暗号ユーザーが暗号アルゴリズムの安全性評価結果をどう活用するか

宇根正志（うねまさし）／黒川貴司（くろかわたかし）／鈴木雅貴（すずきまさたか）／田中秀磨（たなかひでま）

要　旨

金融取引では、データの守秘や一貫性を確保する手段として、RSA やトリプル DES 等の暗号アルゴリズムが利用されている。暗号アルゴリズムの安全性は暗号解読技術や計算機性能の向上とともに低下することから、金融機関は、学界における安全性評価結果を活用して暗号アルゴリズムの安全性の動向を把握しておく必要がある。その際、多岐にわたる評価結果のうち注目すべきものはどれか、また、そうした評価結果をどのようにリスク管理に活用するかの 2 点が課題となる。

本稿では、まず、インターネット・バンキングにおける暗号アルゴリズムの利用事例を基に、どのような評価結果に注目すべきかを説明するとともに、評価結果のリスク管理上の取扱いが暗号アルゴリズムの安全性評価の状態に応じて変化することを示し、「学術的に解読された」場合には各システムでの対応に関する検討の開始が必要であることを説明する。また、評価結果をリスク管理に適用する際、暗号アルゴリズムの安全性を「攻撃実行に必要な計算量」でなく「資金と時間」で表現することが有用であることを説明し、そうした表現方法の一例を示す。

キーワード：暗号アルゴリズム、安全性評価、危殆化、計算量、リスク管理

本稿の作成に当たっては、東京電機大学の佐々木良一教授から有益なコメントを頂いた。ここに記して感謝したい。ただし、本稿に示されている意見は、筆者たち個人に属し、日本銀行あるいは独立行政法人情報通信研究機構の公式見解を示すものではない。また、ありうべき誤りはすべて筆者たち個人に属する。

宇根正志　日本銀行金融研究所企画役（E-mail: masashi.une@boj.or.jp）
黒川貴司　独立行政法人情報通信研究機構情報通信セキュリティ研究センター
（E-mail: blackriver@nict.go.jp）
鈴木雅貴　日本銀行金融研究所（E-mail: masataka.suzuki@boj.or.jp）
田中秀磨　独立行政法人情報通信研究機構情報通信セキュリティ研究センター
（E-mail: hidema@nict.go.jp）

1. はじめに

金融取引に関するデータの守秘や認証、取引相手の本人確認等を実行する手段として、各種の暗号アルゴリズムが利用されている。例えば、インターネット・バンキングの場合、金融機関のサーバーと顧客のクライアントPCとの間で暗号通信プロトコルSSL（Secure Sockets Layer）が利用されるケースが多い。このSSLにおいては、通信相手の認証にRSA等が利用されているほか、データの暗号化にAESやトリプルDESが利用されている（宇根・神田［2006］、神田・山岸［2009］）。

金融機関が暗号アルゴリズムを利用する際には、その安全性のレベルが当該アプリケーションにおいて要求されるレベルを満足しているか否かを適宜再評価する必要がある。暗号アルゴリズムの安全性は、時間の経過に伴う計算機性能や暗号解読技術の向上によって低下するが、その進行度合いを見極めるうえで、学界における評価結果を参考にすることが有用である。金融分野において現在広く利用されている暗号アルゴリズムは、「解読にかかる計算量が莫大であり、現実には実行困難である」という計算量的な安全性に基づいており、それらの安全性評価結果は「攻撃実行に必要な計算量」によって示される。かつて金融分野の代表的な共通鍵暗号であったDES（鍵のサイズは56ビット）の場合、候補となる鍵を1つ1つ試して真の鍵を探索する攻撃（全数探索法）に対する安全性は、2の56乗（約7.2京）回程度のDESの暗号化処理に相当する計算量によって示される。こうした評価結果は、特定のアプリケーションを想定しているわけではなく、評価結果の計算量をそのまま個々のアプリケーションに当てはめることは適当でない。暗号アルゴリズムを利用して情報システムのセキュリティを確保しようするユーザー（以下、暗号ユーザーと呼ぶ）は、個々のアプリケーションに応じて、攻撃実行に必要な計算量をリスク管理上適切に解釈したうえで対策を検討することが求められる。

暗号ユーザーがこうした検討を行う場合、まず、当該アプリケーションにおいて注目すべき攻撃のタイプを取捨選択することが重要である。例えば、平文のメッセージ形式が標準規格等によって規定され、攻撃者が平文を自由に選択することができないケースにおいては、攻撃者が自ら選択可能な平文や暗号文は制限されることから、そうした制限のなかで実行可能な攻撃についての安全性評価結果を重視することが有用であると考えられる。

また、注目すべき攻撃のタイプは当該暗号アルゴリズムの安全性評価の状態にも依存する。例えば、アルゴリズムに何ら欠陥が指摘されていない暗号アルゴリズムの場合、暗号ユーザーにとっての現実的な脅威は鍵の全数探索による解読であり、計算機性能が今後どの程度のスピードで向上していくかが主な留意点となる。一方、当該暗号アルゴリズムにおいてアルゴリズム上の欠陥が多少なりとも報告された場合には、その欠陥をベースとして攻撃方法の高度化に関する研究が急速に進展する可能性が高まることから、計算機性能の向上に加え、攻撃方法の向上にも留意する必要が出てくる。

