Vol.2011-SE-171 No.27
2011/3/15

IC カード開発におけるセキュリティ試験手法に関する提案

平井 康雅† 市原 尚久†

組み込みシステムの一つである IC カードは，電子マネー等の多くのサービスで利用されている．また，これらを安全に実現するため，IC カード内に格納された重要なデータの漏洩，改竄等の多くの脅威への対抗手段として，IC カードには暗号化機能等の様々なセキュリティ機能が実装されている．一方，近年，攻撃手法の高度化に伴い，セキュリティ機能そのものをバイパスさせる攻撃が現実のものとなっている．これらの攻撃は，物理的な特性を利用した攻撃であり，攻撃への耐性を検証する方法として，製造された IC カードに対して擬似的な攻撃を用いた評価を行うことが必要不可欠になっている．しかしながら，これらの検証はコーディング時に実施することは不可能であり，問題が発見された場合には大きな手戻りが発生する．そこで，本稿で我々は，IC カードにおける攻撃方法をモデル化し，これらを用いることにより，コーディング時に対策の有効性を検証できる手法について提案する．

Proposal of Security Test Method for Smart Card Development

Yasumasa Hirai† and Naohisa Ichihara†

Smart Card is one of embedded systems and is used in several services such as electronic money. We have to implement countermeasures against much threat such as data leakage and falsification to Smart Card so that we provide services safely. On the other hand, recently attack techniques are being improved, and attackers are able to bypass security functions. Therefore it is necessary that we execute artificial attacks and evaluate whether Smart Card can prevent these attacks. However we are not able to perform this evaluation in the coding process, and we have to implement again if some attacks are found. In this paper, we propose one of the attacker's models and security property test method which enables to evaluate the effectiveness of security countermeasures in the coding process.

1. はじめに

近年，組み込みシステムの 1 つである IC カードは，入退室管理のための認証用デバイスとして，また，電子マネー等のサービスにおいて幅広く利用されている．しかしながら，これらのサービスでは，例えば他人へのなりすまし，電子マネーの不正な増加といった脅威が存在しており，これらの脅威に対抗し，安全なサービスを提供するために，IC カードにはセキュリティ対策として，認証機能や暗号化機能等のセキュリティ機能が実装されている．

また，ICカードでは，物理的な特性を利用し，ICカードの正規の入出力以外から得られる情報を利用したサイドチャネル攻撃と呼ばれる攻撃手法が存在する．これらに対応するため，上記のセキュリティ機能に加え，ハードウェア，ソフトウェア双方での対策についても求められる．サイドチャネル攻撃には，[1], [2], [3]で示されているように，消費電力，電磁波，温度，処理時間等のICカードから漏れる情報を利用し，ICカード内等に格納されている暗号秘密鍵情報を不正に取得するような受動的な攻撃がある．また，近年，攻撃手法の高度化が進み，上記のような攻撃に加え，ICカードの特性を利用して意図的に誤りを注入し，処理を実行させることにより，セキュリティ機能をバイパスする，あるいは，攻撃者にとって有利な情報を出力させる[4]のような能動的な攻撃（故障利用解析）についても現実のものとなってきている．そのため，サービスから要求されるセキュリティ機能に加え，これらのサイドチャネル攻撃等へのセキュリティ対策を実装することが必要不可欠となっている．また，これらのセキュリティ機能およびセキュリティ対策の有効性を客観的に検証することが重要であり，例えば，現在，ICカード等の安全性を客観的に評価する仕組みとしてCC (Common Criteria)認証[5]におけるAVAクラスでの評価が知られており，多くの製品で活用されている．

まず，セキュリティ機能に対する検証では，実装したソースコードに対して，機能テスト等を通じてセキュリティ機能が期待する機能を正しく提供していることを検証することが可能である．なお，セキュリティ機能に関する試験では，例えば認証が失敗した際の機能の振る舞い等，セキュリティの観点での異常系が発生した場合の振る舞いについてテストを実施することが重要である．これらのセキュリティ機能に関するテストは，IC カード化（ソフトウェアを，半導体工場を経由して IC カードとして製造された状態）された後にも網羅的に実施され，コーディング時と同様にセキュリティ機能が期待する機能を正しく提供していることを確認することが可能である．

一方，故障利用解析等のサイドチャネル攻撃へのセキュリティ対策の有効性は，IC

† 株式会社 NTT データ，〒135-8671 東京都江東区豊洲 3-3-9 豊洲センタービルアネックス
NTT DATA Corporation, Toyosu Center Bldg. Annex, 3-9, Toyosu 3-chome, Koto-ku, Tokyo 135-8671, Japan

