コンピュータ・セキュリティ
——システム・ライフサイクルとセキュリティ・コントロール——

大阪大学教養部　萬　代　三　郎

はじめに

コンピュータ本体、コンピュータ関連施設、コンピュータの処理機能、コンピュータに蓄積された情報に対するセキュリティ侵害行為は、現代社会の基盤を構成するもろもろのシステムを危やうくする可能性が出てきている。(注1) その現状は米国司法省（表１）、日本の警察庁（表２）のデータが示すところである。(注2) 米国での傾向は、1974年プライバシー法や海外不正取引防止法などの法的規制により、一時的に鎮静化することはあっても、依然として漸増的であることである。わが国の場合、警察の認知件数は少いが、普通、このような事件の発覚は氷山の一角であることと、事件による被害額と社会に与える影響とを考えあわせると、決して気をゆるめうる数字ではないと思われる。

侵害者に対抗して防御者はどのような措置をとるべきか。まず第１に行なわれるのは、事実の確認とシステムの点検である。この場合、セキュリティ・チェックリストとか安全対策基準が重要な役割を果し、問題発見に貢献する。(注3) 第２は諸対策の検討と実施である。わが国の実施状況は十分なものとはいい難いが、リスクに見合った投資が行なわれているものと思われる。(注4) 第３は、セキュリティ担当要員の育成である。わが国では、情報処理技術者試験にシステム監査技術者試験が加えられたことがこの傾向を促進することになろう。

さらに法的対応策として、法制審議会に諮問され、答申をうけていた「刑法等の一部を改正する法律案」は、昭和62年３月第108回国会に提出され成立した。検討を要する領域として、(1) 文書偽造罪、(2) 業務妨害罪、(3) 財産利得罪、(4) コンピュータ情報の不正入手・漏示、(5) コンピュータの無権限使用、の５つが考えられたが、(4)、(5)については、検討すべき点があるとして除外して立案された。(注5)

これまでのべてきたような背景の中で、コンピュータ・ユーザ組織とその管理者がコンピュータ・システムとその保有情報の安全性確保のためにとるべき途は多種多様であり、それが持つ意味は重い。本稿では、ユーザ組織が現実にかかえる仕事を、コンピュータを中心とするシステム上で実現するためにアプリケーション・システムを作成する場面を例にとり、セキュリティ・コントロールをどのようにシステムと関連づけてゆくべきか、そのためにどのようなコントロールが必要とされるか、という課題を中心にして、コンピュータ・セキュリティの一側面を参考資料・文献等に蓄えられた知識ベースから繙いてゆくこととしたい。

表1　米国におけるコンピューター濫用事件

犯罪類型別の発生件数と損害額（年ごと）

	第1類型 物理的破壊			第2類型 知的財産の騙取と窃盗			第3特型 財務的な騙取と窃盗			第4類型 サービスの無権限の使用			全類型		
	事件数	損害額（1,000ドル）	1事件当たりの損害額（1,000ドル）	事件数	損害額（1,000ドル）	1事件当たりの損害額（1,000ドル）	事件数	損害額（1,000ドル）	1事件当たりの損害額（1,000ドル）	事件数	損害額（1,000ドル）	1事件当たりの損害額（1,000ドル）	全事件数	全損害額（1,000ドル）	平均損害額（1,000ドル）
1958	−	−	−	−	−	−	1	<1	<1	−	−	−	1	−	−
1959	−	−	−	−	−	−	1	278	278	−	−	−	1	278	277
1962	2	−	−	−	−	−	−	−	−	−	−	−	2	−	−
1963	1	2,000	2,000	−	−	−	1	81	81	−	−	−	2	2,081	1,040
1964	1	−	−	2	2,500	2,500	3	100	100	−	−	−	6	2,600	1,300
1965	−	−	−	1	−	−	4	126	63	3	−	−	8	126	63
1966	1	<1	<1	−	−	−	2	28	14	−	−	−	3	28	9
1967	2	<1	<1	−	−	−	−	−	−	2	10	10	4	10	5
1968	1	−	−	3	7,203	3,602	6	5,251	1,313	2	−	−	12	12,454	2,075
1969	4	2,000	2,000	8	1,003	334	4	6	2	4	2	2	20	3,011	376
1970	8	3,600	900	6	6,843	1,369	13	8,910	810	11	−	−	38	19,353	967
1971	7	−	−	20	9,844	1,641	24	5,943	540	8	351	175	59	16,137	849
1972	17	11,148	2,230	19	180	30	19	3,090	257	18	107	21	73	14,524	518
1973	10	4	2	26	26,782	2,435	28	206,274	11,460	11	7	1	75	233,066	6,474
1974	7	2,010	1,005	20	2,197	439	34	3,952	158	12	3	3	73	8,162	247
1975	5	115	58	21	91,670	13,096	49	6,513	176	9	14	5	84	98,312	2,006
1976	5	1,110	370	19	49,465	7,060	30	2,026	78	5	−	−	59	52,601	1,461
1977	14	2,252	322	16	17,946	2,991	44	47,501	1,319	13	154	77	87	67,853	1,330
1978	10	2,523	841	13	300	50	17	12,384	826	2	−	−	42	15,207	633
1979	2	−	−	11	−	−	4	200	200	3	−	−	20	200	200
総計	97	26,761	836	185	215,932	3,322	284	302,661	1,462	103	646	32	669	546,001	1,685

法務総合研究所研究部資料36集「コンピュータ犯罪」より

表2　コンピュータ犯罪の認知（把握）件数

区分＼年次	46年	47	48	49	50	51	52	53	54	55	56	57	58	59	60	計
不正データの入力	−件	−	1	1	2	1	−	3 (1)	3	−	6	5 (1)	4	3	9 (2)	43 (4)
データ、プログラムの不正入手	1 (1)	−	−	−	−	−	−	1	−	−	1	1	2 (1)	2	1	9 (2)
コンピュータ破壊	−	−	−	−	1	−	−	−	−	−	−	−	−	−	−	1
コンピュータ不正使用	−	−	−	−	−	−	−	−	−	−	3 (1)	−	−	1	−	4 (1)
プログラムの改ざん・消去	−	−	−	−	−	−	−	−	−	−	−	−	−	1 (1)	−	1 (1)
計	1 (1)	−	1	1	3	1	−	4 (1)	3	−	10 (1)	6 (1)	6 (1)	12 (1)	10 (2)	58 (8)