このように攻撃のタイプおよび注目すべき安全性評価結果を絞り込んだ後、暗号ユーザーは、攻撃実行に必要な計算量をリスク管理に適用しやすい形態に翻訳して活用することが求められる。例えば、全数探索法に対する DES の安全性は約 7.2 京回の DES 暗号化処理に相当する計算量によって示されるが、実際に攻撃にかかる時間は攻撃者が調達可能な計算機の性能等に依存し、計算機の高性能化に伴って攻撃の時間は短縮される。こうしたことから、暗号ユーザーは、当該アプリケーションにおいて想定する攻撃者の資金等を基に調達可能な計算機の能力を推定し、攻撃に必要な時間を評価したうえで、攻撃が実際に成功する可能性やその結果発生しうる損害額等を考慮しながら対応方針を検討することが求められる。しかし、攻撃実行に必要な計算量を資金や時間に翻訳する方法が学界で議論されることは稀である。

本稿では、金融機関等の暗号ユーザーが学界における安全性評価結果を活用する際に留意すべきポイントを説明する。まず、2 節において、インターネット・バンキングにおける暗号アルゴリズムの利用形態を考慮し、どのようなタイプの攻撃に特に留意する必要があるか、また、暗号アルゴリズムの安全性低下がどのような影響を及ぼす可能性があるかを説明する。次に、3 節において、注目すべき攻撃や安全性評価結果が当該暗号アルゴリズムの安全性の状態によって変化することを DES の事例に基づいて説明するとともに、各状態におけるリスク管理上の対応のあり方を説明する。また、4 節においては、学術的な評価結果として示される「攻撃実行に必要な計算量」を「攻撃実行に必要な資金や時間」に翻訳する方法を示す。最後に 5 節において、今後の課題等を示しつつ本稿を締め括る。

2. どのような攻撃のタイプに注目すべきか

(1) 暗号アルゴリズムの利用事例：インターネット・バンキングのケース

暗号アルゴリズムの利用を検討する際には、これまで知られているすべての攻撃方法とその安全性評価の考え方を理解することが望ましい。しかし、これまでに提案されている攻撃方法は多岐にわたっているほか、利用環境によっても異なってくることから、暗号アルゴリズムの専門家でない暗号ユーザーがすべてを理解することは実際には容易でない。暗号ユーザーにおいては、当該アプリケーションにおいて重視すべき攻撃方法に焦点を当てて検討することが求められる。本稿では、暗号ユーザーとして主に金融機関を想定し、暗号アルゴリズムを利用する代表的なアプリケーションとしてインターネット・バンキングの例を取り上げて検討する。

まず、インターネット・バンキングのサービスを構成するエンティティとして、口座振込等の金融取引サービスを提供する銀行に対応する「サービス提供者」と、同サービスを利用する「利用者」の二者を想定する。そのうえで、利用者が同銀行における自分の銀行口座から第三者の銀行口座に一定の資金を振り込むという口座振込のサービスを行う場合に絞って検討を進める。

こうした場合に想定される代表的な攻撃として、「サービス提供者と利用者の間の通信に攻撃者が割り込んで不正な行為を行う」という中間侵入攻撃（man-in-the-middle attack）がよく知られている。中間侵入攻撃の細部については個々のインターネット・バンキングのサービスに依存することとなるが、ここでは次のようなタイプの中間侵入攻撃を想定する。

- **口座振込実行時に、利用者に対してはサービス提供者になりすますと同時に、サービス提供者に対しては利用者になりすますことによって、口座振込の振込先や振込額を改ざんして当該金額を不正に得る。**

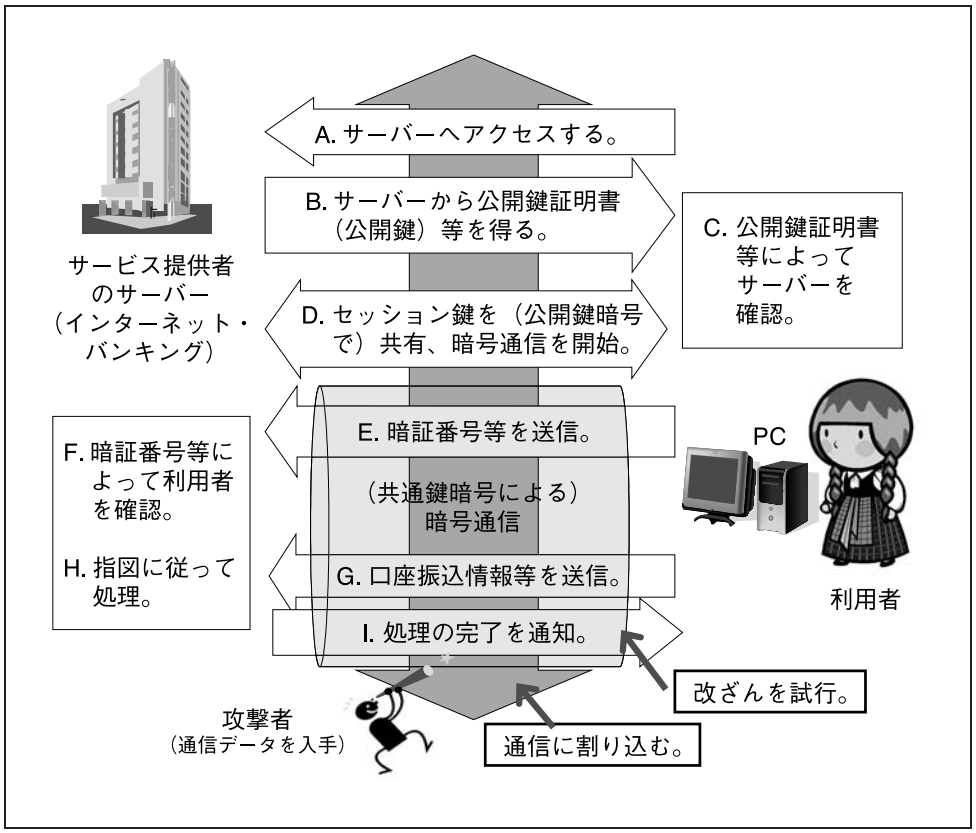
上記の攻撃を防ぐためには少なくとも以下の要件を満足させる必要がある。

- **利用者とサービス提供者は互いに通信相手を確認できる（相互認証）。**
- **利用者とサービス提供者はともに通信データの改ざんを検知できる（データー貫性確認）。**
- **暗証番号等のように、取引実行時に秘密にする必要があるデータについては第三者に対して秘匿したまま通信できる（データの秘匿）。**

