# 平成11年度

# スマートカードの安全性に関する調査

# 調査報告書

平成12年2月29日

情報処理振興事業協会

# 目 次

# 要旨


IC カードは，IC チップ中に CPU を内蔵しているもの (CPU カード) と内蔵しないもの (メモリカード) とに大別され，演算機能や判断機能を有する CPU カードがスマートカードと呼ばれる．このスマートカードでは暗号処理で用いる鍵のような秘密情報が IC チップのメモリに記録され，それを CPU がアクセス制御することで，容易な読み出しを不可能とする耐タンパー性が実現されている．そのためスマートカードは従来よりも安全性の高いクレジットカードや電子マネーの実現手段であり，情報セキュリティのキーデバイスの一つとして期待が寄せられている．

ところが近年，故障利用解析，タイミング解析，電力解析などの各種解析法が提案されたことにより，メモリ上の情報は読み出せないとする従来の考えが覆され，情報セキュリティのキーデバイスとしてのスマートカードに対する期待を崩しかねない状況にある．

本調査では，学会を中心に発表されている攻撃の立場および防御の立場からの解析法に関する研究を調査，整理し，解析法に対する対策をまとめる．これによって，スマートカードといえど万全なセキュリティを保障し得るわけではないという状況を浮き彫りにし，現存する脅威に対する注意を利用者へ促すと共に，これら解析法に対しても耐タンパー性を維持し得るスマートカード開発のための支援レポートとすることを目的としている．

本調査報告書は４部構成であり，それぞれの概要は以下の通りである．

**I. はじめに**

本調査の背景と目的を簡単にまとめる．

**II. スマートカードの概要**

スマートカードの種類，情報化社会における機能と役割について説明し，利用状況に関する国内および海外動向，さらに今後の展望をまとめる．

またスマートカードの規格・標準として，国際規格 ISO 7816，国内規格 JIC-SAP，および，業界標準である EMV，全銀協，JavaCard Forum，Visa Open Platform，MULTOS，Smart Card for Windows，Open Card Framework，PC Smart Card (PC/SC) の計 10 規格について，それらの位置付け，ねらい，現状，相互関係を整理する．

これらの規格・仕様の主な相互関係は，まずスマートカードの物理的・機能的条件に関する国際規格 ISO 7816 をベースとして，その上に金融系スマートカードの国際標準 EMV 仕様が基本 OS として位置付けられる．さらに多目的スマートカード利用の OS として，JavaCard，Visa Open Platform，MULTOS が EMV に準拠して制定されている．またスマートカードの PC での利用を目的として制定された規格が，Smart Card for Windows，Open Card Framework，

PC Smart Card (PC/SC) である．

## III. 耐タンパー性に対する解析法

本報告書ではスマートカードに対する解析法として，破壊型解析法であるプローブ解析，非破壊型解析法である故障利用解析，タイミング解析，電力解析に対して，それぞれの解析原理，各種暗号アルゴリズムに対する適用例，攻撃の実現可能性と対策をまとめる．

まず破壊型解析法は，スマートカードおよびICチップに物理的変形を加える攻撃法であり，チップ内部を直接観測できるという意味で非常に強力な解析手法といえる．プロセッサの回路構成，メモリの内容，バス上のデータの読み取りや，回路自体の改竄等が可能であり，対策としては，チップ表面の加工，読み取り困難なメモリの採用，解析センサの採用等が重要となる．

また代表的な破壊型解析法であるプローブ解析は，ICの内部バスに直接プローブを当ててレジスタのビット値を観測することで秘密情報を得る解析法であり，このようなアクセスが可能でさえあれば，実装されたほとんど全ての暗号方式を解析することができる．ただし攻撃の実行にはデバイスと暗号方式に対する詳細な知識に加え，技術的にも習熟度が要求されるため，攻撃の実現可能性は高いとは言えない．プローブ解析に対する耐タンパー技術としては，周波数，電圧，温度検知回路などのICチップへの標準装備が進められている．

次に非破壊型解析法は，攻撃者が人為的に引き起こした計算誤りを利用する故障利用解析と，設計者の予期しなかった情報 (side-channel information) を利用するタイミング解析および電力解析とに分類される．

故障利用解析は，一過性の故障ないし他の機能に影響を与えない範囲の限定的な障害をスマートカードに与え，攻撃者が意図した異常な処理による出力と正常出力とを比較することで秘密情報を推測する解析法である．故障の発生には，放射線やレーザー照射，高電圧印加など専用の設備を必要とする．故障利用解析の対策としてはプローブ解析への対策と同様なアプローチが有効であり，例えば周波数，電圧，温度検知回路の搭載による適正動作範囲外でのスマートカード使用の防止が重要である．

タイミング解析は，秘密情報に依存して暗号化の処理時間が異なる場合に統計的解析を用いて秘密情報を推測する手法であり，暗号アルゴリズム自体ではなく不注意な実装法に対する攻撃法である．例えばデータのビット値が0か1かで処理に分岐が生じるような場合がタイミング解析の対象となる．有効な対策としては，処理時間を一定化する実装アルゴリズムの採用，ランダムな遅延による処理時間のカムフラージュなどがある．

暗号処理中の消費電力を観測する電力解析は，消費電力スペクトルに対して直接的 (視覚的) な解析を行う単純電力解析と，統計的手法を用いるより強力な

電力差分解析とに大別される．特に電力差分解析では，測定したスペクトルの平均をとってノイズや測定誤差の影響を減らし，秘密情報に依存した消費電力の変化のみを取り出すことで効率的な解析を行う．電力解析に対する対策は，ハードウェアレベル，ソフトウェアレベルなど各設計レベルでとらえることができるが，情報の外部への漏洩を削減することを目的としたハードウェアレベルよりも，ハードウェアからの情報のリークは不可避なものとの仮定に立ったソフトウェアおよびアルゴリズムレベルでの対策が電力解析に対してより効果的とされる．

**IV. 安全性評価の標準化動向**

安全性評価に関する標準化動向として，暗号モジュールの安全性に関する米国政府調達基準 FIPS 140-1 を調査し，暗号モジュールが満たすべきセキュリティ要件 11 項目とセキュリティレベル 4 段階の概要を整理する．

また現在改定作業が進められている FIPS 140-2 について，特に電力解析のような比較的新しい攻撃法に対する対応をまとめる．電力解析に代表される強力な解析法が提案されたことを受けて，FIPS 140-2 では電力解析，タイミング解析，故障利用解析等による攻撃の軽減という新たなセキュリティ要件が規定され，これら解析法に対する注意を促している．しかしこれらの攻撃に対して決め手となるような対策は明示されておらず，今後の技術開発動向に期待する形となっている．

以上述べてきたように，近年提案された各種解析法はスマートカードに対する脅威であり，それらを軽視することは許されない．その一方で解析に対する対処法も多方面から議論・提案されつつあり，有効な対策を施すことで解析法を万能にさせないことも可能になっている．本調査で報告した解析法に対しても耐タンパー性を維持し得るスマートカードの開発のため，今後のさらなる技術開発が重要である．

# 第I部
# はじめに

## I.1　背景

1974年にフランスのロラン・モレロ氏により世界で初めてスマートカードが提案されて以来，スマートカードは情報化社会に不可欠な存在になりつつある．例えば，鉄道の改札を通過する際やビルへ入る際にはスマートカードは身分証明書として利用することができ，電子商取引においては貨幣価値やポイントを保持するための財布として利用することができる．さらに，これら複数の異なった機能を一枚のカードで提供することができる．従来より広く用いられている磁気カードと比べスマートカードには以下のような利点がある．

- アプリケーションに応じた柔軟な処理が可能
- スマートカード内のメモリに外部端末が直接アクセスできない

これらはいずれもスマートカード内にCPUが存在することに起因する．そのため，使い勝手や内部情報を読み出す困難性(耐タンパー性)は磁気カードに対して格段に向上している．

## I.2　調査の目的

スマートカードは携帯性にも優れ利用が簡単なため幅広く普及し始めているが，現在市場に出回っているすべてのスマートカードが如何なる攻撃に対しても内部情報を外部に漏らさない構造を持っているかというと，必ずしもそうとは言えない．スマートカード内のデータを不正に読み出す方法がいくつか報告されており，実際にデータを取り出した事例も報告されている．

本調査では，スマートカード内のデータを不正に読み出す方法，その対策方法，ならびに各解析方法に対する耐性の評価方法について調査する．加えて，スマートカードの規格や安全性評価方法の標準化動向に関する調査も行い整理する．

# 第II部
# スマートカードの概要

## II.1　機能と役割

本章ではICカードの種類について説明するとともに，情報化社会におけるスマートカードの機能と役割，および現在の利用状況の概要ならびに今後の展望についてまとめる [IC97, IC99a, IC99b, IC99c, IC99d, IC99e, YK99].

### II.1.1　ICカードの種類

ICチップが埋め込まれたICカードは演算やデータ処理等の高度な判断機能を備えており，記憶容量に関しても磁気カードと比較して数倍から数百倍の大容量を有している．

ICカードは，まずCPUの有無からメモリカードとCPUカードとに分類される．メモリカードには単にデータの記録を目的としたタイプと，メモリへのアクセスを制御するためのセキュリティ回路を備えたプロテクティッド・メモリカードとがあり，欧州での使い捨てプリペイドカードはこのプロテクティッド・メモリカードに相当する．一方のCPUカードは演算機能や判断機能を有し，情報を記録するだけの受動的メディアに対する能動的メディアとなる．このCPUカードが本報告書で調査対象としているスマートカードである．またCPUカードの内，暗号の高速処理用のコプロセッサを搭載したものをクリプトカードとも言う．メモリカードで先行する欧州に対し，日本のベンダーはCPUカードの製造販売を主流としている．なおメモリカードおよびCPUカード以外の分類として，表示機能，手動入力機能，磁気カードとの兼用機能などを備えた多機能カードがある．

次にインターフェースの違いからICカードを接触型カードと非接触型カードとに分類することができる．カード端末機との接続のための外部端子を有する接触型カードとは異なり，非接触型カードは内部のアンテナを通して電力供給やデータの読み書きを行う．メモリカードが主体となる非接触型タイプは通信距離によって表II.1に示したような4つのタイプに分類され，それぞれの通信距離は，密着型が2mm，近接型が10cm，近傍型が70cm，遠隔型が数m程度までである．また近傍型までが電磁誘導方式，遠隔型がマイクロ波方式となっている．電子乗車券やテレホンカードとして実用化が進んでいるのが近接型タイプである．なお接触型，非接触型の両方の機能を兼ね備えたものはコンビネーションカードないしハイブリッドカードと呼ばれ，例えば電子マネー等の決済分野では接触型で，入退出管理に関しては非接触型でという用途が期待されている．

以上を整理したものを表 II.1 に示す.

表 II.1: IC カードの分類

| | |
|---|---|
| CPU の有無 | メモリカード<br>CPU カード (スマートカード) |
| インターフェースの違い | 接触型カード<br>非接触型カード　密着型<br>近接型<br>近傍型<br>遠隔型 |

### II.1.2　スマートカードの機能と応用分野

スマートカードの特長であるインテリジェント性，高セキュリティ，大記憶容量，多機能性，携帯性などに基づき，その主な機能として

- 認証・識別機能 (個人およびカード)
- 暗号化機能
- データキャリア機能
- オフライン機能

を挙げることができる.

高度情報化社会の成長に伴いスマートカードの役割も様々な場面で広がりを見せつつあり，その応用分野は，金融，流通・サービス，交通・運輸，情報・通信，健康・医療，教育・企業，行政など多岐にわたる．分野毎に必要とされるスマートカードの機能について，主な特徴をまとめると表 II.2 のようになる.

まず認証機能および暗号化機能が最も強く求められる分野がキャッシュカードや電子マネーの金融・決済分野である．認証機能にはカード認証と個人認証の 2 種類があるが，カード認証をメインとするものとしては流通・サービス分野のポイントカード，交通・運輸分野の定期券や有料道路通行カードなどを挙げることができる．さらに個人認証も必要となる分野は ID 分野とも呼ぶことができ，教育・企業での身分証明証，行政分野の運転免許証がある．データキャリア機能およびオフライン機能はほとんど全ての分野に関係し，例えばデータキャリア機能が重要な役割を果たす用途として医療カードや健康保険証がある.

表 II.2: スマートカードの応用分野と機能

| 応用分野 | 用途 | 主に必要とされる機能 |
|---|---|---|
| 金融・決済 | キャッシュカード，クレジットカード<br>電子マネー，ホームバンキング | 個人＋カード認証，暗号化<br>データ記録，オフライン |
| 流通・サービス | ポイントカード，ショッピングカード<br>自動販売機 | カード認証，オフライン |
| 交通・運輸 | 有料道路通行料，駐車場料金<br>乗車券，定期券 | カード認証，オフライン |
| 情報・通信 | テレホンカード，移動体通信<br>有料放送受信料，PC ネットワーク | 個人＋カード認証，暗号化 |
| 健康・医療 | 診察券，医療カード，遠隔診断 | 個人＋カード認証<br>データ記録，オフライン |
| 教育・企業 | 身分証明証，個人情報<br>入退室管理，各種施設利用 | 個人＋カード認証<br>データ記録，オフライン |
| レジャー | テーマパーク，ゲーム<br>メンバーズカード | カード認証，オフライン |
| 行政 | 健康保険証，運転免許証<br>住民基本台帳，パスポート | 個人＋カード認証<br>データ記録，オフライン |

## II.1.3 スマートカードの利用状況と展望

スマートカードの主な利用状況と動向を国内と国外に関してまとめる．

### II.1.3.1 国内動向

**金融・決済分野** セキュリティ面から接触型カードの応用が特に見込まれる分野が金融・決済である．

キャッシュカードに関しては，郵便貯金スマートカード化実証実験 (大宮)，スマートコマース・ジャパン (神戸)，スマートカード・ソサエティ (渋谷)，スーパーキャッシュ (新宿) など，電子マネーとの併用による多機能化が指向されている．

クレジットカードに関しては，世界的クレジット会社がスマートカード化戦略を次々に発表し，VISA インターナショナルは 2002 年までに全発行カードの 3 分の 1 を，MasterCard は 2010 年までに全カードのスマートカード化を目標に掲げている．また JCB はクレジットカードとして世界初の接触/非接触統合型カードを 1998 年に導入している．

1999 年 1 月からサービスが開始された日本版デビットカード J-Debit も，特にオフライン処理によるコスト削減の目的から金融系カードのスマートカード化に弾みをつけると見られている．

**流通・サービス分野**　ポイントカードのスマートカード化として，商店街，ショッピングセンター，ガソリンスタンド等での発行事例がある．従来との差別化としては，クレジットカードとの一体化，顧客管理・サービス機能を持つメンバーズカードとしての利用が挙げられる．今後は特に自動販売機での非接触型カードの普及が進むとの予測もある．

**交通・運輸分野**　金融・決済分野よりもスマートカード化が進んでいるのがこの交通・運輸や情報・通信分野である．
汎用電子乗車券技術研究組合 (TRAMET) による 1998 年からのスマートカード乗車券実証実験，JR 東日本によるスマートカード出改札システムの 2001 年導入予定など，利便性から非接触型カードの応用が期待されている．スマートカード化による効果としては，サービスアップ (定期券を取り出す必要がない)，コストダウン (磁気券処理の低減，メカニカル部分が不要)，セキュリティの向上が見込まれる．
ノンストップ自動料金収受システム (ETC) の導入も進み，2000 年からの利用が予定されている．ETC ではスマートカードは車内設置する無線装置 (車載器) に挿入して使用される．クレジットカードやデビットカードなどとの将来的な機能統合の構想もあると言われる．

**情報・通信分野**　NTT 東日本，NTT 西日本による非接触型スマートカード公衆電話のサービスが 1999 年 3 月から開始され，同年 9 月時点での設置台数は 5000 台に上っている．今後は全国の県庁所在地級都市の主要駅などに設置される．スマートカード化の最大の要因は変造テレホンカード問題への対策であるが，導入によるその他のメリットとして，公衆電話機のコスト削減，利用者への新サービスの提供などがある．今後のスマートカード市場では約半数がテレホンカードで占められると予測されている．

**行政分野**　自治省ではコミュニティ・ネットワーク構想の下，行政サービスシステムの全国的な統一化のためスマートカードによる住民基本台帳番号カードの導入を検討しており，全国 17 市町村がモデル事業地域の指定を受けている．転入転出事務の効率化，住民票の写しの全国的交付などを目的とする．
社会保険庁は 1995 年から熊本県八代市で医療保険カード実験を行っており，1998 年からは八代市以外の全国の医療機関においても医療保険カードのみでの受診が可能になっている．

#### II.1.3.2　海外動向

スマートカード先進国である欧州を中心に金融・決済分野に関してまとめる．

**フランス**　フランス銀行カード協会は 1993 年までに全ての銀行カードのスマートカード化を実施し，現在は IC チップの位置をフランス方式と呼ばれるこれま

でのカードの上部からISO標準へと切替えを行っている．共有化・共通化に関しても，銀行間でのカードの共通化，スタンダードなカードの発行などを進めている．今後は電子財布，インターネット対応，ユーロ対応や，接触型/非接触型コンビネーションカードの導入も近いと予想される．

**ドイツ**　1996年にスタートしたドイツ全金融機関による電子マネーシステムGeldKarteは世界最大のカード発行規模(1998年で5000万枚)を誇る．利用率向上のための展開推進プロジェクトにも取り組んでおり，マクドナルドでの導入や，市内電車での割安賃金の設定などもあり，GeldKarteのターミナルは1999年に全国で7万台以上と見られている．

**イタリア**　1998年にシエナ市で発行が開始されたシエナカードは電子財布および行政カードとしての機能を持ち，行政カードに関しては，個人情報，個人認証，交通違反に対する徴税・罰金などのサービスが行われている．今後，身分証明証，市内バス利用，病院予約，税金支払，インターネットアクセスなどのサービスが追加される予定．

**オーストラリア**　電子財布プロジェクトとして，接触型スマートカードのVISAキャッシュ，モンデックス，クイックリンクの他に，非接触型のCitトランスカードがある．1995年から実証実験が開始されたCitトランスカードは多目的用途であり，公共交通機関の支払，各種自動販売機での支払，パーキングメーター，インターネット上での決済などのサービスを行っている．

#### II.1.3.3　今後の展望

今後は接触型と非接触型をワンチップ化したコンビネーションカードや，複数の機能を搭載した多目的利用の統合型スマートカードの実現が予想される．カードを多目的に使用する場合，カードの管理を誰が行うかなどの様々な問題が生じるため，課題検討，共通プラットフォームの整備など次世代スマートカードシステムの開発が業界を横断して進められている．

## II.2　スマートカード規格およびその動向

本章では，多くの団体，企業によって制定されたスマートカードの標準・仕様について，それらの位置づけ，ねらい，現状，相互関係を整理する．

図II.1に示すように国際的なスマートカード仕様の展開はISO 7816が他の標準・仕様の基盤となる規格としてあり，他の標準・仕様はISO 7816に準拠して制定されている．

ISO 7816ではスマートカードの物理的な部分の規格，電気的特性，標準コマンド，標準セキュリティが定義されている．EMVは，金融系スマートカードの国際標準であ

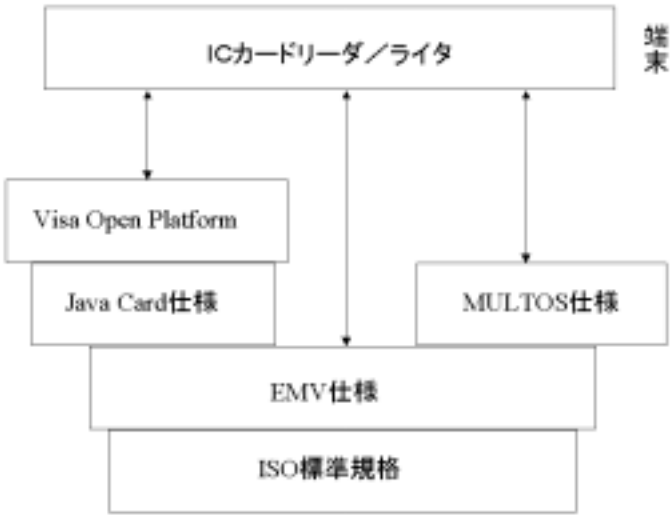


図 II.1: 各規格・仕様の連帯

り，スマートカードの基本 OS と位置づけられる．更に EMV に準拠した形で多目的スマートカード用の OS として，MAOSCO が MULTOS 仕様，SUN Microsystems が JavaCard 仕様を制定した．VISA インターナショナルはスマートカードに JavaCard 2.0 を用いる Visa Open Platform を提案している．

図 II.1 における端末とは，主に金融では POS や ATM などが，多目的利用においては PC や携帯電話など様々なデバイスが想定されるが，スマートカードの PC での利用を目的とする図 II.2 の各標準・仕様がある．

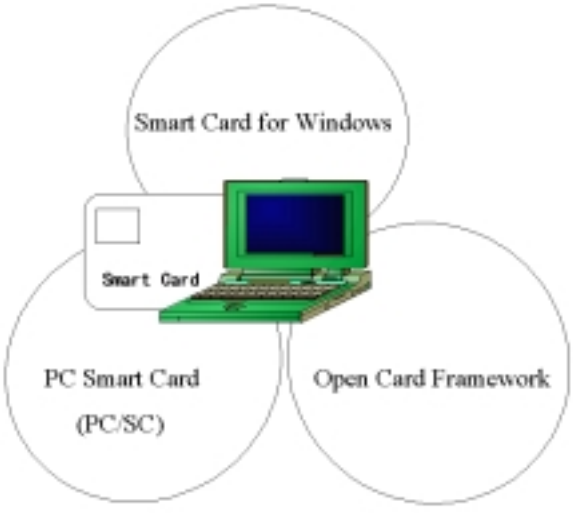


図 II.2: PC でのスマートカード利用

スマートカードの規格・標準化について国内に目を向けると，国際規格である ISO

7816 に基づき，以下の JIS(日本工業規格) が制定されている．

**JISX6303**　外部端子付き IC カードの物理的特性
　外部端子付き IC カードの物理的特性と，外部端子の位置・寸法を規定．

**JISX6304**　外部端子付き IC カードの電気信号および伝送プロトコル
　IC カードとリーダ/ライタ間の電気的信号の授受の手順を規定．

**JISX6306**　外部端子付き IC カード共通コマンド
　基本コマンド，ファイル構造，セキュリティの基本構造を規定．

**JISX6307**　外部端子付き IC カードのデータ要素
　業態間で共通に使用されるデータ要素の内容およびフォーマットを規定．

ISO 及び JIS 標準化を踏まえながら，アプリケーションのインターフェースを標準化することを目的として国内の企業，団体を中心に JICSAP 仕様が開発された．

この他に全日本銀行協会連合会によるスマートカード対応 ATM への互換性や銀行間による相互乗り入れなどの統一を図った国内版 EMV 仕様ともいえる全銀協仕様がある．

## II.2.1　国際・国内規格

### II.2.1.1　ISO 7816

ISO（International Organization for Standardization）の規格検討委員会によって，外部端子付きスマートカードの物理的，機能的条件等について，基本部分の互換性を確保するための国際規格として制定された [IC97, ISO]．利用分野を特定せずに適用されることを想定している．

ISO 7816 は，スマートカードの物理的形状や，リーダライタとの通信プロトコルなどを 11 パートに分けて定めている．以下にそれぞれを列挙する．

**ISO7816-1**　Physical characteristic
　材料，形状，寸法，強度および耐静電気特性を規定．

**ISO7816-2**　Dimension and location of the contacts
　外部端子の数，位置と寸法を規定．

**ISO7816-3**　Electric signals and transmission protocols
　スマートカードとリーダ/ライタ間の電気的信号の授受の手順を規定．
　AM1:T=1 ブロック伝送方式について規定．
　AM2:プロトコル選択基準について規定．

**ISO7816-4**　Use of the secure messaging
　基本コマンド，ファイル構造，セキュリティの基本構造を規定．

**ISO7816-5** Registration of identifier
アプリケーション ID，ID の登録方法，業務の選択基準を規定.

**ISO7816-6** Interindustry data elements
業態間で共通に使用されるデータの要素の ID，名称，内容およびフォーマットを規定.

**ISO7816-7** Interindustry commands for Structed Card Query Language
(SCQL) 関連コマンドについて.

**ISO7816-8** Security architecture and related interindustry commands
セキュリティ関連コマンドについて.

**ISO7816-9** Additional interindustry enhanced commands
PIN に関するコマンドについて.

**ISO7816-10** Operating procedure and answer to reset for synchronous cards
同期式スマートカードと端末間の電気的特性および伝送プロトコルについて.

**ISO7816-11** Security architecture
メッセージ安全保護の具体的な利用方法を含むセキュリティ概念.

#### II.2.1.2 JICSAP

IC カードシステム利用促進協議会（JICSAP : Japan Ic Card System APplication council）が 1997 年 9 月に制定したスマートカード仕様である [JI97]．カード製造者，リーダライタ製造者，システムインテグレータ等のスマートカードシステム関係ベンダ間のアプリケーション層での相互の互換性をとることを目的に制定されている．ISO/IEC JTC1（ISO と国際電気標準会議の合同会議）と ISO/TC68（金融取引標準化委員会）に対応した形でまとめられている．

コマンドや機能については ISO 7816-4 の中から必要最低限のものを抽出して採用している．また，スマートカードでの真正性の確認や端末の相互確認，不正アクセス防止のための自動ロック機構など，セキュリティの仕様も規定している．さらに，異なるサービス提供者による多目的利用を可能にするため，カード内部のファイル構造を規格化した．

仕様は 7 章から成る．以下に章立てとその概要を述べる．

1. 概要
   この仕様の内容の概要および参考規格についてのまとめ.
2. 電気的特性
   各端子の電気的特性は JIS X 6304 第 4 章，スマートカードの動作手順は JIS X 6304 第 5 章に従うことを規定している.

3. 伝送手順制御
   伝送制御手順は，JIS X 6304 第 9 章に準拠することを規定している．
4. 基本構造
   ファイル，データに関する基本構造，アクセス方法を規定している．
   ファイルは，ファイル制御情報を有する専用ファイル（DF）とデータ格納のための基礎ファイル（EF）に分類される．DF は正確な DF 名および名の上位桁からの部分桁からでも参照できる．EF は 2 バイトで符合化した EF 識別子によって参照できる．
   データはレコードまたはバイナリデータとしてアクセスされる．データアクセス方法およびレコード付番方法は，EF の構造に依存する．この方法についてもこの章で規定されている．
5. セキュリティ
   第 4 章で記述したファイル構造へのカード内部の CPU によるアクセス管理に必要な要素として，セキュリティ属性とセキュリティステータスがある．これらは ISO 7816-4 および JIS X 6306 に規定されている．
   セキュリティ属性は，カードに対してアクセスする際の諸条件を決定する情報であり，この規格ではデータに対しアクセスするために必要となるキー（例えば照合キー，認証キーなど）を指定する情報を指す．セキュリティステータスはこの照合（または認証）結果を保持するものである．
6. コマンド
   メッセージはコマンドを送信し，受信側でそれを処理し，レスポンスを送り返すステップにしたがって，接続装置とスマートカードで送受信される．メッセージにはコマンドメッセージおよびレスポンスメッセージがあり，これらは APDU（応用プロトコルデータ単位）と呼ばれる．これらのコマンドは ISO 7816-4 および JIS X 6306 の符号化規則にしたがって定められている．
7. ステータス
   データにおける各数値がまとめられている．

## II.2.2 業界標準

### II.2.2.1 金融

#### II.2.2.1.1 EMV

Europay, MasterCard, VISA のクレジットカード会社 3 社が策定したスマートカードの共通仕様である [IC97, EMV96, EMV98]．スマートカードの相互運用性を図るため，1996 年 6 月に発行された．EMV は ISO 7816 に基礎を置き，ISO 規格に金融アプリケーションに特化した機能を追加できるように規格されている．

EMV の仕様は 3 冊から構成されている．以下にそれぞれの概要を述べる．

1. スマートカード仕様
   決済システムにおいて正確な操作と互換性が確保されるように，スマートカードと端末に要求される最低限の機能を述べている．次の 4 パートからなる．
   (a) 電気的特性，論理インタフェース，伝送プロトコル
   (b) データ要素とコマンド
   (c) アプリケーション
   (d) セキュリティ
2. スマートカード端末仕様
   スマートカード仕様，スマートカードアプリケーション仕様で規定されるスマートカードを処理する端末に必要な必須・推奨・オプション条件を定義する．次の 3 パートからなる．
   (a) 一般的要件
   (b) ソフトウェアアーキテクチャ
   (c) カード保有者，取扱銀行インタフェース
3. スマートカードアプリケーション仕様
   国際規模における決済システムの構築に不可欠な，スマートカードおよび端末の手続きを定義する．

#### II.2.2.1.2　全銀協

全国銀行協会連合会（全銀協）は，銀行業務を中心とする多機能スマートカードの標準化を目的に「全銀協 IC カード仕様」を 1988 年 2 月に制定した [IC97]．国際的互換性を図る観点から，ISO における国際規格に準拠することを前提としているが，制定当時は ISO の規格がまだ検討中であったため，技術的仕様は暫定的内容にとどまっていた．その後 ISO の規格が固まったことや，スマートカードの利用実験が活発化してきたことから全面的な見直しを行い，1997 年 4 月に改訂版を制定した [ZGK97].

