情報セキュリティモジュールの認証製品利用に関するガイドライン

東京大学生産技術研究所　松浦研究室

Version 1.0 (平成 22 年 5 月)

目次

第 I 部　本ガイドラインの位置づけ 3

1 **背景** 3

2 **対象** 3

3 **理論** 4

第 II 部　ガイドライン 8

4 **概要** 8

5 **手順一：採択枠組みの開発** 9

5.1 「採択枠組み」とは何か？ 9

5.2 コストストラクチャの構築 10

5.3 リスクストラクチャの構築 10

6 **手順二：候補の分析** 11

6.1 総コストの算出 11

6.2 リスクスコアの算出 11

6.3 リスク許容境界の設定 14

6.4 文書化の始動 18

7 **手順三：統合** 18

7.1 候補の選択 18

7.2 選択結果の統合 18

7.3 理論に照らした検証 19

7.4 文書の更新 20

8 **手順四：文書化の完了** 21

8.1 セキュリティ仕様の作成 21

8.2　予算案の準備 21
8.3　要件変更への対応 21
8.4　改良改革 21

第 I 部
本ガイドラインの位置づけ

1　背景

情報セキュリティシステムを構築する際に、要件に基づき最適なシステム設計をしたいという要求は強い。しかし残念ながら、技術面だけにとどまらずリスクを科学的に扱う理論基盤に基づいた設計手法や評価手法は存在しない。ただし、IT システム設計案全体を見通しよく比較評価するためにベストプラクティスを文書化した VMM (Value Measuring Methodology)[1],[2] が米国連邦政府で利用実績を積み重ねるなど、一定の文書化さえできれば新たな設計評価手法も利用され得るだけの土壌は整いつつある。

一方、システムの構成要素となるモジュール単位では、要件の客観的な表現を容易にする製品認証制度（モジュール製品を試験し、認証する制度）が注目を集めている。例えば、日本の暗号モジュール試験及び認証制度 (JCMVP: Japan Cryptographic Module Validation Program)[3] は、既に 2006 年 6 月より、試行運用が情報処理推進機構（IPA）により開始された。そして、2007 年 4 月に正式運用が始まり、現在に至っている。JCMVP では、暗号モジュールを 4 つのレベルに分けて試験及び認証している。生体認証モジュールに関しても、研究段階だが、適当な複数レベルに分けた試験及び認証制度を可能とするための評価基準作りが試みられている [4]。

モジュール選択を何度も行いながらシステムを構成する場合、ベストプラクティスに基づいて設計チームを稼働させることが多いだろう。本ガイドラインは、それら多くのモジュール選択結果の間に理論的整合性（ミクロ経済学に基づく解析的な理論モデルとの整合性）があるかどうかを簡易に検証し、代替案比較によるシステム設計を支援するためのドキュメントである [5], [6]。[*1]

2　対象

システム構築には、以下のように様々な立場の利害関係者が関わっている。

1. ソフトウェアモジュールやハードウェアモジュールの提供者
2. モジュールを用いてシステムを構築するシステムベンダ

[*1] 本ガイドラインに至る研究の一部は、新エネルギー・産業技術総合開発機構 (NEDO) 産業技術研究助成事業（若手研究グラント）による助成を受けた。

3. ベンダから最終製品を購入する消費者
4. 当該産業の所轄官庁

本ガイドラインの対象（想定する利用者）は、2. である。

製品認証制度としては、JCMVP が既に運用されていることに配慮し、4 つのレベルに分けて試験及び認証する制度を対象とする。ただし、レベル数が異なっても、基本的に同じ手法で PDCA サイクルを構築できる。

3 理論

ミクロ経済学に基づく情報セキュリティへの最適投資理論が盛んに研究されるようになってから 10 年弱が経過している。本ガイドラインでは、

- 定式化を最適投資以外の問題にも適用でき、一般性が高い。
- 公的データによる実証研究が存在する。
- 豊富な含意が導出されており、それらがベストプラクティスによく整合している。

を全て満たす貴重なモデルである Gordon-Loeb モデル [7] とその拡張 [8] を理論基盤とする。そこでは、次の基本パラメータを定義して定式化が行われる。

λ: 攻撃等の脅威が成功した時の経済的損失。

t: 攻撃等の脅威が生起する確率 $(0 \leq t \leq 1)$。理論展開の中では、表記の簡略化目的で、$L = t\lambda$ と定義されるパラメータ L（潜在損失と呼ばれる）も用いられる。

v: 攻撃等の脅威が生起した際に、生起したという条件の下で、脅威が成功する条件付き確率 $(0 \leq v \leq 1)$。**脆弱性**と呼ばれる。

今、金額 $z \geq 0$ の情報セキュリティ投資を行うことを考える。Gordon-Loeb モデルでは、投資によって脆弱性を低減できると見なす。そして、低減後の脆弱性は投資金額と投資前の脆弱性のみに依存すると仮定し、低減後の脆弱性を $S(z, v)$ と表記する。この $S(z, v)$ を、セキュリティ侵害確率 (SBP: security breach probability) 関数と呼ぶ。最適投資問題は、この脆弱性低減による損失低減の期待値から投資額を差し引いた値 ENBIS(Expected Net Benefits from an investment in Information Security) を最大化する問題として定式化される。SBP 関数としては、いくつかの関数系が検討されている。とくに、$S(z, v) = v^{\alpha z+1}$ で定義される関数系は、「極めて低い脆弱性や高い脆弱性ではなく、中程度の脆弱性に対して重点的に投資すべきである」という投資指針を表現した解析