インターネットにおいては上記の要件を満足させるために暗号アルゴリズムが利用されており、多くの場合には暗号通信プロトコル SSL が使われている。SSL においては、相互認証に公開鍵暗号が利用され、データの一貫性確認や秘匿に共通鍵暗号が利用される。また、共通鍵暗号を使用する際には暗号化用のセッション鍵を通信当事者間で共有する必要があり、そうした鍵共有には公開鍵暗号が用いられる。また、公開鍵暗号の構成要素や共通鍵暗号の鍵生成方法として、ハッシュ関数が利用されるケースが多い。これらの暗号アルゴリズムを利用しながら、以下のような流れで暗号通信による取引が実行される場合を想定する（図表 1 参照）。

A. 利用者が PC からサービス提供者のサーバーにアクセスする。
B. 当該サーバーは利用者の PC に対して公開鍵証明書等を送信する。
C. 利用者の PC のブラウザーは、当該サーバーの確認を公開鍵証明書等によって実施する。
D. 利用者の PC のブラウザーとサーバーとの間で共通鍵暗号用のセッション鍵の共有が公開鍵暗号によって実行され、同セッション鍵によって暗号通信が開始される。
E. 利用者は暗証番号やワンタイム・パスワード等をサーバーに送信する。
F. サーバーは暗証番号等によって利用者の本人確認を行う。
G. 同確認が成功した後、利用者は口座振込に必要な情報をサーバーに送信する。なお、サービス内容によっては、第二暗証番号等による取引実行の確認が行われることもある。
H. サーバーは指図に従って処理を実行する。
I. サーバーは処理の完了を通知する。

図表１　インターネット・バンキングにおける処理の流れと中間侵入攻撃（概念図）



こうした処理に対して中間侵入攻撃を試みる際に、攻撃者は、サービス提供者と利用者との間で交信される以下の情報を入手することができる[1]。

① サービス提供者のサーバーの公開鍵（図表１の B の処理時）
② 利用される暗号アルゴリズムの種類（同 B の処理時）
③ 公開鍵暗号による暗号文（暗号化されたセッション鍵等）（同 D の処理時）
④ 共通鍵暗号による暗号文（暗号化された暗証番号や口座情報等）（同 E、G の処理時）

仮に、共通鍵暗号に問題があった場合、上記④の暗証番号等の情報が漏洩したり改ざんされたりする可能性がある。公開鍵暗号やその構成要素であるハッシュ関数に問題があった場合には、上記③のセッション鍵等が漏洩し、共通鍵暗号の効果が

1 これらに加えて、攻撃者が適当なデータを暗号文として選んでサーバーや利用者の PC に送信し、それらの反応を手掛かりに暗号アルゴリズムの解読等を試みるという攻撃も知られている（Bleichenbacher [1998]）。ただし、当該データの形式を検査して異常なデータが連続して送信される場合には取引を停止するといった運用によって対応が可能であることから、ここでは、そうした運用によって対応済みであると想定して議論を進める。

失われてしまう可能性がある。

(2) 各暗号アルゴリズムにおいて留意すべき攻撃

共通鍵暗号、ハッシュ関数、公開鍵暗号について、学術的な安全性評価の概要と留意すべき攻撃を説明する。

イ．共通鍵暗号

共通鍵暗号は、データ（平文）の暗号化と復号に共通の鍵が用いられる方式であり、秘匿したいデータ本体の暗号化に利用されるケースが多い。金融分野における代表的な共通鍵暗号としては、トリプル DES と AES が挙げられる。

安全性に関する研究では、「攻撃者が当該アルゴリズムを知っている」という前提のもとで、攻撃者が用いる情報の種類と量、および、計算量によって安全性が評価される。具体的には、「攻撃者が平文や暗号文を何ら操作することなく入手した平文・暗号文ペア」（既知平文・暗号文ペア）を利用する場合と、「攻撃者が選択した平文に対する暗号文を入手する場合の平文・暗号文ペア」（選択平文・暗号文ペア）を利用する場合が前提となるケースが多い。

インターネット・バンキングの例の場合、攻撃者は取引時に通信される暗号文（暗号化された取引データや暗証番号等）を利用可能である。このとき、平文が一定のデータ形式に従っている場合を考慮すると、攻撃者が当該暗号文に対応する平文もある程度推定できると想定しておく必要がある。そうした攻撃のなかで最も容易に実行可能なものは全数探索法であり、**サービス提供者にとっては、全数探索法がどの程度の資金と時間によって実行可能かが主な関心事となる**。学界では、全数探索法への安全性は、鍵のサイズを N ビットとすると、「2 の N 乗回の暗号化処理に相当する計算量によって解読できる」と表現される。