カードに対して物理的な疑似攻撃を実行することにより検証する手法が用いられている．このような物理的な評価を実施することにより，セキュリティ対策の有効性を正しく把握することが可能となる一方で，攻撃が成功した場合には大幅な改修が必要となる可能性がある．

そこで本稿では，コーディング時点でセキュリティ対策の有効性について検証し，手戻りを軽減するためのセキュリティ試験手法について提案する．提案手法では，サイドチャネル攻撃の中でも故障利用解析に着目し，故障利用解析に対するセキュリティ試験の実施にあたり，評価する上で必要となるセキュリティレベルを決定するための攻撃者モデルについて示す．また，これらの攻撃者モデルに応じたセキュリティ試験手法について示す．

2. ICカード開発における既存のセキュリティ試験

IC カード開発におけるセキュリティ試験では，サービス実行上の不正行為を防止するために実装されるセキュリティ機能が期待する機能を正しく提供していることを確認するための試験と，脆弱性および攻撃に対する試験（penetration testing）の 2 種類の試験が存在し，双方の試験により IC カードの安全性を確認する必要がある．

本稿では，前者をセキュリティ機能試験，後者をセキュリティ特性試験と定義する．IC カードソフトウェアに対するセキュリティ試験のスコープを図 1 に示す．

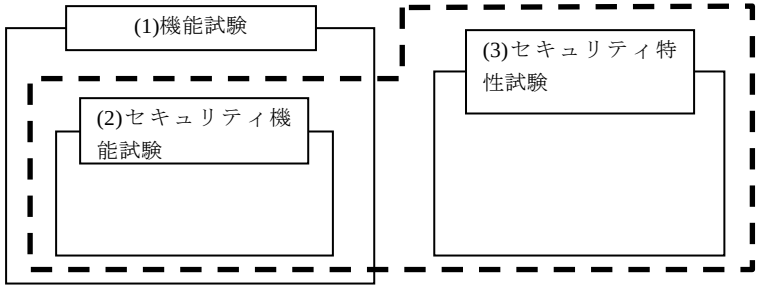

図 1　IC カードソフトウェアに対するセキュリティ試験のスコープ

以下に，図 1 で示す 3 つの試験について，その概要を示す．

(1) 機能試験

機能試験は，ソフトウェアが実現している機能が正しく動作することを確認するための試験であり，プログラムの構造（命令，分岐等）に着目し，網羅的に試験するホワイトボックステストや，入出力に着目し，ソフトウェアに実装されている機能が仕様通りに正しく動作するかを，ブラックボックステストを用いて検証する．

(2) セキュリティ機能試験

セキュリティ機能試験は，機能試験の一部であり，特に暗号機能，認証機能，アクセス制御機能といった，IC カードソフトウェアが安全にサービスを提供する上で必要となるセキュリティ機能が正しく機能を提供しているかを検証するテストである．セキュリティ機能試験では，一般的な試験手法と同様に，開発した IC カードソフトウェアに対し，ホワイトボックステスト，ブラックボックステスト実施することにより，期待する機能を正しく提供していることを検証する．また，IC カード化されたものに対しても，IC カードソフトウェアに対して実行するブラックボックステストと同様のセキュリティ機能試験を実施することにより，ハードウェアとして IC カードに正しく機能が実現されていることを検証する．

機能試験との主な差異は，不正な入力があった場合に，攻撃者に有利な情報が出力されるか等，セキュリティの観点で，セキュリティ機能が正しく動作しているかの検証を実施することである．一般的な機能試験における正常系，異常系に関する試験に加え，攻撃者の振る舞いを想定し，特に異常系の試験を実行する必要がある．例えば CC 認証における ATE クラスの評価においてセキュリティ機能試験が実施されている．セキュリティ機能試験の観点については 2.1 節にて一例を示す．

(3) セキュリティ特性試験

セキュリティ特性試験は，(2)のセキュリティ機能試験において試験対象となるセキュリティ機能に対して，攻撃者が物理特性を利用し，セキュリティ機能が提供している正規の入出力以外から入力を与える（故障注入攻撃等）ことによりセキュリティ機能をバイパスさせる，あるいは出力を取得することにより， IC カード内に格納された秘密情報（暗号機能における秘密鍵等）を不正に入手する攻撃等（電力解析攻撃等）に対し，対策が有効であるかを検証する試験である．上述したように，これらの不正行為は IC カードの物理特性を利用した攻撃であり，物理媒体として製造された IC カードに対して，疑似的な攻撃を実行することにより，セキュリティ対策の有効性を検証する．これらの試験を実施するためには，特殊な試験環境が必要となる．例えば CC 認証における AVA クラスの評価では，セキュリティ特性試験が実施されている．