注：（）内は、警察の認知ではないが、新聞等で報道された事犯であり、内数として示した。

警察庁資料より

1. アプリケーション・システム・セキュリティの目的とシステムの脆弱性

多様なコンピュータ・アプリケーションにとり、不法もしくは望ましくない結果を斎らす事柄がある。それらは、1.データの変更または損壊、2. データの公開、3. データまたはシステム・サービスの利用不能、があげられる。これらの行為はコンピュータ・システムの次の3つのセキュリティ目的に対応するものである。

1. データの保全性（integrity）— データが変更をうけず、データ処理過程が正当で、偶然的または意図的な変更、破壊にさらされない状態が保持される必要がある。
2. データの機密性（confidentiality）—データが機密裡に保有されており、無許可の公開から保護される必要がある。
3. データ処理機能の可用性（availability）— 必要とされるデータ処理サービスが許容できる時間内でえられる必要がある。

（コンピュータ自体と関連施設の保護は、もう1つの重大な問題であるが、本稿のテーマと切り離して取り扱うことにする。）

これらの目的を充足するために多くの効果的なコントロール技法がある。しかし特定のコントロール技法がすべてのセキュリティ目的に有効であるとは考えられず、またアプリケーションによっては取り上げるべきセキュリティ目的がすべてではなく限定されることもありうるので、アプリケーションごとのセキュリティ目的の明確化とセキュリティ技法の有効性の評価がまず検討されるべきであろう。

セキュリティ目的を侵害する行為には、まず偶発的事故、エラー、手抜かりなどの偶然的行為が考えられるが、第2の型の意図的な侵害行為に比べて、発生頻度が高く、一般に予想以上の大きな損失を斎らす。したがって、このような偶然的行為からの有害な結果の潜在的可能性を減少させる安全措置が、第2の型の詐欺や意図的悪用の機会を減らす第1ステップになる。頻繁にエラーを許すシステムは、犯罪行為の肥沃な土壌であり、犯罪行為を覆いつくしてしまう。

意図的な行為に対するコントロール技法は注意深く考えられねばならない。このような行為はコントロールの弱点を利用してくるからである。またシステムの脆弱性を減らすことができないコントロール技法は意図的行為をほとんど抑止できないであろう。アプリケーション・システムは、その内部に犯罪行為を防止するかあるいは少くともそれを発見する内部コントロール機構を保有すること、が必要である。しかし自主的なコントロールが及ばない場合、今回改正された法的制裁が効果的であることは間違いないであろう。

コンピュータ・システムに常駐する情報は、以下にのべるような多種類の危険に対して脆弱である。したがってコントロール技法の選択は、システムのセキュリティ目的だけでなく、アプリケーションのすべての脆弱性に基づいてなされるべきである。ユーザは常識として、これらの脆

弱性に精通しておくことが問題を理解するために必要である。

1. インプット・エラー ── 不正確で、誤解され、一貫性に欠け、不合理なデータが受け入れられる。
2. オープン・システム・アクセス ── 誰がシステムのユーザかがコントロールされない。
3. 貧弱なアクセス承認基準 ── いろいろな情報項目へのアクセス権を誰が保持しているか、を管理要員が知らない。
4. データアクセスの監査不足 ── データアクセスの記録がとられていないので行動の説明ができない。
5. 保護されない情報 ── データが無許可のアクセスから保護されないことがある。
6. ダイアルイン・アクセス ── 無許可の人間がシステムにアクセスできる。
7. プログラム・エラー ── プログラムはエラーを含むものである。エラーは機密扱い情報を変更、公開、さらに破壊させることがある。
8. データの処理ミス ── 処理を間違い、データを更新、消去するかも知れない。
9. ＯＳの欠陥 ── エラーでユーザがＯＳをコントロールできることがある。その間、ユーザはシステムを不能にしたり、監査証跡を消したり、システム情報にアクセスすることができる。
10. プログラムの破壊 ── セキュリティ・コントロールを不能にする。そしてサブプログラムを隠しもつプログラムが発行される。このプログラムは後日の使用を待って、秘密または未確認ファイルへ情報をコピーしたり、データを無許可で修正する。

2. アプリケーション・システム・ライフサイクルとセキュリティ

アプリケーション・システムのライフサイクルは、３つの段階 ── 初期、開発、運用、を踏むものと考える。運用段階にあるものが一定の期間を経過すると、システムは拡張または改訂をうけるのが常である。その時点で改めて前述の３段階が繰りかえされる。アプリケーションは、ハードウエア、コンピュータ施設などの関連要素とできるだけ独立であることが望ましい。各段階の特徴は次のとおりである。

1. 初期段階では、アプリケーションの目的と要件が確認され、目標に対する代替的な接近法が考慮される。実行可能性の検討と潜在的な解の費用便益分析とに基いて、特定のシステム開発の開始を決定する。この際、セキュリティ計画が考慮されないならば、上記計画は重大な歪みをうけ、セキュリティ侵害の遠因を作ることになり、ひいては費用に重大な負担を強いることになる（再設計・開発は高くつき、効率もよくない）。
2. 開発段階は、定義、設計、プログラミング、検査という４つの時期からなる。これらは論

理的には独立であるが、実際上はサブループが作られる。この段階では、どのようなセキュリティ・コントロール技法を含めるべきかの決定が、秩序正しく行なわれなければならない。さらにその結果、必要とされるセキュリティ水準が達成されることを保証することが必要である。

3. 運用段階は、予定したエンド・ユーザにシステムが受け入れられることから始まる。ユーザ組織の使命と責任を満たすためにシステムが存在する。運用中のセキュリティ・コントロールは、日々の運用で強化されねばならず、手控えられてはならない。

上述の説明に特に付言したいのは監査人の参加である。伝統的に監査人はその機能の独立性を維持するために、被監査システムの設計・開発・運用に参画することを拒んできたが、システムの大規模化、複雑化にともない、伝統的な監査方法を修正、拡張、追補をしなければならなくなり、積極的にシステム開発過程に当初から関与するようになった。(注6) とくに内部監査は業務の検討のため組織内で行なわれる独立の評価活動である。これはコントロールの効果を測定し、評価することにより機能するマネジメント・コントロールである。内部監査には、コントロール、標準化、データ処理結果の検査と評価が含まれる。とくにアプリケーションのセキュリティに関する内部監査の1つの側面は、次の事項を保証するコントロールを評価するためのセキュリティ監査活動である。