一方、大量の既知平文・暗号文ペアを利用しつつ、当該暗号アルゴリズムの数学的な性質を巧みに利用して全数探索法よりも効率的に鍵の推定を試みる攻撃も盛んに研究されている。そうした攻撃が提案されると、その暗号アルゴリズムは「学術的に解読された」ものとして扱われる。こうした攻撃が現実の脅威となるか否かを検討する際には、**必要な既知平文・暗号文ペアを入手するためにどの程度の資金や時間が必要になるかを明確にする必要がある**。例えば、利用者がデータを暗号化してサーバーに送信するタイミングで攻撃者が暗号文を盗聴する場合、既知平文・暗号文ペアの入手可能性は、通信回線の帯域、利用者のサービス利用頻度、その他の運用上の制約等に依存する。

また、データの一貫性確認のためにメッセージ認証コード（message authentication code; MAC）として共通鍵暗号が利用されるケースが想定されるが、これらの安全性評価に関しても上記の留意点が当てはまるほか、各メッセージ認証コードに特有の攻撃が提案されているケースもあり、留意が必要である。

ロ．ハッシュ関数

ハッシュ関数は、大きなサイズのデータを一定のサイズ（例えば160ビット）のランダムなデータに変換する関数であり、出力から入力を得ることが困難であるように設計される。このため、ハッシュ関数の安全性評価は、**与えられた出力に対する入力を得ることがどの程度難しいかがポイント**となっており、そうした入力を得るために必要な計算量が安全性の尺度となっている。具体的には、次の攻撃に対する安全性が評価対象となるケースが多い。

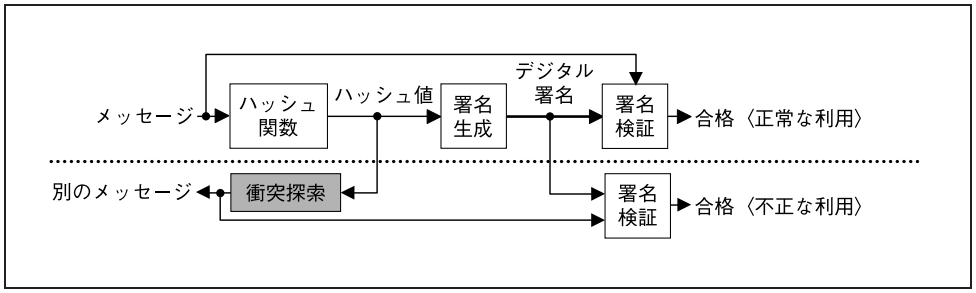
- 出力が同一となるような複数の異なる入力のペア（衝突ペアと呼ばれる）を見つける（衝突探索）。
- 所与の入出力ペアから、同じ出力を得る別の入力（第二原像と呼ばれる）を見つける（第二原像探索）。
- 所与の出力に対応する入力（原像と呼ばれる）を見つける（原像探索）。

上記のどの攻撃に注目するかはハッシュ関数の利用形態に依存する。本稿のインターネット・バンキングの場合、ハッシュ関数はデジタル署名における①署名生成の対象となるデータの生成に利用される場合と、②共通鍵暗号用の鍵として用いられる擬似乱数生成用に利用される場合が主に想定される。

上記①の署名生成対象のデータ生成に利用される場合、出力が同一となる異なる入力が見つかれば、それが特定のデジタル署名に対して通用する場合がある（図表2参照）。このため、衝突探索と第二原像探索に対する安全性が重要となる。ただし、署名生成対象のデータの形式が規定されている場合、データの形式を満足するような入力ペアの探索がどの程度困難かが重要となる。上記②の**擬似乱数生成にハッシュ関数を利用する場合、**擬似乱数生成の種（ハッシュ関数への入力）の秘匿が必要であり、**原像探索がどの程度困難かが重要**となる。

衝突探索、第二原像探索、原像探索に対する安全性は各探索に必要な計算量によって評価される。ハッシュ関数のアルゴリズムに問題がないとすれば、ハッシュ関数の出力のサイズを N ビットとすると、衝突探索の計算量は「2の $N/2$ 乗回程度のハッシュ関数演算」に相当するほか、原像探索と第二原像探索の計算量はともに「2の N 乗回程度のハッシュ関数演算」に相当するとみられている。これらよりも少ない

図表2　デジタル署名におけるハッシュ値の衝突を使った攻撃例



計算量によって攻撃が達成可能であると評価された場合、当該ハッシュ関数のアルゴリズムに問題があるとの学術的な評価が下される[2]。

ハ．公開鍵暗号

公開鍵暗号は、守秘の目的で利用する場合とデジタル署名の目的で利用する場合がある。守秘の目的の場合[3]、復号の鍵を秘密に管理すれば、暗号化の鍵を公開して利用することができるという特徴をもつ暗号化方式といえる。公開される暗号化の鍵（公開鍵）を利用しても復号（逆変換）を実行困難であることが必要であり、「解を求めることが難しい数学の問題」を用いて実現するケースが多い。金融分野の主流であるRSAは、「大きな2つの素数からなる合成数を素因数分解することが困難である」という性質を利用している。合成数（公開鍵に対応）が素因数分解できると秘密の復号用の鍵（秘密鍵）を容易に入手可能であり、素因数分解は合成数のサイズが大きくなるほど難しくなる。

安全性に関しては、共通鍵暗号と同様に、攻撃者が利用する情報の種類と量、解読に必要な計算量に基づいて評価される。本稿のインターネット・バンキングの例においては、攻撃者は、利用者やサービス提供者の公開鍵や、両者の間で通信される暗号文を入手可能なほか、公開鍵を用いて任意の平文に対する暗号文も計算可能であり、選択平文・暗号文ペアも攻撃に利用可能といえる。