仕様は 2 部構成である．以下に概要を述べる．

**第 1 部**　IC カード標準仕様の基本的な考え方

スマートカードの利用業務を

- 銀行間の相互乗り入れが展望される「標準業務」
- 標準業務以外の，個々の銀行内のみで完結する「任意業務」
- 銀行業務以外の業務への利用のため，一部エリアを貸与する「領域貸与業務」

の 3 つに分類し，多目的利用を目指している．また，カードの発行主体が銀行でなければならない，という考え方も示している．

**第 2 部**　IC カードの技術的標準仕様

**第 1 章**　技術標準仕様の基本的な考え方

国際的な互換性を図る観点から，国際標準に準拠することを前提とする．具体的には，ISO/TC68（金融取引標準化委員会）の成果物である ISO 9992 を中心に準拠し，必要に応じて ISO/IEC JTC1（ISO と国際電気標準会議の合同会議）の成果物である ISO/IEC 7816 等を引用する．

また，国際規格において選択的な記述がなされている事項について具体的に特定し，国際規格の規定内容では実装困難なものや，この仕様で想定している機能を実現するために不足しているものについては，独自に定めている．

**第 2 章**　参照規格

ISO 7816, ISO 9992 などを参照している．

**第 3 章**　用語の定義および略語

この規格書で使われる用語の定義．

**第 4 章**　IC カードの物理的特性

この規格におけるスマートカードの物理的特性は，ISO/IEC 7816-1 および 7816-2 に準拠することを規定している．

**第 5 章**　IC カードの電気的特性

この規格におけるスマートカードの電気的特性は，ISO/IEC 7816-3 に準拠することを規定している．

**第 6 章**　伝送プロトコル

この規格におけるスマートカードの伝送プロトコルは，ISO/IEC 7816-3 の T=1 の伝送プロトコルを採用することを規定している．

**第 7 章**　データおよびファイルの基本構造

コマンドを処理する際の，端末側から見たデータおよびファイルの論理構造について定める．それらの実際の格納位置およびその他の構造に関する事項は規定しない．

ファイルは，ファイル制御情報を有する専用ファイル（DF）とデータ格納のための基礎ファイル（EF）に分類される．DF は正確な DF 名および名の上位桁からの部分桁からでも参照できる．EF は 2 バイトで符合化した EF 識別子によって参照できる．

**第 8 章**　論理チャネル

カード内部のファイルアクセスの利便性を考慮し，複数ファイルへの同時アクセスを実現するために論理チャネルを採用している．

論理チャネルには 0 から 3 までのチャネル番号がつけられる．この規格では，最低 2 チャネルをサポートすることと規定．そのうちの 1 つは基本論理チャネルで，国際的な取引を実行する際に常に利用可能とし，チャネル番号は 0 と定めている．

**第 9 章**　セキュリティ機構

第 7 章で記述したファイル構造へのカード内部の CPU によるアクセス管理に必要な要素として，セキュリティ属性とセキュリティステータスがある．またこの規格では，多目的利用を前提としているため，各業務の要求によりそれらの業務が個別に利用の可否を設定できるようにしている．セキュリティ属性は，カードに対してアクセスする際の諸条件を決定する情報であり，この規格ではデータに対しアクセスするために必要となるキー（例えば照合キー，認証キーなど）を指定する情報を指す．セキュリティステータスはこの照合（または認証）結果を保持するものである．

**第 10 章**　IC カードのライフサイクルとセキュリティ管理

銀行が発行するスマートカードのライフサイクルと，各局面におけるセキュリティ管理について規定している．

ライフサイクル，セキュリティ管理を規定する際に，下記を一般原則とする．

- カードのサイクル（製造，発行，使用，終結）にあたっては，個々のカードに対して処理が実行され，その実行により他のカードの利用を拘束しない
- カード発行者がカードのライフサイクルの責任を負う
- セキュリティ監査証跡は，カードのライフサイクル期間中保存される

**第 11 章**　メッセージ構造

メッセージはコマンドを送信し，受信側でそれを処理し，レスポンスを送り返すステップにしたがって，接続装置とスマートカードで送受信される．メッセージにはコマンドメッセージおよびレスポンスメッセージがあり，これらは APDU（応用プロトコルデータ単位）と呼ばれる．これらのコマンドは ISO 7816-4 の符号化規則にしたがって定められている．

**第 12 章**　メッセージの安全保護

メッセージの安全保護は，送受信メッセージの隠蔽や改ざん防止のために行われるが，この規格ではこの機能をサポートすることは任意としている．

**第 13 章**　コマンド

この規格におけるスマートカードのコマンドは ISO 9992 に準拠することを規定しており，その他のコマンドおよび機能をサポートすることは，規定の範囲外とする．

**第 14 章**　管理情報キャラクタ

管理情報キャラクタとは，IC 部の特性に関する数種の情報を，早い段階でカード受け入れ装置に伝達するために設定されているデータである．この規格では，ISO 7816-4 に準拠することを規定している．

**第 15 章**　機能（プロセス・フロー）

金融取引に仕様しうる各機能の構造と用法について以下の機能を規定している．

1. カード・セッションの初期化
2. 共通データファイル（カードに記録された，カード発行者などの共通データ）の認証
3. 適用業務ファイル（サービスをサポートするファイル）の選択
4. 適用業務ファイルの認証
5. カード受け入れ装置の認証
6. スマートカードによるカード保有者の検証
7. カード受け入れ装置による取引の承認
8. 取引の記録
9. 取引暗号化コードの記録

**第 16 章**　データ要素

各データ要素をまとめた表を掲載している．

#### II.2.2.2　カード用 OS

##### II.2.2.2.1　JavaCard Forum

JavaCard Forum とは，スマートカード製造メーカである Schlumberger 社と Gemplus 社が共同で結成した JavaCard のフォーラムである．

JavaCard は JavaCard API によって具体化できる Java ベースのスマートカードであり，JavaCard 1.0 が 1996 年に発行された．実用レベルにバージョンアップされた JavaCard 2.0 は Sun Microsystems 社が 1997 年に発行した [IC97, JC97, JC]．

JavaCard API は Java 言語でかかれたアプリケーションを，スマートカードで利用するための API である．スマートカード上のプログラムはカード OS（COS）上で動作するが，COS に互換性がないと同じプログラムを導入しても異なる機種上では動作しなくなる．これがスマートカードの多目的利用を妨げる原因であった．そこで，Java が動作するチップを使ってどの環境でも動作するようにしたカードが JavaCard である．

JavaCard API 仕様に準拠した JavaCard アプレットはプラットフォームに依存しないため，別のベンダーのカードでも動作させることが可能である．また，複数のアプリケーションを搭載することもできる．これらのアプリケーションは，カード発行後にもインストールすることができる．

JavaCard は ISO 7816 や EMV に準拠している．

##### II.2.2.2.2　Visa Open Platform

Visa Open Platform は「いつでも，どこでもアクセスできるカード」をコンセ

プトに 1998 年 7 月 VISA メンバー各社により VISA キャッシュや VISA スマートクレジット，VISA スマートデビット，コンピュータネットワークセキュリティ，電子商取引，ロイヤリティ（購入得点化プログラムなど），旅行やエンターテイメントといったマルチアプリケーションを持つカードとして Visa Open Platform スマートカードプロジェクトが制定した仕様である [IC97, VISA].

VISA インターナショナルが世界中の VISA メンバー金融機関に対して提供するプログラム「Visa Smart」を行うに当たり，スマートカード，端末，アプリケーション開発サービスの提供を行う．このカード，端末，アプリケーション開発サービスの具体的仕様が Visa Open Platform である．ここでいうスマートカードとは，JavaCard 2.0 仕様に基づくもので，金融以外の多機能性も有している．各カード発行会社が金融や金融以外の機能を自由に選んで組み込めるようにカード発行後のアプリケーションローディングの標準化, プラットフォームの独立性，マルチアプリケーションのポータビリティなどを目指している．

また，カード端末共に ISO/EMV 仕様のカード及び端末との相互互換性を保証し，スマートカードの PC での利用に際してはマイクロソフトの Smart Card for Windows のサポートを行う．

#### II.2.2.2.3 MULTOS

MULTOS は Mondex 社が標準化を提唱しているカード OS である [IC97, MUL]. Mondex 社は 1997 年 5 月に MAOSCO と呼ぶコンソーシアムを結成し，多目的スマートカード向け OS としての MULTOS の応用技術開発を推進している．

MULTOS は 1 枚のスマートカード上に複数のアプリケーションを搭載できるのが最大の特徴である．そのため，決済機能のほか出退勤管理や定期券など，さまざまな機能を 1 枚のカードにまとめられる利点がある．各アプリケーションはパソコンを使ってロードしたり削除したりできる．

チップ上に OS が載り，それらの上にアプリケーションが搭載される MULTOS は C 言語をベースとした MEL（MULTOS Enabled Language）と呼ばれる言語を使った開発環境が用意されているアプリケーションである．

MULTOS の電気仕様や端末との通信コマンドは EMV 仕様に準拠しており，既存の Mondex カードや IC クレジットカードなどと同じ端末で利用可能である．

### II.2.2.3 PC でのカード利用

IC カードを PC の周辺機器としてセキュリティ用途に用いたいという要請に応えることを目的として次のような規格が提案されている．

#### II.2.2.3.1 Smart Card for Windows

Microsoft 社が 1998 年 9 月にスマートカードのための OS として発表した Smart

Card for Windows [SCW99] は，8K バイトの ROM を内蔵したスマートカード用の 8 ビット OS である [Kum99, His00].

Smart Card for Windows の狙いは次の 4 点である.

- スマートカードを，開発ツールと接続性の両方に関して，PC 環境で普及可能にする.
- 開発者が理解でき，それをサポートするソフトウェア開発ツールを提供する.
- カード発行者が希望のコンポーネントを各提供者から選択できるようにする.
- 低価格なスマートカードを配布できるようにする.

Smart Card for Windows の大きな特徴は，Microsoft 社の Visual C++や Visual Basic など，多くの企業が利用している開発ツールをそのまま使えることである．また，Smart Card for Windows は PC/SC プログラムの一部であるため，カードリーダに PC/SC 認定のロゴがついてさえいれば，カードを読み取ることが可能である.

その他の特徴を以下に挙げる.

- スマートカードを使用したシステムを構築する場合に，カード自身は Smart Card for Windows で対応，カードにアクセスするためのアプリケーションは Windows プラットホームで対応する.
- メモリやアプリケーションのニーズに応じて機能を追加／削除し，OS をカスタマイズできる.
  例えば，カードのセキュリティ機能の高低，カードの正常動作に必要なコマンド数を決めたり，チップメーカを指定することができる.
- アプリケーションをダウンロードして追加できる.
  ダウンロード可能なものは言語非依存なバーチャルマシンで，例えば Visual Basic などで開発したプログラムである．カード上にアプリケーションをダウンロードできるという点では JavaCard と同じである.
- Windows ネットワークで動作する.
- 単純な構造の小さな OS のため，開発コストを JavaCard や MULTOS を用いたカードに比べて 1/5 程度におさえることができる.

Smart Card for Windows はスマートカードを使用するすべてのシステムがターゲットである．現在の仕様は 8 ビットカード対応であるが，今後は 32 ビットカードの上位バージョンも予定されている.

#### II.2.2.3.2 Open Card Framework

Open Card Framework (OCF) 1.0 は 1998 年 5 月に CardTech/SecureTech East 98 において発表された規格で，複数のベンダーからなる Open Card コンソーシアムが策定した [Ima99, OC]．現在は OCF1.1 がリリースされている.

現在の市場では多くのスマートカードが出回っているが，各サービスを実現する

プログラムを書く場合，それぞれの機種に応じたプログラムを書く必要がある．すなわち同じサービスを実現するプログラムでも，機種が異なればプログラムを書き換えることが求められる．そのため一度スマートカードの機種を確定してしまうと，機種を変更するのが容易ではなかった．つまりアプリケーション開発者から見たとき，スマートカードの機種依存に関して，少なくとも次の 3 点を考慮する必要がある．

1. カード端末製造者
   各製造者ごとにさまざまなインタフェースやプロトコルを用いている．さらに同じ製造者のカードでも，機能や価格からいろいろなモデルが存在する．そのためカードに依存したインタフェースなどを考慮しなければならない．
2. カード OS プロバイダ
   多くのカード OS，APIs（Application Program Interfaces）があるため，種々のコマンドやレスポンスコードがある．カード OS ごとに，これらに対応させなければならない．
3. カード発行者
   カードに搭載するアプリケーションを，カードのどこに配置するのかを決めるのがカード発行者である．そのためアプリケーションコードは自由に配置できるようにしておかなければならない．

これらの問題に対し，OCF は CardTerminal 層，CardService 層の 2 つの層および ApplicationManagement を提供している．

CardTerminal 層は，カード端末製造者に，カード端末の物理的特性などを与えるものである．また，PC/SC 対応のカード端末にアクセスするための Java API も提供している．

CardService 層は，現存するカード OS とそれらのさまざまな機能を幅広く扱えるようにするためのインフラを与える．インタフェースの設計は Java のプログラミングモデルに適合するような形で行われている．

ApplicationManagement は，1 枚のカードに複数のアプリケーションを搭載するときに，それらのアプリケーションの配置，選択の方法や，リストを作成する方法を与える．これにより，カード発行者に依存していた部分の問題が解決される．

#### II.2.2.3.3 PC Smart Card (PC/SC)

PC/SC は PC 上でのスマートカードを利用するための標準を定めたもので，複数のベンダーからなる PC/SC Workshop により 1997 年 12 月にバージョン 1.0 が公開された [PC]．ISO 7816 と EMV を基礎に作られている．

PC/SC の仕様は大きく分けてスマートカード，カードリーダ，カードリソースを管理するリソースマネージャ，カードを操作するアプリケーションの 4 つである．これらの各層間でのインタフェースを定義することで，カード，リーダ，アプリケーションの組み合わせをできるだけ意識せずにシステムを構築できるようにすること

が，PS/SC の狙いである．

PC/SC を構成する 8 つのパートは以下の通りである．

**Part 1** システムの構成，コンポーネントについての概要．

**Part 2** スマートカードと接続される機器の特性について．

**Part 3** 接続される機器とのインタフェースについて．

**Part 4** 接続される機器について．

**Part 5** リソースマネージャについて．

**Part 6** スマートカードサービスプロバイダのモデルについて．

**Part 7** アプリケーション開発ツールについて．

**Part 8** セキュリティについて．

# 第III部
# 耐タンパー性に対する解析法

## III.1　解析の分類

解析方法として，チップを取り出すなどカードに物理的変形を加えて解析する場合と，カードの通常の利用法の中で解析する２つの手法が考えられる．ここでは，この二つのアプローチをそれぞれ破壊型解析，非破壊型解析と呼び，それぞれについて知られている代表的手法を簡単に整理する．

### III.1.1　破壊型解析法

破壊型解析法は，直接チップを解析できるという意味で非常に強力な解析法であり，さまざまな提案がなされてきた [AK, HPS99, KK99]．本節では代表的な手法としてプローブ解析 [HPS99] を紹介する．

#### III.1.1.1　プローブ解析

プローブ解析は，耐タンパデバイスのパッケージを取り去り直接チップにプローブを当てることを想定した場合の暗号解析手法である．攻撃者はICにプローブを当て，秘密情報を用いた計算がなされているときの任意の1ビットの変化を観察し，秘密情報を取り出す．内部バスにアクセスできる場合，実装されたほとんどすべての暗号方式は解析可能である．

しかしながら，解析には攻撃対象の耐タンパデバイスについての深い知識を要し，攻撃の実現可能性は低いと思われる．攻撃者はまずデバイスのパッケージを適切に取り去り，チップの表面を露出させる．次に線幅に気をつけ，保護膜の除去を行う．攻撃者は解析を行う際どのゲートが目的のものか知る必要がある．また，これらの作業が成功した場合もプローブの特性劣化や信号の同期をとる際の問題などがある．さらに，どのレジスタに観察すべきビットが存在するか知っているかどうかで解析の困難さが大きく変化する．

このような攻撃に対して，セキュリティディテクタの装備，空気に触れると記録内容が消滅する揮発性メモリの採用などの対策が考えられている．

### III.1.2　非破壊型解析法

耐タンパデバイスを用いたセキュリティシステムの設計者は，多くの場合，秘密情報が閉じた信頼できる計算環境の中にあることを仮定する．しかし，実際のコンピュータやマイクロチップは秘密情報を用いて演算を行う際，設計者の予想しなかった情報 (side-channel information) を外にもらしてしまう．このような設計者の予期しない情報を利用してスマートカードの秘密情報の解析を行う方法としてタイミング解析と電力解析が知られている．また，故障利用解析と呼ばれる，攻撃者が人為的に起こした計算誤りを利用して解析を行う手法も非破壊型解析法に分類される．本節ではこれら非破壊型解析法の概要を整理する．

#### III.1.2.1　故障利用解析

故障利用解析は，加熱や放射線照射などによりIC内の回路に人為的に故障を起こさせ，正常な出力と故障時の出力の差から，内部の秘密を推定するという方法である．1996年にBonehら [BDL96, BDL97] により，公開鍵暗号系で用いられている冪乗剰余演算等の代数演算を実装した耐タンパデバイスから格納されている秘密情報を得る方法が提案された．この後，多くの研究者らにより様々な条件の下での適用例やより現実的な攻撃モデルの提案，およびその評価がなされている [BS97, JQ97, MS97].

一般に耐タンパデバイスのICの部分は主に次のパーツからなっている．

- CPU (Central Processing Unit)
- ROM (Read Only Memory):
  読み出し専用の半導体メモリであり，不揮発性である．
- PROM (Programmable ROM; EPROM (Erasable PROM), EEPROM (Electrically EPROM), etc.):
  特別な条件下ではデータの消去及び再書き込みが可能なROMのことであり，不揮発性である．データの消去において紫外線を用いるものをEPROM，電気的に一括消去できるものをEEPROMと呼ぶ．EEPROMでは，数十万〜百万回までしか消去及び書き換えができない．
- RAM (Random Access Memory):
  読み書き可能な半導体メモリであり，ほとんどのものは電源が供給されないとデータを保持できない (揮発性である).
- コプロセッサ
- Random Logic
- その他

これらのうちEPROM, EEPROM, RAMは紫外線を照射したり電圧を加えるなどして書き換えが容易である．最も書き換えが困難なマスクROMでも，ワード線，ビッ

ト線から対象のメモリセルを一度切り離し，外部から反転させた信号を入力できれば攻撃は可能である．しかし，メモリにパリティチェック機能を持たせることで，パリティエラーが起きたときにプログラムを停止させることができる．また，ワークステーションクラスの計算機には ECC (Error Correcting Code) メモリが用いられているものもあり，このような攻撃に対抗する技術も存在する．

#### III.1.2.2 タイミング解析

タイミング解析は，暗号処理のタイミングが処理の最適化のため鍵情報に依存して変化することに着目し，処理のタイミングを統計的に解析して秘密情報を推定する方法である．1996 年に Kocher [Koc96] がはじめてアイデアを提案し，冪乗剰余演算に対する攻撃が成功する確率をコンピュータシミュレーションによって導出した．その後，タイミング解析によって実際にスマートカードから秘密鍵を取り出した報告もある [DK+98].

この攻撃に対し，システムのコスト，パフォーマンスをほとんど損なわずに解析を防ぐ方法がいくつか知られている．Montgomery アルゴリズムを用いた冪乗剰余演算では，アルゴリズムに若干の修正を加えることで解析を困難にすることが可能である [DK+98]． また，電子現金や電子投票において匿名性を確保するために提案されている，ブラインド署名のアルゴリズムを応用して解析を困難にする手法もある [Koc96].

#### III.1.2.3 電力解析

電力解析は，処理を行っているときの耐タンパデバイスの消費電力が秘密情報や処理内容と関係があることに着目し，消費電力を観察することによって秘密情報を推定する方法である．1998 年に Kocher ら [KJJ98] が提案し，現在標準的に用いられている共通鍵暗号である DES (Data Encryption Standard) [DES77] を実装したスマートカードに対して解析を行った．彼らはカードが処理を行っているときの消費電力の変化を直接解析に用いる方法 (単純電力解析) と，観測データに対して統計的処理を行うことで秘密鍵を推定する方法 (電力差分解析) を発表した．彼らの方法は非常に強力でさまざまなアプリケーションに適用できるなどの理由により大きなインパクトを与えている．彼らの発表以降，多くの研究者により様々なシステム下での適用例やその評価がなされており [BS99, CJ+99a, CJ+99b, Cor99, Fah99, GP99, KJJ99, MDS99a, MDS99b]，次世代の共通鍵暗号の標準として選定が進められている AES (Advanced Encryption Standard) の候補に対しても，実装した場合，電力解析に対する耐性があるかどうかの評価が行われている [BS99, CJ+99a].

# III.2　破壊型解析法

## III.2.1　破壊型解析法の脅威と対策

破壊型解析法は，スマートカードおよびそのチップに物理的変形を加えて解析を行う手法である．本節では，[AK, AK97, KK99] により破壊型解析法の手法とその対策を紹介する．はじめに破壊型解析法への準備として，スマートカードの物理的構造について述べる．

一般的なスマートカードは以下に示すように８ビットのマイクロコントローラとROM, EEPROM, RAM などで構成され，シリアルな入出力を持っている．これらは１チップに収められ，プラスチックのカードに封止されている．暗号鍵などのセキュリティ関連情報は EEPROM に入っている．図 III.1 にスマートカードの構成と IC モジュールの外部端子を図 III.2 に IC チップモジュールの内部構成を示す．コプロセッサは，暗号処理などに関してスマートカード CPU を補助する．

1. マイクロプロセッサの構成
   (a) CPU
   (b) 暗号処理用コプロセッサ
   (c) セキュリティ論理回路
   (d) データメモリ (EEPROM)
   (e) 一時記憶メモリ (RAM)
   (f) オペレーティングシステム (ROM)
2. チップの特徴
   (a) ROM : 3K ～ 23K バイト
   (b) RAM : 1K バイト以下
   (c) EEPROM : 1K ～ 32K バイト
   (d) CPU : 8bit マイクロプロセッサが主流だが，16bit CISC や 32bit RISC もある．
   (e) テクノロジ : $1.2\mu$m ～ $0.8\mu$m
   (f) チップ寸法 : $10\text{mm}^2$ ～ $25\text{mm}^2$

破壊型解析法は，以上のようなチップの内部を直接解析できるという意味で非常に強力な解析法である．以下，破壊型解析法の原理，手法，対策について述べる．

### III.2.1.1　破壊型解析の原理

破壊型解析法には，スマートカードからチップを取り出しチップを開封して，各レイヤを剥いでいきプロセッサ内の回路情報や ROM データを顕微鏡を用いて読み出す静的な情報収集と，チップを動作させながらアドレス，あるいはデータバス上の任意の 1 ビットにプローブを当て，その変化を観測する動的な情報収集，ROM 等

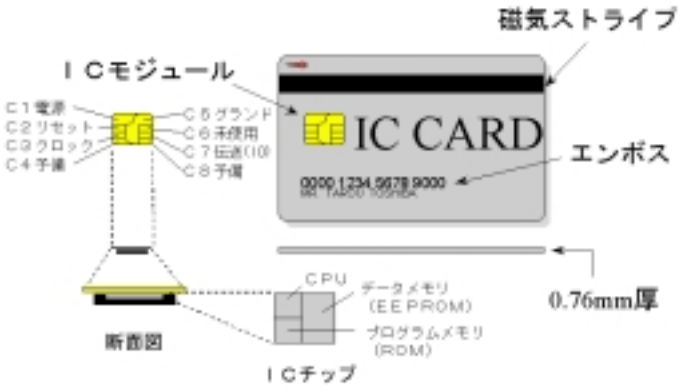


図 III.1: スマートカードの構成

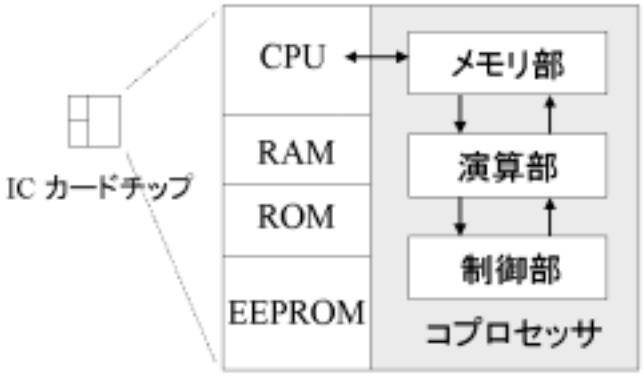


図 III.2: チップの構成

のメモリやゲートのデータを直接書き換える解析方法等がある．

本節では，破壊型解析を行うアタッカの傾向とコスト，破壊型解析により漏洩する可能性のある項目について述べる．

文献 [AK] ではアタッカーは以下の 3 種類に分類されるとしている．

**クラス I**　賢い外部者

**クラス II**　知識を持った内部者

**クラス III**　資金のある団体

Kuhn らは，スマートカードに限らず他の耐タンパーデバイスはクラス II やクラス I のアタッカーに攻撃されやく，クラス I のアタッカーは数千ドル程度の資金で大学の電気工学部などにあるマイクロプローバとレーザーカッタ顕微鏡を用いて解析を行うことが可能であるとしている．

またさらに高度な解析技術を用いることにより，近年僅か数日でプロセッサの重要な部分がリバースエンジニアリングされるようになっている．

一般に破壊型解析法は，特殊な装置と半導体のプロセスおよび回路技術，暗号アルゴリズムなど多岐にわたる知識と技術が要求され，比較的コストも高いため実現性は低いと見られているが，一旦チップが開封されると以下に示すような情報が漏洩する危険度は大きく，破壊型解析法の手法の理解と対策について熟考する必要がある．

**プロセッサの回路構成**　プロセッサの回路構成を得ることにより，命令体系などが明らかになる．

**ROM 等メモリの内容**　メモリに実装された秘密情報や使用されている暗号アルゴリズムが解析される．

**プローブによるバス上のデータの内容**　チップにプローブを当て，回路を動作させながら任意のバス上の 1 ビットの変化を観察して秘密情報を取り出す．プローブによっても実装されたほとんど全ての暗号方式は解析可能である．

**回路そのものの改竄**　次節の破壊型解析の手法で詳しく述べる解析装置を用いることにより回路構成が盗まれるだけでなく，回路そのものを改竄される可能性がある．

#### III.2.1.2　破壊型解析の手法

破壊型解析法の手順として下記が文献で示されている．

1. スマートカードのパッケージ開封
   カードから取り外したチップをある種の溶剤に浸すことでチップ表面の保護膜の除去を行う．

2. ネットリスト作成
   各種顕微鏡を用いて撮影したチップ表面の写真を解析し，トランジスタレベルでのネットリストを得る．
3. マニュアルプロービング
   マニュアルプロービングワークステーションと呼ばれる装置を用いてチップ上のバスラインへのプロービングを行う．