解をもたらし、唯一の実証サポートを有する関数系として注目されている [9], [10]。正の定数 α は、「情報セキュリティの生産性」を表現していると考えられている。

ただし、情報セキュリティ投資の効果には本来、攻撃等の脅威の抑止力も含まれるはずである。そこで、拡張モデルでは、

- 投資に応じて脅威生起確率 t も低減される

と捉えて抑止力を理論に取り込む。投資後の脅威生起確率を $T(z,t) = t^{\beta z+1}$ でモデル化し、非負の定数 β を「脅威低減に関する（情報セキュリティの）生産性」と呼ぶ。これに伴い、先の正の定数 α は、「脆弱性低減に関する（情報セキュリティの）生産性」と呼ぶこととなる。この時、ENBIS 最大化問題

$$ENBIS(z) = vt\lambda - S(z,v)T(z,t)\lambda - z \;\rightarrow\; \max. \tag{1}$$

の解析解は

$$z^* = \frac{\ln\left\{-1/\left(vt\lambda\ln(v^\alpha t^\beta)\right)\right\}}{\ln(v^\alpha t^\beta)} = \frac{\ln\frac{1}{-vL\{\alpha(\ln v)+\beta(\ln t)\}}}{\alpha(\ln v)+\beta(\ln t)} \tag{2}$$

で与えられる。ただし、

$$F(v) \equiv v\ln v + \frac{\beta\ln t}{\alpha}\cdot v + \frac{1}{\alpha L} \geq 0 \tag{3}$$

の場合には、投資額をゼロに限りなく近づけた時の限界効用が限界費用を上回らないので、投資はなされない。$F(v) < 0$ の場合に限り、(2) 式が最適な投資額を与える。

上記の結果は、横軸に α、縦軸に β をとった平面に表現すると体系的に理解できる。すなわち、図 1 のようにまとめられる。この平面（二次元空間）を、**情報セキュリティの生産性空間**と呼ぶ。

図 1 において、左下部の領域（Case I と Case II-A-1）は零投資領域である[*2]。生産性がこの領域内であれば、最適投資額 z^* は脆弱性 v の値にかかわらず、0 となる。導入可能な情報セキュリティ対策が極めて貧弱でどちらの観点でも生産性が低い場合には、対策を導入しても有意義ではない。零投資領域は、この直感に対応している。

続いて、図 1 における右下部の領域（Case II-B-2-a）は中庸脆弱性重視領域である。生産性がこの領域内であれば、脆弱性に関する 2 つの閾値 V_1, V_2 が存在し、低い脆弱性 $(v \leq V_1)$ と高い脆弱性 $(v \geq V_2)$ に対しては最適投資額 $z^* = 0$ となり、中程度の脆弱性

[*2] 「Case I」などの場合分けの番号・記号は、文献 [8] に示された証明に現れる場合分けの番号・記号に対応している。

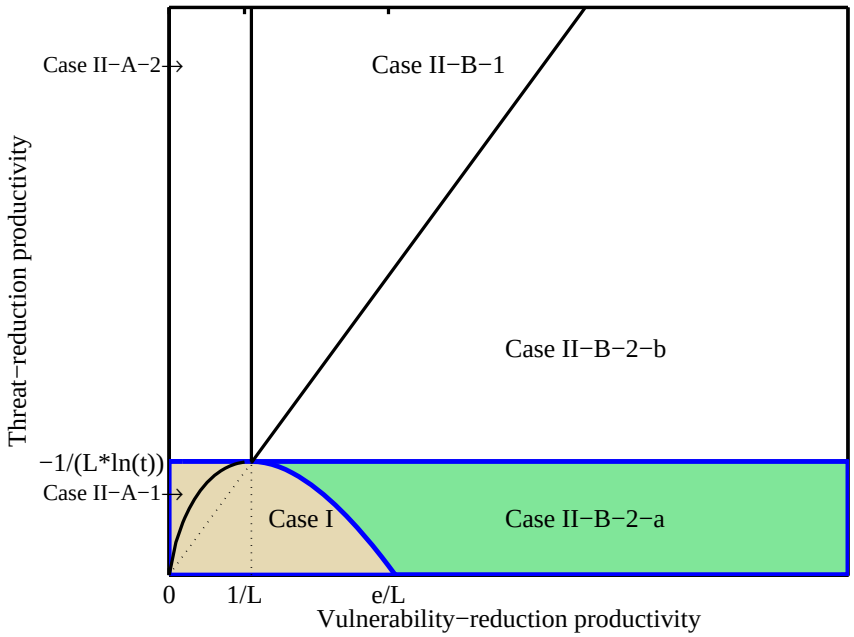

図 1　情報セキュリティの生産性空間.