1. データ処理システムで保有され作成されるデータの正確性と信頼性
2. 情報資産（ハードウェア、データを含む）をすべての重大な意図的脅威、または危険から適切に保護すること
3. データ処理システムのすべての要素の運用上の信頼性と性能の保証

3. 初期段階のセキュリティ計画

この段階でなされる次の決定でセキュリティ問題が無視されてはならない。

1. データ処理システムの役割と、組織がその目的達成のためにデータ処理に依存する度合に関する決定
2. データ処理システムが行う処理を人間が効果的にレビューできる可能性に関する決定
3. システム内に記憶されるデータの性格とその機密度に関する決定

人間の活動とアプリケーションとの間の関係は、1つの活動中のエラーが他の活動によって発見されるように計画されねばならないことである。つまりコンピュータは誤りや偽りのある人間の活動を発見でき、また人間の活動はシステムからの重要なアウトプットが検討できるように計画されねばならない。

新システムの設計に際し、システムの性格の定義づけが不明確であるため、ややもすればセキ

ュリティや保全性（integrity）の問題が無視される傾向がある。このことが適切なセキュリティ目的と一貫性をもたないことを基本的特性とするシステムを作り出す。次の２つの項目はシステムに固有のセキュリティ問題を確認する助けとなる。

3.1　セキュリティの実行可能性

アプリケーションに対する非現実的な計画は、費用効果的でない答をもつセキュリティ問題を作り出す。システム設計者は不適切なセキュリティ・コントロールを作り出すことがある、ことを認識しておく必要がある。このとき次の５つの問題は、セキュリティの実行可能性に関する主要な問題をカバーする。

(1) 原データの正確さ ── 原データは完全ではなく、データがシステムのコントロールに入る前に不正確であることがある。全体的なエラー率が低い場合でさえ、エラーが意図的に導入されるならば、またそれが重要データと関連するならば、わずかのエラーから深刻な損害が起る。人間によるデータの検討が不足しているため、危険が著しく増大するようになる。不正確で不完全なデータを発見し、拒否するためにデータのバリデーションが行なわれ、データ品質の改善に貢献できる。

(2) ユーザの同一性の検証 ── システムのユーザは、その行動が説明できるように適切に確認され、認証をうける。局外者や他の認証をうけていない個人が、システムを利用することを防ぐことができる。

(3) インタフェースの制限 ── 適切なセキュリティが可能であるようにシステムのユーザ・インタフェースは十分に制限されるべきである。現在のＯＳは、悪意で書かれたプログラムがすべてのセキュリティ・コントロールを迂回することを防御できないので、高いリスクを持つアプリケーションは、極めて限られた人間だけが機密扱いプログラムをかけているコンピュータ施設へプログラムを提出することが許されるように計画されるべきである。

(4) 職務の分離（データ処理部門とユーザ部門間）── ２人以上の別々の人物が重要な活動に参加している時に、エラーがよく発見され、詐欺が抑止される。データ処理で支援された部門が、データ処理とそれに関連する人間活動をコントロールしている場合、セキュリティはより容易に保護され、多くの場合、データ収集、支出の承認、資産に対する物理的コントロール等はユーザ部門によって維持される。

(5) 施設のセキュリティ ── 処理施設は安全であるか、セキュリティの一貫性が検討されるべきである。さらにアプリケーション利用上の要件が施設の能力と両立するかどうか、また高いリスクのアプリケーションに対して、その施設で他のアプリケーションを併行して処理することが受け入れられるか。これらの検討課題はセキュリティの基

本にかかわる問題である。

3.2　リスクの初期評価

リスク分析は費用便益分析で考慮される年間期待損害額（セキュリティ侵害の影響の推定値と発生頻度の積）を決定する。コントロールの最大許容費用の推定の第一歩であるリスク分析は、費用効果的なセキュリティ・コントロール・システムの設計の中心であるので、設計の最初の段階で行なわれるべきである。リスクの初期評価は、コントロールが働いているにもかかわらず発生するかも知れない失敗（＝侵害）に起因する損失額を推定する。アプリケーションの計画は、もし可能であれば、このような失敗の潜在的な影響を最小にするように改訂されるべきである。最初のリスク分析から導出された失敗に起因する期待損失額が、代替システム間の費用便益のトレードオフを考える場合に考慮される。

(1) 失敗の影響（セキュリティ侵害の影響）── セキュリティの失敗の影響は、判りやすい簡単な表現で説明されなければならない。例えば、損失金額、個人の不便や困難、人命の損失、社会・経済の混乱度合などである。見過したり、過少評価しないかぎり、潜在的な影響を早急に確認することは可能である。少くとも以下の失敗の影響はシステムで処理される情報の集団ごとに評価されるべきである。

(a) 不正確なデータ ── データとプログラムはエラーで汚染されるが、システムもエラーを含むアウトプットを作り続ける。エラーの潜在的影響を推定する必要がある。回数は少いが重大なエラーの影響と、多数回の小さなエラーの累積効果を考えてみる必要がある。

(b) 偽造データ ── 個人が自己利益のため、何とかして情報を偽造しようとする（例えば収入に結びつく経歴データ）。偽造は人手で発見できないように十分巧妙にされている。

(c) データの公開 ── システム中の機密扱いデータが、一般あるいは特定の個人に利用できるようになる。データの無許可の公開は必ずしも発見されるとは限らない。長期間にわたり発生する影響全体を推定する必要がある（例えば、プライバシーの侵害）。

(d) データとサービスの不能 ── コンピュータと関連施設が使用不能になり、システムを移設させるまで利用できなくなる。

(2) 失敗の頻度（セキュリティ侵害の発生頻度）── 失敗の影響の推定は、失敗の頻度の予測がなければ意味がない。システムの初期計画期間にコントロールの効果を評価することは困難である（コントロールが十分設計されていないので）。しかし粗い推定は可能であろう。 例えば、何年に何回起こるという程度の推定である。

4. 開発段階のセキュリティの形成

開発は、定義、設計、プログラミング、検査の各時期を経過して進行する。各時期ごとに適切なセキュリティ・コントロールが必要である。システムのセキュリティと保全性は、開発努力の全体的な質によって強い影響をうける。開発過程の効果的な管理はセキュリティを改善し、同時に費用を削減するための一方法である。ソフトウエア開発努力が貧弱に組織され、デバッグと品質管理検査に大きく依存している場合、ソフトウエア管理は効果的でなくなり、開発費用が超過するようになる。