また解析に用いる主な装置として次のようなものがある．

1. 顕微鏡

   **共焦点光学顕微鏡**　電子顕微鏡のように試料を真空中に置く必要がなく，普通の光学顕微鏡に比べて解像度とコントラストが良い．

   **電子顕微鏡**　電子顕微鏡には大きく分けて透過型電子顕微鏡 (TEM: Transmission Electron Microscope) と走査型電子顕微鏡 (SEM: Scannig Electron Microscope) の二つがあり，TEM は試料の微細領域の組成や構造の解析を，SEM は試料表面に対する解析を行うことができる．

2. 波形解析装置

   **EB テスタ**　回路内部の AC 波形測定，論理波形測定

   **マニュアルプローバ**　回路内部の絶対電位測定，回路内部への信号印可

3. 解析補助装置

   **簡易 RIE 装置**　保護膜および層間絶縁膜の剥離

   **FIB (Focused Ion Beam) 加工装置**　イオンビームによる集積回路内部の配線の切断・接続．表 III.1 に FIB 加工装置とレーザーカッタの機能の比較を示す．

表 III.1: FIB 加工装置とレーザーカッタとの比較表

| | FIB | レーザーカッター |
|---|---|---|
| 最小加工寸法 | $0.3\mu m$ | $1 \sim 2\mu m$ |
| スループット | 遅い | 早い |
| 深さ方向の制御 | 容易 | 困難 |
| 熱的なダメージ | 殆ど無し | 有り |
| 加工形状 | 矩形 | すり鉢状 |

#### III.2.1.3　破壊型解析の対策

破壊型解析法の対策として，以下の方法を挙げることができる．

1. 周波数センサの強化
   CPU のクロックに同期して観測を行うプローブ解析はプロセッサの動作速度が遅くなる (数キロヘルツ程度) ほど容易になるため，低周波アラームの装備がプローブ攻撃の対策として有効である．高性能な周波数センサには，設定されたタイムリミットを超えてクロックに変化が起こらない場合にトリガーを出すタイプもあり，トリガーを検知したプロセッサは命令をリセットし，全てのバスラインとレジスタを直ちにグランドレベルに落とす．
2. チップ表面の加工
   ニクロム線が組み込まれた IBM の ABYSS チップのように物理的に無理なアクセスが行われた場合にチップ自体を破壊する技術や，チップ表面にシリコンをコーティングすることで溶剤を用いた化学的攻撃への耐性を高める技術等が知られている．
3. 読み取りにくいメモリの採用

   **ROM**　ROM のビットパターンは拡散層にストアされているためチップ表面からの読み取りは通常困難であるが，保護層が除去された場合の対策としてアドレスバスをスクランブルするなどの方法がある．

   **RAM**　観測の繰り返しを要するプローブ解析を RAM へ適用する際，アタッカーが電圧を変化させることで RAM のカウンタを容易にリセットしてしまうことも起こり得る．このような問題への対応策として，DRAM 同様に読み出しによって電荷が消滅する FeRAM などの強誘電体デバイスの採用が有効となる．

4. 解析センサの採用
   ダミーのメッシュ状メタルレイヤを回路上へ配置することもプローブ解析に対する対策として有効であり，スマートカード用の CPU である ST16SF48A, DS5002FPM, DS1954 のようなバッテリーバッファ型の SRAM 内蔵のセキュリティプロセッサなどで実装されている．
5. センスした情報に基づく情報の消去
   例えば上記メッシュセンサは割り込みや回路のショートを常にモニターし，電源電圧の変化を検知した場合は直ちに ROM へ 0 を書き込むことでプローブ解析を阻止している．
6. テスト回路の破壊
   工場出荷時の検査用として，プロービングによってバスラインやコントロールラインに直接アクセスするためのテスト回路が備わっており，これを悪用したプローブ解析を防止するため，工場出荷時にはテスト回路を構造的に破壊する

ことが重要である．

7. アーキテクチャレベルでの対策
   一般的でない命令セットやバススクランブル技術による独自のセキュリティプロセッサの開発も破壊型解析への対策として有効である．コストとのトレードオフを伴うが，チップ，暗号アルゴリズムなどの更新期間の短縮や，単純な回路構成の採用を避けることによっても安全性の向上が実現できる．

## III.2.2 プローブ解析

破壊型解析法としてHandschuhらによって提案されたプローブ解析[HPS99]を紹介する．この手法はICの内部バスに直接探針を当て，暗号処理の内部周期毎に目的のレジスタのビット値を観測することで秘密情報を得る．そのためチップへのプローブが可能な状況さえ攻撃者が一旦作り出してしまえば，解読は暗号方式を問わずほとんどのものに対して可能となる．ただし，実際にプローブ解析を実行するには暗号方式とデバイスの両者に対する詳しい知識を要し，いくつもの技術的課題を克服しなければならないため，攻撃の実現可能性は決して高いとは言えない．

本節ではまずプローブ解析の原理を公開鍵暗号に対して示し，適用例としてRSA，DSA，DES，RC5に対する攻撃法にも触れる．最後に解析の実現可能性と対策を検討する．

### III.2.2.1 プローブ解析の原理

ここではRSA，ElGamal，DSA，Schnorr型署名などの公開鍵暗号に対する攻撃を例にとってプローブ解析の原理を説明する．

公開鍵暗号で用いるべき乗剰余算

$$A = m^d \bmod n$$

において，指数$d$をプローブ解析によって求めることを考える．その際，べき乗剰余算の実装アルゴリズムとして実績の高いSM-1 (Standard Square-and-Multiply Algorithm)を例にとる．このアルゴリズムの処理手順は図III.3に示す通りである．

SM-1では内部ループ毎に$d$を1ビットずつ上位ビットから下位ビットへとスキャンしながら演算処理を実行し，$i$番目のステップが終了した時点でアキュムレータ$A$に格納されている値は

$$A_i = m^{d_i} \bmod n$$

となる．ここで$d_i$は$d$の上位$i$ビットで表現される整数を表す．

プローブ解析では，アキュムレータ$A$の任意の1ビット以上の値がステップ毎に完全な同期をもって測定可能であることを前提条件としている．すなわち，プロー

| | |
|---|---|
| Initialization | |
| | $N \leftarrow n,\ M \leftarrow m$ |
| | $A \leftarrow 1,\ i \leftarrow \|d\|$ |
| Scanning Loop | |
| | While $(1 \leq i)$ |
| | { |
| | $\quad A \leftarrow A \cdot A \bmod N$ |
| | $\quad$ If $(d[i] == 1),\ A \leftarrow A \cdot M \bmod N$ |
| | $\quad i \leftarrow i - 1$ |
| | } |
| Output | $A = m^d \bmod n$ |

図 III.3: SM-1 の処理手順

ブするアキュムレータ $A$ のビット位置を $J \subset \{1, \ldots, |A|\}$ ($|x|$ は $x$ のビット長を表す) とし，観測値のセットを $A(J)$ とすると，攻撃者はステップ $i$ の終了時までに

$$T_i = (A_1(J), A_2(J), \ldots, A_i(J))$$

というシークエンスを得ることになる．このプローブするビット位置 $J$ が既知か否かで解析手順が異なるため，以下それぞれの場合に分けて解説する．

#### III.2.2.1.1 ビット位置が既知の場合

べき乗剰余算の実行において底 $m$ と法 $n$ は既知であるため，$d_i$の推測値を$\delta$とすると，攻撃者は SM-1 を用いて

$$T'_i(\delta) = (A'_1(J), A'_2(J), \ldots, A'_i(J))$$

を容易に計算することができる．推測値$\delta$が正しくなるのは $T_i = T'_i(\delta)$ の場合のみであるため，プローブ解析では $T_i$の測定毎に $T_i$と $T'_i(\delta)$ とを比較し，$\delta$が正しいものであるかどうかを判定する．これを $i = 1, \ldots, |d|$ の間繰り返すと，攻撃が終了した時点で得られている$\delta$の候補には求めるべき値 $d$ が必ず含まれていることになる．以上の攻撃手順をまとめたものが図 III.4 である．処理 (i) では$\delta$の候補が機械的に 2 倍に増やされるが，これは SM-1 のステップ毎に $d_i$が 2 進数で 1 桁ずつ増えていくのに対応した操作である．そして (ii) が$\delta$の判定処理となっている．

このプローブ解析は統計的手法を必要とせず，ポイントはチェックするべき$\delta$の候補数 $|\Delta|$ がステップを重ねる毎に爆発的に増加しないかどうかにかかっている．この候補数の増減則はプローブするビット数 $|J|$ に依存して変わるため，条件毎に $|\Delta|$ を推算する．

$\Delta \leftarrow \{0\}$
For $(i = 1, \ldots, |d|)$
{
  (i) $\Delta^0 \leftarrow \{2\delta,\, 2\delta + 1 \mid \delta \in \Delta\}$
  (ii) $\Delta \leftarrow \{\delta \mid \delta \in \Delta^0 \text{ and } T_i = T_i'(\delta)\}$
}
return $\Delta$

図 III.4: プローブ解析の手順 (ビット位置が既知)

$|\Delta|$ の見積り:　ステップ $i$ 終了時の$\delta$の候補数の平均値 $u_i$は,

$$u_i = 1 + \varepsilon(2u_{i-1} - 1) \tag{III.1}$$

という漸化式に従う．ここで$\varepsilon$はステップ $i-1$ で正しいと判定された$\delta$がステップ $i$ でも正しいと判定される確率であり，測定値 $A_i(J)$ と推測値 $A_i'(J)$ との間に相関が全くないと仮定して

$$\varepsilon = 2^{-|J|}$$

としている．(III.1) 式は，ステップ $i-1$ で生き残った$\delta$の候補には正しい$\delta$が必ず 1 つ含まれ，それ以外の $2u_{i-1} - 1$ 個は次のステップ $i$ で確率$\varepsilon$で生き残ることを意味している．この漸化式 (III.1) を初期条件 $u_0 = 1$ を用いて解くと，ビット数 $|J|$ が 1 か 2 以上かによって $|\Delta|$ の推定値を

$$|J| = 1\ (\varepsilon = \frac{1}{2}) \text{ のとき} \qquad E(|\Delta|) = 1 + \frac{|d|}{2} \tag{III.2}$$

$$|J| \geq 2\ (\varepsilon < \frac{1}{2}) \text{ のとき} \qquad E(|\Delta|) = \frac{1 - \varepsilon - \varepsilon(2\varepsilon)^{|d|}}{1 - 2\varepsilon} \leq \frac{1 - \varepsilon}{1 - 2\varepsilon} \tag{III.3}$$

と求めることができる．すなわちプローブするビット数が $|J| \geq 2$ の場合は候補数は$\frac{1-\varepsilon}{1-2\varepsilon}$以下であり，この値は最も大きくなる $|J| = 2$ で 1.5 と非常に小さいものになっている．また $|J| = 1$ の場合も候補数は $|d|$ のオーダーであり，解析は実行可能であることが分かる．

#### III.2.2.1.2　ビット位置が未知の場合

プローブするビットの位置が未知の場合も基本的な解析手順は変わらないが，ビット位置に関するループが内側に追加されるため攻撃手順は図 III.5 のように拡張される．

$J \leftarrow \{1, \ldots, |A|\}$
$^{\forall} j \in J \quad \Delta_j \leftarrow \{0\}$
For $(i = 1, \ldots, |d|)$
{
  For $(j \in J)$
  {
    (i) $\Delta_j^0 \leftarrow \{2\delta, 2\delta + 1 \mid \delta \in \Delta_j\}$
    (ii) $\Delta_j \leftarrow \{\delta \mid \delta \in \Delta_j^0 \text{ and } T_i \subset T_i'(\delta; j)\}$
  }
  If $|\Delta_j| = 0$ then $J \leftarrow J - j$
}
return $\Delta = \cup_J \Delta_j$

図 III.5: プローブ解析の手順 (ビット位置が未知)

処理 (ii) の $T_i'(\delta; j)$ は，$j \in \{1, \ldots, |A|\}$ である任意のビット $j$ に関して推算した

$$T_i'(\delta; j) = (A_1'(j), A_2'(j), \ldots, A_i'(j))$$

である．$\delta$は $T_i \subset T_i'(\delta; j)$ であれば $j$ に関して正しいと判定され，$\Delta_j$に保存される．そして $|\Delta_j| = 0$ となった時，$j$ はプローブしているビット位置ではないことが分かり，ビット位置に関するループから外される．

$|\Delta|$ の見積り:　プローブしているビットの位置が未知の場合は，前述の漸化式 (III.1) において初期条件が $u_0 = |A|$ となり，$|\Delta|$ の推定値は次のようになる．

$$|J| = 1 \text{ のとき} \quad E(|\Delta|) = |A| + \frac{|d|}{2} \tag{III.4}$$

$$|J| \geq 2 \text{ のとき} \quad E(|\Delta|) = \frac{1 - \varepsilon - (1 - \varepsilon)(2\varepsilon)^{|d|}}{1 - 2\varepsilon} + |A|(2\varepsilon)^{|d|}$$

$$\leq \frac{1 - \varepsilon}{1 - 2\varepsilon} + |A|(2\varepsilon)^{|d|} \tag{III.5}$$

$\delta$の候補数の初期値が $|A|$であるために ($|A|$ は鍵サイズで決まり，512，1024，2048 といった値になる)，(III.5) 式の $|J| \geq 2$ の場合，ビット位置が既知の (III.3) 式と比較して $|A|(2\varepsilon)^{|d|}$という項が付加される．しかしこの項はステップ数 10〜20 程度で急速にゼロへと収束するため，ビット位置が未知の場合でも解析は現実的といえる．また $|J| = 1$ の場合にも候補数は $|A|$ のオーダーであり，解析は実行可能範囲に入っている．ただし $|J|$ が 1 か 2 以上かで $|\Delta|$ の増減則がドラスティックに変わるため，特にビット位置が未知の場合は 2 つ以上のビットのプローブが効率的であると考えられる．

#### III.2.2.2 プローブ解析の適用例

##### III.2.2.2.1 RSA

RSA [RSA78] における処理手続きを図 III.6 に示す.

| | |
|---|---|
| 秘密鍵: | $p$, $q$, $d$ |
| 公開鍵: | $n$, $e$ |
| | $n = pq$ |
| | $(e,(p-1)(q-1)) = 1$ |
| | $ed = 1 \bmod (p-1)(q-1)$ |
| 暗号化: | $C = M^e \bmod n$ |
| 復号化: | $M = C^d \bmod n$ |

図 III.6: RSA の処理

RSA の復号化では秘密鍵 $d$ を指数とするべき乗剰余算が行われるため，III.2.2.1 節で述べた SM-1 への攻撃をそのまま用いることで RSA に対するプローブ解析を実行することができる．復号化のべき乗剰余算では暗号文 $C$ と法 $n$ とが既知であり，プローブ解析から秘密鍵 $d$ を直接求めることが可能である.

##### III.2.2.2.2 DSA

DSA (Digital Signature Algorithm) [DSA95] における処理手続きを図 III.7 に示す.

前述の RSA とは異なり，DSA では秘密鍵 $x$ が署名手続きにおけるべき乗剰余算の指数として用いられていないため，(i) の $r$ の計算にプローブ解析を適用する．$r$ の計算では指数 $k$ 以外の $p, q, g$ は既知であり，III.2.2.1 節で述べた SM-1 へのプローブ攻撃を実行して $k$ を求めることができる．一方 (ii) より

$$x = r^{-1}(sk - H(m)) \bmod q$$

であり，署名 $(r,s)$ およびメッセージ $m$ のハッシュ値 $H(m)$ が既知であるため，プローブ解析で求めた $k$ を用いて秘密鍵 $x$ を計算することが可能である.

##### III.2.2.2.3 DES

DES はアルゴリズムを図 III.8 に示したように 16 段の Feistel 型である [DES77].

プローブ解析では各ラウンド毎に $L$ か $R$ どちらか片方のレジスタのビット値を測定する．今，レジスタ $L$ をプローブするとし，1 組の平文と暗号文を用意して最終 16 段目の鍵 $K_{16}$の 6 ビットを求めることを考える．この鍵 $K_{16}$の 6 ビットを $k_{16}$と書き，$k_{16}$と $F$ 関数等で演算処理が行われるレジスタ $L$ のビットを $l_{16}$と書く.

| | |
|---|---|
| 秘密鍵: | $x\ (0 < x < q)$ |
| 公開鍵: | $p, q, g, y$ |
| | $g = h^{\frac{p-1}{q}} \bmod p\ (0 < h < p, g \neq 1)$ |
| | $y = g^x \bmod p$ |
| 署名: | 乱数 $k$ 生成 $(0 < k < q)$ |
| | (i) $r = (g^k \bmod p) \bmod q$ |
| | (ii) $s = (k^{-1}(H(m) + xr)) \bmod q$ |
| 検証: | $0 < r < q$ および $0 < s < q$ でなければ却下 |
| | $w = s^{-1} \bmod q$ |
| | $u_1 = H(m)w \bmod q$ |
| | $u_2 = rw \bmod q$ |
| | $v = (g^{u_1} y^{u_2} \bmod p) \bmod q$ |
| | $v = r$ ならば検証成立 |

図 III.7: DSA の処理

$(L_0|R_0) = IP(m_L|m_R)$
for $i = 1$ to 16
$L_i = R_{i-1}$
$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$
end for
$c = IP^{-1}(R_{16}|L_{16})$

図 III.8: DES のアルゴリズム

まず $R_{16} = L_{15} \oplus F(R_{15}, K_{16})$ においてプローブ解析から $l_{15}$ と $l_{16}(= r_{15})$ が分かっているため，$k_{16}$の推測値を $k'_{16}$として $r'_{16} = l_{15} \oplus F(r_{15}, k'_{16})$ を計算する．一方，暗号文に初期転置 $IP$ をかけることで $r_{16}$の正しい値を得ることができるため，これと $r'_{16}$とを比較することで鍵の推測値 $k'_{16}$が正しかったかどうかを判断することができる．プローブするビット数に依存すると思われるが，文献 [HPS99] では鍵の 6 ビットを求めるには暗号文が 6 つあればよいとしている．

以上の手続きは 1 段目にも同様に適用可能であるため，1 段目と 16 段目からそれぞれ 6 ビットずつ，合計 12 ビットをプローブ解析から得ることができる．そして DES の鍵 56 ビットの残り 44 ビットに関しては総当たり検索を行うことで鍵の全ビットを求める．

このようにしてプローブ解析の DES への適用は可能であるが，triple-DES に関しては必要な総当たり検索数が $2^{100}$となりプローブ解析は現実的ではない．また DES

への攻撃に関しても，プローブするビット位置は任意ではなく求める鍵のビットと対応していなければならない等の技術的困難さが含まれる．

#### III.2.2.2.4 RC5

RC5は平文および暗号文のブロックサイズ$2w$ビット ($w$=16, 32, 64)，鍵サイズ$b$バイト ($0 \leq b \leq 255$)，段数$r$ ($0 \leq r \leq 255$)が指定可能で，RC5-$w/b/r$と表記される [Riv95]．RC5のアルゴリズムを図III.9に示す．$x <<< y$は，$y$の下位$\log w$ビットで表される整数値分だけ$x$を左巡回シフトすることを表す．また図III.9の表記では，1段がRC5本来の1段の半分に相当している．

$L_1 = L_0 + S_0$
$R_1 = R_0 + S_1$
for $i = 2$ to $2r + 1$ do
$L_i = R_{i-1}$
$R_i = ((L_{i-1} \oplus R_{i-1}) <<< R_{i-1}) + S_i$

図 III.9: RC5のアルゴリズム

RC5に対するプローブ解析ではDESへの適用と同様にレジスタ$L$か$R$どちらか片方のビット値が各ラウンド毎にプローブ可能であることを前提条件とし，ここではレジスタ$R$のビット$p$ (上位ビット側から$p$番目)をプローブするとする．また暗号化の前処理としてRC5では秘密鍵から拡張鍵テーブルを作成するが，プローブ解析では秘密鍵そのものではなく拡張された鍵$S_i$の解読を目的とする．RC5への攻撃手順は以下の通りである．

**ステップ 1** 最終段$2r + 1$では$L_{2r} \oplus R_{2r}$が$R_{2r} (= L_{2r+1})$の下位$\log w$ビットの値で左巡回シフトされるため，下位$\log w$ビットの値が$w - p$であるような暗号文$L_{2r+1}$に対しては，$(L_{2r} \oplus R_{2r}) <<< R_{2r}$の最下位ビットがちょうど$p$になる．この最下位ビットの値は$R_{2r-1} (= L_{2r})$および$R_{2r}$へのプローブ解析から知ることができるため，これと暗号文$R_{2r+1}$とから鍵$S_{2r+1}$の最下位ビットの値を求める．

**ステップ 2** 次に下位$\log w$ビットの値が$w - p + 1$である暗号文$L_{2r+1}$に対しては，ビット$p$はシフトによって$(L_{2r} \oplus R_{2r}) <<< R_{2r}$の最上位ビットへ移るため，ステップ1と同様な手続きから鍵$S_{2r+1}$の最上位ビットの値を得る．$S_{2r+1}$の残りの$w - 2$ビットについても異なる暗号文を用いて順次求める．

**ステップ 3** 得られた最終段の鍵$S_{2r+1}$を用いて$L_{2r+1}$および$R_{2r+1}$を復号化し，ステップ1，2と同様な手続きを$S_{2r}$に対して行う．この操作を繰り返すことで$S_4$までの鍵を得る．

**ステップ 4**　残った鍵 $S_3, S_2, S_1, S_0$は 2 段のブロック暗号に対する直接的な解析から求める．

Handschuh らは用意する平文/暗号文は $w$ の数倍程度でよく，解析は複雑でないとしている．しかし DES の場合と同様にプローブするビット位置が既知であることを前提としている点などからは，技術的には困難が伴なうと予想される．

#### III.2.2.3　プローブ解析の対策

以上説明してきたようにプローブ解析は目的とするレジスタへのアクセスが可能な状態を準備することさえできれば，秘密情報の解読はほとんどの暗号方式に対して比較的容易に実行できると考えられる．しかしこれを実現するためには攻撃者は次に挙げるような技術的課題を全てクリアしなければならず，プローブ解析の実現可能性は決して高いとは言えない．

1. IC チップの前処理
   チップ表面の露出，保護膜の除去など
2. デバイス構造の把握
   目的とするレジスタ位置・ビット位置の見極め，各種解析センサの回避など
3. プローブ中の制御
   探針の特性劣化の対策，攻撃する暗号方式の内部ループとの同期など

1 番目に関してはそれほど高度な専門性は必要としない場合があるとしても，2 番目，3 番目に関しては暗号方式およびそれが実装されたデバイスに関する詳細な知識に加え，技術的にもかなりな習熟度が要求されると考えられる．

プローブ解析に対する耐タンパー技術としては，周波数検知回路，電圧検知回路，温度検知回路などのカード用 IC への標準装備が進んでおり，IC チップカタログに低・高電圧センサー，低周波数センサー，コンタクト異常検出機能などの各種セキュリティ回路の搭載を記載しているベンダーもある．

また耐タンパーの新技術として，IC チップに Si コーティングを施すことでプローブ解析のような IC チップへの物理的攻撃を防ぐ技術 (例えば物理的に無理なアクセスをするとチップ自体が破壊される) も開発されている [Pio98]．これは従来の製造工程を用いてコスト効率良く実現でき，リバースエンジニアリングや，プロービング，化学的処理，電子顕微鏡，フォーカスイオンビーム (FIB) などの解析から IC チップを保護するものである．

# III.3　非破壊型解析法

本章では非破壊型解析法として，故障利用解析，タイミング解析，電力解析についてまとめる．

## III.3.1　故障利用解析

### III.3.1.1　故障利用解析の原理

#### III.3.1.1.1　概要

故障利用解析は，IC カード等の耐タンパーデバイスの計算誤りを利用した解析方法である．故障利用解析は，IC カードに一過性の故障あるいは他の機能に影響を与えない範囲の限定的な障害を与え，IC カードに攻撃者の望む異常な処理を行わせる攻撃である．故障利用解析は直接物理的な攻撃を与える場合とは異なり，攻撃対象とする IC カード自体を利用できることを前提とする．

まず Bellcore の Boneh らにより公開鍵暗号系で用いられているべき乗剰余演算等の代数演算を実装した耐タンパーデバイスに対する方法が提案された [BDL96]．この方法では，放射線の照射や高電圧を加えたり，瞬間的にクロック周波数や駆動電圧を変動させることにより故意にエラーを発生させ，その結果得られる誤った計算結果と正しい計算結果から秘密鍵の情報が得られるというものである．

また，Biham らは共通鍵暗号系においても同様に故意にエラーを発生させることにより，秘密鍵の情報が得られるとの考えを提案した．その後も，他の暗号方式への適用や，さまざまな条件下での適用例，IC カード等の耐タンパーデバイスの特性を考慮した攻撃モデルの提案などがなされている．

本文書では，共通鍵暗号の DES に対する 2 種類の故障利用解析，RC5 に対する解析，公開鍵暗号の RSA および ElGamal 署名への適用を紹介する．また，実現可能性や技術課題と対策についても述べる．

### III.3.1.2　DES への適用

本節では，まず Biham らが提案した故障差分攻撃と故障非差分攻撃それぞれの DES への適用を述べる．

#### III.3.1.2.1　故障差分攻撃

Biham と Shamir による故障差分攻撃 (*Differential Fault Analysis*: DFA) は，IC カード等の耐タンパーデバイスにおいて同一の平文を暗号化する処理中に故障を発生させ，エラーを含んだ暗号文と正しい暗号文の差分から，故障の発生位置とビットの誤り内容を推測する方法である [BS97]．DFA により，DES の場合 50〜200 個の

暗号文情報 (平文は知らなくても良い) から全ての鍵ビットを求めることができる．さらに DFA によると，DES の 3 倍の鍵長を持ちより安全である Triple DES であっても，同数の暗号文から 168 ビットのすべての鍵ビット (あるいは，768 ビットの拡大鍵の全ビット) を求めることが出来る．

DFA の手順は以下の通り [BS97, MS97]

**DFA の手順**

1. 耐タンパーデバイスを同じ鍵を用いて，同じ平文を繰り返し暗号化して暗号文を観測する．この処理中に，耐タンパーデバイスに物理的影響を与え，エラーを引き起こす．この結果，複数の誤った暗号化結果と正しい暗号化結果が得られる．
2. 何段目でエラーが起ったか推測する．たとえば 16 段目の F 関数の入力側でエラーが起った場合，出力差分の伝播と，暗号化結果の差分の一致からエラーの発生を絞り込むことができる．
3. 14 段目の右 32 ビットでエラーが起った場合も同様に考えられる．
4. 上記観測により，計算機シミュレーションでは 200 個以下の暗号文から 16 段目に入力される 48 ビットの鍵すべてを導出できる．残りの 8 ビットは全数探索して求めれば良い．

先の例の F 関数の入力での故障発生以外にも F 関数の内部での故障発生も仮定できる．ただし攻撃が成功する確率は同程度である．またデータ暗号化処理部ではなく鍵生成部の内部での故障発生も考えられる．

Biham らによると，故障発生の位置と発生のタイミングを自由に選択できるならば攻撃を効率化できる．たとえば攻撃者が最後の数段で一様に故障を発生させられるなら必要な暗号文数は 10 個である．さらに最も的確な故障発生位置では，必要な暗号文の数は 3 つまで減少させられる．

#### III.3.1.2.2 故障差分攻撃の応用

Biham らは，DFA の種々の応用についても考察を行っている [BS97].