2.1 セキュリティ機能試験

本節では，既存の IC カードソフトウェアに実装されるセキュリティ機能に対する試験の一例について示す．なお，セキュリティ機能として，認証機能である外部認証（External Authentication）を例に示す．

(4) 外部認証（External Authentication）

外部認証は，不正なシステムが正当なシステムになりすますことによる IC カード内の秘密情報への不正なアクセス等の脅威に対抗するための機能であり，国際標準規格 ISO/IEC 7816-4 において規定されている．外部認証機能は公開鍵暗号方式あるいは共通鍵暗号方式を用いて実現される．IC カードは外部認証機能を実行し，正当なシステムのみが知り得る秘密鍵を認証対象システムが確かに保有していることを署名検証等により確認することで，接続されるシステムの真正性を検証することが可能となる．

図 2 に，公開鍵暗号方式を用いた外部認証の例を示す．

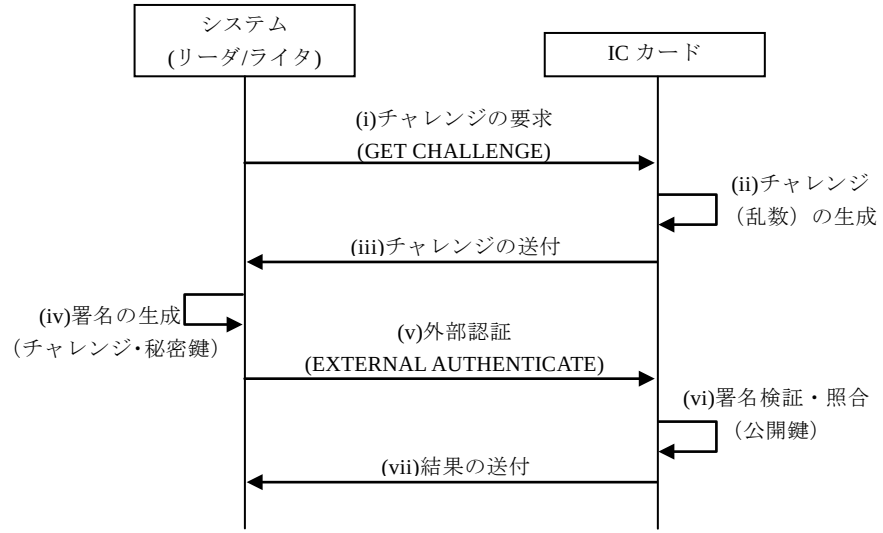

図 2　公開鍵暗号方式を用いた外部認証の例

外部認証では，図 2 に示す以下のステップをシステム-IC カード間で実行する．

- 実行ステップ

(i) チャレンジの要求
　認証を実行したいシステム（リーダ/ライタ）は，認証に用いるチャレンジを IC カードに要求

(ii) チャレンジ（乱数）の生成
　IC カードは，外部認証実行時の署名対象とするチャレンジとして，乱数を生成し，検証用に IC カード内に保存

(iii) チャレンジの送付
　IC カードは，チャレンジをシステムに送付

(iv) 署名の生成
　システムは，IC カードから送付されたチャレンジと，自らの正当性を検証するために IC カード内に格納している公開鍵と対となる秘密鍵を用いてチャレンジに対してディジタル署名を付与し，署名を生成

(v) 外部認証
　システムは，ステップ(iv)で付与した署名を IC カードにレスポンスとして送付

(vi) 署名検証・照合
　IC カードは，システムから送付された署名と，ステップ(ii)で保存したチャレンジ，予め格納されているシステムの公開鍵を用いて署名の検証・照合を実施

(vii)結果の送付
　IC カードは，署名検証結果をシステムに送付

セキュリティ機能試験では，ホワイトボックステストおよびブラックボックステストを用いて，上記のステップを実行するプログラムの構造や，入出力に着目した試験を実施し，正常系，異常系の双方で期待した動作となるかを検証する．また，これに加え，攻撃者を想定した場合の試験を実施する必要がある．攻撃者を想定した場合のセキュリティ機能試験において求められる試験観点の例について以下で示す．