4.1 セキュリティ要件の定義

多くのセキュリティ問題は、貧弱なセキュリティ計画やソフトウエアの使命の定義にまでさかのぼって追跡することができる。どんなセキュリティ規制が必要か、またどの程度までソフトウエアによって強化されうるか、の決定は、ソフトウエア開発が進行する前になされ、文書化されねばならない。セキュリティ要件は、システム開発に経験のあるコンピュータ専門家だけでなく、サービスをうける組織の要員によっても定義されるべきである。コントロールの適切度の評価に責任をもつ内部監査人はセキュリティ要件の検討のため援助を求められる。セキュリティ要件の定義は重要なことであるが、この仕事の完成は予想以上に困難である。以下の項目はセキュリティ要件を定義し、詳細なセキュリティ仕様に到達するために使用される。

(1) アプリケーション・システム・インタフェース ── アプリケーション・システムとそれに関連する各ジョブ機能を確認し、両者の間のすべてのインタフェースを確認し定義することが必要である。さらに他のシステムとのすべてのインタフェースをも確認し定義することもまた必要である。これらの作業で、次のような重要なジョブ機能を含めるように留意する。原データの収集、インプットの準備、データの投入、アウトプットの配布、データベース管理、セキュリティ計画とコントロール、内部監査、プログラム保守、記録(archival)とバックアップ・データの保存。

(2) 各インタフェースに関連する責任 ── それぞれのアプリケーション・システム・インタフェースに対して、それを通してアプリケーション・システムと相互的に関係する個人の責任を定めておく必要がある。またインタフェース利用上の制約、利用上発生するエラーの可能性、意図的な悪用、を考慮しておく必要がある。適切なコントロールがなければ、意図的な悪用は予想されるインタフェースの利用法と全く関係のない機能をも実行することができる。インタフェースが正しく利用されることを保証するのに利用できるコントロールを確立すべきである。

(3) 機密扱いの対象と作業 ── インプット・データ、記憶データ、アウトプット・データを含め処理されるデータ対象を論理的に確認する。さらに、データ対象の機密性と資

産価値を決定する。データの保全性、信頼性、可用性の目的から、それぞれのデータ対象または関連するデータ対象に対してセキュリティ要件を定義しておくことが必要である。

(4) エラーの許容度 ── アプリケーションのエラーの許容度は、データに期待される信頼性と妥当性、アプリケーションの目的を考慮して決定される。例えば、電子送金システムはエラーが直接ドル金額で表現されるので、エラーの許容度は低い。他方、航空管制システムのようなリアルタイム・コントロール・システムは、事が人命につながるのでエラーの余裕はない。エラーに対するアプリケーションの許容度と、エラー・レベルを受け入れ可能な許容度の範囲に保持するための要件とは、セキュリティ要件の中で定義されねばならない。

(5) 可用性の要件 ── システムが利用できないために重大な損害が発生した場合、可用性要件の定義が重要になる。この要件は、ユーザ・ニーズに依存して、サービス妨害の最大期間、それの発生頻度、さらにこれら二つの要因の組合せで表現される。

(6) 基本的コントロールへの要求 ── それぞれのアプリケーション・システム・インタフエースまたは各データ対象について、基本的コントロールのそれぞれに対する詳細な要件を定義する必要がある。(注7)

4.2 セキュリティの設計

設計時期は、基本的コントロールがどのようにして実施されるかを決定する時期である。セキュリティの設計は、システム設計に付加すればよいという単純な問題ではなく、全設計段階を通して浸透しなければならない。すべての設計は、必ずしも最も安全なやり方をとる必要はなく、ふさわしい水準のセキュリティを達成すればよい。各設計段階のトレードオフでセキュリティに適切なウェイトが与えられることが必要である。次の(1)～(5)のような基本的設計上の決定はシステムの脆弱性を劇的に減少させる。セキュリティ問題の第一次の対象は、正しく実施できず、保守できず、監査できず、過度に複雑な、設計である。どのようなシステムもそうであるが、高性能という目標は極めて複雑な設計のための弁明になりうるが、簡単化がとくに重要である。すべての機密扱いのアプリケーションに対して、設計期間の最初にリスク分析が行なわれ、これを活用した適切な費用効果的なコントロールがシステム設計に不可欠である。

(1) 不必要なプログラミング ── ユーザは、普通、こっそりとすべてのセキュリティ・コントロールを迂回させたがる。ＯＳはこれを防ごうとするが、そのコントロールは賢明なユーザ・プログラムで迂回または回避される。大抵のユーザはセキュリティを打破するのに必要な知識をもっていないが、ユーザの創意の不足にたよることは貧弱なセキュリティ防御である。

アプリケーションやＯＳのプログラム中の任意の箇所へ挿入されたトラップ、ある指定された事象で発動するトラップは、端末から入れられたデータをプログラムとして実行させる。このコントロールはプログラム開発者の発見努力と、プログラムにトラップを埋めこむ機会を持ちうる第三者の誠意とに依存している。ユーザが不必要なプログラミング能力をもつ危険を最小化する試みが、曝露の機会を減らすことになる。

(2) ユーザインタフェースの制約 ── ユーザ要件を充足させるために、ユーザ・インタフェースが特別に作り替えられることがある。この不必要な柔軟性のためユーザがシステムを受け入れることを難しくしている。またこの柔軟性がセキュリティを傷つけている。ユーザとセキュリティ分析家の両者によって、あまり複雑にならず、ユーザの現実の要求を簡単に充すインタフェースが設計されるべきである。

(3) 共用コンピュータ ── アプリケーションのデータとプログラムは、他のアプリケーションとコンピュータ施設を共有しない場合には保護は容易である。アプリケーションがかかっているコンピュータからすべてのプログラムの開発活動を排除することは特に役に立つ。機密扱いのアプリケーションは自組織のコンピュータで単独に働かせることが最も費用効果的である。

(4) 重要なプログラムの分離── セキュリティにとって重要なプログラムとシステム・データは、容易に監査され保護されるように充分に確認されるべきである。もし可能であれば、セキュリティ・コントロールはアプリケーション・プログラムとほとんど相互作用のないモジュール内に分離されねばならない。これはモジュールを監査し稼動中に無許可の修正からそれらのモジュールを保護することを容易にするためである。