**一時的なハードウェア障害** DFA の一形態として，瞬間的にクロック周波数や駆動電圧を変動させることにより一時的なハードウェア障害を発生させる攻撃がある．設計者はこの攻撃に対しては処理を二重化し暗号化結果の比較を行うという対策をとることができる．しかし，この対策は不十分である．仮に 512 箇所の故障発生位置の候補があったとすると，故障が一致する確率は 1/512 である．つまり，その場合には従来の攻撃では 200 個であった暗号文を 100000 個に増やせば良い．

**拡大鍵への適用** 暗号の実装では on-the-fly ではなく拡大鍵を事前に計算しておくことがある．この時，任意の拡大鍵ビットに故障の影響を与えることが容易で

ある．ただし必要な暗号文の数は変らない．

**処理改竄**　処理がデバイス中のソフトウェアで実行されている場合，故障によりプログラムカウンタやループカウンタを改竄することもできる可能性がある．この場合，より少ないラウンド数で暗号を攻撃することも可能となる．

**鍵スケジュール部への攻撃**　DES の鍵スケジュール部では中間的な鍵にローテート演算が行われ，16 段分の処理で最初の鍵と一致する．16 段処理した後の鍵を次の暗号化の鍵としても用いる場合，DFA によりローテート量を変化させることができれば，相関鍵解読法 (related key cryptanalysis)[Bih94] や相関鍵差分解読法 (differential related key cryptanalysis)[KSW96] が可能となる．また，線形解読法や差分-線形解読法も効率化できる．

**利用モードへの攻撃**　DFA の応用は，暗号の様々な利用モードで用いられる鍵を求めるのにも利用できる．この攻撃は CFB モードへの差分解読法 [PN+93] と同様の効果がある．

**アルゴリズム非公開暗号への適用**　DFA を適用した際の全ビットが 0 である鍵 $k_f$ を 1 ビットずつ変化させた鍵とオリジナルの鍵 $k_0$ の暗号結果の一致を調べる手間を $O(n)$ とすると，$O(n^2)$ で正しい鍵が求まる．さらに，Sbox など演算箇所に DFA を適用することにより，秘密の Sbox の内容を解析することが可能となる．

#### III.3.1.2.3　故障非差分攻撃

故障差分攻撃 (DFA) に対しては，一時的な故障を発生させるのは現実的でないといった批判がある．故障非差分攻撃 (*Non-Differential Fault Analysis*: NDFA) は，電子ビームなどによりシリコン上の回路に操作を加えて CPU のレジスタ上のビットに常に 0(あるいは常に 1) となる誤りを引き起こす攻撃法である [BS97].

NDFA は，DFA とは異なり純粋な意味での暗号文単独攻撃が可能となる．平文も故障が発生せず正常に処理された暗号文も必要としない．NDFA にはループ繰り返し構造を持ったコードへの基本的な攻撃，ループが展開されたコードへの適用などがあり，それぞれ以下の方法で攻撃を行う．

**基本的攻撃**　ループ繰り返し構造を持った DES の IC カードへの実装に対する適用を考える．すなわち DES の暗号化では図 III.10 の 1 段の構造が 16 回繰り返される．

左側のレジスタの LSB を破壊することにより，レジスタの LSB は常に 0 となる．したがって，最終段での処理は図 III.11 で表される．

左側のレジスタの LSB は破壊されているため，$L_{15}$ および $L_{16}$ は常に 0 である．したがって，暗号文の LSB と F 関数出力の LSB が一致する．このビットは S7 の出力であり，S7 への入力の鍵 6 ビットは，およそ 6 個の暗号文から組合わせ

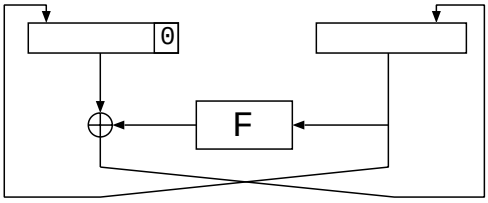


図 III.10: DES のループ繰り返し構造

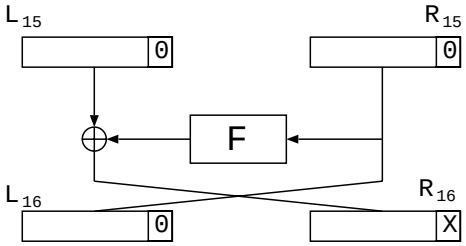


図 III.11: DES の最終段

により求められる．残りの鍵のビットは，故障箇所を追加していくことにより求められる．

**ループ展開型コードへの適用**　高速化などの為にループ展開されたコードの場合，攻撃はより容易となる．一段だけのデータの破壊が可能なため，$L_{15}$の LSB を破壊することにより，2 個の暗号文から対応する鍵が求まる．また，破壊する位置によってはより効率的である．鍵 XOR 後の Sbox の両端のビットを破壊した場合，Sbox による置換が 1 対 1 になるため，出力から入力が一意に求まる．

他の方法として，最初の 2 段以外の 14 段の拡大鍵のレジスタを全て破壊して 0 にした場合，暗号文を拡大鍵 0 で 14 段分復号すれば 2 段 DES の解読と等価となり，容易に解読できる．この攻撃は，他のデータ暗号化部と鍵スケジュール部が独立した暗号にも適用できる．

#### III.3.1.3　RC5 への適用

DES の場合と同様に，RC5 に対する故障利用解析 (DFA) が提案されている [MS97]．提案されている攻撃法では，攻撃者が適用できる DFA のレベルとして，拡大鍵の 1 ビットにビットエラーの発生 (もしくは書き換え) が可能と仮定する．

RC5 はブロック長，段数，鍵長を可変パラメータとしてもつブロック暗号である．2w ビットブロック長，r 段，k バイトの鍵を持つ RC5 を RC5-w/r/k と記する．説明

を容易にするために，図 III.12 の half round の概念を用いると，RC5 アルゴリズムは以下のように記述できる．

$$
\begin{array}{l}
L_1 = L_0 + S_0; \\
R_1 = R_0 + S_1; \\
for\ i = 2\ to\ 2r + 1\ do \\
\qquad L_i = R_{i-1}; \\
\qquad R_i = ((L_{i-1} \oplus R_{i-1}) <<< R_{i-1}) + S_i;
\end{array}
$$

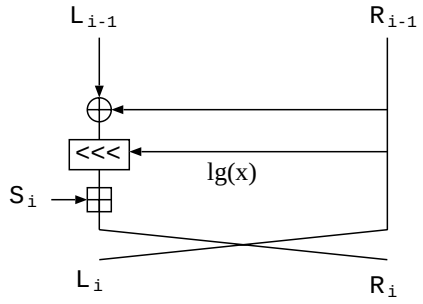


図 III.12: RC5 の half round

ここで，”$\oplus$”は排他的論理和，”$<<<$”はデータ依存ローテーション，”$+$”は加算を表す．

RC5 の拡大鍵生成アルゴリズムはある種一方向性であるため，一部の拡大鍵から他の拡大鍵を得ることは困難である．また，全ての拡大鍵が分かったとしても元の鍵を得ることは困難である．DFA による攻撃では，元の鍵は求められないが，暗号化あるいは復号に用いられている全ての拡大鍵を求めることを目的とする．

RC5 への DFA による攻撃は以下の方針で拡大鍵を導出する．

- 拡大鍵 $S_i$の t ビット目に起こしたビットエラーによる DFA の影響を受けた出力 $L'_i, R'_i$を元に，$S_i[t]$ を決定する．
- 拡大鍵 $S_{i-1}$の t ビット目に起こしたビットエラーによる DFA の影響を受けた出力 $L'_i, R'_i$を元に，$S_{i-1}[t]$ を決定し，さらに $S_i$の候補を絞り込むための情報を得る．

拡大鍵の推定には，正しい i 段目の出力 $L_i, R_i$と DFA による影響を受けた出力 $L'_i, R'_i$のどのビットが異なっているかの情報を元に，ビット誤りの位置と加算のキャリーの伝播を考慮しつつ場合わけと検証を繰り返し，拡大鍵 $S_i$の値の候補を絞って行く．攻撃手順と場合わけの詳細については文献 [MS97] で発表されているが，非常に複雑でありここでは解説しない．

#### III.3.1.4 RSA への適用

RSA のべき乗剰余算における故障利用解析 (DFA) を述べる．RSA は共通の法を用いて互いに素な $e_1$と $e_2$を用いて同一のメッセージ $m$ を暗号化した場合，メッセージ $m$ が導出できてしまうことが知られている．メッセージ $m$ の暗号文をそれぞれ $c_1 = m^{e_1} \bmod n$, $c_2 = m^{e_2} \bmod n$ とすると，$gcd(e_1, e_2) = 1$ より $ue_1 + ve_2 = 1$ を満たす $u, v \in \mathcal{Z}$ が存在し，$m$ は次式で求められる．

$$m = m^{ue_1+ve_2} \equiv c_1^u c_2^v \bmod n$$

RSA に対する DFA の適用では，DFA によって同様の状況を生じさせることにより，RSA への攻撃が可能となる [JQ97].

べき指数 $e$ の二進化表現を $e = \sum_{i=0}^{t-1} e_i 2^i$とする．$j$ ビット目にエラーを引き起こしビットを反転させたとする．メッセージ $m$ を暗号化した結果は本来 $c = m^e \bmod n$ であるが，ビットのエラーにより$\widehat{c} = m^{\widehat{e}} \bmod n$ となる．ここで，

$$\widehat{e} = \begin{cases} e + 2^j & e_j = 0 \text{ のとき} \\ e - 2^j & e_j = 1 \text{ のとき} \end{cases}$$

である．また，$\delta = gcd(\widehat{e}, e)$ とすると，$\delta = gcd(e \pm 2^j, e)$ より$\delta$は $2^j$の約数である．RSA では $e$ は奇数であるため，$\delta = 1$ が成り立つ．すなわち$\widehat{e}, e$ は互いに素である．

したがって，共通の法を用いた場合同様に $c$ と$\widehat{c}$からメッセージ $m$ の導出が可能である．また，$e$ を 2 ビット以上変化させる場合この攻撃法が適用出来るかは $gcd(\widehat{e}, e)$ に依存する．ただし $e$ に素数を用いた場合，この攻撃は (MSB より大きなビットを操作しないならば) 常に適用できる．

#### III.3.1.5 ElGamal への適用

故障利用解析を利用して ElGamal 署名に使用された秘密鍵を導出する方法を述べる．以下の ElGamal 署名において，乱数 $x$ を直接操作することにより，RSA 同様に秘密鍵 $x_a$を導出することができる [And96, BD+96].

まず，DSA や Schnorr 署名などでも用いられている ElGamal 署名方式を説明する．メッセージ $m$ に対する ElGamal 署名は，署名用の秘密鍵 $x_a$と乱数 $x$ を用い，次の $r$ と $s$ が署名となる．

$$\begin{aligned} r &= g^x \bmod p \\ s &= \frac{hash(m) - x_a \cdot r}{x} \bmod (p-1) \end{aligned}$$

ここで，乱数 $x$ は，$(p-1)$ を $x$ で割り切れない必要がある．また，乱数 $x$ はメッセージごとに毎回異なる独立な値を用いなければならない．

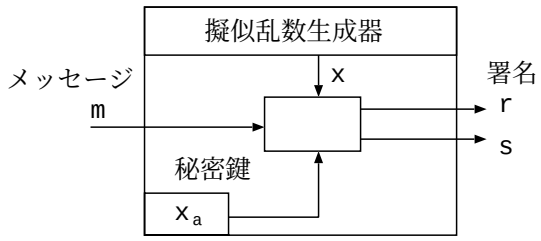


図 III.13: ElGamal 署名

ElGamal 署名で用いる乱数生成に着目すると，擬似乱数生成がソフトウェア (カード上のコード) で行われる場合の乱数直接操作攻撃や，ハードウェアによる擬似乱数生成の場合の乱数固定攻撃などが考えられる [ZM97].

これら IC カード中の擬似乱数生成器を操作する攻撃方法に対しては，フォールトトレラント技術を用いたり，擬似乱数生成器の生成結果の検証を行うことにより擬似乱数生成器の改竄を困難にすることができ，耐タンパーな LSI を設計するのと同じぐらい重要であると指摘している.

**乱数直接操作攻撃 (ソフトウェア)** 擬似乱数生成器の内部ステータスを操作することにより，

1. 擬似乱数生成の内部ステータスを保持しているメモリの位置を探す (鍵など重要な情報は RAM の若いアドレスに配置されることが多いと言われている [And96]).
2. 物理的ストレスなどにより，擬似乱数生成器の元の内部ステータス $S_{info}$ を全ビット 1 など自明な値に置き換える.
3. 署名用鍵 $x_a$ を次の手順で求める.
   (a) 既知の内部状態 (全ビット 1 など) の擬似乱数生成器から生成される乱数 $x_0$ を計算する.
   (b) 既知の乱数 $x_0$ より署名鍵 $x_a$ を計算する.

$$x_a = \frac{hash(m) - s \cdot x_0}{r} \bmod (p-1)$$

**乱数直接操作攻撃 (ハードウェア)** 定格よりも低い駆動電圧を与えたり，定格外の温度環境など，ハードウェアの擬似乱数生成器は異常な環境では予測可能な乱数を生成することがある [AK]. この場合，乱数を直接操作できるため，以下のように署名用の鍵が求まる.

$$x_a = \frac{hash(m) - s \cdot x_0}{r} \bmod (p-1)$$

**乱数固定攻撃**　これまでに紹介した方法では，既知の乱数値を用いていたが，固定乱数値 (値は既知でない) を用いても攻撃ができる．攻撃者は固定の乱数 $x_0$を用いて二つのメッセージ $m_1$と $m_2$に署名を行わせる．すなわち，

$$r_0 = g^{x_0} \bmod p, s_1 = \frac{hash(m_1) - x_a \cdot r_0}{x_0} \bmod (p-1)$$

$$r_0 = g^{x_0} \bmod p, s_2 = \frac{hash(m_2) - x_a \cdot r_0}{x_0} \bmod (p-1)$$

$m_1$と $m_2$に対する署名の結果から $x_0$が求まる．

$$x_0 = \frac{hash(m_1) - hash(m_2)}{s_1 - s_2} \bmod (p-1)$$

したがって，署名に用いた秘密鍵は

$$x_a = \frac{hash(m_1) - s_1 \cdot x_0}{r_0} \bmod (p-1)$$

この乱数を固定させた攻撃方法は，擬似乱数生成器を操作し既知の乱数を出力させる攻撃方法よりも適用が容易であると思われる．

#### III.3.1.6　故障利用解析の対策

本節では故障利用解析の実現可能性と対策についてまとめる．

以上説明してきたように故障利用解析は，攻撃者が望む一過性の故障を IC カードに的確に与える必要がある．

故障利用解析では，鍵などの情報を保持しているメモリの物理的な位置の把握と改竄が必要となる．耐タンパーデバイスに対してこのような高度な改竄が可能であるならば，同様にプローブ解析なども不可能ではないと思われる．また，故障利用解析では IC カードを利用できる状態に保ったまま攻撃を行わなければならないため，いくつか技術的課題をクリアしなければならない．たとえば，レーザ照射などはカードにチップが搭載された状態のまま行えなければならない．また，その他の技術的課題をいくつか挙げると

1. 故障発生の手段
   放射線やレーザ照射といった故障発生手段には専用の設備が必要となる．
2. 不可逆障害の影響
   Biham らの故障非差分攻撃では，暗号アルゴリズムの利用するメモリに不可逆的な障害を与えているが，暗号アルゴリズムのコードが中間データの保持にワークメモリを利用している場合，他のアプリケーションにも影響が及ぶ．

3. 故障の発生のタイミング
   一過性の故障を適切なタイミングで，適切な時間 (数$\mu$秒) だけ発生させなければならない．耐タンパーデバイスへのクロックが内部で供給されている場合，困難が伴うと思われる．
4. 攻撃位置の特定
   配線の多層化やアドレススクランブルなどが行われている耐タンパーデバイスに対して攻撃すべき RAM や ROM の位置を的確に把握する技術が必要である．
5. 故障の発生方法
   たとえばチップが多層である場合，破壊的な解析では上層を除去することで対応できるが，故障利用解析では上層に障害を与えることなく目的の箇所を操作できる必要がある．
6. メモリの特性
   ECC (Error Correcting Code) メモリが使われることは稀であるが，RAM や PROM や EEPROM など書き込み可能なメモリについてはパリティチェック機能がある．
7. クロック/電圧/温度変動
   現在の IC カードには，クロックや電圧や温度等の動作環境検知回路を搭載し，異常を検出したならばカードを停止あるいは使用不能にする機能をもったものもある．

対策としては，プローブ解析への対策と同様の対策が有効である．外界からの影響を取り除くため，周波数検知回路，電圧検知回路，温度検知回路などを搭載し，誤動作を引き起こす異常な環境での使用を許さないなどの対策が必要である．また，レーザによる任意の回路形成や破壊などを防ぐには多層配線など耐タンパー技術が有効である．

内蔵プログラムに関しては，故障利用解析は内部状態や中間データを保持するメモリへの攻撃を行うため，その処理固有のワーク領域の使用を避けることも有効である．ワーク領域を共有することにより，特定の処理以外にも改竄の影響を波及させ，正常な IC カードとしては利用できなくする効果もある．

共通鍵暗号に関しては，鍵スケジュール部の処理を事前に行い拡大鍵を保持する方法よりも，旧来の RAM 領域の少ない IC カード等で用いられていた on-the-fly 鍵生成の方が解析が困難となる．拡大鍵を ROM に保持することは避ける方が望ましい．

また，実装について IC カード設計者は，RAM の若いアドレスに重要な情報を配置する傾向があるとの指摘がある [And96]．後方のアドレスはスタック領域として用いられるため，自ずと前方に変数などが配置されることが多いと思われるが，安全性を高めるために対応が望まれる．

## III.3.2　タイミング解析

### III.3.2.1　タイミング解析の原理

タイミング解析は，暗号化の処理時間が秘密鍵データに依存して異なるとき，その差を統計的に解析して秘密情報を推定する攻撃法である．1996 年に Kocher によってそのアイディアが発表され，同時に Diffie-Hellman [DH76]，RSA [RSA78] などのべき乗剰余算，中国剰余定理を用いた RSA，さらに DSS (Digital Signature Standard) についての攻撃法の原理が提案された [Koc96]．その後，タイミング解析によって実際に秘密情報を推定した報告もなされている [DK+98]．

タイミング解析は，暗号アルゴリズム自体ではなく，スマートカードへの不注意な実装方法に対して行われる攻撃法である．暗号アルゴリズムを実装する場合，処理時間の短縮やプログラムサイズの縮小を狙って，処理の最適化を行うことが多い．最適化とは，例えばデータのあるビットが 1 であるか 0 であるかによって，行う必要のない処理をスキップしたり，分岐して異なる処理を行ったりすることである．このような方法で実装すると処理時間がデータに依存して異なってくる．そのため処理時間を見ることで秘密情報を推定することができる．これがタイミング解析の原理である．

タイミング解析を行う上で，次の仮定をおく．

- 秘密情報は攻撃中は不変である
- 攻撃者は実装されているアルゴリズムを知っている
- 攻撃者は暗号化 1 回の処理時間を正確に計測可能である

上のような状況のとき，攻撃は次のように行われる．

1. 実装アルゴリズム中で，処理時間差を生じる部分のうち 1ヶ所に注目する．
2. 計測できるのは暗号化全体の時間だけなので，注目した部分以外の時間差が (統計的に) 無視できるまでの量のメッセージ (平文) を入力し，暗号化の処理時間を計測する．
3. その計測データから秘密鍵情報を推定する．

III.3.2.2 節以降で，現在までに報告されている各暗号方式への適用例を述べる．

### III.3.2.2　タイミング解析の適用例

#### III.3.2.2.1　RSA, Diffie-Hellman

Dhem らによって提案された，RSA, Diffie-Hellman で用いられているべき乗剰余算 $m^d \bmod n$ へのタイミング解析の適用を説明する [DK+98]．ここで $n$ は公開鍵，$m$ はメッセージ，$d$ は秘密鍵であり，攻撃者から見たパラメータの関係は表 III.2 のとおりである．

表 III.2: 攻撃者から見たパラメータ (べき乗剰余算)

| | $m$ | $n$ | $d$ |
|---|---|---|---|
| 既知／未知 | 既知 | 既知 | 未知 |
| 入力パラメータ | ○ | | |
| 解析対象パラメータ | | | ○ |

べき乗剰余算は通常，バイナリ法を用いて剰余乗算の繰り返しで実現されている．さらに剰余乗算の高速化手法として知られる Montgomery の方法 (Montgomery 乗算と呼ぶ) を用いていると仮定する．図 III.14 に Montgomery 乗算 $mont(x, y, n)$ の，図 III.15 に Montgomery 乗算を用いたべき乗剰余算のアルゴリズムを示す．この節では図 III.15 の $x = mont(x, x, n)$，$x = mont(x, m', n)$ をそれぞれ単に 2 乗算，乗算と呼ぶことにする．

入力：$x, y, n$
出力：$w = xyR^{-1} \bmod n$ （$R$：Montgomery 定数）

$w = xy + (xy(-n^{-1}) \bmod R)n$
$w = w/R$
$If\ (w > n)\ then$
　$w = w - n$　(減算 A)
$return\ w$

図 III.14: Montgomery 乗算 mont(x,y,n)

図 III.14から分かるとおり，Montgomery 乗算には条件分岐があり，減算 $w = w - n$ が 1 回多く行われる場合がある．したがって Montgomery 乗算はデータによって処理時間が異なるため，タイミング解析の対象となる．以後この減算を減算 A と呼ぶことにする．

秘密情報 $d$ の推定は MSB (Most Significant Bit, 最上位ビット) から順に 1 ビットずつ進めていく．$d_{k-1}$ から $d_{k-i+1}$ までの $(i-1)$ ビットを知っているときに $d_{k-i}$ の値を推定する方法を述べる．

まず，メッセージ $m$ を入力して最初の $(i-2)$ 回のループを行い，$d_{k-i}$による乗算の手前まで計算が進んだ状況を考える．そのときの $x$ の値を $x_{temp}$ とする．$d_{k-1}$から $d_{k-i+1}$までの値を知っていると仮定しているので，攻撃者は $x_{temp}$の値を知ることができる．

次にメッセージ全体の集合を，$d_{k-1}$の値と，Montgomery 乗算での減算 A の有無により 4 つに分ける．

$d_{k-i} = 1$ のとき．次に行われる演算は乗算と 2 乗算である．

入力：$n, m, d$ $(d = (d_{k-1}d_{k-2}\cdots d_0)_{(2)}, d_{k-1} = 1)$
出力：$x = m^d \bmod n$

$m' = mR \bmod n \quad (R$：Montgomery 定数)
$x = m'$
$For\ i = k-2\ downto\ 0$
　$x = mont(x, x, n)$
　$If\ (d_i == 1)\ then$
　　$x = mont(x, m', n)$
$Endfor$
$x = mont(x, 1, n)$
$return\ x$

図 III.15: 左バイナリ法+Montgomery 乗算によるべき乗剰余算

1. $x = mont(x_{temp}, m', n)$
2. $x = mont(x, x, n)$

2 乗算で減算 A が行われるメッセージ $m$ の集合を $M_1$，行われないメッセージの集合を $M_2$とする．

$d_{k-i} = 0$ のとき．乗算は行われずに 2 乗算 $x = mont(x_{temp}, x_{temp}, n)$ のみが行われる．この 2 乗算で減算 A が行われるメッセージ $m$ の集合を $M_3$，行われないメッセージの集合を $M_4$とする．

このとき，$M_1$と $M_2$の間の処理時間の差が大きければ $d_{k-i} = 1$ であり，$M_3$と $M_4$ の間の処理時間の差が大きければ $d_{k-i} = 0$ であると推定できる．

この方法を実際に適用した結果，128 ビットべき乗剰余算では約 10,000 個のメッセージで，512 ビットべき乗剰余算は約 350,000 個のメッセージで秘密鍵 $d$ が完全に復元でき，その復元の速度は，前者が 4 ビット/秒，後者が 1 ビット/分であったと報告されている [DK+98].

#### III.3.2.2.2 Rijndael

Koeune らに提案された Rijndael へのタイミング解析の適用を説明する [KQ99].

Rijndael の暗号化アルゴリズムの概要を図 III.16 に示す [DR98].

暗号化は 1 番目の拡大鍵と加算を行った後にラウンド関数に入る．ラウンド関数は，ByteSub, ShiftRow, MixColumn, AddRoundKey と呼ばれる 4 つの変換部からなる．それぞれバイト単位での変換であり，ByteSub は S-box による置換，ShiftRow はある定数によるシフト，AddRoundKey は拡大鍵との加算を行うもので，処理時間はデータによらず一定となる．残りの MixColumn では，処理の中に GF($2^8$) 上での'02' 倍算がある．(ここで，GF($2^8$) の元を'02' のように表している．) Rijndael で

| |
|---|
| 入力：平文，拡大鍵 |
| 出力：暗号文 |
| AddRoundKey() |
| *For* $i = 1$ *to* $N_r - 1$　($N_r$: ラウンド数) |
| 　ByteSub() |
| 　ShiftRow() |
| 　MixColumn() |
| 　AddRoundKey() |
| **Endfor** |
| ByteSub() |
| ShiftRow() |
| AddRoundKey() |

図 III.16: Rijndael の暗号化アルゴリズム

は，GF($2^8$) を定義する多項式を $x^8 + x^4 + x^3 + x + 1$ としているため，'02' 倍算を次のように行うことができる．

1. 1 ビット左シフト
2. キャリーが発生したら，'1B' と排他的論理和 (XOR) をとる

このように実装したとき，キャリーが発生したかどうかで XOR が行われるので，処理時間にデータに依存した差が生じる．これは'02' 倍算されるバイトの最上位ビットの値に依存している．

タイミング解析により 1 番目の拡大鍵 ($R_1$とする，最初の AddRoundKey で平文に加算される拡大鍵) を推定する方法を述べる．

平文を入力し，第 1 ラウンドの MixColumn での'02' 倍算の手前まで計算が進んだ状況を考える ('02' 倍算は平文のすべてのバイトに対して行われるが，任意に選んだある 1 つのバイトに注目する)．ByteSub，ShiftRow での変換は暗号化鍵に依存していないので，'02' 倍算されるバイトの値は $R_1$が分かれば攻撃者は知ることができる．逆に'02' 倍算されたバイトの値が分かれば $R_1$を知ることができる．

平文のうち，第 1 ラウンドの'02' 倍算に影響するバイトの値が $j$，その他のバイトはランダムであるようなものの集合を $S_j$とする．$S_j$に含まれる平文を多数入力して暗号化時間を計測することで，'02' 倍算において'1B' との XOR が行われたかどうか推定できる．つまり'02' 倍算されるバイトの MSB，したがって $R_1$のある 1 ビットの値を推定できる．$k \neq j$ なる $S_k$ に対して同じことを行い，$R_1$の別の 1 ビットを推定できる．これを繰り返すことで $R_1$を完全に推定できる．

$R_1$以外の拡大鍵についても同様であり，すべての拡大鍵を推定することができる．

128 ビットブロック，128 ビット鍵の場合にこの攻撃を適用した結果，鍵 1 バイトあたり約 3000 個の平文で鍵を完全に復元できた，と報告されている [KQ99].