$(V_1 < v < V_2)$ に対して (2) 式に従う投資 z^* が推奨される。この領域に相当する数値例を、図 2 に示す。図 2 には、文献 [8] に記されている解析解導出方法の理解を助けるため、最適投資曲線[*3]だけでなく曲線 $-\alpha v \ln v - \beta v \ln t$ も併記した。重要な含意は、

- 抑止効果があまり高くない場合には、費用対効果の観点で最適性を考えると、中程度の脆弱性対策に重点を置くべきである

ということである。

最後に、図 1 における上部領域（Case II-A-2、Case II-B-1、そして Case II-B-2-b）は高脆弱性重視領域である。生産性がこの領域内であれば、ある閾値 V_1 以下の脆弱性 v までは最適投資額 z^* が 0 であり、その閾値を越える高い脆弱性に対しては、(2) 式に従う投資 z^* が推奨される。この領域に相当する数値例を、図 3 に示す。重要な含意は、

- 抑止効果が充分高い場合には、費用対効果の観点で最適性を考えても、高い脆弱性に投資する余裕が出てくる

[*3] 横軸に脆弱性 v、縦軸に最適投資額 z^* をとった曲線を最適投資曲線と呼ぶ。脅威 t ではなく脆弱性を横軸にとる理由は、脆弱性は基本的に自己アセスメントに基づくからである。これに対し、脅威は、環境を分析しなければアセスメントできない。

ということである。元の Gordon-Loeb モデルにおいても、SBP 関数として別の関数系を採用すれば、高脆弱性重視の最適投資曲線を導出できる。

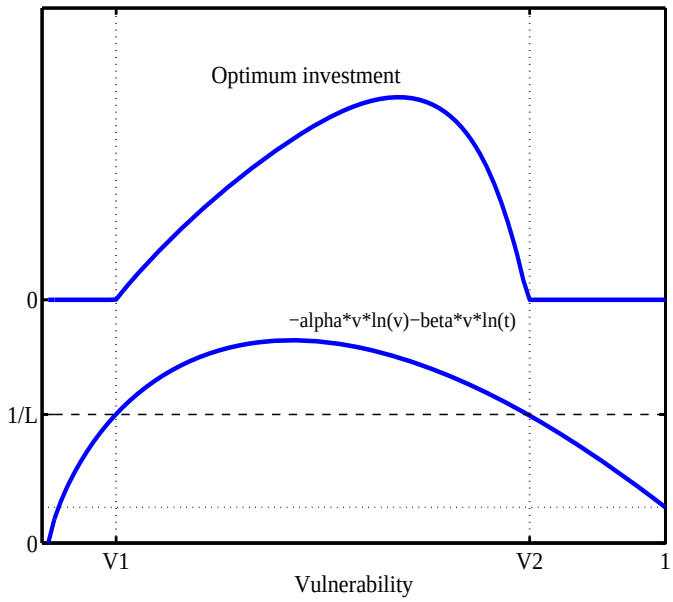

図 2　中庸脆弱性重視領域の数値例と最適投資曲線 ($\alpha = 0.00001$, $\beta = 0.000001$, $t = 0.5$, $\lambda = 800000$).

(2) 式を被害額の期待値 vL で除して、$x = -vL(\alpha \ln v + \beta \ln t)$ とおけば、

$$\frac{z^*}{vL} = -\frac{1}{x} \ln \left(\frac{1}{x} \right) \tag{4}$$

となる。(4) 式の右辺は、$x = e$ において最大値 $1/e$ をとる。ゆえに、最適投資額は、被害額の期待値の高々 $1/e$ 倍すなわち 37% である。コストが被害額の期待値の 37% を越える場合には、そのままでは過剰投資の恐れがある。この含意は、元の Gordon-Loeb モデルにおいても導出できる。

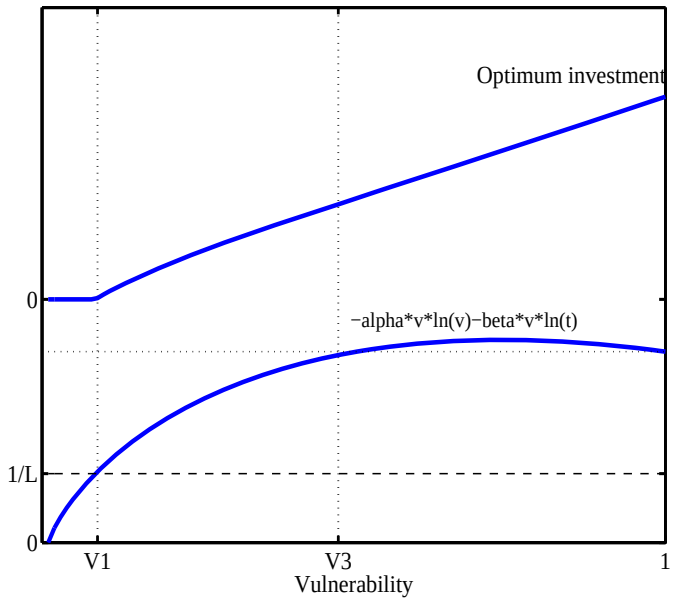

図 3　高脆弱性重視領域の数値例と最適投資曲線 ($\alpha = 0.00001$, $\beta = 0.00001$, $t = 0.5$, $\lambda = 800000$).

