryptoClose()` を介して, CSP を明示的に破棄した場合.
2. C-SELECT を利用しているプログラムが C-SELECT をアンロードした場合.
3. C-SELECT が 9 章で示す自己テストに失敗した場合.

なお, 物理的境界内で不要になった CSP は, HDD 等のコンピュータプラットフォームに接続されている不揮発性ストレージをフォーマットすることで破棄されるものとする.

8.4 鍵の格納

C-SELECT は, 不揮発性メモリを持たず, 鍵を保持しない.

9 自己テスト

C-SELECT は, パワーアップ自己テストおよび条件自己テストの機能を有する. 9.1 節にパワーアップ自己テストに関して示し, 9.2 節に条件自己テストに関して示す.

9.1 パワーアップ自己テスト

C-SELECT はダイナミックリンクライブラリとして提供される. したがって, プログラムが C-SELECT をロードしたときに, 自動的にパワーアップ自己テストが実行される. パワーアップ自己テストの結果は,「状態出力インタフェース」である `CryptoFIPSControl()` によってモジュールから出力できる. パワーアップ自己テストの失敗時, C-SELECT は `CryptoFIPSControl()` を介した状態出力だけしかできない Loading Error 状態へ遷移する. Loading Error 状態からの回復は, C-SELECT をアンロードし, 再度 C-SELECT をロードする方法だけである.
また, C-SELECT は, USER により, `CryptoFIPSControl()` を介してオンデマンドでパワーアップ自己テストを実行することもできる. オンデマンドでのパワーアップ自己テストの結果も,「状態出力インタフェース」である `CryptoFIPSControl()` によってモジュールから出力できる. なおオンデマンドでのパワーアップ自己テストの失敗時, C-SELECT は `CryptoFIPSControl()`, `CryptoClose()`, `CryptoPluginFini()`, および `CryptoFree()` を介した制御入力または状態出力しかできない Operational Error 状態へ遷移する. Operational Error 状態から回復する方法は, (1) C-SELECT をアンロードし, 再度 C-SELECT をロードする方法, (2) ソフトウエアリセットを実行する方法, の 2 通りの方法で回復することができる.
C-SELECT は, パワーアップ自己テストとして, 暗号アルゴリズムテストとソフトウエア完全性テストを実装しており, 以下に示す全ての暗号アルゴリズムテストおよびソフトウエア完全性テストにパスした場合に, パワーアップ自己テストにパスしたと判断する. なお, C-SELECT は重要機能テストを実装していない.

9.1.1 暗号アルゴリズムテスト

C-SELECT は, 暗号アルゴリズムテストとして, 既知解テスト（以下, KAT と示す）または鍵ペア整合性テスト（以下, PWCT と示す）を実装している. Table 9-1 に, C-SELECT が実装している暗号アルゴリズムテストの種別を暗号アルゴリズムごとに示す.

Table 9-1 JCMVP モードにおける暗号アルゴリズムテストの一覧

サービスのタイプ	アルゴリズム	暗号アルゴリズムテストの種別
秘密鍵暗号	TDES (3 keys)	KAT
	AES	KAT
	Camellia	KAT
デジタル署名	DSA	KAT
	ECDSA	KAT
	RSASSA-PKCS1-v1_5	PWCT
ハッシュ関数	SHA	KAT
メッセージ認証	HMAC	N/A
乱数生成	PRNG based on SHA-1 for general purpose in FIPS 186-2 (+ change notice 1) Appendix 3.1	KAT
鍵共有	DH	KAT
	ECDH	KAT
鍵包装	RSA-OAEP	PWCT
	RSAES-PKCS1-v1_5	PWCT

本モジュールはパワーアップ自己テストとしてソフトウエア完全性テストを実施しており，ソフトウエア完全性テストは HMAC-SHA1 によって実現されている．したがって，SHA 系暗号アルゴリズムテストは，SHA-256 および SHA-512 に対してだけ実施しており，SHA-1 に対しては実施しない．

9.1.2 ソフトウエア完全性テスト

C-SELECT は，HMAC-SHA1 によって，モジュールの完全性テストを実現している．HMAC-SHA1 の秘密鍵は C-SELECT 内部に保持し，HMAC 値は異なる独立したファイルに保持している．C-SELECT は，自身であるダイナミックリンクライブラリを対象に HMAC-SHA1 を計算し，独立ファイルに保持している HMAC 値と比較し，一致するか否かによって完全性を検証する．