#### III.3.2.2.3 その他

以下の例は Kocher による攻撃法の原理である [Koc96]．しかし，実際に攻撃に成功したという報告はなく，実用的であるかどうかは分かっていない．

##### III.3.2.2.3.1 中国剰余定理を用いた RSA

中国剰余定理を用いた RSA の処理手順は図 III.17 のとおりであり，攻撃者から見たパラメータの関係は表 III.3 のとおりである．

| |
|---|
| 入力：$m, p, q, d_p, d_q, A(= p^{-1} \bmod q)$ |
| 出力：$x = m^d \bmod n$ |
| $m_p = m \bmod p$ |
| $m_q = m \bmod q$ |
| $s_p = m_p^{d_p} \bmod p$ |
| $s_q = m_q^{d_q} \bmod q$ |
| $x = ((s_q - s_p)A \bmod q)p + s_p$ |

図 III.17: 中国剰余定理を用いた RSA

表 III.3: 攻撃者から見たパラメータ (中国剰余定理を用いた RSA)

| | $m$ | $p$ | $q$ | $d_p$ | $d_q$ | $A$ |
|---|---|---|---|---|---|---|
| 既知／未知 | 既知 | 未知 | 未知 | 未知 | 未知 | 未知 |
| 入力パラメータ | ○ | | | | | |
| 解析対象パラメータ | | ○ | ○ | | | |

中国剰余定理を用いた RSA の処理でもべき乗剰余算があるが，今回は法が秘密鍵で攻撃者にとって未知のため，III.3.2.2.1 節で述べた方法は使えない．

注目するステップは最初の 2 ステップである．$m_p = m \bmod p$ という除算の処理時間は $m > p$ か $m < p$ かで処理時間が異なる．前者のほうが後者より処理時間が大きい．そのため，その処理時間を計ることで $p$ の大きさをある程度絞り込むことができる．$q$ についても同様である．

##### III.3.2.2.3.2 DSS

DSS (Digital Signature Standard) の処理手順は図 III.18 のとおりであり，攻撃者から見たパラメータの関係は表 III.4 のとおりである．

| |
|---|
| 入力：$m, p, q, g, x$ |
| 出力：$r, s$ |
| 乱数 $k$ を生成 |
| $r = (g^k \bmod p) \bmod q$ |
| $s = k^{-1}(h(m) + xr) \bmod q$ |

図 III.18: DSS

表 III.4: 攻撃者から見たパラメータ (DSS)

| | $m$ | $p$ | $q$ | $g$ | $x$ | $k$ |
|---|---|---|---|---|---|---|
| 既知／未知 | 既知 | 既知 | 既知 | 既知 | 未知 | 未知 |
| 入力パラメータ | ○ | | | | | |
| 解析対象パラメータ | | | | | ○ | |

秘密鍵データ $x$ を推定するために注目するステップは $s$ の計算である．$s$ の計算は通常，$k^{-1} \bmod q$ と $(h(m)+xr) \bmod q$ との剰余乗算で行われるが，$(h(m)+xr) \bmod q$ の処理時間のほうに注目する．ここで $h(m)$ は $m$ のハッシュ値であり，攻撃者はその処理時間を計算することができる．また，この値は $q$ 程度であるため，$(h(m)+xr) \bmod q$ の処理時間はほとんど $xr \bmod q$ とみなせる．$r$ は署名の一部であり攻撃者にとって既知である．したがって，$xr \bmod q$ の処理時間から，$xr$ の値をある程度絞ることができる．すると $x$ の上位ビットを推定することができる．$x$ の上位ビットが分かると $xr$ の上位ビットも分かるため，さらに $x$ の次の上位ビットの値を推定していくことが可能である．このように繰り返していくことで，$x$ を推定することが可能となる．

#### III.3.2.3 タイミング解析の対策

タイミング解析は暗号アルゴリズムの実装方法に対する攻撃法なので，慎重に実装することで防ぐことができると考えられる．実装の際のタイミング解析への対策として 3 つの方法を挙げる．また，この対策を行っているかがタイミング解析への耐性の 1 つの判断基準になる．

- すべての演算がデータによらず一定時間で終わるようにする
- 時間計測を正確に行えないようにする
- ブラインド署名のアルゴリズムの応用

1 番目の対策は最も基本的かつ重要である．これは実装アルゴリズムから分岐処理をなくすことで実現できる．例えば Rijndael での'02' 倍算は，以下のような手順で行えば分岐がなくなり，常に同じ時間で処理できる．ここで各レジスタは 8 ビッ

トであるものとする．

| | |
|---|---|
| 入力：$A$ | |
| 出力：'02'$A$ | |
| $A \ll 1$ | (キャリーを$C$とする．$C = 0$ *or* $1$) |
| $B = (-C)$ & '1B' | ($-C = 0$ or 'FF') |
| **return** $A \oplus B$ | ($\oplus$はXORを表す) |

図 III.19: 変更した'02'倍算アルゴリズム

2番目の対策は，例えばランダムに遅延を挿入し，暗号化の処理時間を実際の時間と異なるように見せることで実現できる．

3番目の対策のブラインド署名のアルゴリズムの応用とは，内部の処理において，攻撃者が入力したメッセージとは異なるメッセージを用いる方法である．例えばRSAで$x = m^d \bmod n$を計算するときに

1. $M = mv_i \bmod n$
2. $X = M^d \bmod n$
3. $x = Xv_f \bmod n$

で計算するようにする．ここで$(v_i, v_f)$は$v_f^{-1} = v_i^d$をみたすもので，秘密に保持しておく．このようにすると攻撃者の入力したメッセージ$m$とは別の$M$がべき乗計算時に用いられているため，前節までに述べた攻撃法が適用できなくなる．$(v_i, v_f)$は暗号化のたびに変更することが望ましく，暗号化を行うときに，$v_i = v_i^2 \bmod n, v_f = v_f^2 \bmod n$を行って更新していく方法が提案されている[Koc96]．

## III.3.3 電力解析

電力解析は消費電力の変化に着目して鍵や処理内容を解析しようという方法であり，単純電力解析(Simple Power Analysis: SPA)，電力差分解析(Differential Power Analysis: DPA)，高次電力差分解析(High-Order Differential Power Analysis: HO-DPA)というカテゴリに大別される．ここでは電力解析のDES，AES候補，RSA，楕円曲線暗号に対する適用について解説し，電力解析への対策を最後にまとめる．

### III.3.3.1 DESに対する電力解析

#### III.3.3.1.1 単純電力解析

本節では，Kocherら[KJJ98, KJJ99]が発表した単純電力解析について，彼らがスマートカードに対して実際に行った単純電力解析の原理と測定結果，および攻撃

の有効性についてまとめる．対処方法については III.3.3.5.3.1 節を参照のこと．

#### III.3.3.1.1.1　解析原理

スマートカードなどの耐タンパデバイスのほとんどはトランジスタで構成された論理回路からなり，ゲートに電圧が加えられたときに電流が流れ，電力が消費される．一般に回路の消費電力は，実行している演算と用いられているデータの値に関係する．例えば，乗法演算は 0 を書き込む場合よりも 1 を書き込む場合のほうが消費電力が大きくなり，乗法演算と平方演算ではそれぞれ異なる電力を消費する．これより秘密情報を用いた演算を行っているデバイスの消費電力の変化を観察することで，秘密情報に関してハミング重みなどの情報を得ることができ，エントロピーを小さくすることができる．消費電力の変化を直接解析に用いる方法を単純電力解析と呼ぶ．

#### III.3.3.1.1.2　測定

デバイスの消費電力は，デバイスと電源または接地との間に抵抗を直列に挿入し，抵抗を流れる電流値から求めることができる．実際に DES の演算を行っているスマートカードに対して消費電力測定を行うと，測定波形より DES の 16 段の演算がはっきり確かめられる．さらに，消費電力波形を詳しく解析することにより鍵レジスタの交換等の情報を得ることができる．

#### III.3.3.1.1.3　評価

このように単純電力解析はデバイスが実行している一連の演算を反映するため，例えば次のような演算から情報を得ることができる．

**鍵スケジュール**　DES の鍵生成部では各段においてビットシフトを行う．このときシフトされる鍵のビットが”0”か”1”かで電流がそれぞれ異なる経路を流れる場合，消費電力の違いから単純電力解析が有効となる．

**転置**　DES のアルゴリズムでは多くの置換が行われる．置換されるデータによって置換アルゴリズムが異なる場合，置換時の消費電力の違いから情報を得ることができる．

**比較**　記号列や数値の比較を行ってミスマッチが生じた時，条件分岐がしばしば起きる．このような条件分岐は消費電力の大きな違いとなって外から観察可能である．

**乗算**　乗法剰余演算回路は演算するデータについて，非常に多くの情報を外に漏らす．具体的にそれがどのような消費電力の変化となるかは回路構成によるが，オペランド値やハミング重みなどの情報が得られることが知られている．

**冪乗演算**　単純な冪乗剰余演算は冪指数を上位桁から順にスキャンしていき，ビット値が”1”のとき乗算を行いながら平方演算を繰り返す．もし，平方演算と乗

法演算がそれぞれ異なる電力を消費したり，異なる計算時間をとったり，異なる計算回路を用いる場合，冪指数についての情報を得ることができる．

#### III.3.3.1.2　電力差分解析

本節では，Kocher ら [KJJ98, KJJ99] が発表した電力差分解析について解析の原理，詳細，測定例，解析の改善法についてまとめる．対処方法については III.3.3.5.3.1 節を参照されたい．

##### III.3.3.1.2.1　解析原理

耐タンパデバイスの消費電力は一般に演算内容と演算に用いられている秘密情報に依存する．しかしこれらの内容に依存した消費電力の変化は小さく，測定誤差やノイズなどから見分けることは一般に困難である．

そこで Kocher らは大量の測定値の平均をとって測定誤差やノイズなどの影響を小さくし，全データの平均値との差分を取ることで演算プロセスによる電力消費の影響を除いて，用いられる秘密情報による消費電力の変化のみを取り出す方法 (電力差分解析) を提案した．

彼らは DES に対する適用例を示している．まず，第 1 段 (または第 16 段) に入る鍵の一部のビットについて予想し，第 1 段 (または第 16 段) の最後にメモリに書きこまれる (と予想される) データの 1 ビットの値に注目して，その値に従って消費電力の観測データを分類する．次にそれぞれのグループについて測定値の平均をとり，それらの差分をとる．予想が正しい場合注目したビットが演算に用いられるとき消費電力の差分が大きくなる．予想が異なる場合目立った差分は確認されない．

##### III.3.3.1.2.2　解析の詳細

1. $m$ 回暗号化プロセスを観測し，それぞれ第 16 段の消費電力の変化 $T_1, \cdots, T_m$ を観測する．さらに，暗号文 $C_1, \cdots, C_m$ を記録する．第 16 段の消費電力の変化を解析に用いる場合，平文の情報は必要ない．なお， $m$ は 1000 程度で十分である．
2. 鍵に依存した分配関数 $D(K_s, C)$ を選択する．ただし， $K_s$ は何らかの鍵情報，$C$ は暗号文である．例えば，最終段の S ボックス 1 の出力の，1 ビット目の値に着目し，S ボックス 1 に供給される 6 ビットの部分鍵を推定する場合を考える．この場合，関数 $D$ は次式で与えられる．

$$D(C_1, C_6, K_{16}) = C_1 \oplus SBOX_1(C_6 \oplus K_{16})$$

ただし $K_{16}$ は第 16 段に S ボックス 1 に供給される 6 ビットの部分鍵の予想値，$C_6$ は $K_{16}$ と XOR される暗号文の 6 ビット， $SBOX_1(x)$ は S ボックス 1 に 6

ビット $x$ が供給された場合の出力結果の 1 ビット目， $C_1$ は $SBOX_1$の出力結果に XOR される暗号文の 1 ビットである．

3. 関数 $D$ を用いて $T_1, \cdots, T_m$ を 2 つのグループに分ける:

$$\begin{aligned} S_0 &= \{T_i | D(\cdot,\cdot,\cdot) = 0\} \\ S_1 &= \{T_i | D(\cdot,\cdot,\cdot) = 1\} \end{aligned}$$

次に，それぞれのグループについて消費電力の平均値を取る．

$$\begin{aligned} A_0 &= \frac{1}{|S_0|} \sum_{T_i \in S_0} T_i \\ A_1 &= \frac{1}{|S_1|} \sum_{T_i \in S_1} T_i \end{aligned}$$

ただし， $|S_0| + |S_1| = m$ である．

4. $A_0$ と $A_1$ の差分をとり，電力差分信号 $\Delta_D$ を得る．

$$\Delta_D = A_0 - A_1$$

5. 部分鍵の予想値 $K_s$ が正しくない場合， $D(\cdot,\cdot,\cdot)$ は暗号文に対してほぼランダムに"0"と"1"を出力する．従って十分多くのサンプルを取ると $\Delta_D$ の値は 0 に近づいていく．(実際には正しい予想値 $K_s$ との相互作用のため， $\Delta_D$ の波形は完全にはフラットにならない．) $K_s$ が正しい場合は， $D(\cdot,\cdot,\cdot)$ は注目したビットの実際の値と同じ値を取るため， $m \to \infty$ とすることで $\Delta_D$ は注目したビットを用いるときに消費する電力に近づいていく．他のデータ値や測定誤差など $D(\cdot,\cdot,\cdot)$ に依存しないものは 0 に近づいていく．消費電力はデータのビット値に依存するため， $D(\cdot,\cdot,\cdot)$ の波形は注目したビットが用いられる領域でパルスを見せ，それ以外の領域では平坦になる．
6. 以上を繰り返し，S ボックス 1 に供給される部分鍵を推定する．反復の最大値は $2^6 = 64$ 回である．
7. 同様の作業を残り 7 つの S ボックスについて行い，秘密鍵について 48 ビットの情報を得る．残りの 8 ビットの鍵情報は全探索によって求める．

#### III.3.3.1.2.3 高次電力差分解析

ここで高次電力差分解析について解説する．

上に述べた電力差分解析はサンプルの一つのイベントに基づいた情報に対して解析を行っているが，高次電力差分解析は複数のイベントに基づいた情報を関連付けて解析に用いる．分配関数 $D$ はサンプルごとにそれぞれ異なる重み付けをしたり，2 つ以上のグループ分けをしたりすることができる．そのような関数は多くの防御策を封じ，平文や暗号文の情報が不完全な場合でも解析が可能な場合がある．また，特徴的な統計的性質を持つサンプルに対しては，単純に平均をとるのではなく別の処理を行うことが有効である．

#### III.3.3.2　AES 候補に対する電力解析

本節では，AES 候補に対する電力解析の解析原理，各 AES 候補に対する適用例について解説する [BS99, CJ+99a, DR99]．対処方法については III.3.3.5.3.2 節を参照されたい．

##### III.3.3.2.1　解析原理

スマートカード等の耐タンパデバイスに対する電力解析である単純電力解析と電力差分解析は，Kocher により提案され [KJJ98]，非常に強力な攻撃法として注目を集めている．攻撃者は，暗号化処理やデータ処理時におけるスマートカード内の消費電力を観測することにより，スマートカードの内部情報を取り出す．これらの攻撃法は DES (Data Encryption Standard) にも適用できることから，スマートカードを用いた多くのアプリケーションにとって大きな脅威となり得る．

単純電力解析について，Kocher が提案した攻撃法では攻撃者は全ての入力値または出力値を必要とする．しかし現実のアプリケーションを考慮した場合，攻撃者が全ての入力値または出力値を知ることはほぼ不可能であるので，この攻撃法は現実的ではない．しかし，入力値または出力値に関する情報を得る必要がない攻撃法 [BS99] も提案されている．以下にその具体的な攻撃法を示す．なお，解析の対象となるプロトコルについて，プロトコル内のサブプロトコルは同一の順序で実行され，プロトコルの実行に要するクロックサイクルが常に一定であると仮定する．

攻撃者の第一目標は，得られた電力消費グラフの中で鍵スケジュール部に関係する部分を特定することであり，その攻撃は二つのステップに分けられる．

**ステップ 1:** 単一のスマートカードを用いて，入力項目を変化させプロトコルを多数回実行する．(入力項目の変化とは，例えば，支払い金額や支払い回数を変えることであり，実際の入力値を知る必要はない．) 得られた複数の電力消費グラフ間の同一サイクルを比較し，電力消費値に大きなばらつきが存在するサイクルを鍵スケジュール部に関係しないサイクルとして除外する．これは，単一のスマートカードにおいて，鍵スケジュールは同一であり，鍵スケジュール部に関係するサイクルにおける電力消費値は入力項目に依存しないからである．

**ステップ 2:** ステップ 1 で行った比較を複数のスマートカードに対して行う．ステップ 1 で絞り込まれたサイクルにおいて，得られた複数の電力消費グラフ間の同一サイクルを比較し，電力消費値のばらつきが小さいサイクルを鍵スケジュール部に関係しないサイクルとして除外する．これは，各スマートカードにおいて，鍵スケジュールは異なると考えられ，鍵スケジュール部に関係するサイクルにおける電力消費値はスマートカード毎に異なるからである．

鍵スケジュール部に関係するサイクルを特定する際に問題となるのは，鍵スケジュール部に関係するサイクル候補として，莫大なサイクル数の中から上述の攻撃

が可能な程度のサイクル数にどうやって絞り込むかということである．しかし，例えばDESでは，16のラウンドがほぼ認識可能なので，これはそれほど重大な問題ではないと思われる．

攻撃者の第二(最終)目標は，上述の攻撃により特定したサイクルから秘密鍵を取り出すことである．スマートカードは，その性能上，全てのサブ鍵をRAMに保持しておくことは困難であるので，暗号化処理において各サブ鍵は用いられる直前にそれぞれRAMへ書き込まれる．書き込み処理時における電力消費量は，"1"を書き込んだ回数に関係するので，サブ鍵の書き込み処理時の電力消費量からサブ鍵に含まれる"1"の個数が判明する．つまり，サブ鍵の書き込み処理時の電力消費量を観測することにより，サブ鍵のハミング重みが計測される．8ビットのハミング重み計測一回当たり平均2.54ビットの情報が取り出される．このハミング重み計測には，多くの誤りが含まれている可能性があるが，単一のスマートカード(同一の秘密鍵)での消費電力観測を多くの回数行い，平均化することで誤りを減少させることができる．

秘密鍵の取り出しについて，話を具体的にするためDESの場合を取り上げる．DESは56ビットの秘密鍵，16のラウンドを持ち，ラウンド$n$ $(n = 1,2,\ldots,16)$において，48ビットの秘密鍵$K_n$は6ビットのサブ鍵に分割される．(各ラウンドにおいて生成されるサブ鍵の数は8である．)一度にスマートカード内のRAMへ書き込む単位を8ビットと仮定すると，RAMに保持される8ビットのハミング重み計測を各ラウンドにつき6回行うことができる．つまり，攻撃者は16ラウンドにわたるサブ鍵のハミング重み計測により，秘密鍵の各ビットを変数とする(56変数の)96個の線形方程式を得ることができる．得られた方程式を解くことにより56の変数の値，つまり秘密鍵が判明する．この攻撃法の問題点として，得られた方程式の解が不定になる可能性が存在することと，ハミング重み計測誤りを効率良く検出することが挙げられるが，前者について，DESの構造上，得られた方程式の解は唯一解であり，後者について，例えば以下の二つの方法が考えられる．

- 求める変数が56個であるのに対し，得られる方程式は96個であることから，誤り訂正符号において用いられている手法を利用する．
- DESの構造上，得られた方程式を28変数の，48個の方程式に分離させることができるので，28ビットを全数探索する．

電力差分解析は，算術演算や論理演算等の処理において，スマートカードの消費電力パターンと入力ビットに何らかの相関があることを利用した電力解析攻撃である．簡単化のため，入力$n$ビット$(op_1, op_2, \ldots, op_n)$の中で$op_1$が処理$I$における消費電力$P_I$と相関を持つとする．$op_1$の値により処理$I$における消費電力は次式で表される．

$$P_I(0) \quad = \quad E_{op_2,\ldots,op_n}[P(I,0,op_2,\ldots,op_n)]$$

$$P_I(1) \quad = \quad E_{op_2,\ldots,op_n}[P(I,1,op_2,\ldots,op_n)]$$

ここで，$E_{op_2,\ldots,op_n}$は$op_1$以外の入力$op_2,\ldots,op_n$を全通り入力したときの平均である．処理$I$における消費電力$P_I$と$op_1$が相関を持つとは，言い換えると，$P_I(0) \neq P_I(1)$である．以下，$D = P_I(0) - P_I(1)$とする．$D$が雑音に比べて十分大きい場合，電力差分解析は非常に強力な攻撃となる．

基本的に電力差分解析は以下の三つのステップに分けられる．

**ターゲット指定**　III.3.3.2.1.1節に挙げた処理の内で，あるサイクルの，ある処理に注目する．始めに，スマートカード内の秘密鍵の予想値$K$を設定する．次に，入力値の$a$ビット目のみ変えた場合の消費電力値の差$D_a$を計算する．$D_a$が大きい入力値を攻撃に用いる．

**データ収集**　様々な入力値を用いて消費電力パターンのサンプルを多数入手する．

**データ解析**　入力値の$a$ビット目が"0"のグループと"1"のグループをそれぞれ，$G_0, G_1$とし，二番目のステップで得られた消費電力パターンを$G_0, G_1$に分類する．$G_i\ (i = 0, 1)$に含まれる各サンプルでの消費電力値の和を$P_{G_i}$とすると，$|P_{G_0} - P_{G_1}|$を最大にする秘密鍵の予想値$K$が正しい値となる．

予想値$K$が正しい場合，$|P_{G_0} - P_{G_1}| \simeq D \cdot n/2$となり，逆に誤っている場合，$|P_{G_0} - P_{G_1}|$は0に近づく．

#### III.3.3.2.1.1　スマートカードの電力モデル

暗号を実際のアプリケーションに応用する場合，暗号自体への攻撃に対する耐性のみではなく実装されるデバイスへの攻撃に対する耐性も考慮しなければならない．

多くのスマートカードではCMOSが使われており，その特性として，チップ上で何らかの変化が生じた場合だけ電力が消費され，状態を維持するための消費電力は少ない．

スマートカードにおいて，以下に示す処理が特徴的な電力消費パターンを示す．

- 加算，減算，乗算
- ビット毎の論理演算
- RAMへの書き込み
- EEPROMへの書き込み
- RAMやEEPROMからの読み込み

また，一般にスマートカードは，内部あるいは外部のクロックにより駆動され，通常は全ての動作が次のクロックが立ち上がる前に終了する．なお，チップ上のノイズ生成器等はクロックに関係なく連続的にランダムにかつ小さく電力消費を行う．

クロックは，電力を消費する一連のイベントをチップ内部のマイクロ命令に従って駆動する．1 つのクロックサイクルの間の一連のイベントを決定するのはチップ内部の状態であり，これを関連状態 (relevant state) と呼ぶこととする．

クロックのエッジからのチップ全体の瞬間電力消費は，各々のイベントの瞬間電力消費の組み合わせで得られる．しかし各イベントの消費電力はカップリング効果ばかりでなく，基盤やレイアウト，気温や電圧等にも依存するため，複雑である．また電力ばかりでなく，そのタイミングも複雑である．簡単化のため，ここではとりあえず電力消費の総和が各イベントの電力消費の総和であると仮定する．

何らかの固定コードの実行パスの中のある命令のあるサイクルを考える．$S$ を制御がこのサイクルに到達する際の関連状態 (relevant state) の集合，$E$ をこのサイクルの中で起こりうる全てのイベント空間とする．各 $s \in S$ 及び $e \in E$ について，$occurs(e,s)$ を状態 $s$ においてイベント $e$ が起きたとき 1，そうでないとき 0 となる 2 値関数とする．$delay(e,s)$ を状態 $s$ においてクロックのエッジからイベント $e$ が起きるまでの遅延とし，さらに $f(e,t)$ をイベント $e$ が起きたときの電力消費インパルスの時間 $t$ に関する関数とする (このときイベントが起きる時間が $t=0$ であり，また $t<0$ に対し $f(e,t)=0$ である)．このときこのチップのこのサイクルの状態 $s$，クロックのエッジからの時間 $t$ における電力消費関数 $P(s,t)$ は以下のように表せる．

$$\begin{aligned} P(s,t) &= N_c(t) + \sum_{e\in E}(f(e,t-delay(e,s)+N_d(e,s)) \\ &\quad +N(e,t)) * occurs(e,s); \end{aligned} \tag{III.6}$$

ここで，$N(e,t)$ はイベント $e$ の電力消費に伴うガウス雑音，$N_d(e,s)$ は遅延関数に影響するガウス雑音，$N_c(t)$ は外部ガウス雑音を表す．

数式 (III.6) はこのサイクルにおいて電力消費関数と状態 $s$ が強い関係にあることを示している．電力差分解析ではこの電力消費の非対称性を利用する．特に安価なスマートカードでは状態空間が小さく，この非対称性は顕著となるため攻撃に弱くなる．

#### III.3.3.2.2 AES 候補への適用例

AES の候補となっている各暗号方式をスマートカードに実装した場合の電力解析攻撃の影響を調査する．表 III.5 に各 AES 候補の単純電力解析 (SPA) と電力差分解析 (DPA) に対する耐性及びその対処の実装可能性をまとめる．なお，対処の実装可能性について，ここでは各暗号方式において行われている処理のみを考慮している．つまり，電力解析攻撃を受けにくい (または防御が容易である) 処理のみを用いている暗号方式に対して電力解析攻撃への対処の実装が容易であるとしている．

各暗号方式で行われている基本的な処理のうち，ビット毎の論理演算，加算，減算，及び乗算処理は，消費電力が計測されやすく，また単純電力解析に対する対策

表 III.5: 各 AES 候補の電力解析攻撃に対する耐性及び対処法の実装可能性

| | 電力解析攻撃に対する耐性 vs. SPA | vs. DPA | 対処法の実装 |
|---|---|---|---|
| Mars | ○ | △ | × |
| RC6 | ○ | △ | × |
| Rijndael | △ | × | ○ |
| Serpent | ○ | × | ○ |
| Twofish | ◎ | × | △ |
| 記号の意味 | ◎：非常に強い<br>○：強い<br>△：やや脆弱<br>×：脆弱 | | ○：容易<br>△：やや困難<br>×：困難 |

を講じることが困難である (特に乗算は非常に困難である)．従ってこれらの処理を用いないことが望ましい．

以下に，各暗号方式で行われている処理とともに，それぞれの単純電力解析，電力差分解析との関係を列挙する．

#### III.3.3.2.2.1 Mars

行われている処理

テーブル検索 (table-lookup)，シフト演算，ビット毎の論理演算，加算，減算，乗算

vs. SPA

Mars では，ハミング重み計測により，鍵スケジュール部に関係するサイクルにおいて RAM に書き込まれる 8 ビットの内，平均 2.54 ビットの情報が取り出されるが，RAM に書き込まれる各ビットがスマートカード内の秘密鍵と直接的に対応していないため，スマートカード内の秘密鍵を求めることは困難である．また，ハミング重み計測に誤りが生じた場合，得られた情報をスマートカード内の秘密鍵の特定に用いることはさらに困難である．

vs. DPA

Mars におけるホワイトニング鍵は Twofish と同じ方法で得ることができる．しかし鍵スケジュール部が複雑であるために，全てのサブ鍵を得るために攻撃されるべき中心ラウンド数は膨大な数になる．各々の次の展開ボックスに対し，排他的論理和の鍵が電力差分解析で攻撃可能であり，かつ乗算鍵が bit-by-bit 電力差分解析で攻撃可能でなければならない．

#### III.3.3.2.2.2 RC6

行われている処理
シフト演算，ビット毎の論理演算，加算，減算，mod$2^{32}$上における二乗演算
vs. SPA
RC6 では，Mars と同様の理由によりスマートカード内の秘密鍵を求めることは困難である．
vs. DPA
加算入力および出力ホワイトニング鍵は bit-by-bit 電力差分解析で取り出すことが可能である．また，加算ラウンド鍵も同様に取り出すことができる．鍵スケジュールが複雑であるため，全てのサブ鍵を得るためには全てのラウンドを攻撃しなければならない．

#### III.3.3.2.2.3 Rijndael

行われている処理
テーブル検索 (table-lookup)，シフト演算
vs. SPA
Rijndael では，Mars と同様の理由によりスマートカード内の秘密鍵を求めることは困難であるが，RAM に書き込まれる各ビットとスマートカード内の秘密鍵との関係が Mars や RC6 に比べてより直接的であるので，Mars や RC6 よりは容易であると考えられる．
vs. DPA
電力差分解析では 128 ビットの Rijndael の全てのサブ鍵が得られたあとに排他的論理和がなされる 0 ラウンド鍵が得られる．Rijndael の提案者は 34 バイトのスマートカードであれば電力差分解析に耐えると主張している．しかし，現在のところそのようなスマートカードは存在しない．

#### III.3.3.2.2.4 Serpent

行われている処理
シフト演算，ビット毎の論理演算
vs. SPA
Serpent では，Mars と同様の理由によりスマートカード内の秘密鍵を求めることは困難である．
vs. DPA
Serpent は DES と同じくある種の電力差分解析 (すなわち，鍵の排他的論理和のあ