第 II 部
ガイドライン

4　概要

複数のモジュール候補から一つのモジュールを採用する際、候補の中には、認証を受けていないモジュール（以下では、非認証候補という）や、特定のセキュリティレベル[*4]で認証を取得したモジュール（レベル i を取得した場合、以下では、Lvi 認証候補という）が存在する。本ガイドラインでは、認証候補あるいは非認証候補を取捨選択しながら IT システムを開発する過程を支援する。

進めるべき手順は、4 段階から成る。

- 手順一：採択枠組みの開発
  1. コスト・ストラクチャの構築
  2. リスク・ストラクチャの構築

[*4] レベル 1 からレベル 4 まであるとする。

- 手順二：候補の分析
    1. 総コストの算出
    2. リスクスコアの算出
    3. リスク許容境界の設定
    4. 文書化の始動
- 手順三：統合
    1. 候補の選択
    2. 選択結果の統合
    3. 理論に照らした検証
    4. 文書の更新
- 手順四：文書化の完了
    1. セキュリティ仕様の作成
    2. 予算案の準備
    3. 要件変更への対応
    4. 改良改革

5　手順一：採択枠組みの開発

5.1　「採択枠組み」とは何か？

代替案比較による取捨選択を体系的にするためには、どのように要素分解し統合できるのか（構造: ストラクチャ）を明らかにしなければならない。このような構造化のメカニズムを、採択の枠組み（以降では、単に「採択枠組み」）と呼ぶ。本ガイドラインにおける採択枠組みは、コストストラクチャとリスクストラクチャから構成される。

1. コストストラクチャの構築：　コスト要素の構造を明らかにする。総コストを算出できるようにするのが主な目的であるが、開発チームが作業に体系的に取り組む体制（心構えも含む）となるのを促すことも、目的に含まれる。
2. リスクストラクチャの構築：　リスク要因の相対的優先度を表現したリスクファクタを設定する。モジュールの採用による合計リスクの指標である「リスクスコア」を算出できるようにするのが主な目的であるが、開発チームが作業に体系的に取り組む体制（心構えも含む）となるのを促すことも、目的に含まれる。

5.2 コストストラクチャの構築

JCMVP のように情報セキュリティ分野で「モジュール」と見なされ得る製品には、システムの構成要素というよりもむしろそれ自体をシステムと見なしてよいようなものも含まれる。そのため、コストがどの程度複雑な構造を持つかは、ケースバイケースである。ここでは、例として、次のコストストラクチャが構築されたとする。

- 1.0 開発費用
  - 1.1 ハードウェアの導入による費用
  - 1.2 ソフトウェアの導入による費用
  - 1.3 サポートサービスの利用による費用
- 2.0 実装費用
  - 2.1 調達に際して発生する費用
  - 2.2 人件費
- 3.0 運用保守
  - 3.1 新人教育による費用
  - 3.2 メインテナンスによる費用

5.3 リスクストラクチャの構築

リスクストラクチャの構築では、優先度で重み付けをしてリスク要因を統合できるようにすることが肝要である。開発過程では、実装ミス[*5]や、あるレベルの要件に未対応などの可能性を、想定しなければならない。それらの不具合の発生確率を、ここではセキュリティ不具合率（または単に不具合率）と呼ぶ。これはモジュール選択における大きなリスク要因であって、脆弱性のあるモジュールをシステムに埋め込んでしまうと、システムの安全性に悪影響が及ぶ。リスクストラクチャの五つのリスク要因（Risk Factor）は、

- 実装対象の技術自体（以降では、単に「対象技術」と記す)[*6]の不具合率と、4 つのセキュリティレベル[*7]それぞれに対応する不具合率を、相対的にどの程度重視するか

[*5] 例えば暗号モジュールの場合には、暗号アルゴリズムの実装ミス。
[*6] 暗号モジュールの場合は、暗号アルゴリズム自体。
[*7] JCMVP の場合には、JIS X 19790 暗号モジュールの 4 つのセキュリティレベル。

というバランスを表現する。この重み付けは、製品の利用環境などに合わせなければならない。

- RiskFactor$= (p_0, p_1, p_2, p_3, p_4)$ と定義する。ここに、p_0 は対象技術の優先度を表し、p_i はレベル i（$1 \leq i \leq 4$）の優先度を表し、$p_0 + p_1 + p_2 + p_3 + p_4 = 100\%$ とする。例えば、$(0, 0, 100\%, 0, 0)$ は、レベル 2 のみ重視することを表す。$(30\%, 60\%, 10\%, 0, 0)$ は、「レベル 1 を最も重視し、レベル 2 もある程度望ましいが、執着しない。かつ、対象技術（自体に不具合がないこと）もある程度重視している。」ということを表す。
- もし、ある高い安全性レベルが必要ならば、より低い安全性レベルの優先度を 0 にする。例えば、レベル 2 の安全性が絶対条件ならば、$p_1 = 0$ に設定する。

システム設計結果は、RiskFactor の設定に著しく依存する可能性がある。JCMVP の場合には、JIS X 19790 暗号モジュールセキュリティレベルと要件要約を参照し、各レベルの要件を徹底的に理解した上で合理的な重み付けをすることが、極めて重要である。また、その際、次のような過不足に留意する：

- 脅威低減効果があまり高くないと思われる場合には、高いレベルへの過剰投資になっていないか注意する。
- 脅威低減効果が充分高いと思われる場合には、高いレベルへの投資不足になっていないか注意する。

6 手順二：候補の分析

6.1 総コストの算出

コストストラクチャに基づいてコスト要素の試算を行い、統合する。ここでは、特段の理論は用いず、単純に総和をとる。例として、レベル 1 で認証を取得したあるモジュールのコストを見積もり、表 1 のような結果が得られたとする。また、レベル 2 で認証を取得したあるモジュールのコストを見積もり、表 2 のような結果が得られたとする。

6.2 リスクスコアの算出

モジュールのセキュリティレベルが 5 種類（レベル 4、レベル 3、レベル 2、レベル 1、非認証）しかないため、各候補のとり得るリスクスコアは、せいぜい 5 つの値にしか分か