9.2 条件自己テスト

C-SELECT は通常動作中に，条件自己テストとして，以下の鍵ペア整合性テストと連続乱数生成器テストを実行する．これらのテストが実行されるタイミングは，それぞれ 9.2.1 節または 9.2.2 節に示す．条件自己テストの結果は，パワーアップ自己テストの結果同様，「状態出力インタフェース」である `CryptoFIPSControl()` によってモジュールから出力できる．条件自己テストの失敗時，C-SELECT は `CryptoFIPSControl()`, `CryptoClose()`, `CryptoPluginFini()`，および `CryptoFree()` を介した制御入力または状態出力しかできない Operational Error 状態へ遷移する．Operational Error 状態から回復する方法は，(1) C-SELECT をアンロードし，再度 C-SELECT をロードする方法，(2) ソフトウエアリセットを実行する方法，の 2 通りの方法で回復することができる．
C-SELECT は，条件自己テストとして，鍵ペア整合性テストと連続乱数生成器テストだけを実装している．

9.2.1 鍵ペア整合性テスト

C-SELECT は，Table 9-2 に示す暗号アルゴリズムにおいて鍵ペア整合性テストを実装している．鍵ペア整合性テストは，8.1 節で示した鍵生成ごとに実行し，生成した鍵を検証する．なお，DH および ECDH の鍵ペア整合性テストは実施しない．

Table 9-2 条件自己テストにおける PWCT テストの一覧

サービスのタイプ	アルゴリズム
デジタル署名	DSA
	ECDSA
	RSASSA-PKCS1-v1_5
鍵包装	RSA-OAEP
	RSAES-PKCS1-v1_5

9.2.2 連続乱数生成器テスト

C-SELECT は，乱数生成サービスにおいて連続乱数生成器テストを実装している．連続乱数生成器テストは［FIPS186-2］に基づいて生成した乱数に対してテストを実施するため，乱数生成サービスの実行のごとに実行する．

10 その他の攻撃への対処

C-SELECT は，その他の攻撃へ対処しない.

11 Secure operation

11.1 インストール手順

インストール手順の概要を示す.

C-SELECT が動作する OS は, パスワードで利用者を識別・認証する機能を有する. CO および USER の認証には, 当該識別・認証機能を利用する. このため, CO は, OS に CO および USER 権限を有するアカウントをそれぞれ作成する. 権限は, アカウントごとに付与してもよいし, アカウントが属するグループごとに付与してもよい. なお, OS の管理者と CO を同じにする必要はない.

C-SELECT はインストーラを有さない. そこで, CO は, インストール後のモジュールが改変されることを防止するために配布されたファイルを適切なフォルダへコピーし, コピーしたファイルに対して適切なアクセス権限 (e.g. CO に対してリード・ライト権限, USER に対してリード権限) を設定する.

アカウント管理方法およびアクセス権限管理方法は, それぞれの環境に応じた手順に従う.

11.1.1 Windows 環境

Microsoft Windows 2000 SP4, Microsoft Windows XP SP2 および Windows Vista の環境において, 開発者は, CO へ crypto.dll, crypto.lib, crypto.dll.sha1, api_def.h, api_prt.h, crypt_com.h の 6 つのファイルを配布する. CO は, これらのファイルを任意のフォルダにコピーすることができる.

11.1.2 Linux 環境

Linux の環境において, 開発者は, CO へ libcrypto.so.0.0, libcrypto.so.0.0.sha1, api_def.h, api_prt.h, crypt_com.h の 5 つのファイルを配布する. CO は, 一般的には, libcrypto.so.0.0 および libcrypto.so.0.0.sha1 を /usr/local/lib へ, ヘッダーファイルを /usr/local/include へコピーする.