- 攻撃の例：IC カードで生成される乱数の偏りを利用した攻撃

　外部認証を実行する際に生成される乱数に偏りがあった場合，攻撃者は，過去に正当なシステムが外部認証を実行した際にICカードから生成されたチャレンジと，そのチャレンジに対して付与された署名の対を収集する．次に，攻撃者は，正当なシステムになりすまし，外部認証を実行し，ICカードにチャレンジを要求

する．ICカードから送付されるチャレンジが，攻撃者が予め収集しているチャレンジに含まれる場合，攻撃者は，対応する署名をICカードに対し外部認証処理として送付することにより，秘密鍵を知ること無く外部認証処理を通過し，正当なシステムとして誤認識される．

- 攻撃の例に対する試験観点
  上述した攻撃例では，攻撃者は，外部認証におけるチャレンジとして用いられる乱数の偏りを利用することにより，外部認証を不正に通過させることが可能となる．そのため，外部認証のプログラムを実行し，実行結果として得られるチャレンジがセキュリティの観点で見た場合に安全であるようランダムに分布していることを確認する．これらの乱数について検証する方法として，例えば疑似乱数の評価であるFIPS140-2 [6]で示されている乱数検定方法等がある．

3. 既存のICカード開発におけるセキュリティ試験の課題

本章では，2 章で示した IC カード開発におけるセキュリティ試験手法の課題について述べる．

2 章で述べたように，IC カード開発におけるセキュリティ試験では，検証を実施するセキュリティ上のリスクに応じて，セキュリティ機能試験とセキュリティ特性試験の 2 種類のセキュリティ試験を実施する必要がある．また，セキュリティ特性試験では，IC カードから漏れだす消費電力量に関する情報を利用して秘密情報を推定する電力解析攻撃等の受動的な攻撃（[1]等）と，IC カードの外部から電力や光等を与えることにより IC カード内の状態を変化させる故障注入攻撃等の能動的な攻撃（[4]等）の双方について検証する．

表 1 にセキュリティ機能試験とセキュリティ特性試験の比較について示す．

表 1　セキュリティ機能試験とセキュリティ特性試験の比較

試験対象	セキュリティ機能試験	セキュリティ特性試験	
		①出力を不正入手する受動的なサイドチャネル攻撃に対する試験 (電力解析攻撃等)	②入力を不正に加える能動的なサイドチャネル攻撃に対する試験 （故障注入攻撃等）
ソースコード	○	×	×
物理媒体 （IC カード等）	○	○	○

セキュリティ機能試験は，ホワイトボックステスト，ブラックボックステストを用いて，ソースコードに対して実施することが可能である．また，ソースコードに基づき製造された物理媒体（IC カード）に対しても，ソースコードに対して実施したブラックボックステストと同様の試験項目について試験を実施することにより，セキュリティ機能が正しく動作していることを検証可能である．

一方，セキュリティ特性試験で対象としている攻撃は，IC カードの物理特性を利用した攻撃であり，その対策はソースコード上に実装されるが，対策が正しく振る舞い，セキュリティ機能のバイパスを防いでいるか等を検証するためには，物理媒体として製造された IC カードに対して検証を実施する必要がある．

そのため，IC カード化し，セキュリティ特性試験を実施した後に，ソースコードへの結果の反映を繰り返し実施するスパイラルアップでの開発を実施する方法が考えられるが，IC カード化するためには多大なコストや時間が必要となり実施は困難である．また，IC カード化された段階でセキュリティ特性試験により初めて脆弱性が発見された場合，大幅な改修が必要となる可能性がある．

そこで，本稿で，我々は物理媒体に対して実施していたセキュリティ特性試験をソースコードの段階で実施可能とする新たなセキュリティ試験手法を提案する．また，セキュリティ試験手法の提案にあたり，試験対象が満たすべきセキュリティレベルを設定する上で最も重要となる攻撃者モデルを，実際の攻撃手法および攻撃の困難さに基づき提案する．なお，表 1 で示したセキュリティ特性手法で対象とする 2 種類の攻撃のうち①の受動的な攻撃は多くの場合，物理的な対策により防がれるため，本手法では，②で示す能動的な攻撃に対する試験を対象とする．