(5) 利用可能なコントロールの活用 ── ＯＳとファシリティ・マネジメンは、次のようないろいろなコントロールを提供できる ── ユーザの同一性検査、システム・ファイルへのアクセス承認、ＯＳ活動の記録、バックアップと修復手続き ── 。アプリケーション・システムは、普通、これらのコントロールを補う必要があるが、利用できるものは可能な限り全面的に活用すべきである。ＯＳのセキュリティ・コントロールは、絶対的に信頼できるものではないので、アプリケーション・システムは重要なデータが変更されたかどうかを確認するために、なんらかのデータ保全性チェックを使用すべきである。しかし一般的にアプリケーションで実施されるコントロールが、ＯＳによって提供されるものに比べて信頼できるという保証はない、ということを知っておく必要がある。

(6) 設計の検討 ── この時点で確認されたセキュリティの設計上の手抜かりと不適切さが、費用のかかるソフトウエアの修正を必要とする。アプリケーションに対する全セキュ

リティ計画は、設計を担当しなかった専門家グループによって検討されるべきである。これはアプリケーションのライフサイクルを通じて容易に変更されえないすべての決定を検討するためにとくに必要である。

4.3　セキュリティのためのプログラミング

ソフトウエア工学の教えに準拠したプログラミングはセキュリティにとって不可欠である。プログラミング・エラーが重大な損失の一般的な原因であるからである。さらにプログラムに意図的に挿入されたトラップは、しばしば詐欺や横領を実行するために利用される。プログラム・エラーやトラップを発見することが難しいので、最も効果的な方法はその発生を防ぐことである。プログラミング期間中にアプリケーション・システムのセキュリティを強化するため以下の項目の実施が推賞される。

(1) 仲間による検討 ── プログラマがプログラムを完成すると１人以上の仲間により検討が加えられる。典型的なやり方は検討者がプログラム中にどのようなセキュリティ・エラーも含まれていないことを確認するだけでなく、次のことを検討することである。

(a) プログラムがすべての設計仕様を満足している。

(b) プログラムが効率的である。

(c) プログラムが容易に保守できる。

少くとも１人のプログラマが完全にそのプログラムを理解できるまでプログラムを検討することが望まれる。また残存するエラーについて最初の執筆プログラマーと等しい責任を負わされる場合、最も効果があるであろう。この方法はプログラマがプログラムに虚偽の修正を加えることを防ぐ最も効果的な方法である。

(2) プログラム・ライブリーによる管理 ── テイブラリーはプログラム・モジュールが開発されるにつれて、それらのすべての版へのアクセスを登録しコントロールする。コントロール機能は人間または自動化手段の何れかにより実行される。ライブラリーはプログラミング期間中、次のようなセキュリティ・コントロールを提供できる。

(a) 承認された人だけにプログラム・モジュールへのアクセスを許す。

(b) プログラム・モジュールへのすべてのアクセス、とくに修正を記録する。

(c) コントロール・データの変更の発見を容易にするためプログラム・モジュールを統合する。

(d) 変更されたレコードを確認するために、前の版のモジュールと現行版との比較をする。

プログラム・モジュールの完成が近づくにつれ、より厳密なコントロールが必要である。

(3) セキュリティ関連プログラムの文書化 ── 一般にプログラムの文書化はすべてのソフトウエア開発期間中に必要とされる。セキュリティ・コントロールがその効果を検討している場合と、ソフトウエア保守後、セキュリティ・コントロールが元のまゝ効果的である場合、プログラムの文書化はセキュリティのために必要である。以下のセキュリティ関連モジュールとプログラムの一部は、はっきりと確認され、完全に文書化されねばならない。

(a) セキュリティ・コントロールを実施するプログラム

(b) 重要な処理（例えば、リアルタイム・コントロール）を行うプログラム

(c) 実行中に重要または機密扱いデータをアクセスするプログラム

(4) プログラム開発ツール ── プログラム言語とプログラミング・ツールの選択は、最終製品の信頼性と正確さを増加させることにより、セキュリティを強化することができる。このようなツールの正しい選択と利用は、プログラミング・エラーがソースプログラムへ入ることを防ぐ手助けになる。

4.4 セキュリティ・ソフトウエアの検査と評価

検査と評価は、システムが信頼することができ、仕様書の条件を満しており、ユーザ要件を充足している、ことを示すことが目的である。また注意深くて十分な検査と評価は、システム設計とプログラムのエラー、手抜かり、その他の欠陥をえぐり出すことにより、システム・セキュリティを改善することができる。しかし注意深くて十分な検査と評価でさえも、システム中のすべての欠陥を見つけ出すという保証はできない。

セキュリティが開発サイクルを通して考慮されなかった場合、また欠陥を避けようとする試みが最初から行なわれなかった場合には、検査と評価によって確認された欠陥の除去によってさえも、セキュリティの改善をあまり期待することはできない。

(1) 検査計画 ── セキュリティの検査計画は、何が検査され、どのような検査方法が用いられ、どのようなツールが必要とされるかを説明すべきである。この計画は、データのインプットと処理中に存在する異常な、普通でない、ありそうもない違法なシステム応答を確認する検査を含むべきである。要件の文書化は、最新版が保有される場合、検査計画を正式にするためのベースを提示する。

(2) 静的評価 ── システムの文書化とプログラムの検査・分析に関連する技法は、意図的トラップや他の無許可の修正を発見する最も効果的な方法である。しかし多くのシステムの複雑さと技法・ツールの限界のため、静的評価を使い、システムを完全に分析することは今のところ実際的ではない。さらにこの方法は生きているシステムを検討しないので、実行環境のエラーは発見されない。

(3) 動的検査 ── この方法はテスト・データを用いて、システムまたはその部分を実行させ、実際の結果と既知の結果を比較する。当然のことであるが、この検査中本番データ・ファイルは検査データとしては使われない。また他のすべての重要ファイルは回避され、検査期間中は保護されるべきである。これはこれらのファイルが破壊されるか、あるいはその内容がうっかり公開されるリスクをなくすためである。どんな事例を検査対象とするか、どれ程の検査事例が必要かは、検討されるシステムと検査チームの経験に依存するであろう。

5. 運用段階のセキュリティ

開発に引続いて運用段階に入る。アプリケーションに組み込まれた第一次コントロールが機能し始める。期待した保護を提供する場合でさえも、この段階を通してセキュリティ対策は緩められるべきではない。セキュリティ・コントロールを実施し、それを補填するために使われる以下のような多数の手続きがある。

5.1 データのコントロール

データの保護、コントロールと保全性は、アプリケーションでデータが通過するすべての段階で求められるものである。

(1) インプット検査 ── これは原データの機械可読形式への正確な変換を保持する過程である。多くの場合、ＣＲＴ端末のような入力装置へ原データを手でキーインすることで行われる。以下はその検査方法である。