11.2 利用手順

C-SELECT を利用するプログラムの開発者は, C-SELECT をダイナミックライブラリとして利用し, かつ, C-SELECT のロード後に `CryptoFIPSControl()` をコールし JCMVP モードに設定することにより, JCMVP モードで C-SELECT を利用することができる. モジ

ュールのモードを指定可能なタイミングは, C-SELECT の起動時, つまり C-SELECT のロード時だけである. 一旦 C-SELECT を特定のモジュールのモードに設定した後は, 異なるモジュールのモードへ変更することはできない. 異なるモジュールのモードへ変更するためには, C-SELECT をアンロードし, C-SELECT をロードして, `CryptoFIPSControl()` を用いてモジュールのモードを設定しなければならない. なお, C-SELECT のロード後, `CryptoFIPSControl()` をコールすることにより, モジュールのモードを確認することができる.

12 略語

Acronyms	Definitions
SP	Security Policy
API	Application Programming Interface
RFC	Request for Comments
EMI	Electromagnetic interference
EMC	Electromagnetic compatibility
JCMVP	Japan Cryptographic Module Validation Program
CMVP	Cryptographic Module Validation Program
IPA	Information-technology Promotion Agency
CO	Crypto Officer
OS	Operating System

13 参考文献

[MSR] (独) 情報処理推進機構, "JCMVP 暗号モジュールセキュリティ要件 MSR-01", 平成 18 年 10 月 16 日

[MTR] (独) 情報処理推進機構, "JCMVP 暗号モジュール試験要件 MTR-01", 平成 18 年 10 月 16 日

[ATR] (独) 情報処理推進機構, "JCMVP 暗号モジュール試験要件 ATR-01", 平成 18 年 10 月 16 日

[ANSI X9.30] ANSI X9.30-1997, "Public Key Cryptography for the Financial Services Industry: Part 1: The Digital Signature Algorithm (DSA)", January 30, 1997

[ANSI X9.42] ANSI X9.42-2001, "Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography", March 9, 2001

[FIPS46-3] FIPS Publication 46-3, "DATA ENCRYPTION STANDARD (DES)", October 25, 1999

[FIPS113] FIPS Publication 113, "COMPUTER DATA AUTHENTICATION", May 30, 1985

[FIPS186-2] FIPS Publication 186-2 (+Change Notice 1), "DIGITAL SIGNATURE STANDARD (DSS)", October 5, 2001

[FIPS180-2] FIPS Publication 180-2 (+Change Notice to include SHA-224), "SECURE HASH STANDARD", August 1, 2002

[FIPS197] FIPS Publication 197, "ADVANCED ENCRYPTION STANDARD (AES)", November 26, 2001

[SP800-38A] NIST Special Publication 800-38A, "Recommendation for Block Cipher Modes of Operation Methods and Techniques", December 2001

[SP800-38B] NIST Special Publication 800-38B, "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication", May 2005

[SP800-38C] NIST Special Publication 800-38C, "Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality", May 2004

[SP800-67] NIST Special Publication 800-67, "Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher", May 2004

[AESKW] "AES Key Wrap Specification", November 16, 2001

[RFC1319] B. Kaliski, "The MD2 Message-Digest Algorithm", RFC 1319, April 1992

[RFC1320] R. Rivest, "The MD4 Message-Digest Algorithm", RFC 1320, April 1992

[RFC1321] R. Rivest, "The MD5 Message-Digest Algorithm", RFC 1321, April 1992

[RFC2268] R. Rivest, "A Description of the RC2(r) Encryption Algorithm", RFC 2268, March 1998

[RFC2628] V. Smyslov, "Simple Cryptographic Program Interface (Crypto API)", RFC 2628, June 1999

[RFC3217] R. Housley, "Triple-DES and RC2 Key Wrapping", RFC 3217, December 2001

[PKCS#1] RSA Laboratories PKCS #1 v2.1, "RSA Cryptography Standard", June 14, 2002

[Camellia] "128 ビットブロック暗号 Camellia アルゴリズム仕様書" ,第 2 版, September 26, 2001

[Elgamal] Daniel Bleichenbacher, “Generating ElGamal Signatures Without Knowing the Secret Key”, Advances in Cryptology - EUROCRYPT '96, Lecture Notes in Computer Science, Springer-Verlag, vol. 1070, pp. 10-18, May 1996

[SEC1] "SEC 1: Elliptic Curve Cryptography", Ver.1.0, September 20, 2000