とのS-ボックスの探索)に対して弱い．鍵スケジュールの仕方により，マスター鍵を得るには始めの2ラウンドのサブ鍵があれば十分であることが明らかである．

#### III.3.3.2.2.5　Twofish

行われている処理
テーブル検索(table-lookup)，シフト演算，加算
vs. SPA
Twofishの鍵スケジュール部は複雑であるので，ハミング重み計測からスマートカード内の秘密鍵の直接的な情報を得ることは困難である．
vs. DPA
[CJ+99a]にはTwofishに対して行った電力差分解析の実験の様子が示されている．ここではホワイトニング鍵(128ビット)をまず求め，これをもとにマスター鍵(128ビット)の候補を絞り込む手法が採られている．ホワイトニングプロセスには特徴的な電力消費パターンが存在するため特定が容易である．このためホワイトニング鍵については電力差分解析によりたった100個の電力サンプルから全てのビットが正しく得られたという．さらに，Twofishの仕様に基づきマスター鍵を絞り込む場合，ホワイトニング鍵がわかっていればその候補は$9^8$個以下に減らすことができるという．これは全探索が困難でない個数である．

### III.3.3.3　RSAに対する電力解析

本節では，RSAに対する電力解析の原理を解説する[MDS99b]．対処方法についてはIII.3.3.5.3.3節を参照されたい．

#### III.3.3.3.1　解析原理

スマートカードを用いた公開鍵暗号システム（RSA暗号や楕円曲線暗号など）において，個人の秘密鍵に相当する秘密情報がスマートカード内に保管され利用される．スマートカードはその利用の際に，カードリーダーにより認証される．ここで重要なことはこのリーダーはある別のメンバにより製造されたものであるため，利用者にとって信用できるデバイスではないということである．従って，カードをカードリーダーに入れ，カードリーダーによるカードのコントロールを許した場合にでも，カード内部に保管される秘密情報の秘匿性は維持されるべきである．RSA暗号を用いてカードの認証を行う場合，カードリーダはランダムなチャレンジをカードに対して与え，その値に対してカード内部に秘密に保管されている秘密冪指数を用いて冪乗演算するように要求する．従って，例え不正に製造されたカードリーダーがカードにアクセスし，カードの冪乗演算実行中の電力消費を観測したとしても，カード内部の秘密情報が明かされないようにすることが必要である．

剰余冪演算は公開鍵暗号を実装する際には極めて重要な技術である．剰余冪演算を行う最も代表的なアルゴリズムは「平方乗算法 (square-and-multiply algorithm)」である．また，楕円曲線暗号においてはこれによく似た「倍加算法 (double-and-add algorithm)」を用いる．平方乗算法としては次の二つのアルゴリズムが代表的である．これを図 III.20，III.21 に示す．図 III.20 は最上位のゼロでないビットから始まり下位のビットに向かって演算が進んでいく．逆に図 III.21 は最下位のゼロでないビットから始まり上位のビットに向けて演算が進んでいく．図 III.21 では図 III.20 に比べて余分なメモリを必要とする．

$R = M$
for $(i = n - 2$ down to $0)\{$
  $R = R^2 \bmod N$
  if ($i$th bit of $e$ is a 1)
  $R = R \cdot M \bmod N$
$\}$
return $R$

図 III.20: exp1(M,e,N)

$R = 1$
$S = M$
for $(i = 0$ to $n - 1)\{$
  if ($i$th bit of $e$ is a 1) $\{$
    $R = R \cdot S \bmod N$
  $\}$
  $S = S^2 \bmod N$
$\}$
return $R$

図 III.21: exp2(M,e,N)

文献 [MDS99b] において示されている後述の 3 つの攻撃法はこれら二つのアルゴリズムのいずれにも原理的には適用可能である．とりわけ MESD 及び SEMD は平方乗算法に対しての攻撃であるが，この演算方法は公開鍵暗号に対しては何らかの形で実装されているので適用は可能である．ZEMD 攻撃を成功させるには攻撃者がスマートカード内でどのような手法で演算が行われているのかを知っている必要があるが，とりうる演算手法はそれほど多くはないので，あらゆる可能性のある演算

方法を想定してこの攻撃を適用することにより，攻撃を成功させることは可能であろう．

ここで述べられる攻撃の目的は，スマートカード内に秘密に保管される秘密の冪指数 $e$ を導出することである．攻撃者はスマートカードの動作を完全に制御することが可能であるとする．つまり，スマートカードは秘密鍵 $e$ を出力する以外のあらゆる攻撃者の命令に従うものとする．スマートカードに最も必要とされる機能は「内部認証 (internal authenticate)」，即ち入力 $M$ に対して，$M^e$ を出力する機能である．スマートカードの中にはオペレーションを行う際に PIN(Personal Identification Number) の入力を要求するものもあるが，ここではそれについては考慮しない．さらに攻撃者の計測回数に関しても制限しない．これらは現実の実装の観点から見て正当な仮定と言える．

まずはじめに，単純に乗算命令による電力信号とカードから出力される電力信号との相関をとることにより，$e$ を決定することが可能かどうかを見てみよう．これにより冪乗演算全体に要する電力信号の相関を調査することにより，単一の剰余乗算もしくは剰余平方を区別することが可能かを見ることができる．ここで，剰余乗算の電力信号を $S_m[j]$，剰余冪演算の電力信号を $S_e[j+\tau]$ とする．このとき，相関信号 $S_c[j]$ は次のように計算できる．

$$S_c[j] = \sum_{\tau=0}^{W} S_m[\tau] S_e[j+\tau]$$

ここで，$W$ は乗算信号のサンプル総数とする．$T_m$ を剰余乗算に要する時間，$T$ をサンプルレートとすると，$W = T/T_m$ となる．攻撃者は事前にスマートカードの設計書等を解析することにより，攻撃にとって適切な $W$ を知ることが出来る．

実際のスマートカードにおいて入力値を固定し，5000 回の出力の測定値を測定ノイズ除去のために平均化した結果を測定する実験で，この攻撃の可能性が検証されている [MDS99b].

あらかじめ既知の冪係数について測定した場合と，未知の冪係数について測定した場合の電力信号と相関信号を観測した場合に，剰余乗算と剰余平方の位置がどのように現れるかを調べたところ，確かにこれらの計算が実行されている部分で，電力相関信号はピークを形成しているが，これら二つの間で大きな相違を見ることは出来ない．つまり，ここで述べられている電力相関の測定は剰余乗算と剰余平方のいずれが行われているのかを特定するためには用いることができない．だが，これらの測定により平方情報アルゴリズムに要する時間情報が決定できることは興味深い．この情報は電力攻撃とタイミング攻撃をともに用いる可能性を示している．相関信号は全ての中間処理のタイミングを明らかに出来るので，この攻撃は単純な時間攻撃に比べ非常に強力であるといえる．

#### III.3.3.3.1.1 SEMD (Single Exponent Multiple Data) 攻撃

SEMD 攻撃はスマートカードが二つの冪指数（一方は秘密指数で一方は公開指

数）を用いて任意の数のランダムな値に対して冪演算を実行する場面を想定する．このような場面はISO7816[ISO] 標準の「外部認証コマンド (external authenticate command)」をサポートするスマートカードシステムで起こりうる．内部認証コマンド (internal authenticate command)が秘密指数を用いて冪演算を行うのに対して，外部認証コマンドは特定のスマートカードリーダに関連付けられた公開鍵を用いて冪演算を実行する．攻撃者はこの公開鍵のビットを知っているものとする．

この攻撃の大前提は未知の冪指数を用いた実行された冪演算の電力信号と，既知の冪指数を用いて実行された冪演算の電力信号ををを互いに比較することにより，攻撃者は二つの冪指数のビットの違いを測定して秘密鍵を解き明かすことが可能であるということである．実際には，平方乗算アルゴリズムの中間データの結果が電力信号に大きな影響を及ぼすので，この比較は単純ではない．単純な電力差分解析は秘密冪指数を用いて $L$ 個のランダムな値を冪乗させ，それらの電力信号 $S_i[j]$ を収集することにより開始する．同様に，公開冪指数に対しても $L$ 個の電力信号 $P_i[j]$ を測定する．これらを用いて電力差分解析 (DPA) バイアス信号 $D_j$は次のように構成できる．

$$D[j] = \frac{1}{L}\sum_{i=1}^{L} S_i[j] - \frac{1}{L}\sum_{i=1}^{L} P_i[j] = \bar{S}[j] - \bar{P}[j]$$

中間データに依存する$\bar{S}[j]$と$\bar{S}[j]$の分布は同じ値に平均化される．つまりデータに依存するサンプル点 $j$ においては次が成立する．

$$\bar{S}[j] = \bar{S}[j] \approx \mu$$

だが，冪指数に依存するサンプル点 $j$ では，剰余乗算が行われるか剰余平方が行われるかに従って異なる値 $(\mu_s, \mu_p)$ に平均化されるはずであり，従ってDPAバイアス信号 $D[j]$ もゼロでない値となるであろう．これを用いて以下のように平方と乗算の位置を抽出することが可能である．

$$D[j] \approx \begin{cases} 0 & (j \text{における冪演算処理が一致するなら}) \\ 0 \text{以外} & (j \text{における冪演算処理が異なるなら}) \end{cases}$$

SEMD 攻撃の可能性を実験で検証した結果が報告されている [MDS99b]．実験では，法および冪指数を64ビットと短いビット長を用いている．短いビット長の冪指数を用いた場合，より多くの電力信号をデジタルオシロスコープに収めることが出来るため，一回の測定で，より長いビットに対して攻撃を行うことが出来る．実際のサイズ（例えば1024ビットなど）の冪指数に対しては一度の測定で冪指数の一部だけが攻撃可能である．一度に攻撃可能なビット長は攻撃者の持つデジタルオシロスコープの解像度（メモリ）に依存する．測定結果を観察したところ，DPAバイアス信号は，平方と乗算とで異なる処理の行われている部分において，DPAバイアス信号が強く増幅されていることが分かった．出力に対して適切なフィルタリングを

行うことにより，この相違点は簡単に特定可能であり，SEMD 攻撃の有効性は実証されている．つまり，スマートカードシステムを実装する際には SEMD 攻撃への対応を考慮することが重要であろう．

#### III.3.3.3.1.2 MESD (Multiple Exponent Single Data) 攻撃

MESD 攻撃は SEMD 攻撃よりも強力であるが，スマートカードに対していくつかの仮定を追加する必要がある．前述の SEMD 攻撃は攻撃者の側に複雑な処理を必要としない単純な攻撃であるが，DPA バイアス信号を頻繁に測定することはしばしば困難である．MESD 攻撃は SEMD 攻撃の信号対雑音比 (SNR) を改善する．MESD 攻撃の仮定は「攻撃者が自由に選択した冪指数を用いて，スマートカードに対して定数（攻撃者が知らなくても可）を冪乗演算させることが出来る」というものである．このような仮定は現実的に起こりえないものではない．何故なら，しばしばスマートカードは新しい冪指数をセットアップできるように構成されていることがあるからである．スマートカードは無限のメモリを持っていないので，既に冪乗された値のリストを全て保持することは出来ない．従ってカードの側では定数を冪乗するように繰り返し依頼されたとしても，それを検地することは出来ない．

MESD 攻撃のアルゴリズムは図 III.22 に示される．アルゴリズムの第一ステップは任意の値 $M$ を選択し，$M$ を秘密指数 $e$ を用いて冪乗させ，対応する平均電力信号 $S_M[j]$ を収集することである．次に，アルゴリズムは $e$ の第一ビットから順に最後のビットまで進んでいく．$i$ 番目のビットを攻撃するために，攻撃 $\mathrm{sy}$ は $i$ 番目のビットが 0 か 1 であると推測し，それぞれの予想値を用いて冪演算をカードに実行させる．攻撃者は既に $i-1$ 番目までのビットを推測していると仮定すると，$i-1$ 番目までのビットによる演算の中間データは測定結果と予想結果は一致するだろう．従って $i$ 番目のビットの予想が正しければ，中間結果の予想は $i$ 番目でも一致するはずであり，もし予想が正しくなければ結果は異なってくるはずである．この相違は相関電力を調べることで確認できる．$M$ を $e_g$を用いて冪乗するのに要する平均電力信号を，$e_g$の $i$ 番目のビットが 1 である場合 $S_1[j]$，0 である場合 $S_0[j]$ とする．二つの DPA バイアス信号は次のように計算できる．

$$D_1[j] = S_M[j] - S_1[j] \qquad D_0[j] = S_M[j] - S_0[j]$$

予想が正しい場合，電力信号は秘密冪による電力信号と一致するためバイアス電力は 0 になる．

実際のスマートカードを用いて，この攻撃の有効性を検証したところ，SNR は明らかに SEMD 攻撃に比べ向上している．SNR が高いということは MESD 攻撃の実現に要する試行測定回数が少なくてすむことを示す．また，攻撃者は一度だけ DPA バイアス信号を計算すればよい．例えば，攻撃者は全てのビットが 1 であると予測するとする．予測が正しければバイアス信号は 0 のまま続き，予測がずれた時点で 0 以外の値になる．このテクニックにより攻撃アルゴリズムの実行時間は効率的に

$M$ = arbitrary value and $e_g = 0$
Collect $S_M[j]$
for ($i = n - 1$ to 0){
  guess ($i$th bit of $e_g$ is a 1) and collect $S_1[j]$
  guess ($i$th bit of $e_g$ is a 0) and collect $S_0[j]$
  Calculate two DPA bias signal :
    $D_1[j] = S_M[j] - S_1[j]$ and $D_0[j] = S_M[j] - S_0[j]$
  Decide which guess was correct using DPA result
  update $e_g$
}
$e_g$ is now equal to $e$ (the secret exponent)

図 III.22: MESD 攻撃アルゴリズム

削減される．実際，この測定結果は冪指数一ビットにつき 200 回の試行により測定された．MESD 攻撃は可能な状況では，カードを実装する際にこの攻撃を考慮する必要がある．

#### III.3.3.3.1.3 ZEMD (Zero Exponent Multiple Data) 攻撃

ZEMD 攻撃は MESD 攻撃に似ているが仮定に相違がある．ZEMD 攻撃の仮定は「攻撃者が多くのランダムなメッセージに対する秘密冪指数を用いた冪演算をスマートカードに実行させることが出来る」というものである．この攻撃では攻撃者は冪指数に関する予備知識をまったく必要としない（このため Zero Exponent と呼んでいる）が，そのかわりに攻撃者はオフラインのシミュレーションにおいて平方乗算アルゴリズムの中間結果を予測できなければならない．そのため，攻撃者はカード内部で冪演算を実行するために行われているアルゴリズムの詳細を知っている必要がある．だが，これらのアルゴリズムの候補は数少ないので全数探索的に全てのアルゴリズムを考慮して攻撃を行うことは可能であろう．

ZEMD 攻撃は図 III.23 に示される．ZEMD 攻撃は第 1 ビットから順に最終ビットまで実行される．このアルゴリズムでは変数 $e_g$が徐々に秘密指数に等しくなっていく．それぞれの繰り返しで，冪のある 1 ビットを求めることができ，$e_g$は次々に更新されていく．アルゴリズムを $i$ 回目に繰り返したとき，$e_g$の $i-1$ 番目のビットまでは正確であると仮定する．アルゴリズムは $i$ 番目のビットが 1 であると仮定し，それが正しいかどうかを DPA バイアス信号を用いて推測していく．DPA バイアス信号はランダムな入力 $M$ を選び，消費電力をシミュレーションにより求めることにより決定する．消費電力は冪指数のハミング重みにより推測可能である．

実験による検証で，実際に ZEND 攻撃成功させるためには，200 個程度のランダ

```
e_g = 0
for (i = n − 1 to 0) {
   guess (ith bit of e_g is a 1)
   for (k = 1 to L){
      choose random value : M
      simulate to the ith set the calculation of M^{e_g} mod N
      if (multiplication result has high Hamming weight)
         run smartcard and collect power signal : S[j]
         add S[j] to set S_high
      if (multiplication result has low Hamming weight)
         run smartcard and collect power signal : S[j]
         add S[j] to set S_low
   }
   Average the power signals and get DPA bias signal :
   if DPA bias signal has spikes
      the guess was correct : make ith bit of e_g equal to 1
   else
      the guess was wrong: make ith bit of e_g equal to 0
}
e_g is now equal to e (the secret exponent)
```

図 III.23: ZEMD 攻撃アルゴリズム

ムな電力信号の測定で十分であることが報告されている [MDS99b].

#### III.3.3.4 楕円曲線暗号に対する電力解析

Kocher らによって提案された電力差分攻撃は電力信号を計測することでスマートカード内の秘密情報を不正に入手することを可能とする強力な攻撃方法である．スマートカード上の DES の解析に対し，Kocher らは電力差分解析を適用し 1000 回の暗号化処理を行うことで秘密鍵の導出に成功している．[Cor99] において，楕円曲線暗号系に対する電力差分解析を一般化し，それを楕円曲線上の Diffie-Hellman 鍵配送や楕円曲線上の ElGamal 型暗号に適用した．これらの攻撃を利用した場合，スマートカード内の秘密鍵を取り出すことが可能である．また，これらの攻撃に対する対応方法についても検討が行なわれている．本節では，これらの手法の紹介を行なう．対処方法については III.3.3.5.3.4 節を参照のこと．

楕円曲線の暗号への応用は 1985 年に Miller と Koblitz によって初めて提案された．

それ以来，様々な楕円曲線暗号系の研究は活発に行われている．楕円曲線は，従来の離散対数暗号系で利用されている群構造と置き換え可能な群構造を提供する．位数 $n$ の元 $g$ による巡回群での離散対数問題は，$G$ の元 $y = g^x$ に対し $x$ を導出することである．楕円曲線上の離散対数問題は有限体上の乗法群のような他の群より大幅に困難であると考えられている．超特異楕円曲線以外の楕円曲線上の離散対数問題においては，準指数オーダーの計算時間での計算アルゴリズムは知られていない．このため，楕円曲線暗号系においては秘密鍵のサイズは 160 ビット程度で充分となる．

[Cor99] においては，スマートカード上に実装された楕円曲線暗号に対し，電力消費量を計測することで攻撃を試みている．電力差分解析は電力消費量に応じた秘密情報の漏洩を利用した強力な攻撃である．電力差分解析は DES に対して適用され，1000 回の暗号化処理によって秘密鍵を不正に入手することに成功している．さらに，AES の候補に挙がっている暗号系のスマートカード上の実装に対しても電力差分解析が効果的であることが示されている．これらの結果により楕円曲線暗号の単純な実装に対しても電力差分解析が効果的であることが確認された．

#### III.3.3.4.1 解析原理

##### III.3.3.4.1.1 楕円曲線上の乗算

楕円曲線上の点 $P$ の $d$ 倍を導出する処理をスカラー乗算と呼び，$dP$ と表記する．楕円曲線上のスカラー乗算は $\bmod m$ による整数の乗法群に類似している．

$dP$ の計算は倍加算法 (double-and-add algorithm) をそのまま適用することで行うことができる．$d = \{d_{l-1}, \cdots, d_0\}$ とビット列表記し ($d_{l-1}$ を最上位ビットとする)，次のアルゴリズム 1 によってこれを示す．

```
input P
Q ← P
for i from l − 2 to 0 do
        Q ← 2Q
        if di = 1 then Q ← Q + P
output Q
```

図 III.24: アルゴリズム 1

スカラー乗算を高速に行うための様々な手法が存在している．$P$ が既知の場合，$P$ の乗算に関する表を事前に計算することは有効的である．楕円曲線において減算と加算のコストは等しいため，倍加算法は次の加減算法 (addition-subtraction algorithm)

に改良することができる．

$$d = \sum_{i=0}^{l-1} c_i 2^i, \quad c_i \in \{-1, 0, 1\}$$

$d$ の NAF (non-adjacent form) とはすべての $i \geq 0$ に対して $c_i \times c_{i+1} = 0$ となるような，$d$ の符号付二進表記である．すべての正整数は NAF によってユニークに定められ，なおかつ NAF は $d$ の符号付二進表記のうち非ゼロ係数の数を最小にすることが知られている．アルゴリズム 2 に加減算法を示す．

```
input P
Q ← P
for i from l − 2 to 0 do
        Q ← 2Q
        if ci = 1 then Q ← Q + P
        if ci = −1 then Q ← Q − P
output Q
```

図 III.25: アルゴリズム 2

最小の (楕円曲線上の) 計算処理回数で dP を計算する方法を見つけることは，$d$ の最も短い加減算鎖を見つけることと同値である．加減算鎖とは次のような正整数の系列である．

$$a_0 = 1 \to a_1 \to a_2 \to \cdots \to a_r = d, \quad \forall i \in \{1, 2, \cdots, r\}, \quad a_i = \pm aj \pm ak, \quad k \leq j < i$$

$d$ の最も短い加減算鎖は $a_1 P, a_2 P, \cdots, a_r P = dP$ と計算することで，最小の計算処理回数で $dP$ を導出することができる．

#### III.3.3.4.1.2　電力消費量を利用した Q=dP の d の導出

1998 年，Kocher は DES に対する単純電力解析と電力差分解析を著した．単純電力解析は暗号アルゴリズム一つの実行とその電力消費量の観測により行われる．電力差分解析はより洗練された強力な攻撃方法である．これは，一つの暗号アルゴリズムに対してさまざまな入力を行い，その結果に対して統計的な解析をする．本節では，$Q = dP$ の計算中で消費される電力を観測することで d を導出する手法を示す．まず，スカラー乗算の単純な実装に対しる単純電力解析の有効性を議論し，その対応手法を併せて示す．その後，スカラー乗算に対する，電力差分解析による攻撃方法を示す．

##### III.3.3.4.1.2.1　単純電力解析に対する対処

電力消費量を観測することで，視覚的にさまざまな重要な特性を識別することができる．たとえば，アルゴリズム 1 における乗算と加算の判別などが可能である．単純電力解析への対処方法としてアルゴリズム中の命令が扱われているデータに依存しないようにしなければならない．一つの方法として，アルゴリズム内にデータに依存した分岐を持たせないことが考えられる．この手法でアルゴリズム 1 に改良を加えたものが次のアルゴリズム 1′である．

input $P$
$Q[0] \leftarrow P$
for $i$ from $l-2$ to 0 do
　　$Q[0] \leftarrow 2Q[0]$
　　$Q[1] \leftarrow Q[0] + P$
　　$Q[0] \leftarrow Q[di]$
output $Q$

図 III.26: アルゴリズム 1’

#### III.3.3.4.1.2.2　倍加算法に対する電力差分解析

本節では，アルゴリズム 1′に対する電力差分解析を示す．アルゴリズムは一定時間で終了することを仮定する．この仮定が成立しない場合，時間攻撃や単純電力解析が容易に行われる．

DES への電力差分解析は電力消費量と鍵に依存した特定のビットの相関を利用している．たとえば，ひとつの最初のラウンドの SBOX の出力におけるビット $b$ は入力メッセージと鍵の 6 個の未知のビットに依存している．そこで，6 個の未知のビットの取りうる値に対する出力 $b$ と電力消費量の相関を調べてやることで 6 個の未知のビットの値を推測することが可能となる．これを残りの SBOX に対しても適用してやることで合計 48 ビットの値が求められる．未知のままの残り 8 ビットは全数探索で導出すればよい．

アルゴリズム 1′に対する電力差分解析は $j$ ステップ目において $Q$ の値が $(d_{l-1}, \cdots, d_j)$ にのみ依存していることを利用している．ここで，点の値がメモリ内でどのように表されているかわかっているものとする．点 $Q$ がアルゴリズム中で取り扱われるとき，$Q$ のある特定のビットと電力消費量に相関が生じる．アルゴリズム中で取り扱われない点と電力消費量には相関は生じない．これゆえ，スマートカード内で取り扱われている点を推測することが可能である．

$d$ の最上位ビットから二ビット目 $d_{l-2}$ は電力消費量と $4P$ の二進表現によるビット列中の特定のビットとの相関を計算することにより導出される．もし，$d_{l-2} = 0$ であれば，$4P$ がアルゴリズム 1′中に出現する．一方，$d_{l-2} = 1$ であれば，$4P$ は絶対

に出現しない．この手法により，$d_{l-2}$を求めることができる．同様の手法により残りのビットも導出される．

アルゴリズム 1′が点 $P_1, P_2, .., P_k$に対して適用され，$Q_1 = dP_1, \cdots, Q_k = dP_k$の計算が行われるものとする．$C_i(t)$ を $Q_i = dP_i$の計算を実行したときの電力消費量とする．$s_i$を $4P_i$のある特定のビットとすると，$s_i$と $C_i(t)$ の相関関数は次のように表される．

$$g(t) =< C_i(t) > i = 1, 2, \cdots, k|_{s_i=1} - < Ci(t) > i = 1, 2, \cdots, k|_{s_i} = 0$$

$4P_i$が $t = t_i$で実行されたとき，その電力消費量 $C_i(t_1)$ は $s_i$と相関が生じると仮定する．$s_i = 1$ のときの平均消費電力は $s_i = 0$ のときの平均消費電力と異なったものとなる．そのため，$t = t_i$において $g(t)$ にピークが現れる．$4P_i$がアルゴリズム中で出現しなかったものとすると $g(t)$ 中にピークは現れない．

#### III.3.3.4.1.2.3　任意のスカラー乗算アルゴリズムへの拡張

ここでは上記の電力差分解析を加減算鎖アルゴリズムに適用する手法を示す．つまり，

$$a_0 = 1 \to a_1 \to a_2 \to \cdots \to a_r = d, \quad \forall i \in \{1, 2, \cdots, r\}, \quad a_i = \pm a_j \pm a_k, \quad k \leq j < i$$

に対して $a'_i = \pm a_j \pm a_k, k \leq j < i$を満足するすべての $a'_i$ を $a_i$のとりうる値として挙げる．これらの $a'_i$に関して，$a'_i P$ と電力消費量との相関を調べる．もし，そこに有意な相関が観測されれば，$a'_i P$ がアルゴリズム中に出現したことを意味し，$a_i = a'_i$ であるといえる．この手法により，$d = a_r$を $O(r^2)$ の計算時間で導出することができる．

#### III.3.3.4.1.3　楕円曲線暗号への攻撃

本節では，楕円 ElGamal 暗号と楕円 Diffie-Hellman 鍵配送に対する電力差分解析の適用方法を示す．楕円曲線 DSA では固定された指数ではなくランダムな指数を用いているため電力差分解析の適用はできない．

##### III.3.3.4.1.3.1　楕円 ElGamal 暗号

楕円 ElGamal 暗号での処理手続きを以下に示す．

**システムパラメータ:**

$\varepsilon$: $GF(p)$ もしくは $GF(2^n)$ 上の楕円曲線曲線

$\sharp\varepsilon$: $\varepsilon$の位数

$q$: $q|\sharp\varepsilon$を満足する大きい素数

$G$: $\varepsilon$上の位数 $q$ の元

**鍵生成:**

秘密鍵: $d \in [1, \cdots, q-1]$

公開鍵: $Q = dP$

**メッセージ $m$ の暗号化：**

乱数 $k \in [1, \cdots, q-1]$

$kP = (x_1, y_1),\ kQ = (x_2, y_2),\ c = x_2 + m$ を計算.

$(x_1, y_1, c)$ が暗号文

**復号化：**

$(x'_2, y'_2) = d(x_1, y_1),\ m = c - x'_2$.

楕円 ElGamal 暗号への電力差分攻撃では，さまざまな暗号文 $(x_1, y_1, c)$ を用いて復号化を行い，$d(x_1, y_1)$ の計算に電力差分解析を適用することで秘密鍵 $d$ を導出することが可能である.

#### III.3.3.4.1.3.2　楕円曲線 Diffie-Hellman 鍵配送

楕円曲線 Diffie-Hellman 鍵配送法での処理手続きを以下に示す.

**システムパラメータ:**

$\varepsilon$: $GF(p)$ もしくは $GF(2^n)$ 上の楕円曲線曲線

$\sharp\varepsilon$: $\varepsilon$の位数

$q$: $q|\sharp\varepsilon$を満足する大きい素数

$G$: $\varepsilon$上の位数 $q$ の元

**鍵生成:**

Alice, Bob の秘密鍵: $s_A, s_B \in [1, .., q-1]$

Alice, Bob の公開鍵: $Q_A = s_A P, Q_B = s_B P$

**鍵共有：**

$P_{AB} = s_A Q_B = s_B Q_A$を計算.