(a) 視覚による検査 ── キーインされたデータは目で原データと比較される。

(b) キー検査 ── 原データは２回キーインされ、前にキーインされた変換ずみの原データと自動的に比較される。発見されたエラーは補正され、再キーインされ検査される。

(c) チェック・ディジット ── 原データ項目は、特別のアルゴリズム、項目中の他のすべての文字から導出された特別のチェック・ディジットを含んでいる。原データがキーインされる時、チェック・ディジットを計算し、原データの一部としてキーインされたチェック・ディジットと結果が比較される。

(d) コントロール・トータル ── 原データは小グループまたは同じ型の仕事別に分割される。個々の合計はグループ中の重要な項目を監視している。合計はグループと一緒にキーインされインプットの原レコードを形成する。このキーイン過程の後で、キーインされた項目は再合計され、結果はこれらの合計を含んでいる原レコードと比較される。不一致はキーイン操作の誤りか、最初のコントロール・トータルの計

算ミスを示す。

(2) データ保管管理 ── これはデータ記憶媒体の保守と、それへのアクセスを取扱う。記憶媒体へおかれたデータは、偶然的・意図的修正、破壊、無許可の公開という被害を被る。重要な親ファイルか、バックアップと修復データかが変更または破壊された場合、当該組織の使命は、混乱させられる。次のようなコントロール技法がある。

(a) 保管領域へのアクセス制御 ── 未使用のデータ記憶媒体は、物理的アクセスがコントロール（出入り管理）されている領域におかれ保持されるべきである。アクセス権をもつ唯一の人が管理権をもつ人である。

(b) データ承認 ── 媒体は使用の承認を得ている人だけに開放されるべきである。

(c) 機密データのバックアップ ── データの破壊と変更に対する最も確かな形の保護は、他の場所（オフサイト）での追加コピーの保持である。バックアップ・データの作成とオフサイトへの移動は定められた手続きによって取扱われる。

(d) データの暗号化 ── 本号の別稿と、本稿の文献を参照されたい。

(3) アウトプット配布のコントロール ── 機密データを処理するプログラムからのアウトプットの配布のコントロールは、データがコンピュータを離れた後、データに対する継続的保護を保証する助けになる。コンピュータや記憶媒体から盗むのに比べて・ハードコピーの形で情報を盗むのはより容易である。それ故にデータはアウトプット段階でとくに脆弱である。次に方法によって何らかの保護が与えられる。

(a) 受領書への署名 ── コピー数を限定し、番号をつけ、各コピーに署名を求めることは、意図したところに配布される唯一の絶対的方法である。

(b) メールによる配布 ── メール配送システムを通してアウトプットを配布することは、アウトプットが意図したところへ届くことを保証するもう一つの方法である。

(c) ラベル ── 機密扱いのアウトプットの各頁に自動的に適切な機密分類のマークをつけることで、コピーを受け取ることを承認されていない人への警告になる。

（電子メール配布システムを採用する時にはソフトウエアによるアクセス制御をうけけることは当然である。）

5.2 要員の管理

運用段階に関与する要員に対するセキュリティ・コントロールは、採用段階から離任段階に至るまで隙き間なく行なわれるべきであるが、主なものを次に列挙する。

(a) 文書による業務割当て ── 労働慣習の相違によって受け入れ難いことからも知れないが、個人に割当てられたすべての仕事を書面で列挙し、指定する。仕事の責任と権限が明確にされよう。

(b) 職務の分離 ── この先例はオートメーション時代に由来するが、今ではきわめて効果的なリスク減少技法である。以下にリストされた各職能の責任は可能な範囲まで個別の人に割当てられる。分離が不可能な場合でも、プログラム設計、プログラミング、プログラム検査、プログラム保守のような密接に関連する職能は分離するように努力すべきである。また職能間で責任の重り合いがないように職能を明確に定義することが重要である。 ── 職能例のリスト ── データ収集、データ準備、データ投入、データベース管理、特定目的のデータ利用承認、データ処理作業、プログラミング、プログラム設計、プログラム検査、プログラム保守、データ処理施設と設備のセキュリティ、データ通信、アウトプットの配布、内部監査、セキュリティ計画と調整。

(c) 協力活動 ── 2人以上の人間が、同じ仕事の別々の部分を指定された時間的制約内に仕上げる。不正や他の無許可活動をうけやすい仕事に対して要員間の共謀が必要な場合、このような活動は成功確率を減少させる。

(d) 休暇の強制 ── 緊張と不安を取除くことによって、要員の健康に貢献するであろう。緊張して不安な要員は仕事中に事故をおこし易く、エラーや見過しをしがちである。これだけでなく要員に依存する不正計画を1人で設定する確率が小さくなり、横領の抑止になる。

(e) 仕事の配置転換 ── 要員を一つの仕事から他の仕事へランダムな期間をおいて移動させることは、詐欺に対して休暇の強制策と同じ抑止効果をもつ。

(f) 会計上の制約 ── プロジェクトの勘定につけられた資金量と、その勘定で使用できるコンピュータ時間に上限をおくことにより、私的使用また不正使用のためのコンピュータ時間が抑制され、横領の企てが抑止される。

(g) 物理的アクセスの制約 ── コンピュータ施設への出入りは障害物（例えば、電子ロック）によって注意深く管理されるべきである。また単純な理由（例えば、仕事の依頼、伝達）によってその場に居合わせた人が、その区域を動き回ることができなくなる。とくにプログラマはコンピュータ室への入室権限をもつと思われがちであるが、それは厳禁されるべきである。

(h) データ承認の制約 ── データへのアクセスは、地位とかレベルという前例に従うよりも、むしろそのアクセスの個々の必要性を熟慮して与えられるべきである。

(i) 監督者による監視 ── 要員がセキュリティ許可を持っているということは、その要員を熟知し、その行動を観察できる監督者による継続的観察の必要性を排除するものではない。

5.3 要員の離任時の手続き

要員の離任に際し、退職の条件または形式の如何をとわずに監視が行なわれるべきである。

(a) すべての承認の取消し ── 離任する要員に与えられてきたすべての承認（例えば、データ、ファイル・アクセス承認）は取消されねばならない。またすべての承認リストから彼の名前は除去されるべきである。