表 1　某 Lv1 認証候補のコスト試算と集計（単位：万円）

コスト見積もり	2010 年度	2011 年度	2012 年度以降	合計
1.0 開発費用	**6**	**1**	**1**	**8**
1.1 ハードウェア	2			2
1.2 ソフトウェア	3			3
1.3 サポート	1	1	1	3
2.0 実装費用	**4**	**3**	**2**	**9**
2.1 調達	1	1	1	3
2.2 人件費	3	2	1	6
3.0 運用保守	**1**	**2**	**3**	**6**
3.1 新人教育		1	2	3
3.2 メインテナンス	1	1	1	3
トータル	11	6	6	**23**

表 2　某 Lv2 認証候補のコスト試算と集計（単位：万円）

コスト見積もり	2010 年度	2011 年度	2012 年度以降	合計
1.0 開発費用	**14**	**3**	**3**	**20**
1.1 ハードウェア	6			6
1.2 ソフトウェア	5			5
1.3 サポート	3	3	3	9
2.0 実装費用	**5**	**4**	**2**	**11**
2.1 調達	2	2	1	5
2.2 人件費	3	2	1	6
3.0 運用保守	**2**	**2**	**3**	**7**
3.1 新人教育		1	2	3
3.2 メインテナンス	2	1	1	4
トータル	21	9	8	**38**

れない。ここに、リスクスコアとは、モジュールの採用による合計リスクの指標であって、0 から 100 までの値を取り得る。不具合率の重み付け総和をとったということを踏まえて % を添えて表記するが、何か具体的な確率を意味するわけではない。

リスクスコアの算出は単純である。各リスク要因の優先度と想定される不具合率を掛け算し、その結果をそのリスク要因のリスクスコアとし、そして５項目のリスクスコアの総和を当該候補のリスクスコアとする。

ここで、Lvi 認証候補を考える。JCMVP におけるセキュリティレベルの定義に合わせ、レベルが上がれば要件が強まるとする。簡単のため、「上位レベルの認証を取得できるにもかかわらず下位レベルの認証しか取得していない」ということはないと考え、Lvi 認証候補に関しては、レベル i までのリスク要因の不具合率を 0%、レベル $i+1$ 以上のリスク要因の不具合率を 100% と見なす。

リスクスコアは、不具合率をリスク要因の優先度で重み付けした総和をとって算出する。認証候補の不具合率の値は、認証を信頼して一意的に設定する。非認証候補の不具合率の値には、認証制度の運営機関により公開されたレベル別の実装不具合率を用いる。ただし、公開データが粗く、全てのレベル分けに対応していない場合には、簡易に（しかしやや慎重に）考えて「等分した値以上である」と推定する[*8]。手順三の統合作業で必要となるので、候補に非認証モジュールが含まれていない場合でも、かりに非認証候補があればリスクスコアがいくつになるかを算出しておく[*9]。

- 例として、2009 年のデータ [11] を参照してみる。レベル 1 とレベル 2 の合計不具合率が 50% であり、レベル 3 とレベル 4 の合計不具合率が 75% であった。等分して下限を決め、非認証候補に関しては、レベル 1 およびレベル 2 のリスク要因の不具合率をいずれも 25%+ と設定し、レベル 3 およびレベル 4 のリスク要因の不具合率をいずれも 37.5%+ と設定する。同じ実データによれば、認証を受けようとする暗号モジュールですら、アルゴリズムの実装不具合率は 8% にも及んだ。よって、非認証候補に関しては、暗号アルゴリズムの不具合率を 8%+ と設定する。ここで、“x%+” という表記は、不具合率が x% 以上であることを意味する。

非認証候補に関しては、入手容易なデータで不具合率を設定するならば、下限値しか設定できない。そのため、非認証候補のリスクスコアが x%+ で認証候補のリスクスコアが y の時、$x < y$ であることも十分起こり得る。候補の分析では、x に相当する数値と y に相当する数値を直接的に比較することのないよう、注意しなければならない。手順三では、異なるモジュール選択課題の暫定的な分析結果を、x のみに着目して統合する。

もう一点の注意事項として、候補のリスクスコアが 0% となることは、決して「当該候補を採用すれば構成されるシステムから脆弱性が排除される」ことを意味しているわけではない。リスクスコアの定義から明らかなように、リスクスコアが 0% となることは、当該候補の採用がユーザの IT セキュリティシステム構築方針を最大限に満たすことだけを

[*8] 例えば、レベル 1 とレベル 2 の不具合率の合計が公開されていてその値が 60% である時には、レベル 1 とレベル 2 の不具合率をそれぞれ「30% 以上」と推定する。

[*9] 統合時に必要になってから算出するのではなく、この段階で算出しておく。

意味する。言い方を変えると、リスクスコアが 0% となる候補を採用しても脆弱性は依然として存在するかもしれないが、ユーザに重要視されていない、と考えてよい。

例１： ユーザが設定したリスクファクタを、$(30\%, 60\%, 10\%, 0\%, 0\%)$ とする。この時、表 3–表 7 のようにリスクスコアを算出すればよい。計算した結果、Lv2〜Lv4 のモジュールであればいかなる認証候補でもリスクスコアは 0% になり、Lv1 のモジュールであればいかなる認証候補でもリスクスコアは 10% になり、認証を取得していないモジュールであればいかなる非認証候補でもリスクスコアは 19.9%+ となる（$19.9 > 10$）。なお、表 1 では「某 Lv1 認証候補」と記したが、表 3–表 7 においては「某」を削除して単に「Lv1 認証候補」「Lv2 認証候補」などと記した。これは、同じレベルのモジュールであればいかなる候補でも同じリスクスコアとなるためである。