4 章で，②の能動的な攻撃の例を示し，5 章で提案する攻撃者モデルおよび試験手法について述べる．

4. セキュリティ特性試験で対象とする能動的な攻撃の例

本章では，セキュリティ特性試験で対象とするサイドチャネル攻撃のうち，能動的な攻撃についての例を示す．能動的な攻撃は，ICカード内で処理が実行されている間に，外部から電力や光等を加えることにより処理の誤りを発生させ，その結果を利用して，セキュリティ機能をバイパスさせる攻撃等のである．近年，これらの攻撃は高度化しており，例えば[7]で示されているような，レーザーを用いて故障を発生させる手法が知られている．以下で，CRT-based-RSA署名に対する故障注入攻撃[4]および，認証機能に対するバイパス攻撃について示す．

4.1 CRT-based RSAに対する故障注入攻撃

CRT-based-RSAは, RSA cryptosystem[8]の秘密鍵を用いた復号あるいは署名における, べき乗剰余演算を実施する上で, 中国人剰余定理(Chinese Remainder Theorem)を用いて演算処理を高速化する手法である．RSAの演算とCRT-based-RSA署名の演算の主な差異は，RSA cryptosystemでは秘密鍵 d を生成する際にのみ使用される秘密鍵 p,q (公開鍵 $n = p \cdot q$)を復号演算で用いることである．高速に演算処理が実行可能であるため，ICカード等の省リソースデバイスにおいて広く利用されている．

以下に，CRT-based-RSA を用いたディジタル署名の概要について示す．なお，ここでは native な RSA 暗号アルゴリズムを用いることとする．

【前提】RSA 署名において，2 つの大きな素数を p,q（秘密鍵）とし，公開鍵 N は $N = p \cdot q$ であるとする． また，署名鍵を s する．

このとき，ある任意のメッセージ x（x は1から N の整数とする）に対して署名鍵 s を用いて施した署名を $S = x^s \bmod N$ とする．

※署名鍵 s は秘密情報であり，正当な署名者のみが保有している場合，署名 S を偽造することが困難である．

【CRT を用いた署名演算の高速化】

上記の RSA 署名 $S = x^s \bmod N$ に中国人剰余定理を適用する．

(1) $E_1 = x^s \bmod p$， $E_2 = x^s \bmod q$ を計算

(2) $\begin{matrix} a \equiv 1 (\bmod p) & b \equiv 0 (\bmod p) \\ a \equiv 0 (\bmod q) & b \equiv 1 (\bmod q) \end{matrix}$ を満たす a,b を計算

(3) $E = aE_1 + bE_2 \bmod N$ を計算

$E = S$ となり，小さな法 p,q に基づき高速な実行が可能となる．

IC カード内の署名演算において，ある入力（上記 x）を IC カードに与え，IC カード内に格納されている署名鍵（上記 s）を用いて署名を施し，署名（上記 S）を出力するような処理が考えられる．ここで，署名鍵 s は秘密鍵であり，s が漏洩した場合，署名が偽造されてしまう．

以下で，上記で示した CRT-based-RSA を用いた署名に対し，故障注入攻撃の例について示す．

【CRT-based-RSA を用いた署名に対する故障注入攻撃】

(1) 攻撃者は，任意のメッセージ x を入力として IC カードに与え， x に対して署名鍵 s を用いて施された正しい署名 E を取得する．

(2) 攻撃者は，(1)と同様のメッセージ x を入力し，IC カードに以下の署名処理を実行させる．

(a) IC カードは， $E_1 = x^s \bmod p$ を計算する

(b) **攻撃者は， E_1 に対して故障を注入し，任意の値 $\hat{E}_1$ に変更する**

(c) IC カードは， $E_2 = x^s \bmod q$ を計算

(d) IC カードは， $\hat{E} = a\hat{E}_1 + bE_2 \bmod N$ を計算

(e) IC カードは，誤った署名 $\hat{E}$ を出力する

(3) 攻撃者は，事前に取得している正しい署名 E と，故障注入により取得した誤った署名 $\hat{E}$ を用いて以下の計算を実施し，署名鍵 s を不正に取得する．

(a) $\begin{aligned} E - \hat{E} &= (aE_1 + bE_2) - (a\hat{E}_1 + bE_2) (\bmod N) \\ &= a(E_1 - \hat{E}_1) \end{aligned}$ を計算

(4) a は q の倍数であり，公開鍵 N との $E - \hat{E}$ の最大公約数を計算することにより p,q を求め，これに用いて攻撃者は署名鍵 s を計算する．

4.2 認証機能に対するバイパス攻撃

本節で対象とする認証機能は，IC カードに対して外部から入力されたパスワード等の認証子が，IC カード内に格納された認証子と一致するかを確認することにより，利用者の認証を行う場合を想定する．ここで認証子を 4Byte とし，1Byte 毎に入力された認証子と IC カード内に格納され認証子が一致しているかを比較する方法とする．