(b) 鍵、バッヂ等の返却 ── 構内、装置、情報へのアクセスを得るために使うすべての鍵、バッヂ、その他の装置、マニュアル類は離任要員から返却させねばならない。

(c) 錠の変更 ── 離任要員が知っているすべての錠の組み合せはすぐに変更されねばならない。定期的変更だけでは不十分であろう。

(d) パスワードの取消し ── 離任要員のパスワードが取消されるだけでなく、リストから除去されねばならない。またこのパワードは再利用できないようにする。

5.4 セキュリティ訓練

この訓練とは、要員に割当てられたセキュリティ業務の達成を教育するために必要な訓練と、セキュリティの責任を教える訓練をいう。

(a) タスク訓練 ── セキュリティ責任上のタスク訓練にはすべての技量とセキュリティ関連義務を適切に遂行するためにとられるべき手段の概略とが含まれる（例えば、消火器具の操作、電源事故後のコンピュータのリスタート等）。

(b) セキュリティ意識訓練 ── すべての要員に特定のセキュリティ責任が割当てられていることを教えこむ訓練で継続的に行なわれ、技術が進むにつれて新しい課題を取扱い、セキュリティの必要性の定期的検討を行う。

5.5 ソフトウエアの修正とハードウエア保守

この問題はすべてのシステムの有用性を継続させるために必要であるが、これらの活動の結果が、エラー、事故、意図的活動をひきおこすリスクを増大させることを認識すべきである。

(1) ソフトウエア修正 ── ソフトウエアの修正は業務上不可避的であるが、前述の（4.3）セキュリティのためのプログラミングがこの場合にも適用される。さらに次のような業務を監視することにより変更中のシステムに追加的保護を与える。

(a) プログラムの再調整 ── 挿入される新しいプログラムは、入れかえられるプログラム、変更の仕様と比較されねばならない。

(b) 元のプログラムの有用性 ── 取替えられたプログラムは、新しいプログラムの試用中に発生するトラブルに備えて妥当な期間（4～6週）、ライブラリー中ですぐに利用できる状況にしておくべきである。

(c) 文書化 ── すべての変更は、その結果、変更をした理由、データとアウトプットへの特別の効果、とくに装置構成から生じた変更を含めて、十分に文書化されるべき

である。

(d) テストデータの利用 ── すべての変更は、予測し期待する結果が得られたかどうかについてテストをうけるべきである。しかし本番データをこのために使用してはならない。

(e) テスト ── 修正されたソフトウエアは最初のソフトウエアに対するのと同じテストをうけるべきである。

(2) ハードウエア保守 ── ハードウエアが保守されている間に、オンラインでつながっているデータは不注意で消去されたり、機密扱いデータが保守員（外部の人間）の目にさらされる。

(a) 本番データの扱い ── すべての本番データは、保守期間中退避させるか、止むをえない場合は、保守と切り離して稼動させる。

(b) 記憶装置のクリア ── データとソフトウエアの残滓を除去するために記憶装置にはダミーが書きこまれる。

(c) 定期保守 ── 貧弱な保守から発生するエラーを防ぐために規則正しく計画され達成されるべきである。

5.5 コンティンジェンシー・プランニング(注8)

この計画は、システムの重要機能の継続を保証するために必要とされる事前計画と配置とからなる。計画は運用の全面的あるいは部分的休止、データベースや物理的施設の破壊等のすべての事象をカバーすべきである。またこの計画は装置の可用性を保持し、要員の効率的配置を保証する手続きを含めるべきである。アプリケーションの二つの大きな脆弱性は、処理の可用性の喪失とデータの重大な損失とであるが、この問題に対して、支援されている使命の重大さとシステム・アウトプットの要件の緊急性とが考慮されねばならない。以下の計画がアプリケーションの緊急事態と関連する。

(1) 重要機能の確認 ── 当該組織にとってのアプリケーション中の重要機能が確認されるべきである。どの機能が常に重要であるか、他の機能がどれ程の時間的許容性をもつているかを分析すべきである。

(2) 代替サイトでの運用 ── 同じまたは互換しうる装置を使い、緊急期間中、重要機能に時間を配分できる他組織のサイトを用意すべきである（コールドシエル、ホットシェル、協定によるサイト等）

(3) 限定された処理の手動化 ── すべてではないが、重要な機能の中には自動データ処理化以前の手続きに短期間戻す必要があるものも出てこよう。

(4) データ、プログラムのバックアップ ── データとプログラムを損失から保護するもっ

とも容易な方法は、アクセスが容易で安全な場所へ追加コピーを保管することである。データとプログラムの選択、保管場所、保有期間の決定は、データの脆弱性、データベースの大きさ、アプリケーションの重要度に依存する。

(5) データの修復 ― 修正され損失をうけたデータは世代変更により修復することは可能である。このための要員、ソフトウエアの準備、禁止事項の部分的解除等が必要になる。

(6) 施設の復元作業 ― 施設が破壊または損傷をうけた場合、施設の復元か移転の計画がなければならない。このような計画は、本来設備関係者の責任であるが、アプリケーション側の管理者は、仕事の負荷計画、スペース要件、設備要求、移転作業等の領域で援助を求められることがある。これらの計画活動に必要な情報を事前に準備し、定期的に更新しておくことが、計画の完成と適時性を保証する。

むすび

表題の下で関連するセキュリティ上の諸項目を概観してきたが、その中から読みとられるように、コンピュータ・セキュリティは、若い未成熟の技術・管理分野である。また個別技術の集合であり、未だ十分に構造づけられていない知識体系でもある。さらに技術進歩に対応して（例えば、ＯＡ技術の開発とオフイスへの導入、マイクロ・メインフレーム・リンクスに伴う諸問題等)、より一層の充実が期待される分野でもある。

一般的にいえば、100％のセキュリティ保持は無限の投資を伴わないかぎり不可能である。すべてのシステム資源を対象にして、すべての脅威がつねに起るものとは考えられない。したがってわれわれが保護すべき資源は、それぞれのリスクに見合った投資をすることにより、相応した脅威からシステム資源を保護することは可能である筈である。

他面からみると、最近のシステムでは利益を獲得するための投資とは異なり、保守とか監査、セキュリティといった、どちらかといえば後向きの投資、つまり損失を最小にする投資が重要視されている傾向がある。これは複雑で大規模化したシステムの一つの特徴であろうが、受動的にこのような投資を行うのではなく、損失の機会を認識し積極的に投資を行うことが、投資効果を高める方法でもあり、セキュリティ技術を生かす方法でもあると考えられる。