例２： ユーザが設定したリスクファクタを、$(10\%, 0\%, 80\%, 10\%, 0\%)$ とする。この時、表 8–表 12 のようにリスクスコアを算出すればよい。計算した結果、Lv3〜Lv4 のモジュールであればいかなる認証候補でもリスクスコアは 0% になり、Lv2 のモジュールであればいかなる認証候補でもリスクスコアは 10% になり、Lv1 のモジュールであればいかなる認証候補でもリスクスコアは 90% になり、認証を取得していないモジュールであればいかなる非認証候補でもリスクスコアは 24.55%+ となる（$24.55 < 90$）。

表 3　Lv4 認証候補のリスクスコア（例１）

リスク要因	優先度	不具合率	重み付けしたリスクスコア
アルゴリズム	30%	0%	0%
レベル 1	60%	0%	0%
レベル 2	10%	0%	0%
レベル 3	0%	0%	0%
レベル 4	0%	0%	0%
合計	100%	N/A	**0%**

6.3 リスク許容境界の設定

算出されたそれぞれのリスクスコア（認証候補の各レベルのリスクスコア、そして非認証候補のリスクスコア）に対して、投資できる最大予算（コスト）を見積もる。ただし、

表 4　Lv3 認証候補のリスクスコア（例 1）

リスク要因	優先度	不具合率	重み付けしたリスクスコア
アルゴリズム	30%	0%	0%
レベル 1	60%	0%	0%
レベル 2	10%	0%	0%
レベル 3	0%	0%	0%
レベル 4	0%	100%	0%
合計	100%	N/A	**0%**

表 5　Lv2 認証候補のリスクスコア（例 1）

リスク要因	優先度	不具合率	重み付けしたリスクスコア
アルゴリズム	30%	0%	0%
レベル 1	60%	0%	0%
レベル 2	10%	0%	0%
レベル 3	0%	100%	0%
レベル 4	0%	100%	0%
合計	100%	N/A	**0%**

表 6　Lv1 認証候補のリスクスコア（例 1）

リスク要因	優先度	不具合率	重み付けしたリスクスコア
アルゴリズム	30%	0%	0%
レベル 1	60%	0%	0%
レベル 2	10%	100%	10%
レベル 3	0%	100%	0%
レベル 4	0%	100%	0%
合計	100%	N/A	**10%**

具体的なモジュールを想定してそれに対して投資できる最大予算を考えるのではなく、現在のモジュール選択問題の置かれた状況のもとで、それぞれのリスクスコアに対していくら投資できるかを考える。必要に応じて、同じシステムにおける他のモジュール選択問題との間で情報共有をして差し支えない。結果的に、各候補のコストとかけ離れた多額のコストを許容できるという結論になっても、見積もりを再度行う必要はない。

表 7　非認証候補のリスクスコア（例 1）

リスク要因	優先度	不具合率	重み付けしたリスクスコア
アルゴリズム	30%	8%+	2.4%+
レベル 1	60%	25%+	15%+
レベル 2	10%	25%+	2.5%+
レベル 3	0%	37.5%+	0%
レベル 4	0%	37.5%+	0%
合計	100%	N/A	**19.9%+**

表 8　Lv4 認証候補のリスクスコア（例 2）

リスク要因	優先度	不具合率	重み付けしたリスクスコア
アルゴリズム	10%	0%	0%
レベル 1	0%	0%	0%
レベル 2	80%	0%	0%
レベル 3	10%	0%	0%
レベル 4	0%	0%	0%
合計	100%	N/A	**0%**

表 9　Lv3 認証候補のリスクスコア（例 2）

リスク要因	優先度	不具合率	重み付けしたリスクスコア
アルゴリズム	10%	0%	0%
レベル 1	0%	0%	0%
レベル 2	80%	0%	0%
レベル 3	10%	0%	0%
レベル 4	0%	100%	0%
合計	100%	N/A	**0%**

なお、見積もっているのはリスクへの対策として許容できるコストの値であるが、この作業を「コスト許容境界」ではなく「リスク許容境界」の策定と呼んでいる。これは、単純に予算があるかどうかで考えるのではなく、リスク管理の観点から許容範囲を策定すべきという認識を開発チームに徹底させるためである。

また、総コストとリスクスコアを算出してからリスク許容境界を設定するという順序が

表 10　Lv2 認証候補のリスクスコア（例 2）

リスク要因	優先度	不具合率	重み付けしたリスクスコア
アルゴリズム	10%	0%	0%
レベル 1	0%	0%	0%
レベル 2	80%	0%	0%
レベル 3	10%	100%	10%
レベル 4	0%	100%	0%
合計	100%	N/A	**10%**

表 11　Lv1 認証候補のリスクスコア（例 2）

リスク要因	優先度	不具合率	重み付けしたリスクスコア
アルゴリズム	10%	0%	0%
レベル 1	0%	0%	0%
レベル 2	80%	100%	80%
レベル 3	10%	100%	10%
レベル 4	0%	100%	0%
合計	100%	N/A	**90%**