認証機能の例を図 3 に示す．

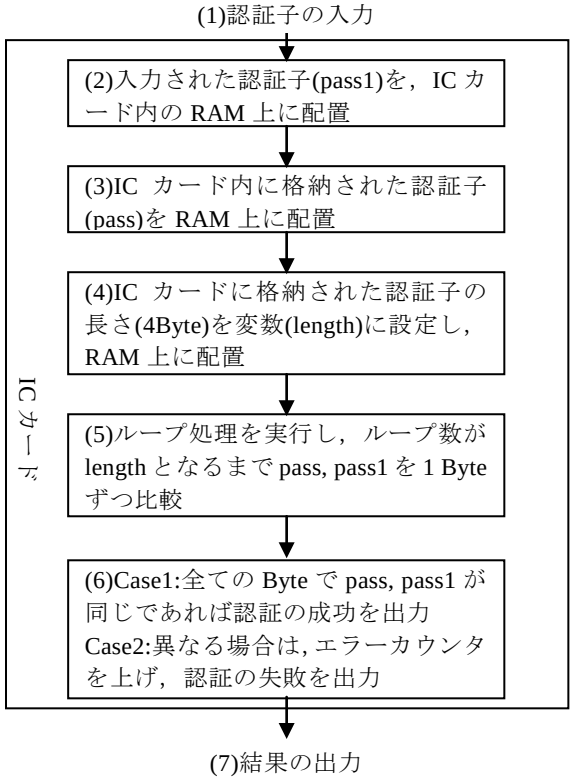

図 3　認証機能の例

ここで，攻撃者は，正当な認証子（pass）を知らないこととする．攻撃者は，以下のセキュリティ機能をバイパスする攻撃を実行し, IC カード内に格納された pass を推定する．

【認証機能に対するバイパス攻撃】
IC カード内に pass (0x31323334)が格納され，エラーカウンタ（最大攻撃回数）は 3 回に設定されていたとする．攻撃の手順は以下の通りである．

(a)　攻撃者は，passA を”0x00000000”に設定する
(b)　攻撃者は，IC カードに passA を認証子として入力する
(c)　攻撃者は, 図 3 の認証機能の処理(6)において, Case2 に分岐した場合, 故障注入攻撃を実施し, エラーカウンタを上げる処理をバイパスさせる
(d)　IC カードは，認証失敗を出力する
(e)　攻撃者は，結果が成功に分岐する”0x31323334”となるまで，passA に対する総当たり攻撃，(a)～(b)の手順を実施する

攻撃者はエラーカウンタにより 3 回しか試行できないが, 故障注入攻撃によりエラーカウンタを無効化し，認証機能をバイパスすることが可能となる．

5. 提案するセキュリティ試験手法

本章では，4 章で例示したような能動的な攻撃に対して有効な対策が実現されているかをコーディング時に検証可能とするセキュリティ試験手法について提案する．

提案するセキュリティ試験では，想定する攻撃者に対して，セキュリティ対策が有効であるかを検証する．セキュリティ試験を実施するにあたり，攻撃者が無限の能力を有すると想定した場合, 全てのデータは攻撃者の思い通りに変更可能であり, 且つ, セキュリティ機能についても思い通りにバイパス可能であるため，多くのセキュリティ対策を無効化することが可能となる．そのため，例えば，このような無限の能力を有する攻撃者を想定した場合に，セキュリティ試験を通過するためには，必要以上のセキュリティ対策を実装する必要が生じる．このようなことが生じないよう， IC カードに対して実施されるセキュリティ特性試験に基づき，セキュリティ試験において確認するセキュリティのレベルを決定するための攻撃者モデルについて 5.1 節にて示す．また，これらの攻撃者モデルに基づきソフトウェアに対してテストを実行するための手法を 5.2 節で示す．

5.1 提案する攻撃者モデル

本節で示す攻撃者モデルは，想定する攻撃者の能力を定義するものであり，IC カードに対する能動的な攻撃に関するセキュリティ特性試験に基づき定義する．攻撃者が能動的な攻撃を実行した場合の，考慮すべき誤った振る舞いは 2 つに分類できる．