公開鍵のみを用いた攻撃として、安全性の根拠となる数学の問題を解いて秘密鍵を得るという攻撃がまず想定される。本攻撃への安全性は数学の問題を解くために必要な計算量によって評価され、RSAの場合、公開鍵の素因数分解に必要な計算量によって示される。仮に、本攻撃が現実の脅威となったとすれば、当該利用者だけでなく、同じサイズの公開鍵を利用する他の利用者にも影響が及ぶ可能性があることから、サービス提供者には、**公開鍵暗号の安全性の根拠となっている数学の問題の高速解法の動向に特に留意する必要がある。**

数学の問題を直接解くのではなく、既知平文・暗号文ペアや選択平文・暗号文ペアを巧みに利用する攻撃も提案されている。ただし、これらの攻撃に対しては、一定の条件のもとで十分な安全性を有することを数学的に証明可能な方式が提案されている。安全性が成立するための条件が現実の環境において満足されていることを確認する必要があるものの、こうした方式の採用によって未知の攻撃に対しても一定の安全性を確保できると期待される。

このように、**サービス提供者は、安全性の根拠となる数学の問題の解法に留意するとともに、安全性証明可能な方式の採用を検討することが有用である。**なお、ハッシュ関数を構成要素として利用する公開鍵暗号の場合、その安全性は当該ハッシュ関数の安全性にも影響されることとなる。

2 金融分野で広く使われているSHA-1の場合、衝突探索に必要とみられている計算量が安全なハッシュ関数における計算量よりも少なく、衝突探索に対して安全性上問題があるとの評価が一般的となっている。もっとも、実際の衝突は現時点では示されていない。

3 以下の議論は、公開鍵暗号をデジタル署名として利用する場合も当てはまる。

以上を簡単にまとめると次のとおりとなる。

	代表例	主な利用形態	主に留意すべき攻撃
共通鍵暗号	トリプル DES AES	守秘、認証（MAC）	全数探索法、既知平文・暗号文ペアを利用した攻撃
ハッシュ関数	SHA-1	デジタル署名生成	衝突探索、第二原像探索
		疑似乱数生成	原像探索
公開鍵暗号	RSA	守秘、デジタル署名	安全性の根拠となる数学の問題の高速解法

3. 安全性の状態に応じて安全性評価結果をどう活用するか

前節では、アプリケーションの形態から注目すべき攻撃のタイプをどのように選択するかを説明した。注目すべき攻撃のタイプや安全性評価結果は、当該暗号アルゴリズムの安全性評価の状態にも依存する。本稿では、DES のケースを例に取り上げて、安全性評価の状態に応じて注目すべき安全性評価結果が変化することを説明するとともに、どのような対応が求められるかを DES の事例に基づいて説明する。なお、同様の検討をハッシュ関数（MD5）、公開鍵暗号（RSA）に関しても行っており、具体的な内容については補論を参照されたい。

(1) 暗号アルゴリズムの安全性評価の状態

暗号アルゴリズムの安全性評価の状態は以下の 4 つに分類することができる。

- 状態 1：暗号アルゴリズムの欠陥を利用した攻撃方法が知られていない。
- 状態 2：暗号アルゴリズムの欠陥を利用した攻撃方法が提案されているものの、部分的な攻撃成功にとどまっており、学術的な解読に至っていない。
- 状態 3：暗号アルゴリズムの欠陥を利用した攻撃方法が提案されており、学術的に解読されている。
- 状態 4：暗号アルゴリズムを利用した実際のシステムやアプリケーションにおいて現実的な脅威となる攻撃方法が示されている。

これらのうち、状態 2 における「部分的な攻撃成功」の状態は、公開鍵暗号の場合、安全性の根拠となっている数学的問題の解決の糸口が発見されているものの、暗号アルゴリズムそのものの安全性が脅かされているわけではないという状態に対応する。共通鍵暗号の場合には、当該攻撃方法を適用するためには暗号アルゴリズムを構成する関数や段数を幾分変更する必要があるとか、膨大な量の平文・暗号文ペアを入手する必要があるといった条件が存在する状態に相当する。状態 2 から徐々に

状態 3 へ遷移するケースが多く、状態 2 での安全性評価結果の監視が重要となる[4]。

状態 3 において提案される攻撃においては、攻撃実行に必要な既知平文・暗号文ペアの数が膨大となるケースが多く、「学術的に解読された」としても鍵の更新の頻度を高めるなどの運用面での対応も可能であるとみられる。

また、状態 4 は、中規模研究所レベルの計算機環境[5]によっていくつかのアプリケーションにおいて暗号アルゴリズムを攻撃可能となっており、暗号アルゴリズム単体によるセキュリティの効果が喪失している状態といえる。

(2) 共通鍵暗号 DES の事例

上記の状態 1〜4 を DES の事例に基づいて説明する。

イ．DES とその安全性評価の流れ

DES は、64 ビットの平文と 56 ビットの鍵を入力として 64 ビットの暗号文を出力する暗号アルゴリズムであり、F 関数と呼ばれる関数を 16 回繰り返す構造（「16 段」と呼ばれる）を有している。DES は、1977 年に FIPS（Federal Information Processing Standards）として米国政府標準暗号に制定され、その後トリプル DES や AES に移行するまでの間、金融分野の標準的な暗号アルゴリズムとして広く利用されてきた。DES の主な安全性評価結果等は次のとおりである。