もし $P_{AB} = 0$ であればエラーを出力する.

共有鍵は $P_{AB}$の $x$ 座標を用いる.

楕円曲線 Diffie-Hellman 鍵配送への電力差分攻撃では，さまざまな利用者の公開鍵に対して鍵共有のための計算 $P_{AB} = s_A Q_B = s_B Q_A$を実行し，それらに電力差分解析を適用することで秘密鍵 $s_A, s_B$を導出することができる.

#### III.3.3.5　電力解析の対策

ここでは電力解析への対処方法として，主に電力差分解析に関して各設計レベル毎に整理する．

##### III.3.3.5.1　トランジスタレベルでの対策

ゲートレベルにおける論理回路デザインの工夫により，外部に漏れる情報を大幅に削減する方法が開発されている [KJJ98]．また，現在のところトランジスタにとって代わるデバイスは実用化されていないが，光コンピューティング等の新しい計算技術を実用化することで問題を解決することもできる．

##### III.3.3.5.2　回路，プロセッサ等ハードウェアレベルでの対策

物理的に大きなシステムでは，電力供給にフィルタリングを施したり，遮蔽シールドで保護するなどの対策が考えられる．コストやサイズの制約があるスマートカードなどのシステムでは，主に次のような対策が考えられる．

- 消費電力信号の振幅を小さくする
- 消費電力測定に対して雑音を付加する
- 秘密情報と内部変数の相関を小さくする
- 消費電力と演算内容の相関を一時的になくすような工夫をする

消費電力信号の振幅を小さくする具体的な方法として，一定のパスコードを用いる，消費電力の変化が小さい命令を用いる，ハミング重みと状態遷移のバランスをとる，等を挙げることができる．また，雑音を付加する方法として，解析に必要なサンプル数を実際には不可能な程度まで大きくする，実行のタイミングやオーダーに関する情報を外部から検出不可能にする，等の対処方法を用いることができる．ただし，これらの対策は，攻撃者が長時間の解析を行えない場合において有効であるが，消費電力と秘密情報を用いた演算内容との相関を完全になくすには至らないため，根本的な解決策とはいえない．

##### III.3.3.5.3　ソフトウェアおよびアルゴリズムレベルでの対策

最も効果的な解決策は，ハードウエアが情報を漏らしてしまうことを不可避なものとする立場に立った対処方法である．非線形な鍵の更新は電力解析を困難にする．例えば SHA を用いて 160 ビットの鍵のハッシュ値をとり，それを用いて演算を行うと解析を困難にすることができる．また，冪指数や剰余の法などのパラメタを頻繁に更新することも有効である．鍵の使用回数を記録する装置を装備するのもよい．

以下，電力解析の適用例として本報告書で調査したDES，AES候補，RSA，楕円曲線暗号における対処方法をまとめる．

#### III.3.3.5.3.1 DESにおける対処方法

**単純電力解析** 単純電力解析を防ぐ方法は比較的簡単に実装することができる．条件分岐演算について秘密の中間プロセスや鍵を用いることで多くの単純電力解析の特徴を隠すことができる．しかし条件分岐を本質的に仮定しているアルゴリズムでは，新たな符号化を要し重大なパフォーマンスの低下を招いてしまう．

さらに，オペランド値に大きく依存した電力消費をするシステムでは，実行パスを同一にしても単純電力解析により解析が可能である．

ただ，現在の共通鍵暗号を実装したデバイスのほとんどは電力消費量が小さく，単純電力解析により鍵情報を得ることが困難である．

**電力差分解析** 電力差分解析を防ぐ方法は大きく3つに分類することができる．初めの2つが，III.3.3.5.2節で述べた消費電力信号の振幅を小さくする方法と消費電力測定に雑音を付加する方法であり，最後の1つが本節の冒頭部で述べたハードウェアからの情報のリークは不可避なものとの仮定に立った対処方法である．

#### III.3.3.5.3.2 AES候補における対処方法

文献[CJ+99a]では，あらゆる電力解析攻撃への対処方法は不可能であるとしているものの，現実的な観点からは高階電力差分解析，単純電力解析，電力差分解析などの一般によく知られた攻撃に耐える実装であれば十分であるとしている．この前提のもとに挙げられている対処方法を以下に挙げ，解説する．

- コードの実行パスの鍵やデータからの独立化・可能な限りのノイズの生成

  **対象** 単純電力解析

  **特徴** 攻撃者に電力差分解析や高階電力差分解析などの静的攻撃を用いざるを得なくさせる．

- 各イベントの相補的なイベントを恣意的に生成する方法

  **対象** サンプル数の少ない電力差分解析

  **方法** 電力消費を起こすイベントを，その補足的なイベントを起こすことで隠蔽．

  **弱点** 相補的なイベントであっても，気温，電圧，外部クロック等の実行環境によって電力消費やタイミングのずれが顕著になるため，効果が期待できない．

- 処理順序，遅延のランダム化

  **弱点**　信号処理のツールを用いて元に戻すことが可能．順序は完全に戻せなくてもサンプル数が多ければ同じ電力関数によるサンプルを識別することは可能．

- コードの改変

  **対象**　電力差分解析

  **方法**　実行に関わる部分以外にコードに改変を加える．

  **特徴**　攻撃者が改変を正しく予想する可能性は極めて低い．

  **問題**　データや鍵に直接依存するビットはどのサイクルでも操作されてはならないが，この制約を満足することは容易でない．たとえば，ブロック暗号ではこれを満たすような手法は存在しない．

- 秘密分散を用いた手法

  **対象**　電力差分解析，高階電力差分解析

  **方法**　全ての隠蔽すべきビットをランダムに$k$個のシェアに分ける．計算はシェアに対し行うことで安全になる．攻撃者が出力にもアクセスできるのならば，シェアへの分割は計算の最後にもなされなくてはならない．

  **特徴**　攻撃者は$k$階差分攻撃を行わざるを得なくなる．攻撃者の得るべきサンプル数を$k$に対し指数関数的に増やすことができる．

  **問題**　スマートカードのコードやメモリーへの要求が非常に厳しく，実現が困難．

#### III.3.3.5.3.3　RSA における対処方法

RSA に対する 3 種類の電力解析攻撃に対して採りうる対策としては，タイミング解析を防ぐために用いられるアプローチと同様のものが考えられる．ブラインド署名 [Cha82] のテクニックを用いるという Kocher [Koc96] のテクニックは電力解析攻撃を防ぐためにも有効である．冪演算を実行する前に，メッセージは乱数$v_i$でブラインド化され，冪演算実行後に$v_f = (v_i{}^{-1})^e \bmod N$を乗ずることによりアンブラインド化 (ブラインドの影響を取り去ること) が行われる．Kocher はこの効率的な手法を提案している．

メッセージのブラインド化は MESD 攻撃と ZEMD 攻撃を防ぐものの，SEMD 攻撃を防ぐことはできない．SEMD 攻撃を防ぐためには冪指数自体をブラインド化することが必要になる．RSA 暗号においては冪指数に$\phi(N)$の倍数を加えることにより冪指数のブラインド化を実現できる．これらをまとめてブラインド処理を施した冪演算アルゴリズムは図 III.27 の通りである．

別の方法は冪演算アルゴリズム自体をランダムに変化させることである．exp1 アルゴリズム (図 III.20) と exp2 アルゴリズム (図 III.21) をランダムなポイントで切り

| | |
|---|---|
| 1. Blind the message $M$: | $\hat{M} = (v_i M) \bmod N$ |
| 2. Blind the exponent $e$: | $\hat{e} = e + r\phi(N)$ |
| 3. Exponentiate: | $\hat{S} = (\hat{M})^{\hat{e}} \bmod N$ |
| 4. Unblind the result: | $S = (v_f \hat{S}) \bmod N$ |

図 III.27: ブラインド化による冪乗演算アルゴリズム

替えて用いることである．このランダム化により達成できる安全性は冪指数のビット長に依存するもののビット長が十分長ければ事実上電力解析攻撃は不可能である．

表 III.6: 電力解析攻撃の分類

| 攻撃名 | 試行回数 | 仮定 | 対処法 |
|---|---|---|---|
| SEMD | 20,000 | 攻撃者がひとつの公開冪指数を知っている | 冪のブラインド化 |
| MESD | 200 | 攻撃者がひとつの冪指数を自由に設定できる | メッセージのブラインド化 |
| ZEMD | 200 | 攻撃者は冪演算アルゴリズムを知っている | メッセージのブラインド化 |

#### III.3.3.5.3.4　楕円曲線暗号における対処方法

ここでは，III.3.3.4 節で示された楕円曲線暗号系に対する電力差分解析への対処方法について述べる．[Cor99] において示された乱数を有効に利用する三つの手法を紹介する．これらの対処方法は実装が容易であり効率性に支障をきたすことはない．しかし，この対処方法により楕円曲線暗号系に対するすべての電力差分解析に対処可能であるかは不明である．

##### III.3.3.5.3.4.1　秘密鍵のランダム化を用いた対処法

$\sharp\varepsilon$を楕円曲線上の点の数とする．$Q = dP$ の計算を次のように行うことで電力差分解析を防ぐ．

1. $n$ ビット (20 ビット程度) の乱数 $k$ を選ぶ．
2. $d' = d + k\sharp\varepsilon$を計算．
3. $Q = d'P$ を計算．$\sharp\varepsilon P = 0$ であるから，$Q = dP$ となる．

この手法を用いた場合，毎回異なる $d'$を用いて $d'P$ を計算することになるため電力差分解析の適用は不可能であり，攻撃を防ぐことができる．

#### III.3.3.5.3.4.2　P のブラインド化を用いた対処法

RSA 暗号に対する Chaum のブラインド署名技術を応用する方法を示す．ランダムに点 R を選び，$S = dR$ を計算する．スカラー乗算は $d(R + P) - S$ により行われる．$R$ と $S$ ははじめからスマート内に収められており，新しい計算の度に $R \leftarrow (-1)^b 2R, S \leftarrow (-1)^b 2S$ を計算し $R, S$ を更新する ($b$ はランダムな 1 ビット)．攻撃者は $P' = P + R$ の値を知らないため電力差分解析を適用することはできない．

#### III.3.3.5.3.4.3　射影座標のランダム化を用いた対処法

$P = (x, y)$ の射影座標 $(X, Y, Z)$ は次のように与えられる．

$$x = X/Z, y = Y/Z$$

$(X, Y, Z)$ はユニークではなく，有限体上$\lambda$すべて $(\lambda \neq 0)$ に対して $(X, Y, Z) = (\lambda X, \lambda Y, \lambda Z)$ は $P$ の射影座標となっている．これを利用し，計算を実行するたびにランダムに$\lambda$に対する $P$ の射影座標を求め，それを次の計算に利用する．この手法により $P$ の射影座標の特定のビットと電力消費量との間の有意な相関は表れないものとなる．したがって，電力差分解析を防ぐことができる．

# 第IV部
# 安全性評価の標準化動向

米国標準技術院(NIST)によって1994年に策定された暗号モジュールの安全性に関する米国政府調達基準FIPS (Federal Information Processing Standard) 140-1 [FIP94]は，コンピュータや通信システムにおける暗号モジュールの技術標準および運用標準を規定したものであり，米国およびカナダ連邦政府(CES)で採択されている．暗号モジュールとは各種暗号化機能を実装したハードウェア，ソフトウェア，ファームウェアおよびその組み合わせ製品と定義され，統一基準の下，いくつかのセキュリティレベルに格付けされて評価される．これまでに60を超えるモジュールがFIPS 140-1の承認を得ている．

本章ではFIPS 140-1に関して，この標準が実現しようとしていること，およびその適用方法をまとめる．また現在改定作業が進められているFIPS 140-2 [FIP99]について，FIPS 140-1からの変更点を特に電力解析攻撃のような比較的新しい攻撃法への対応に関してまとめる．

## IV.1 FIPS 140-1

FIPS 140-1 [FIP94]は，コンピュータや音声を含む通信システムにおける機密扱いではない情報の保護を目的とした暗号モジュールが満たすべきセキュリティ要件11項目を規定し，各セキュリティ要件に対して4段階のセキュリティレベルを定めている．

### IV.1.1 セキュリティレベル

暗号モジュールが用いられる幅広い応用や環境をカバーすべくFIPS 140-1で規定されているセキュリティレベル4段階は次の通りである．

- セキュリティレベル1

  **要約** 最低限のセキュリティレベルであり，製造グレードでの装備を超えた物理的なセキュリティ手段は必要とされない．

  **詳細** 最低限のセキュリティレベルを規定するレベル1では，例えば暗号アルゴリズムとしてFIPS認証のアルゴリズムを用いる等の暗号モジュールに対する基本的なセキュリティ要件が要求される．しかし製造グレードでの装備を超えるような物理的なセキュリティ手段は必要とされていない．

レベル 1 のシステムの例として，IC カードとその関連セキュリティ製品が挙げられており，IC カードはシステムの安全性を高め，暗号鍵の配布時のセキュアな記憶媒体として使用されるとしている．さらに PC の暗号ボードもレベル 1 システムの例として挙げられており，NIST は IC カードと暗号ボードでの NIST 暗号標準の適正な実行を認証している．※ただし IC カードに関しては IV.2.2 節でも触れるように FIPS 140-2 のドラフトでは見直しが行われており，レベル 1 の例から削除されている．

また一般利用目的の PC でのソフトウェア暗号機能もレベル 1 に相当する．PC 暗号ソフトウェアの実行はハードウェアに基づくシステムよりもコスト効率で優れるため，ハードウェアが高価すぎるという理由で暗号によるデータ保護が実施され難いような場合に有効となる．

- セキュリティレベル 2

  **要約**　レベル 1 の暗号モジュールの物理的セキュリティに加え，タンパリングの痕跡が残るコーティングやシール，こじあけ防止ロックなどが必要．

  **詳細**　レベル 2 ではタンパリングの痕跡が残るコーティングやシールを採用することにより，それらを破ることなしにモジュール中の暗号鍵や他の重要セキュリティパラメータ (用語解説 1 参照) へ物理的にアクセスすることを不可能にする．またこじあけ防止ロックは権限のない物理的アクセスを防止するためにカバーやドアに設置される．これらは物理的セキュリティをローコストで実現可能であり，不透明ハードコーティング，高価なタンパリング検知回路や，ゼロ化 (用語解説 3 参照) 回路など，より高いセキュリティレベルで要求されるコストの削減を可能にする．

  レベル 2 は役割ベース (role-based) の認証を規定する．この役割ベースの認証とは，オペレータの役割やその役割で実行可能なサービスをモジュールが認証するものである (セキュリティ要件 3 の役割とサービスを参照のこと)．

  またレベル 2 では，マルチユーザ時分割処理システム (TSS) でのソフトウェア暗号が，TCSEC (用語解説 4 参照) [TCS85] の C2 ないしそれと等価な信頼性を持つ OS 上で実行される場合に，このセキュリティレベルに該当するとしている．ハードウェア暗号に匹敵するセキュリティレベルでのソフトウェア暗号の実行には信頼性の高い OS を必要とすることがセキュリティ専門家から指摘されており，レベル 2 でコスト効率が実現される場合，マルチユーザ TSS でのソフト的な暗号処理が可能となる．

- セキュリティレベル 3

  **要約**　暗号モジュール内の秘密情報への不正アクセスを防ぐため，カバーやドアなどの物理的セキュリティが必要．権限なくカバーやドアが開けられようとした場合，秘密情報をゼロ化する．

**詳細**　レベル 3 では，現存の商業用セキュリティ製品における強化された物理的セキュリティが要求される．暗号モジュールへのタンパリングを防止するためのロック，コーティング，シールを必要とするレベル 2 に対し，レベル 3 はモジュール内部の重大なセキュリティパラメータへの不正アクセスを妨げることを目的とする．例えばマルチチップ内臓モジュールには強固な遮蔽 (enclosure) による封じ込めが要求され，カバーが除去されたりドアが開けられたりした場合は，重要セキュリティパラメータをゼロ化する．その他，モジュール内部への不正アクセスを防止するためにモジュールは不透明ハード容器に密閉されなければならない．

レベル 3 は，レベル 2 での役割ベースの認証よりもより安全性の高い ID ベース (identify-based) の認証を規定する．すなわちモジュールはオペレータの ID を認証し，さらに，その ID 認証されたオペレータが特定の役割を担い，その役割に関連したサービスを実行する権限を有していることを認証する (セキュリティ要件 3 の役割とサービスを参照のこと)．

またレベル 3 は重要セキュリティパラメータの入力や出力に関してより強いセキュリティ要件を規定する．重要セキュリティパラメータ用のデータポートは他のデータポートから物理的に分離されている必要があり，さらに，重要セキュリティパラメータは暗号化された形式でモジュールへの入出力が行われるか，他のインターフェースを介することなく直接モジュールへの入出力が実行されなければならない．

レベル 3 は，TCSEC による B1 ないしそれと同等な信頼性を持つ OS が重要セキュリティパラメータの入出力用の信頼性あるデータパスと共に用いられる場合に，マルチユーザー TSS でのソフトウェア暗号を認めている．信頼できるデータパスを有する B1 ないしそれ以上の信頼性を持った OS は，同じシステム上で実行される信頼性に欠けた他のソフトウェアから暗号ソフトウェアや重要セキュリティパラメータを保護することが可能であり，平文と暗号文との混合や暗号鍵の意図しない転送を阻止し得る．

- セキュリティレベル 4

**要約**　暗号モジュールの周囲全体に保護用の遮蔽 (envelope) が必要．レベル 3 はカバーやドアからの侵入を防ぐのに対し，レベル 4 はあらゆる方向からの侵入を検知し，秘密情報が盗まれる前にゼロ化する．また電圧や温度などの適正作動範囲からの変動を検知し，秘密情報をゼロ化する機構も要求される．

**詳細**　レベル 4 は最も高いセキュリティレベルを規定する．現存する製品の大半はこのレベルに対応していないが，商業用製品にもレベル 4 のセキュリティ要件の多くに準拠するものが存在する．レベル 4 は暗号モジュール周囲のプロテクト用遮蔽に関する物理的セキュリティの規定を行っており，迂

回路があり得る低いレベルでのタンパリング検出回路に対し，レベル 4 はあらゆる方向からの暗号モジュールへの侵入を検知・防止することを目的とする．例えば，暗号モジュールの遮蔽を切断する攻撃があった場合，この攻撃は検知され全ての重要セキュリティパラメータはゼロ化されなければならない．レベル 4 は，侵入者がデバイスをいじり得るような物理的に保護されていない環境での暗号モジュールの利用に対して特に有効である．

またレベル 4 は，環境条件の変動や，モジュールの適正動作範囲からの電圧や温度などの変動によってモジュールの安全性が損なわれることを防ぐことも目的とする．電圧や温度などの適正動作範囲からの逸脱は攻撃に対するモジュールの防衛機能を損なう可能性があるため，変動を検知し重要セキュリティパラメータをゼロ化するための保護機構，ないし，適正動作範囲から外れるような変動によってもモジュールが影響を受けないことを検証するための検査機構を備えることが規定されている．

レベル 4 では，TCSEC による B2 ないしそれと同等な信頼性を持つ OS が用いられる場合に，マルチユーザ TSS でのソフトウェア暗号が認められる．B2 に準拠した OS は，OS の安全面での適正運用を保証するものである．

以上のレベル 1〜4 が次節に挙げるセキュリティ要件に対してそれぞれ評価される．

### IV.1.2　セキュリティ要件

暗号モジュールの設計，実装，運用に関わるセキュリティ分野を網羅して FIPS 140-1 ではセキュリティ要件 11 項目が定められている．セキュリティ要件とセキュリティレベルの対応についてまとめたものが表 IV.1 である．

1. 暗号モジュール
   モジュールの物理的構成やセキュリティ方針などの規定，文書化
   暗号モジュールとは，暗号処理を行うハードウェア，ソフトウェア，ファームウェア，およびそれらの組み合わせと定義される．また暗号バウンダリーとは，暗号モジュールの物理的結合を明確に定義する連続した境界を指し，暗号モジュールがソフトウェアないしファームウェアを含む場合は，コードを実行するプロセッサを含む形で暗号バウンダリーが定義される．
   暗号モジュールの物理的形態として，シングルチップ・モジュール，マルチチップ内臓モジュール，マルチチップ・スタンドアローン・モジュールという 3 つが規定されており，これらに関してはセキュリティ要件 5 の物理的セキュリティで説明する．
   ドキュメントでは，暗号モジュールのハードウェア，ソフトウェア，ファームウェアの構成要素，それらの暗号バウンダリー，および，モジュールの物理的

表 IV.1: セキュリティ要件とセキュリティレベル

| | レベル 1 | レベル 2 | レベル 3 | レベル 4 |
|---|---|---|---|---|
| 暗号モジュール | ・暗号モジュールと暗号バウンダリーの規定<br>・ハードウェア，ソフトウェア，ファームウェアを含む暗号モジュールの解説<br>・モジュールのセキュリティ方針の規定 | | | |
| モジュール・インターフェース | ・必須およびオプションのインターフェースの規定<br>・全てのインターフェースとデータパスの規定 | | ・他のデータポートと分離された重要セキュリティパラメータ用のデータポート | |
| 役割とサービス | ・必須およびオプションの役割とサービスの論理的分離 | ・役割ベースのオペレータ認証 | ・ID ベースのオペレータ認証 | |
| 限定状態マシンモデル | ・有限状態でのマシンモデルの規定<br>・必須およびオプションの状態の規定<br>・状態変移ダイヤグラムと状態変移の規定 | | | |
| 物理的セキュリティ | ・製造グレードでの装備 | ・ロックないしタンパリングの痕跡が残るもの | ・カバーやドアへのタンパリング検知および応答 | ・タンパリング検知および応答用遮蔽 |
| ソフトウェア・セキュリティ | ・ソフトウェア設計の規定<br>・有限状態マシンモデルとソフトウェアの関連付け | | ・ハイレベル言語での実行 | ・公式モデル<br>・前後処理条件 |
| OS セキュリティ | ・実行コード<br>・認証<br>・シングルユーザ，シングルプロセス | ・制御されたアクセス保護 (TCSEC の C2) | ・ラベル化された保護 (TCSEC の B1)<br>・信頼性のある通信パス | ・構造化された保護 (TCSEC の B2) |
| 鍵管理 | ・FIPS 認定の鍵生成/配布技術 | | ・鍵の暗号形式での入出力，または直接的な入出力 | |
| 暗号アルゴリズム | ・機密扱いでない情報を保護するための FIPS 認定暗号アルゴリズム | | | |
| EMI/EMC | ・FCC Part 15, Subpart J, Class A への準拠 | | ・FCC Part 15, Subpart J, Class B への準拠 | |
| 自己検査 | ・パワーアップテストおよび条件付テスト | | | |

構成の規定が求められる．また暗号モジュールのセキュリティ方針 (用語解説 2 参照)，すなわちモジュールの運用上のセキュリティルールの完全な規定が必要とされ，特に，FIPS 140-1 およびその他の規格・標準によるセキュリティ要件から生じるセキュリティルールに準拠しなければならない．

2. モジュール・インターフェース

モジュールとの全ての情報の流れおよび物理的アクセスのコントロール

暗号モジュールは，全ての情報の流れと物理的アクセスがモジュールからの全入出力を定義する論理的インターフェースによって制限されるように設計される必要があり，論理的インターフェースとして次の 4 つを備えなければならない．

**データ入力インターフェース**　モジュールへ入力され処理される全てのデータ (平文データ，暗号文データ，暗号鍵，鍵運用データ，認証データ，他のモジュールからのステータス情報) がこのデータ入力インターフェースを介

して入力される．

**データ出力インターフェース**　モジュールから出力される全てのデータ (平文データ，暗号文データ，暗号鍵，鍵運用データ，認証データ，他のモジュールのコントロール情報) がこのデータ出力インターフェースを介して出力される．エラー発生時や自己検査時には全データの出力が抑制されなければならない．

**制御入力インターフェース**　モジュールの動作をコントロールするための全ての入力コマンド，シグナル，データ (スイッチ，ボタン，キーボード等によるマニュアル制御を含む) がこの制御入力インターフェースを介して入力される．

**状態出力インターフェース**　モジュールのステータスを表示するための全ての出力信号，データ (光，LED，ブザー，表示装置等のステータスコードや物理的表示を含む)がこの状態出力インターフェースを介して出力される．

セキュリティレベル 1 と 2 では，暗号鍵，認証データ，他の重要セキュリティパラメータに用いられるデータ入出力ポートは，モジュールの他のポートと共有されるが，セキュリティレベル 3 と 4 では，平文形式暗号鍵，平文形式認証データ，その他の保護されていない重要セキュリティパラメータに用いられるデータ入出力ポートは，モジュールの他のポートから物理的に分離されていなければならない．さらにこれらのポートは，平文形式暗号鍵，平文形式認証データ，その他の保護されていない重要セキュリティパラメータを直接入出力できなければならない．

また暗号モジュールは上記の必須インターフェースに加え，次のインターフェースをオプションとして備える．

**パワーインターフェース**　全ての外部電力がこのパワーインターフェースを介して入出力される．暗号モジュール内部で全ての電源が供給・維持される場合は不要となる．

**保守管理アクセスインターフェース**　モジュールの保守，サービス，修理を行うためのあらゆるデータ，コントロール，ステータス情報がこの保守管理アクセスインターフェースを介して入出力される．カバーやドアの除去などを含む全ての物理的アクセスのパスは，この保守管理アクセスインターフェースの一部として定義されなければならない．

3. 役割とサービス

認証された役割とそれに対応したサービスのサポート

暗号モジュールは，認証された役割とその役割が実行可能なサービスをサポートするように設計される必要がある．モジュールが同時に複数のオペレータを

サポートする場合は，各々のオペレータの役割とそれが行うサービスがモジュールで内部的に分離されていなければならない．

- 役割
  暗号モジュールがサポートすべき認証された役割として次の 3 つが規定されている．

  **ユーザー**　認証されたユーザーによるセキュリティサービスの受益，暗号処理やその他認証された機能の実行

  **暗号管理者**　認証された暗号管理者による暗号初期化ないし管理機能の実行

  **保守管理者**　認証された保守管理者による保守管理アクセスインターフェースへのアクセス，保守管理テストの実行，モジュールのメンテナンス・サービス・修理のためのデータの取得

  3 番目の保守管理アクセスは，モジュールが上述の保守管理アクセスインターフェースを備える場合にサポートされるものである．
- サービス
  暗号モジュールで実行される全てのサービス，処理，機能がここでいうサービスに属する．
  暗号モジュールは，ステータス表示 (現在の状態の出力) と自己検査 (自己検査の初期化と実行) をサービスとして最低限備える必要がある．またオプションとして，暗号処理を実行することなしにサービスを提供するためのバイパスを有し，バイパスの活性・非活性を行うサービスを要する．
- オペレータ認証
  暗号モジュールは特定の役割やサービスの実行に関する権限がオペレータに与えられているか否かを確認するために，セキュリティレベル 2 においては役割ベース認証ないし ID ベース認証を，セキュリティレベル 3 と 4 においては ID ベース認証を行う．それぞれの認証は以下の通りである．

  **役割ベース認証**　暗号モジュールは暗黙的ないし明示的に 1 つまたはそれ以上の役割をオペレータに選択させ，選択した役割とそれに関連したサービスを実行する権限がオペレータに与えられていることを認証する．モジュールは個々のオペレータに対して個別の認証を行う必要はない．