- 処理のバイパス
  IC カードにおいて処理実行中に，故障の注入により本来実行されるべき命令が実行されないことにより，処理がバイパスされる

- メモリ上に配置されるデータの改変
  IC カードにおいて処理実行中に,メモリに対して故障が注入されることにより，メモリ上のデータが改変される

これらの事象に対し，攻撃者の能力に応じて変化するパラメータを定義する．

なお，図 4 で示すように，サービス実施上，重要な機能を実現しているモジュールが保証すべき処理を「クリティカルプロセス」と定義する．また，サービス実施上，重要なデータおよびクリティカルプロセスで使用するデータを「クリティカルデータ」と定義する．攻撃者は，これらに対して攻撃を試みる．

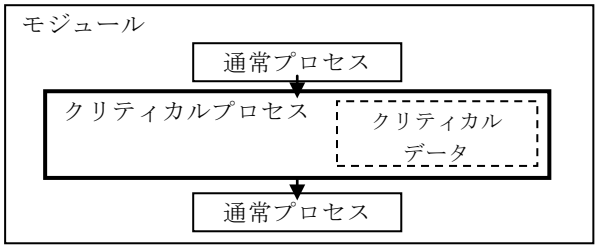

図 4　クリティカルプロセスとクリティカルデータ

処理のバイパスに関するパラメータを表 2 に示す．

表 2　処理のバイパスに関するパラメータ

項番	パラメータ	値
1	バイパス位置	クリティカルプロセス開始～終了
2		クリティカルプロセス実行中（特定位置）～クリティカルプロセス終了
3		クリティカルプロセス開始～クリティカルプロセス実行中（特定位置）
4	バイパス回数	1, ... ,n 回
5	バイパス実行間隔	1step, ... ,n step

まず，「バイパス位置」は，クリティカルプロセスの実行時間や処理内容に応じて攻撃の困難さが変化するが，処理が短い場合，nsec 単位での処理となるため，実行中のクリティカルプロセスを特定位置からバイパスさせることは困難である．また，故障による影響時間の制御することはさらに困難であり，特定位置までバイパスさせる項番 3 が最も高い能力を有する攻撃者を想定していると考えられる．

また，「バイパス回数」は，セキュリティ特性試験における故障を注入可能な回数に対応する．故障を注入するタイミングにも依存するが，一般的に一度の故障の注入による影響は，ある程度の時間継続されると考えられており，例えば，複数回の故障を注入しても，IC カードへの影響は 1 度に限定される場合が考えられる．そのため，より多くのバイパスを実行可能な攻撃者が高い能力を有する攻撃者として想定可能であると考える．

「バイパス実行間隔」も，「バイパス回数」と同様に，1 度の故障の注入が IC カードに影響を及ぼす時間に関係するため，バイパス実行間隔が短い方がより高い能力を有する攻撃者を想定していると考えられる．

次に，メモリ上に配置されるデータの改変に関するパラメータを表 3 に示す．

表 3　メモリ上に配置されるデータの改変に関するパラメータ

項番	パラメータ	値
1	変更可能な変数	1, ... , n 種類
2	変更頻度	1, ..., n 回
3	変更可能データ長	1 bit, ..., 1 Byte, ..., 変数長, ... ,
4	変更可能な値	データ長 1bit: 0 to 1, 1 to 0
5		データ長 1Byte：”? to all 0”, “? to all 1”, “? to 乱数”,” ? to 任意のデータ”
6		データ長 変数長：”? to all 0”, “? to all 1”, “? to 乱数”, “? to 任意のデータ”

まず，「変更可能な変数」は，故障を注入する対象となるメモリ箇所に，また，「変更頻度」は，故障を注入する回数に対応する．これら 2 つのパラメータは依存関係にあり，「1 種類の変数に対して，1 度の変更を加える」方法から，「複数種類の変数に対してそれぞれ複数回の変更を加える」方法まで,その組み合わせによって決定される．変更可能な変数が増加することは，物理媒体に対するセキュリティ特性試験では，故障を注入するための機器（レーザー等）が複数存在するか，あるいは，故障を注入したい別のメモリの位置に機器を移動させることを意味している．前者は攻撃対象が増加することにより，攻撃に要するコストが増加する．また，後者は，nsec 単位の時間

での正確な機器の移動が必要となる．そのため，より多くの変数を変更可能な攻撃者が，より高い攻撃能力を有すると想定できる．また，同一変数に対し，複数頻度での故障の注入については，表 2 の「バイパス回数」と同様に，1 度の故障注入によるある程度の時間の影響が考えられるため，攻撃者にとって都合が良い，意図された処理実行時における複数回変更が可能な攻撃者の方が高い能力を有していると考えられる．