- 1977 年：DES が米国政府標準暗号（FIPS 46）として制定された。
- 1990 年：Biham and Shamir [1991] によって差分解読法が提案され、約 29 京（＝ 2 の 58 乗）回の暗号化処理に相当する計算量によって解読可能であることが示された（15 段に変更した DES の場合、全数探索法と同程度の計算量によって解読可能）。
- 1993 年：Matsui [1993] によって線形解読法が提案され、約 8.8 兆（＝ 2 の 43 乗）回の暗号化処理に相当する計算量で解読可能であることが計算機実験によって示された。
- 1998 年：Shimoyama and Kaneko [1998] によって線形解読法の改良版が提案され、約 6.5 兆回の暗号化処理に相当する計算量で解読可能であることが計算機実験によって示された。
- 1998 年：米国の非営利団体 EFF（Electronic Frontier Foundation）によって全数探索法の専用ハードウエア DES Cracker（EFF [1998]）が開発され（予算は約 25 万ドル）、実験によって約 56 時間で鍵の探索が成功した。

4 NIST や CRYPTREC など多くの評価機関は、状態 3 に至った時点でその暗号アルゴリズムの停止を呼び掛ける場合が多い。これは、対策を講じる前に状態 4 に至った場合には実際のアプリケーションのリスクが急激に増大する可能性があり、対策実施に必要な時間を考慮して早めの対応を促す狙いがあるとみられる。

5 Blaze *et al.* [1996] における “Corporate Department” の規模である 30 万ドルの資金によって準備される計算機環境を想定している。

- 2008 年：Güneysu *et al.* [2008] によって、約 6 日で全数探索法による鍵の探索を実行可能な専用ハードウエア COPACOBANA（予算は約 1 万ユーロ）が提案された。

これらの評価結果をベースに状態 1〜4 がどの期間に対応するかを考えると、以下のとおり整理することができる。

【状態 1】1977 年（DES の FIPS 制定時）〜1989 年頃
【状態 2】1990 年頃（差分解読法の提案時点）〜1992 年頃
【状態 3】1993 年頃（線形解読法の提案時点）〜1998 年頃
【状態 4】1998 年（DES Cracker 提案時点）以降

ロ．状態 1（暗号アルゴリズムの欠陥を用いた攻撃方法の提案はない）

1977 年から 1990 年頃までは、全数探索法以外の攻撃が提案されておらず、安全性評価は計算機能力の向上の度合いの評価が中心であった。実際にいくつかの研究成果によって、当時の計算機能力のもとで DES が十分な安全性を確保していたとの見方が示されている（Blaze *et al.* [1996]、Lenstra and Verheul [2001]）。この状態では、計算機能力の向上による攻撃の実行可能性の評価結果に留意することが求められていた。

ハ．状態 2（「学術的に解読された」状態には至っていない）

1990 年に、新しい攻撃方法として差分解読法が提案された。差分解読法は、通常の DES（16 段）を攻撃する場合には全数探索法以上の計算量が必要であったが、16 段を 15 段に変更した DES であれば全数探索法よりも少ない計算量で攻撃が可能であった。この状態では、計算機能力の向上に加え、攻撃方法の向上による攻撃の実現可能性の評価結果に留意する必要があった。特に DES は、当時技術仕様が公開されていた数少ない暗号アルゴリズムであり、差分解読法提案後に暗号研究者の注目が DES に集中することが予想され、攻撃方法の向上が急速に進む可能性が高いとみられていた。

ニ．状態 3（「学術的に解読された」状態に至っている）

1993 年に、差分解読法よりも強力な攻撃手法として線形解読法が提案された。線形解読法は、全数探索法よりも少ない計算量によって DES（16 段）を攻撃可能であったものの、攻撃実行に必要なデータ量が膨大（約 560 兆個の既知平文・暗号文ペア）であったことから、現実の脅威にまでは至らなかった。1998 年には線形解読法の改良版が発表されたものの、攻撃実行に必要なデータ量が膨大であるという点は変わらなかった。

一方、インターネットの普及や計算機性能の向上が着実に進み、計算機能力の向上による攻撃の実行可能性を評価する試みが行われた。1998 年には、RSA 社が DES

の解読コンテスト DES Challenge を開催し、インターネットを介した分散処理（平均して約 4 万台の PC を動員）によって約 39 日で解読が成功したほか、約 25 万ドルの専用ハードウエア DES Cracker によって約 56 時間で解読が成功した。この結果、比較的実現可能性が高い資金と時間によって DES の解読が可能であることが実証され、全数探索法が現実的な脅威として認識された。

このように、1993 年頃から 1998 年頃においては、計算機能力の向上による攻撃の実行可能性と攻撃方法の向上による攻撃の実現可能性の両方に関する評価結果がポイントとなった。NIST は、こうした流れを踏まえて、1997 年初に次世代の共通鍵暗号 AES の選定を開始する方針を発表しており、より安全性の高い暗号アルゴリズムへの移行について検討が開始されたといえる。

ホ．状態 4（実システムへの攻撃方法が提案されている）

DES Cracker による攻撃成功によって、大半の暗号研究者にとって DES が研究対象外となったとみられる。金融分野においては、ISO/TC68 における暗号アルゴリズムに関連する国際標準において DES に代わってトリプル DES が規定されるようになり、実システムで DES が単体で利用されている事例も減少したとみられる。ただし、大規模なシステムにおける暗号アルゴリズムの移行には相応の時間が必要となる点を踏まえると、1998 年以降、ある程度の期間は DES が使用され続けていたと考えられる。

このように、現実のアプリケーションへの影響を検討することが必須となっており、運用上の弱点を利用した攻撃の実行可能性の評価結果に留意することが最も重要になったといえる。例えば、同一の鍵で処理できる平文の数の上限を設定したり、鍵の交換の頻度を高めたりするといった鍵管理上の運用等において不十分な点があれば、「運用上の弱点」として現実の脅威につながる可能性がある[6]。