(注１) セキュリティ関係用語は、ＪＩＳ情報処理用語（Ｃ6230)に十分含まれていないので、ここでは米国商務省基準部の作成したFIPS PUB 39 *Glossary for Computer System Security*, 1976 を参照することにする。（他に米国会計検査院のグループが書いたものがある。H.J.Podell *et al* ; "A Computer Security Glossary for the Advanced Practioner", *Computer Security Journal*, Vol. 4 No.1. pp.69-88）

(注２) 英国においても、コンピュータ犯罪の発生年度別件数は、59年 1件、60年代 31件、'70年～'75年 15件、'76年～'79年 73件、'80年～'83年 35件、'83年～'84年 34件、'85年～'86年 32件という英国民間会社の調査がある（日経コンピュータ、'87年6月8日号 pp.141 149）。

(注３)

1. 通産省：電子計算機システム安全対策基準（改訂版)、昭和59年８月。
2. 通産省：システム監査基準、昭和60年１月。
3. 総務庁：電子計算処理に係るデータの保護について、昭和51年１月29日。（旧行政管理庁行政管理局）
4. 自治省：(1) 地方公共団体が電子計算機処理を委託する場合における保護について、昭和51年２月21日。
   (2) 地方公共団体におけるコンピュータ・セキュリティ対策チェックリスト、昭和61年３月。
5. 郵政省：情報通信ネットワーク安全・信頼性基準、昭和62年２月。
6. 警察庁：情報システム安全対策指針（中間報告)、昭和61年１月。
7. (財)金融情報システムセンタ：金融機関等コンピュータシステムの安全対策基準、昭和62年12月。

(注４) 日本情報処理開発協会：コンピュータ・セキュリティ実態調査報告書、昭和61年３月。

(注５) ジュリスト、1987年５月15日号 (No.855), pp.4 ～20 。

(注６) 米国内部監査人協会（I.I.A.）：*SAC Study Report*, 3Vols, 1977 。

(注７) 紙数と時間上の制約のため、基本的コントロールについての詳細な論議を割愛せざるを得ない。

(注８) 本項に関しては、大阪大学教養部研究集録（人文・社会)、第36集、昭和62年12月刊（予定）掲載稿で詳細に論議する。

参考文献（和）

1. 鵜沢昌和：コンピュータ・犯罪とエラー、日本経済新聞社、昭和53年８月。
2. 岡本行二：コンピュータのための機密保護と安全管理、オーム社、昭和49年６月。
3. AFIPS編、横山・萬代監訳：セキュリティ、秀潤社、昭和52年２月。
4. 岡本行二：コンピュータ安全保護マニュアル、オーム社、昭和55年10月。
5. 上園忠弘：コンピュータ・セキュリティ、近代科学社、昭和56年10月。
6. D.B.Parker著、日本情報処理開発協会監訳：コンピュータ・セキュリティ、企画センター、昭和57年５月。
7. L.I.Krauss他著、伊藤訳：コンピュータ不正とその対策、清文社、昭和57年８月。
8. 一松　信監修：データ保護と暗号化の研究、日本経済新聞社、昭和57年８月。
9. D.W.Davis他著、上園忠弘監訳：ネットワーク・セキュリティ、日経マグロヒル社、昭和60年12月。
10. H.J.Highland著、上園忠弘訳：パソコン・セキュリティ、啓学出版、昭和61年４月。
11. B.J.Wilkins著、渡部・宇佐美訳：システム監査人のための情報セキュリティ入門、日刊工業新聞社、昭和61年４月。
12. 土居・小山編：コンピュータ・セキュリティ、共立出版、昭和61年10月。
13. M.M.Wofsey編、萬代・内藤・久保訳：コンピュータ・セキュリティの進展、晃洋書房、昭和61年12月。
14. J.Lobel著、与那嶺訳：コンピュータ・セキュリティ、マグロヒル・ブック、昭和62年３月。
15. 青山監査法人編：コンティンジェンシー・プランニング、日経マグロヒル社、昭和60年11月。
16. 池野・小山：現代暗号理論：電子通信学会、昭和61年９月。
17. C.H.マイア、S.M.マティア著、細具他訳：暗号、自然社、昭和61年２月。

参考文献（欧）

1. J.Martin : *Security, Accuracy, and Privacy in Computer System*, Prentice-Hall, 1973.
2. L.J.Hoffman : *Security and Privacy in Computer System*, Melville, 1973.
3. L.J.Hoffman : *Modern Methods for Computer Security and Privacy*, Prentice-Holl, 1977.
4. R.A.Demillo *et al* (ed.) : *Foundations of Secure Computation*, Academic Press, 1978.

5. C.T.Dinardo(ed.): *Computers and Security*, AFIPS Press, 1978 (National Computer Conference から抜粋論文集).
6. D.K.Hsiao *et al* : *Computer Security*, Academic Press, 1979.
7. *Proc. of IEEE Symposium on Security and Privacy*, IEEE, 1980年以降各年。
8. L.I.Krauss : *SAFE*, AMACOM, 1980.
9. E B.Fernandez *et al* : *Database Security and Integrity*, Addison-Wesley, 1981.
10. INFOTECH: *Computer Systems Security*, Pergamon Press, 1981.
11. R。Turn (ed.) : *Advances in Computer System Security*, Artech House, 1981.
12. R.E.R.Denning : *Cryptography and Data Security*, Addison-Wesley, 1982.
13. J。A.Sehweitzer : *Managing Information Security*, Butterworth, 1982.
14. *Proc. of IFIP International Conference on Computer Security*, North-Holland, 1983年以降各年。
15. A.R.D.Norman : *Computer Insecurity*, Chapman & Hill, 1983.
16. R.P.Fisher : *Information System Security*, Prentice-Hall, 1984.
17. J.A。Schweitzer : *Computer Crime and Business Informatin*, Elsevier, 1986.
18. American Bankers Association : *Contingency Planning Manual*, 1985.
19. CHANTICO Series : *Disaster Recovery*, North-Holland, 1985.
20. 米国商務省基準部Federal Information Processing Standards Publication (FIPS PUB) No.31 *Guideline for Automatic Data Processing Security and Risk Management.*
    No. 39 *Glossary for Computer System Security.*
    No. 41 *Computer System Guidelines for Implementing the Privacy Act of 1974.*
    No. 46 *Data Encryption Standard.*
    No. 65 *Guideline for Automatic Data Processing Risk Analysis.*
    No. 73 *Guideline for Security of Computer Applications.*
    No. 87 *Guidelines for ADP Contingency Planning.*
21. A.A.Garcia : *Computer Security*, J.Wiley, 1987.