また，「変更可能なデータ長」については，故障を注入した場合に，隣接する他のメモリへの影響が考えられるため，より小さなデータ（例えば 1bit）に対して限定的な改変ができる攻撃者が，より能力の高い攻撃者であると想定できる．

「変更可能な値」については，攻撃者にとって有利な任意のデータに変更可能な攻撃者が最も能力が高い攻撃者であると考えられる．また，乱数に変更する攻撃では，プログラム上意味のあるデータへの改変は含まないものとする．

なお，IC カードに対するセキュリティ特性試験では，物理特性を利用して故障を注入するため，理論通りの結果を得ることが困難であり，また，IC カード内部での故障注入対象の処理やメモリの位置を正確に特定することは困難であるため，4 章で示すような攻撃が容易に成功するわけではない．しかしながら，セキュリティ試験においては，上記で示した攻撃者モデルに基づき，攻撃者は，正確に攻撃が可能であるものと仮定し，セキュリティ試験を実施することとする．

5.2 提案するセキュリティ試験の実施方法

本節では，5.1 節で示した攻撃者モデルに基づき，ソースコードに対して，セキュリティ特性試験を実施する方法について示す．なお，Fault Injectionに対する試験を実施する方法として[9]等で示されている方法がある．

以下で，手順を示す．

(1) 試験対象となるソースコードにおいて，サービス実施上，重要な機能およびデータを特定する
(2) (1)に基づき，クリティカルプロセスおよびクリティカルデータを定義する
(3) 最新の攻撃手法に基づき，5.1 節の攻撃者モデルにおけるパラメータを設定し，セキュリティ試験で確認するセキュリティのレベルを決定する
(4) 攻撃者モデルのパラメータに従い，セキュリティ試験を実施する

セキュリティ試験は，例えばデバッガ上で実施し，処理のバイパスはプログラム実行中のプログラムカウンタ（PC）の変更，データの改変はプログラム実行中のメモリ上のデータの改竄により疑似的に実現する．

以下で，セキュリティ試験の実施例を示す．

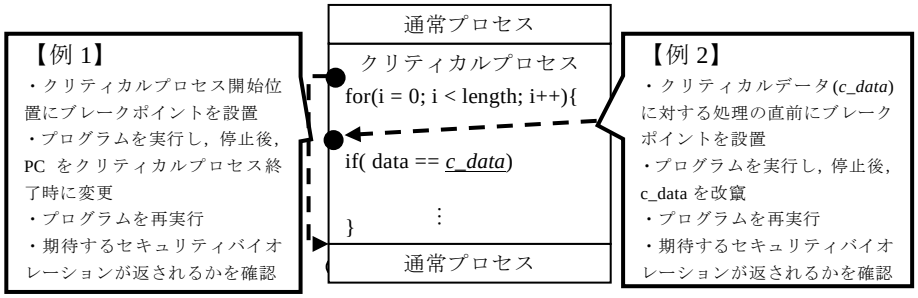

図 5 セキュリティ試験の実施例

6. おわりに

本稿では，従来，物理媒体に対して実施されていたセキュリティ特性試験をソースコードに対して疑似的に実行するセキュリティ試験手法に関する提案方式を示した．また，提案方式では，攻撃の高度化に伴い変化する攻撃者の能力をモデル化した攻撃者モデルを示し，攻撃者モデルに基づいた試験実施方法の例を示した．今後，本手法に基づき，セキュリティ特性試験との比較評価を実施し，手法の有効性を検証する．

参考文献

1) Paul Kocher, Joshua Jaffe, Benjamin Jun.: Differential Power Analysis, CRYPTO'99, pp.388-397, 1999.
2) K. Gandolfi, C. Mourtel, F. Olivier.: Electromagnetic analysis: concrete results, CHES2001, pp.251-261, 2001.
3) Paul Kocher.: Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems, Crypto'96, pp.104-113, 1996.
4) D. Boneh, R. A. DeMillo, R. J. Lipton.: On the Importance of Checking Cryptographic Protocols for Faults, Eurocrypt'97, 1997.
5) Common Criteria: http://www.commoncriteriaportal.org/
6) FIPS140-2: http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf
7) 藤原，諸藤，成吉，塚本，山梨，川村: レーザーアタックの優位的評価手法について，SCIS2011.
8) R. Rivest, A. Shamir and L. Adleman.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM, 21 (2), pp. 120-126, 1978.
9) 黒田，柴山: Software Fault Injection を用いた開発時テスト支援環境，日本ソフトウェア科学会第 22 回大会論文集