(3) 考察

上記の DES の事例を踏まえ、状態 1～4 において特に留意すべきポイントをまとめると以下のとおりである。

- **状態 1 においては、計算機能力の向上による攻撃の実行可能性の評価結果に留意することが必要である[7]。**
- **状態 2 においては、計算機能力の向上による攻撃の実行可能性の評価結果に加え、攻撃方法の向上による攻撃の実行可能性の評価結果に留意することが必要**

6 例えば、ある程度短期間で DES の鍵候補の半数を探索可能なハードウエアが実現できれば、鍵の交換の頻度を上げたとしても、同ハードウエアを利用した鍵の探索によって無視できない確率で正しい鍵を推定できるケースがある（Kusuda and Matsumoto [1997]）。

7 公開鍵暗号については、安全性の根拠となっている数学的問題の高速解法の向上の動向もフォローすることが必要である。詳しくは、補論(2)（RSA の事例）を参照されたい。

である。

- 状態 3 においては、計算機能力の向上および攻撃方法の向上に加えて、運用上の弱点による攻撃の実行可能性の評価結果についても留意する必要がある。その際、既存の対策が不十分と判断される場合には、追加的な運用による対応や暗号アルゴリズムの移行等の技術的な対応の方針の検討を開始することが望ましい。
- 状態 4 においては、既に暗号アルゴリズムのセキュリティ効果が失われており、運用上の弱点による攻撃の実行可能性の評価結果に留意することが必須である。仮に当該システムにおいて弱点が見つかった場合には、運用等で応急処置を施しつつ、暗号アルゴリズムの移行を早急に検討する必要がある。

こうした各状態の先行きを考えるうえで、状態の遷移がどの程度のスピードで進行するかが重要となってくる。本節で取り上げた DES の場合には、FIPS として制定された後 10 年以上は安全性の問題が指摘されなかったものの、差分解読法が提案されて状態 2 に移行した後、3 年足らずで線形解読法が提案されて「学術的に解読された」状態（状態 3）に至っている。これは、**致命的でなくてもいったん暗号アルゴリズムの安全性上の問題が提起される（状態 2）と、その後安全性評価の研究が活発化し、短期間で安全性低下が顕著となる（状態 3 に移行する）可能性があること**を示唆しているといえる[8、9]。

また、**当該暗号アルゴリズムが広範囲に利用され、システム更改のタイミングや複数のシステム間における相互運用性の確保等によって暗号アルゴリズムの移行に相当の時間がかかるとみられるケースでは、状態 2 であったとしても、暗号アルゴリズムの移行を意識して対応を開始することが有用であると考えられる**。例えば、①当該暗号アルゴリズムが利用されているシステムと利用目的を把握するための調査の実施、②暗号アルゴリズムの移行にかかわる関係者の特定、③関係者間での問題意識の共有、④当該暗号アルゴリズムの安全性評価の実情に関する専門家へのヒアリングといった対応が考えられる。

上記の点に加えて、安全性評価結果の先行き見通しを考えるうえで、**安全性評価結果が報告されていない場合であっても、「当該暗号アルゴリズムの安全性は時間とともに低下している」との認識を常にもっておく**ことが重要である。特に、状態 3

8 MD5 の場合には、提案後 1 年ほどで安全性の問題が指摘された後、状態 2 が 10 年程度継続したものの、状態 3 から状態 4 への移行は 1 年程度と短期間であった。また、RSA を利用した守秘方式の 1 つである RSAES-PKCS1-v1_5 の場合、状態 3 を経ずに状態 2 から状態 4 へ遷移している。これらの詳細は補論を参照されたい。

9 ただし、AES のように、学界を通じて全世界的に安全性評価の対象となり、その結果として高い評価を得た暗号アルゴリズムの場合、また、安全性評価結果を踏まえて改良された暗号アルゴリズムの場合には、状態 2 であったとしても、状態 3 には短時間では至らないケースが少なくないと考えられる。先行きの動向をどのように見通すかについては、状態 2 において提案されている攻撃の内容や当該攻撃に必要な計算量（あるいは資金や時間）等に依存するとみられることから、暗号研究の専門家による知見を活用するなどの対応が求められる。例えば、AES の暗号アルゴリズム選定コンテストは、こうした状態 2 の見極めを世界中の暗号研究者によって実施することを目的としているとみることができる。

の「学術的に解読された」暗号アルゴリズムの場合、計算機能力の向上等によって攻撃の実行可能性が向上しているにもかかわらず、安全性評価を実施してその結果を発表する誘因が暗号研究者において急速に失われ、安全性評価結果の報告が行われなくなるケースが多い。こうした状況を「安全性評価結果の発表件数が少ないのは安全性評価が定まったためである」と判断するのではなく、「当該暗号アルゴリズムの安全性低下がどのように進行しているかを判断する手掛かりがなくなった」と考えることが必要である[10]。トリプル DES のように、学術的に解読された暗号アルゴリズムであっても相互運用性確保の観点から広く利用されているものが少なくなく、実務的にはそうした暗号アルゴリズムを対象にした安全性評価を継続して行うことが重要である。

このように、安全性評価の状態に応じて留意すべきポイントが変化していくことを認識し、リスク管理上どのような対応が望ましいかを検討する際には、暗号アルゴリズムの安全性評価に関する研究動向について、暗号ユーザーの問題意識を暗号研究者等の専門家に適宜伝え、専門家からの助言を参考にすることが重要である。わが国の場合、電子政府推奨暗号リストを管理する CRYPTREC の成果[11]を参照したり、暗号アルゴリズムをはじめとする情報セキュリティ技術の調査・研究を行っている公的研究機関（例えば、情報処理推進機構セキュリティセンター、情報通信研究機構情報通信セキュリティ研究センター）に相談したりすることが考えられる。