表 12　非認証候補のリスクスコア（例 2）

リスク要因	優先度	不具合率	重み付けしたリスクスコア
アルゴリズム	10%	8%+	0.8%+
レベル 1	0%	25%+	0%
レベル 2	80%	25%+	20%+
レベル 3	10%	37.5%+	3.75%+
レベル 4	0%	37.5%+	0%
合計	100%	N/A	**24.55%+**

大切である。開発チームの当該ミッションに関する習熟度が上がってから、また、システムに関するできるだけ新しい情報のもとで、設定することが望ましいからである。

ここでは、Lv1 認証候補のリスクスコアに対して許容できる最大コストが例 1 で 28 万円、例 2 で 10 万円となったとする。また、Lv2 認証候補のリスクスコアに対して許容できる最大コストが例 1 で 40 万円、例 2 で 50 万円となったとする。

6.4 文書化の始動

コストストラクチャの構築方法、リスク要因の設定根拠、リスク許容境界を決定する流れに関わる全ての情報と仮定を、明確かつ正確に文書化する。

7 手順三：統合

7.1 候補の選択

分析した候補から何を選択するかを、最適投資理論で直接は支援しない。しかし、必要な利害関係者を集め、経験的ではあっても費用対効果を強く意識した協議によって選択する。総コストがリスク許容境界におさまらない候補は選択しない。また、このモジュールが破られることに起因する被害額の期待値を見積もることができる場合、総コストが被害額の期待値の 37% を越える候補も選択しない。

7.2 選択結果の統合

システム開発においてモジュール選択（採択）が一回あるいは少数回しか行われない場合、選択結果の統合は行わず終了する。

モジュール選択（採択）が多数回行われる場合には、選択結果の統合を行う。あるモジュール選択を行った結果選択された候補の総コストを Z、許容できる最大コストを Z_{max}、その選択に至る分析で算出した非認証候補のリスクスコアを R とする。横軸にリスクスコア、縦軸にコストをとり、点 (R, Z) と点 (R, Z_{max}) をマーキングして、両者を結ぶ線分を記す。この作業を、全てのモジュール選択に関して（同じ平面で）実施する。ただし、選択に際して明らかにリスク回避的[*10]な意志決定をした場合には、マーキングをせずに省略する（すなわち、統合作業に含めない）。

[*10] リスク回避的とは、投資家の投資リスクに対する選好特性の一つである。投資の期待収益に第一の基準を置かず、極力リスクの小さい投資機会を選択する特性を、リスク回避的という。一方、リスクに関わりなく、より期待収益が見込める投資機会を選択する特性のことをリスク中立的という。本研究の最適投資理論では、リスク中立性が仮定されている。

7.3 理論に照らした検証

例えば、例１では某 Lv1 認証候補、例２では某 Lv2 認証候補が選択されたとする。さらにいくつも選択を行い、全てを一つの図（平面）に統合した結果が図 4 のようになった場合（統合例１）、図 5 のようになった場合（統合例２）、そして図 6 のようになった場合（統合例３）、を考える。

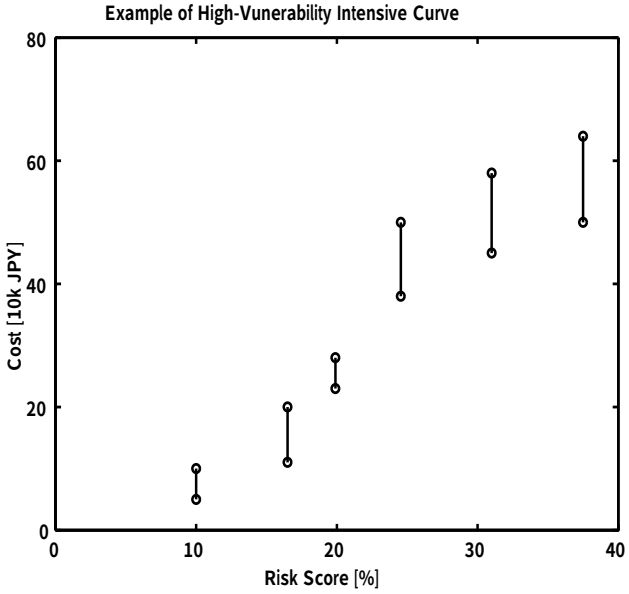

図 4　多数回のモジュール選択の統合例１.

統合例１のように、各線分を通る高脆弱性重視型の最適投資曲線を引くことが可能な場合や、統合例２のように、各線分を通る中庸脆弱性重視型の最適投資曲線を引くことが可能な場合には、とりわけ問題視せず、統合結果を受け入れる。

統合例３のように、それらのいずれでもない場合には、意志決定の基礎となる考え方や基準が首尾一貫していなかった恐れがある。よって、統合結果を受け入れず、現在の文書を基に信頼性の低い作業を洗い出して分析を再度行うなど、PDCA サイクルを回して再度統合する。追加サイクルを経て統合結果が受け入れられたら、終了する。このように「まずは経験的選択を信頼し、要所で体系的に検証して進むべき方向性を定める」という方針は、情報セキュリティ投資の効果を計量する米国連邦政府の取り組み [12] でも使われ、「trust but verify approach」と呼ばれることがある。