  **ID ベース認証**　暗号モジュールはオペレータの ID を認証し，その認証されたオペレータが特定の役割を担う権限を与えられていることを確認する．オペレータは個別に認証される必要がある．

4. 限定状態マシンモデル
   有限個の状態におけるモジュールの使用

暗号モジュールは，モジュールの全ての処理状態およびエラー状態を明確にするため，次に挙げる有限個の状態でのマシンモデルを用いて設計されなければならない．

**電源オン/オフ状態**　第一，第二，予備電源の状態

**暗号管理者状態**　暗号初期化や鍵管理機能などの暗号管理者機能が実行されている状態

**ユーザーサービス状態**　認証されたユーザーがセキュリティサービスの受益，暗号処理の実行，ないし，その他の認証されたユーザー機能を実行している状態

**自己検査状態**　モジュールにおける自己検査実行状態

**エラー状態**　自己検査の不成功，鍵や重要セキュリティパラメータが失われている状況での暗号化など，エラーが発生した状態

**未初期化状態**　モジュールにセキュリティパラメータがロードされていない状態

**アイドル状態**　暗号鍵やセキュリティパラメータがロードされており，モジュールがデータや制御の入力を待っている状態

**セイフティ状態**　暗号鍵やセキュリティパラメータはロードされているが，処理は行われない状態 (オペレータが一時的に不在になるような場合に不正アクセスからモジュールを保護するために用いられる)

**バイパス状態**　暗号処理を行うことなくサービスを提供する状態

**保守管理状態**　モジュールの保守管理を行う状態

以上の全ての状態が，モジュールの FIPS 140-1 への準拠を検証するために詳細に明示されなければならない．

5. 物理的セキュリティ

モジュールへの権限のない物理的アクセスの制限・抑止

暗号モジュールは，モジュール内部への不正な物理的アクセスを制限し，権限のない利用や書き換えを検知するための物理的セキュリティ機構を備えるように設計される必要がある．

暗号モジュールの物理的セキュリティ機構は，モジュールの物理的形態に大きく依存し，物理的セキュリティ要件はシングルチップ・モジュール，マルチチップ内臓モジュール，マルチチップ・スタンドアローン・モジュールの３つに分類される．以下，３つの形態について説明し，それぞれの形態で要求されるセキュリティレベル毎のセキュリティ要件を表 IV.2〜表 IV.4 にまとめる．

**シングルチップ・モジュール**　スタンドアローン・デバイスや物理的に保護されていないモジュールないし遮蔽に内蔵されて利用されるシングル IC チッ

表 IV.2: シングルチップ・モジュールに対するセキュリティ要件

| | |
|---|---|
| レベル 1 | ・製造グレードチップ |
| レベル 2 | ・レベル 1+<br>・タンパリングの痕跡が残る不透明コーティング |
| レベル 3 | ・レベル 1+2+<br>・タンパリングの痕跡が残る不透明ハードコーティング |
| レベル 4 | ・レベル 1+2+3+<br>・除去防止不透明ハードコーティング<br>・温度・電圧に対する EFP/EFT |

表 IV.3: マルチチップ内臓モジュールに対するセキュリティ要件

| | |
|---|---|
| レベル 1 | ・製造グレードのチップおよびマルチチップ |
| レベル 2 | ・レベル 1+<br>・タンパリングの痕跡が残る不透明コーティング |
| レベル 3 | ・レベル 1+2+<br>・不透明ハード容器，取り去り不可能な遮蔽，ないし，<br>検知回路とゼロ化回路を備えた取り去り可能なカバー<br>・保護された通気孔 |
| レベル 4 | ・レベル 1+2+3+<br>・タンパリング応答回路およびゼロ化回路を備えた<br>タンパリング検知用遮蔽<br>・温度・電圧に対する EFP/EFT |

表 IV.4: マルチチップ・スタンドアローン・モジュールに対するセキュリティ要件

| | |
|---|---|
| レベル 1 | ・製造グレードのチップ，マルチチップ，および遮蔽 |
| レベル 2 | ・レベル 1+<br>・カバーやドアに対する機械的ロックないし<br>タンパリングの痕跡が残るシールを備えた不透明遮蔽 |
| レベル 3 | ・レベル 1+2+<br>・不透明ハード容器，ないし，カバーやドアに対する<br>タンパリング応答やゼロ化回路を備えた強固な遮蔽<br>・保護された通気孔 |
| レベル 4 | ・レベル 1+2+3+<br>・ゼロ化回路を備えたタンパリング検知/応答用遮蔽<br>・温度・電圧に対する EFP/EFT |

プ．スマートカードや，暗号処理のためにシングルICチップが組み込まれたシステムがこれに該当する．

**マルチチップ内臓モジュール**　2個以上のICチップが接続され，物理的に保護されていないモジュールないし遮蔽に内蔵されたもの．

**マルチチップ・スタンドアローン・モジュール**　2個以上が結合されたチップを内蔵するシステム全体が物理的に保護されたモジュール．

表IV.2〜IV.4のレベル4におけるEFP (Environmental Failure Protection)とは，温度や電圧などの環境条件を常に測定する回路やデバイスを有し，暗号モジュールの適正動作範囲から外れた場合は直ちにモジュールをシャットダウンするか秘密情報のゼロ化を行う機構である．またEFT (Environmental Failure Testing)は，温度や電圧が適正動作範囲から外れた場合でも暗号モジュールの安全性を保障するための解析やシミュレーションなどの検査を指す．

6. ソフトウェア・セキュリティ

モジュールのセキュリティ方針に沿ったソフトウェアの使用

セキュリティレベル毎に次のようなソフトウェア・セキュリティ要件が要求される．

**セキュリティレベル 1&2**　ドキュメントにおいて次の項目を明記する必要がある．

- ソフトウェア設計の詳細説明
- ソフトウェア設計と暗号モジュールのセキュリティ方針との対応関係の詳細説明
- モジュール中の全てのソフトウェアがリストアップされた完全なソースコード

**セキュリティレベル 3**　セキュリティレベル1と2に加え，全てのソフトウェアのハイレベル言語での実行が必要．ただし，モジュールのパフォーマンスに本質的に影響する場合や，ハイレベル言語が利用できない場合は，ローレベル言語の使用が限定的に認められる．

**セキュリティレベル 4**　セキュリティレベル1，2，3に加え，セキュリティレベル4では次の項目が要求される．

- 暗号モジュールのセキュリティ方針に関する公式モデル (数式による厳密な表記)のドキュメント
- 公式モデルとセキュリティ方針との対応関係の詳細ドキュメント
- 個々のソフトウェアモジュール，ソフトウェア機能，ソフトウェア手続きに関して，モジュールで実行される処理の前後条件を明記したソースコード

- ソフトウェア設計と公式モデルとの対応関係の詳細なドキュメント

7. オペレーティングシステム・セキュリティ
   暗号ソフトウェアの実行の保護
   オペレーティングシステムのセキュリティ要件はセキュリティレベル毎に次のように規定されている.

   **セキュリティレベル 1** レベル 1 ではマルチユーザー，マルチプロセス・システムは元々対象外となっており,
   - 全ての暗号ソフトの実行コードでのインストール
   - FIPS 認定技術の暗号ソフトウェアへの適用
   - シングルユーザーのみでの暗号モジュールの利用
   - 暗号処理に限定した利用

   が要求される.

   **セキュリティレベル 2** セキュリティレベル 1 に加え,
   - コントロールされたアクセス保護を行う OS (TCSEC の C2，ないし，それと同等な FIPS 認定) 上での，暗号ソフトウェア，暗号鍵，その他の重要セキュリティパラメータ，コントロール・ステータス情報の利用
   - 不正アクセスから平文データ，暗号ソフトウェア，暗号鍵，認証データ，およびその他の重要セキュリティパラメータを保護するためのアクセス制御機構

   が要求される.

   **セキュリティレベル 3** レベル 1，2 に加え,
   - 暗号ソフトウェア，暗号鍵，その他の重要セキュリティパラメータ，コントロール・ステータス情報のラベル化と，ラベル化された保護を行う OS (TCSEC の B1，ないし，それと同等な FIPS 認定) 上での利用
   - 暗号鍵，認証データ，その他の重要セキュリティパラメータ，制御入力，状態出力の，信頼性のあるメカニズム (専用 I/O ポートやパス) を介してのやり取り
   - 暗号鍵，その他の重要セキュリティパラメータ，制御入力，状態出力を検査する機能の提供

   が要求される.

   **セキュリティレベル 4** セキュリティレベル 1，2，3 に加え
   - 暗号ソフトウエア，暗号鍵，その他の重要セキュリティパラメータ，コントロール・ステータス情報のラベル化と，構造化された保護を行う OS (TCSEC の B2，ないし，それと同等な FIPS 認定) 上での利用

   が必要となる.

8. 鍵管理

暗号鍵のライフサイクル全般にわたる暗号鍵セキュリティの確保

鍵管理は秘密鍵，公開鍵共に全ての暗号モジュールにおいて必要であり，秘密鍵やプライベート鍵 (秘密鍵とは秘密鍵暗合における秘密鍵を，プライベート鍵とは公開鍵暗合における秘密鍵を指す) は権限のない開示，変更，置換から，公開鍵は権限のない変更や置換から保護されなければならない．

**鍵生成**　暗号モジュールはオプションとして内部鍵の発生機能を持ち，その際 FIPS 認定の鍵生成アルゴリズムを採用する必要がある

**鍵配布**　鍵配布の方法には手動，自動，および両者の組み合わせがあり，暗号モジュールは FIPS 認定の鍵配布技術を採用しなければならない

**鍵入出力**　暗号鍵の手動入力では，暗号モジュールに鍵を入力している間，手動入力テストを用いてその正確性を確認する必要がある

**鍵記憶**　秘密鍵やプライベート鍵が平文形式で暗号モジュールに記録される際は，外部のモジュールからそれらの鍵へのアクセスが可能であってはならない

**鍵廃棄**　暗号モジュールは，モジュール中の全ての平文形式暗号鍵とその他の保護されていない重要セキュリティパラメータをゼロ化する機能を備える必要がある

**鍵保管**　暗号モジュールは保管のための鍵の出力をオプションとして備える．その際，鍵は暗号化されて出力されねばならない．

9. 暗号アルゴリズム

FIPS 認定の暗号アルゴリズムの使用

10. 電磁気的インターフェース (EMI)/電磁気的互換性 (EMC)

米国連邦通信委員会 (FCC) [FCC] 規格への準拠

暗号モジュールはセキュリティレベル毎に次のような FCC の EMI/EMC 要件に最低限準拠しなければならない．

**セキュリティレベル 1&2**　ビジネス利用である FCC Part 15, Subpart J, Class A への準拠

**セキュリティレベル 3&4**　ホーム利用である FCC Part 15, Subpart J, Class B への準拠

11. 自己検査

暗号処理の正常動作の確認と保障

暗号モジュールが正常に機能していることを確認するための自己検査機能は，パワーアップテストと条件付テストとに分類される．FIPS 140-1 で規定する以外の自己検査をオプションとして装備する場合もある．

自己テストが不成功に終わった場合，モジュールはエラー状態の入力とエラー表示の出力を状態インターフェースを介して行う．エラー状態の間は暗号処理を行ってはならず，データ出力インターフェースからのデータの出力も許されない．

**パワーアップテスト**　暗号モジュールがパワーアップされた後，モジュールは自己検査状態に入り次のようなテストを実行する．

- 暗号アルゴリズムテスト
  正しい出力が分かっているデータを用いてアルゴリズムを実行することでテストを行う (既知回答テスト)．モジュールが 2 つの独立な暗号アルゴリズムを含み，暗号アルゴリズムの正しい動作を確認するためにそれらが常に比較されている場合は，暗号アルゴリズムテストは省略してもよい．
- ソフトウェア/ファームウェアテスト
  エラー検知コード (EDC) ないし，データ認証コードや NIST デジタル署名アルゴリズムなどの FIPS 認定技術をソフトウェアおよびファームウェアへ適用する．
- 重要機能テスト
  モジュールのセキュリティ面での動作への影響が大きく，パワーアップテストの一部として検査が可能なその他全ての機能に対する検査．
- 統計的乱数生成テスト
  乱数ないし擬似乱数生成のランダム性の統計的テスト．セキュリティレベル 1，2 では必要とされないが，セキュリティレベル 3 では要求に応じて，セキュリティレベル 4 ではパワーアップ時ないし要求に応じて検査が行われる．

**条件付テスト**　次のテストがそれぞれのテストに応じた条件下で実行される．

- ペア一致テスト
  公開鍵やプライベート鍵を生成する暗号モジュールでは，比較によるテストを行う．公開鍵を平文に適用し，得られた暗号文を元の平文と比較して，両者が異なることを確認する．次にプライベート鍵を暗号文に適用してオリジナルの平文と比較し，両者が等しいことを確認する．

- ソフトウェア/ファームウェア・ロードテスト
  暗号モジュールへ外部からロードが可能なソフトウェアやファームウェアに対してFIPS認定の認証技術(データ認証コードやNISTデジタル署名アルゴリズムなど)による暗号機構を適用する.
- 手動鍵入力テスト
  暗号鍵が手動で暗号モジュールへ入力される場合は,鍵はパリティチェック値などのエラー検知コードを有するか,入力された鍵の正確性を確認するために二重に入力される必要がある.
- 連続乱数生成テスト
  乱数ないし擬似乱数の生成を行う暗号モジュールでは,一定値が発生していないかどうかのテストを行う.$n$ $(n > 15)$ビットのブロックを発生する場合,パワーアップ後に生成された初めのブロックは使用せずに次のブロックとの比較を行うために保存される.新しいブロックを生成する度に,前のブロックとの比較を行い,二つが等しくないことを確認する.

### IV.1.3 用語解説

以下にFIPS 140-1における主な用語の解説を示す.

1. 重要セキュリティパラメータ
   平文形式等の保護されていない形式での暗号鍵,認証データなどのセキュリティ関連の情報で,それの公開や変更が暗号モジュールの安全性やモジュールで保護されている情報の安全性を損ない得るもの.
2. セキュリティ方針
   暗号モジュールが準拠すべきFIPS 140-1のセキュリティ要件に基づくセキュリティルール,および,メーカーが補足的に規定したセキュリティルール.
3. ゼロ化(zeroization)
   データの復旧が不可能なように記憶装置の内容を変更することでデータを電気的に消去すること.
4. TCSEC (Trusted Computer System Evaluation Criteria)
   米国国防総省が1985年に定めた情報システムのセキュリティ機能を評価するための基準であり,通称,オレンジブックと呼ばれる.

## IV.2 FIPS 140-2

1994年に発表されたFIPS 140-1は5年以内の見直しが定められていた.NISTは1998年にパブリックコメントを求め,それに基づいて改定版であるFIPS 140-2の

ドラフト [FIP99] を 1999 年 11 月に公開した．2000 年 2 月 15 日までがパブリックコメントの募集期間となっており，承認された後は，6ヶ月後から FIPS 140-2 が有効に，さらにその 6ヶ月後までが FIPS 140-1 からの移行期間となる．FIPS 140-1 からの主な改定点について次節でまとめる．

### IV.2.1　FIPS 140-1 からの主な改定点

セキュリティレベル 4 段階の位置付けは現行の FIPS 140-1 とほぼ変わらないが，FIPS 140-2 ではセキュリティ要件の 2 項目について見直しが行われている．IV.1.2 節で挙げたセキュリティ要件 11 項目の内，6 のソフトウェア・セキュリティと 9 の暗号アルゴリズムがなくなり，代わりに

- 設計保障
  コンフィグレーション管理，配布と運用，開発，ガイダンス文書，
  機能テストの規定
- その他の攻撃の軽減
  現時点でセキュリティ要件が明確になっていない攻撃の軽減

が加えられた．削除された項目に関しては，ソフトウェア開発が付録にて，暗号アルゴリズムがセキュリティ要件 1 の暗号モジュールの仕様においてそれぞれ扱われている．以下では新しいセキュリティ要件であるその他の攻撃の軽減に関して説明する．

### IV.2.2　新しい攻撃法に対する処置

NIST は電力解析による暗号モジュールへの攻撃に関する声明を 1998 年 12 月に出している [FIP98]．それによると，電力解析攻撃はユーザー自身が攻撃者となるような攻撃を受け易い環境において，暗号モジュール内部の重要な秘密情報を読み出すことが可能であり，現行の FIPS 140-1 によるセキュリティ要件では対応できないとしていた．

これを受け FIPS 140-2 では，新たなセキュリティ要件として上述したその他の攻撃の軽減が追加された．そして攻撃法の例として，電力解析，タイミング解析，故障利用解析，TEMPEST(過渡電磁気パルス解析) に関する解説を行い，ベンダーへの注意を促している．それぞれの解析に対して FIPS 140-2 で言及されている対処法は次の通りである．

**電力解析**　外部 (直流) 電源を用いる暗号モジュールは電力解析に対してリスクが大きく，この基準が制定された時点では電力解析を完全に防ぎ得る対処法は分かっていない．しかしある程度電力解析を回避し得る方法として，電力消費を

平均化するコンデンサーの使用，内部電源の使用，暗号処理中の電力消費を平均化するアルゴリズムやプロセスの採用を挙げている．

**タイミング解析**　対処方法の 1 つとして，暗号処理中のタイミングの変動を抑えるアルゴリズムやプロセスの採用がある．

**故障利用解析**　物理的セキュリティが十分でない場合に故障利用解析に対するリスクが大きくなるため，適切な物理的セキュリティの処置が必要．

**TEMPEST**　TEMPEST は暗号モジュールからの電磁気的シグナルを測定することで，キー入力情報，スクリーン上の情報，暗号鍵などの重要セキュリティ情報を得る攻撃法である．ネットワークケーブルを含む全ての構成物のシールドが必要とされる．

また FIPS 140-1 では，スマートカードは安全性に優れた記録媒体であるとしてセキュリティレベル 1 の暗号モジュールの例に挙げられていたが，この記述は今回の改定で削除されている．

各種攻撃に対する上述した以上に具体的な対策法は FIPS 140-2 においても規定されておらず，1 つ以上の攻撃を軽減するために設計された暗号モジュールであればセキュリティ機構を備えると述べるに留まり，今後の技術開発動向を見守る形をとっている．

# 参考文献


[AK] R. Anderson, M. Kuhn, "Tamper Resistance – a Cautionary Note", *2nd USENIX Workshop on Electronic Commerce*, available at http://www.cl.cam.ac.uk/ mgk25/tamper.pdf

[AK97] R. Anderson, M. Kuhn, "Low Cost Attacks on Tamper Resistant Devices", *Security Protocols, 5th International Workshop*, 1997.

[And96] R. Anderson, "A Serious Weakness of DES", news:CMM.0.90.1.847310320.risko@chiron.csl.sri.com, 2 Nov, 1996.

[BD+96] F. Bao, R. Deng, Y. Han, A. Jeng, T. H. Nagir, and D. Narasimhalu, "A New Attack to RSA on Tamperproof Devices", news, 2 Nov, 1996.

[BDL96] D. Boneh, R. A. DeMillo, and R. J. Lipton, "A New Breed of Crypto Attack on "Tamperproof" Tokens Cracks Even the Strongest RSA Code", 1996.

[BDL97] D. Boneh, R. A. DeMillo, R. J. Lipton, "On the Importance of Checking Cryptographic Protocols for Faults", *Advances in Cryptology: Proceedings of Eurocrypt '97*, Springer-Verlag, 1997, pp.37-51.

[Bih94] E. Biham, "New Types of Cryptanalytic Attacks Using Related Keys", *Journal of Cryptology*, Vol.7, No.4, 1994.

[BS97] E. Biham, A. Shamir, "Differential Fault Analysis of Secret Key Cryptosystems", *Advances in Cryptology: Proceedings of CRYPTO '97*, Springer-Verlag, 1997, pp.513-525.

[BS99] E. Biham, A. Shamir, "Power Analysis of the Key Scheduling of the AES Candidates", *Proceedings of the Second Advanced Encryption Standard Candidate Conference*, 1999.

[Cha82] D. Chaum, "Blind Signatures for Untraceable Payments", *Proceedings of Advances in Cryptology - CRYPTO '82*, 1983, pp.199-203.

[CJ+99a] S. Chari, C. Jutla, J. Rao, P. Rohatgi, "A Cautionary Note Regarding Evaluation of AES Candidates on Smart-Cards", *Proceedings of the Second Advanced Encryption Standard Candidate Conference*, 1999.

[CJ+99b] S. Chari, C. Jutla, J. Rao, P. Rohatgi, "Toward Sound Approaches to Counter Power Analysis Attacks", *Advances in Cryptology: Proceedings of CRYPTO '99*, Springer-Verlag, 1999, pp.398-412.

[Cor99] J.-S. Coron, "Resistance Against Differential Power Analysis for Elliptic Curbe Cryptosystems", *Proceedings of CHES '99*, Springer-Verlag, 1999, pp.292-302.

[DES77] National Bureau of Standards, "Data Encryption Standard", Federal Information Processing Standards Publication 46, 1977.

[DH76] W. Diffie, M. E. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, IT-22, 6, Nov 1976, pp.644-654.

[DK+98] J.-F. Dhem, F. Koeune, P.-A. Leroux, P. Mestré, J.-J. Quisquater, J.-L. Willems, "A Practical Implementation of the Timing Attack", *UCL Report*, 1998, CG1998-1, available at http://www.dice.ucl.ac.be/crypto/techreports.html

[DR98] J. Daemen, V. Rijmen, "AES Proposal: Rijndael", AES submission, 1998.

[DR99] J. Daemen, V. Rijmen, "Resistance against Implementation Attacks: A Comparative Study of the AES Proposals", *Proceedings of the 2nd AES Candidate Conference*, 1999, pp.122-132.

[DSA95] U.S. DOC, Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186, 1994; American National Standard Institute, ANSI X9.30: Public Key Cryptography Using Irreversible Algorithm For the Financial Services Industry: Part I: The Digital Signature Algorithm (DSA), 1995.

[EMV96] "EMV '96 Integrated Circuit Card Specification for Payment Systems Version 3.0", 1996.

[EMV98] "EMV '96 Integrated Circuit Card Specification for Payment Systems Version 3.1.1", 1998.

[Fah99] P. N. Fahn, "IPA: A New Class of Power Attacks", *Proceedings of CHES' 99*, Springer-Verlag, 1999, pp.173-186.

[FCC] http://www.fcc.gov/

[FIP94] FIPS 140-1 "Security Requirements for Cryptographic Modules", NIST, 1994 January 11, available at http://csrc.nist.gov/fips/fips1401.htm

[FIP98] "Attacks on Cryptographic Modules based on the Analysis of Power Consumption", NIST, 1998 December 8, available at http://csrc.nist.gov/cryptval/140-1/poweranalysis.pdf

[FIP99] Draft FIPS 140-2 "Security Requirements for Cryptographic Modules", NIST, 1999, available at http://csrc.nist.gov/fips/dfips140-2.pdf

[GP99] L. Goubin, J. Patarin, "DES and Differential Power Analysis", *Proceedings of CHES '99*, Springer-Verlag, 1999, pp.158-172.

[His00] 久武, "マイクロソフトの IC カード戦略", Card Wave 2000 年 1 月号, 株式会社シーメディア, 2000, pp.52-55.

[HPS99] H. Handschuh, P. Paillier, J. Stern, "Probing Attacks on Tamper-Resistant Devices", *Proceedings of CHES '99*, Springer-Verlag, 1999, pp.303-315.

[IC97] "IC カード総覧 '97-'98", 株式会社シーメディア, 1997.

[IC99a] "大特集 立ち上がり直前, 国内 IC カード市場最前線", Card Wave 1999 年 3 月号, 株式会社シーメディア, 1999, p.17-.

[IC99b] "カードビジネスガイド '99 第 2 章カードビジネスを支えるテクノロジー", Card Wave 1999 年 5 月号, 株式会社シーメディア, 1999, p.72-.

[IC99c] "特集 ETC 実用化間近で, 活発化する IC カード市場動向", Card Wave 1999 年 9 月号, 株式会社シーメディア, 1999, p.10-.

[IC99d] "特集 '99 カードビジネス総決算", Card Wave 1999 年 12 月号, 株式会社シーメディア, 1999, p.10-.

[IC99e] "次世代 IC カードの需要予測と LSI&実装技術動向", ジャパン マーケティング サーベイ, 1999.

[Ima99] 今泉, "多種多様なカードターミナルに適用可能な OpenCard Framework Java をベースにカードサービスの新時代を作り上げるか？", Card Wave 1999 年 3 月号, 株式会社シーメディア, 1999, pp.54-55.

[ISO] "ISO/IEC 7816 Identification Cards - Integrated Circuit(s) Cards with Contacts".

[JC] http://www.sun.co.jp/javacard/

[JC97] "JavaCard 2.0 Application Programming Interfaces", Sun Microsystems, 1997.

[JI97] "日本工業規格準拠 JICSAP 外部端子付き IC カード仕様対応 発行ライブラリ仕様", IC カードシステム利用促進協議会, 1997.

[JQ97] M. Joye, J.-J. Quisquater, "Faulty RSA Encryption", *UCL Report*, 1997, CG1997-8, available at http://www.dice.ucl.ac.be/crypto/tech_reports/CG1997_8.ps.gz

[KJJ98] P. Kocher, J. Jaffe, B. Jun, "Introduction to Differential Power Analysis and Related Attacks", 1998, available at http://www.cryptography.com/dpa/technical/index.html

[KJJ99] P. Kocher, J. Jaffe, B. Jun, "Differential Power Analysis", *Advances in Cryptology: Proceedings of CRYPTO '99*, Springer-Verlag, 1999, pp.388-397.

[KK99] O. Kommerling, M. Kuhn, "Design Principles for Tamper-Resistant Smartcard Processors", *USENIX Workshop on Smartcard Technology*, 1999.

[Koc96] P. C. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems", *Advances in Cryptology: Proceedings of CRYPTO '96*, Springer-Verlag, 1996, pp.104-113.

[KQ99] F. Koeune, J.-J. Quisquater, "A Timing Attack against Rijndael", *UCL Report*, 1999, CG1999-1, available at http://www.dice.ucl.ac.be/crypto/techreports.html

[KSW96] J. Kelsey, B. Schneier, D. Wagner, "Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES", *LNCS1109, CRYPTO '96*, 1996.

[Kum99] 熊谷, "マイクロソフトにおけるスマートカードへの取り組み", Card Wave 1999 年 7 月号，株式会社シーメディア, 1999, pp.16-19.

[MDS99a] T. S. Messerges, E. A. Dabbish, R. H. Sloan, "Investigations of Power Analysis Attacks on Smartcards", *USENIX Workshop on Smartcard Technology*, 1999.

[MDS99b] T. S. Messerges, E. A. Dabbish, R. H. Sloan, "Power Analysis Attacks of Modular Exponentiation in Smartcards", *Proceedings of CHES '99*, Springer-Verlag, 1999, pp.144-157.

[MS97] 盛合, 志帆, "故障利用暗号攻撃によるブロック暗号の解読", *Proceeding of SCIS '97*, 1997, 6A.

[MUL] http://www.multos.com/

[OC] http://www.opencard.org/

[PC] http://www.pcscworkgroup.com/

[Pio98] O. Piou, "SiShell, the First Hardware Solution that Enhances Smart Card Security", *Proceedings of CARTES 98*, 1998.

[PN+93] B. Preneel, M. Nuttin, V. Rijmen, J. Buelens, "Cryptanalysis of the CFB Mode of the DES with a Reduced Number of Rounds", *LNCS, CRYPTO '93*, 1993.

[Riv95] R. L. Rivest, "The RC5 Encryption Algorithm", *Fast Software Encryption - Second International Workshop, Leuven, Belgium, LNCS 1008*, Springer-Verlag, 1995, pp.86-96.

[RSA78] R. L. Rivest, A. Shamir, L. M. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems", Communications of the ACM, 21, 1978, pp.120-126.

[SCW99] "Microsoft Smart Card of Windows Workshop 資料", Microsoft, 1999 年 8 月.

[TCS85] DOD 5200.28-STD, TCSEC "Department of Defense Trusted Computer System Evaluation Criteria", National Computer Security Center, December 1985.

[VISA] http://www.visa.com/

[YK99] 夕田, 木下, "スマートカード ガイドブック", 中央経済社, 1999.

[ZGK97] "全銀協 IC カード仕様 (改訂版)", 全国銀行協会連合会, 1997.

[ZM97] 鄭玉良, 松本勉, "暗号システムに対する乱数操作攻撃", *Proceeding of SCIS '97*, 6B, 1997.