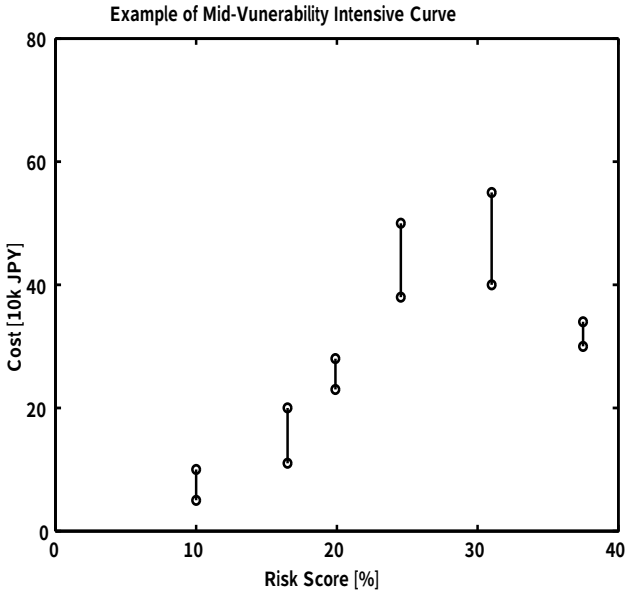

図 5　多数回のモジュール選択の統合例 2 .

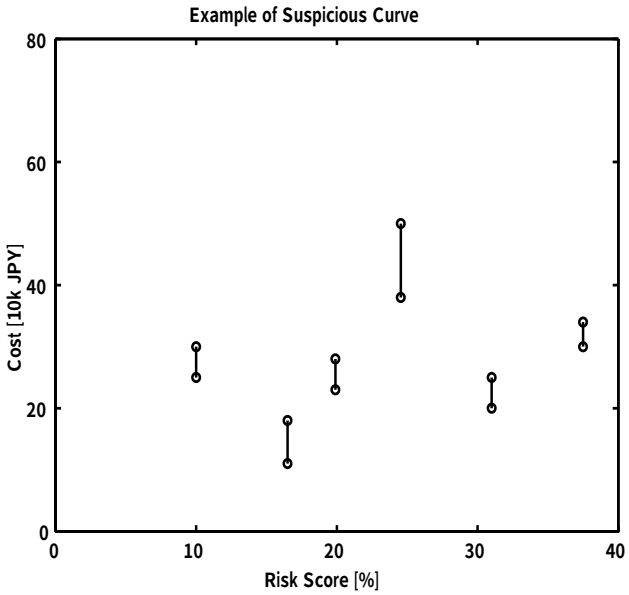

図 6　多数回のモジュール選択の統合例 3 .

7.4　文書の更新

統合作業も含めた追記を行い、文書を更新する。

8　手順四：文書化の完了

8.1　セキュリティ仕様の作成

システム全体に関して首尾一貫したセキュリティ仕様を明確にし、文書に反映させる。

8.2　予算案の準備

システム全体に関してほぼ揃った精度で予算案を準備し、文書に反映させる。

8.3　要件変更への対応

手順三と手順四の実行中に、急遽システムへの要件が大きく変わるなどの事態が生じる恐れがある。その場合、現在の文書を基に、要件変更の要請に対応すべきか棄却すべきかを速やかに決定する。

8.4　改良改革

要件変更の要請に対応する場合には、サイクル増加を有効利用すべく、要請に該当しない事項に関しても改良を試みる。

参考文献

[1] CIO Council Best Practices Committee: “Value Measuring Methodology How-To-Guide”. CIO Council, 2002.

[2] 平成 15 年度情報経済基盤整備・情報システムの政府調達の高度化に関する調査研究「政府調達のための IT 投資評価に関する調査研究」報告書，経済産業省，2004.

[3] 情報処理推進機構: “暗号モジュール試験及び認証制度”. http://www.ipa.go.jp/security/jcmvp/index.html

[4] 井沼 学: “バイオメトリクスセキュリティに関する研究”. 産業技術総合研究所情報セキュリティ研究センター平成 21 年度研究成果報告会, May 2010.

[5] 楊鵬, 松浦幹太: “JCMVP に関するユーザ向けガイドライン試作”. 情報処理学会創立 50 周年記念全国大会論文集, March 2010.

[6] P. Yang and K. Matsuura: “An Introduction of A Users’ Guideline to Japan Cryptographic Module Validation Program”. 5th ACM Symposium on Information, Computer and Communications Security (demo session), April 2010.

[7] L. A. Gordon and M. P. Loeb: “The economics of information security investment”. ACM Trans. on Info. & Sys. Sec., Vol.5, No.4, pp.438–457, 2002.

[8] K. Matsuura: “Productivity Space of Information Security in an Extension of the Gordon-Loeb’s Investment Model”. In Johnson, M. Eric (ed.) *Managing Information Risk and the Economics of Security*, pp.99–119, Springer, 2009.

[9] H. Tanaka, K. Matsuura and O. Sudo: “Vulnerability and Information Security Investment: An Empirical Analysis of e-Local Government in Japan”. The Journal of Accounting and Public Policy, Vol.24, Issue.1, pp.37–59, 2005.

[10] W. Liu, H. Tanaka and K. Matsuura: “Empirical-Analysis Methodology for Information-Security Investment and Its Application to Reliable Survey of Japanese Firms”. 情報処理学会論文誌, Vol.48, No.9, pp.3204–3218, 2007.

[11] Communications Security Establishment, Canada: “CMVP annual report”. 2009.

[12] A. Paller and J. Streufert: “Developing Metrics for Cybersecurity Programs”. Federal Office Systems Exposition 2010 Conference, March 2010.