4. 計算量を資金や時間でどのように表現するか

2 節と 3 節において説明したように、各アプリケーションや安全性評価の状態に応じて留意すべき安全性評価結果を特定できたとする。次に暗号ユーザーにとって必要となるのは、そうした安全性評価結果の内容をリスク管理に活用することである。そのためには、通常安全性評価結果として示される「攻撃実行に必要な計算量」を「攻撃実行に必要な資金と時間」に翻訳することが必要となる。本節では、まず、こうした翻訳の必要性を説明する。そのうえで、翻訳方法に関する検討の一例として、状態 1、2、3 において留意すべきであると説明した「計算機能力の向上による攻撃の実行可能性」に関する評価結果の翻訳方法を検討する。ここでは、共通鍵暗号における全数探索法を取り上げ、安全性評価結果の翻訳方法を検討する。

10 MD5 の場合、学界の関心事が AES に集中していた状態 2 の期間（1993～2003 年）において安全性評価結果の報告が少なかったものの、その後相次いで安全性低下の指摘が行われて安全性が急速に低下したという経緯がある。詳しくは補論を参照されたい。

11 CRYPTREC の成果は http://www.cryptrec.go.jp/ から入手することができる。

(1) 学術的な安全性評価結果を暗号ユーザーが活用するうえでの課題

仮に、中間侵入攻撃によって口座振込先と振込金額の改ざんが成功したとすれば、当該金額を攻撃者が不正に入手し、利用者やサービス提供者が同金額分の損害を被る結果となる[12]。また、サービス提供者においては、攻撃成功に伴う当該サービスに対する信頼低下というリスクも存在する。サービス提供者としての金融機関は、こうしたリスクを評価して当該暗号アルゴリズムの使用を継続するか否かについて意思決定を行うことになる。

こうした検討は、リスク管理上、次の手順で進められることが望まれる。

(I) 当該アプリケーションにおいて想定される攻撃に基づき、攻撃実行に必要な資金や時間、および、攻撃成功時の不正な利得額を見積もる。

(II) 上記(I)の結果を踏まえて攻撃成功確率[13]を見積もり、不正な利得額と攻撃成功確率の積から「不正な利得額の期待値」を推定する。

➢ ここで、不正な利得額の期待値が攻撃実行に必要な資金を超える場合、攻撃が実行される公算が高い。

(III) 攻撃が実行される公算が高いと判断する場合、攻撃成功によってサービス提供者が被る損害額を見積もり、攻撃成功確率との積によって「損害額の期待値」を推定する。

➢ ここで、損害額の期待値がリスク管理上許容できるレベルを超える場合、損害額の期待値を下げるための対応が必要となる。

(IV) 損害額の期待値を下げるための対策が必要と判断する場合、対策に伴うコストや利便性の低下等を考慮しながら、望ましい対策を検討する。

➢ 例えば、暗号アルゴリズムの更新等の技術的な対応、鍵の使用期間の短縮等の運用的な対応や、これらを組み合わせた対応も考えられる。

上記の手順を実行する場合、暗号ユーザーは、まず「攻撃に必要な資金や時間」を見積もる必要がある。そのためには、当該暗号アルゴリズムに関する学界での安全

12 こうした損害額の期待値は、攻撃の具体的な方法、その成功確率、1 回の攻撃の被害額に依存する。例えば、共通鍵暗号のセッション鍵が解読された場合には当該セッション内での被害に限定されるが、当該銀行のサイトの公開鍵証明書が偽造された場合には、偽サイトを検知できず、1 回の攻撃成功で多数の利用者が被害に遭う可能性が考えられる。このように、攻撃の方法によって損害額の期待値は変化することとなる。

13 攻撃成功確率の見積りを単純化して説明すると、例えば次のような方法が考えられる。まず、上記(I)において見積もられる「不正な利得額」の上限値を攻撃のために投入される資金の上限値とみなし、その金額によって攻撃を実行したときに攻撃成功に必要な時間を求める。攻撃成功に必要な時間が当該アプリケーションにおけるサービス提供期間（例えば、デジタル署名付きメッセージを利用する場合、当該署名の有効期間）の範囲内に収まれば、攻撃は有効となり、攻撃成功確率は 100% となる。範囲内に収まらない場合、攻撃成功確率は 100% を下回る値となる。例えば、全数探索法の場合、鍵の全候補数のうちどのくらいの割合まで探索可能かに応じて攻撃成功確率を近似的に算出することが考えられる。

性評価結果（計算量）を資金や時間に翻訳する必要があるが、そうした翻訳方法に関する体系的な検討結果が報告されていない。そのため、暗号ユーザーは、現状の計算機能力を見積もり、計算機の調達・運用費用を独自に算出することが必要となる。また、数年先に攻撃が実行されることを想定する場合、計算機の性能や暗号解読技術の向上の先行きも評価する必要がある。こうした暗号ユーザーにとってのハードルを低くし、望ましいリスク管理の実行を支援するうえで、計算量を資金や時間に翻訳する方法の検討が重要である。

（2）共通鍵暗号に対する全数探索法に関する既存の評価結果

全数探索法に対する共通鍵暗号の安全性評価結果のうち、「計算量から資金と時間への翻訳」との考え方に沿った既存研究として次の 4 つが挙げられる。

① Blaze *et al.* [1996]

1995 年頃主流であった鍵のサイズが 40 ビットと 56 ビットの共通鍵暗号（DES を想定）を対象に、PC よりも高速で鍵の探索が可能な専用ハードウエア[14]を用いた場合に必要な資金と時間を推計した（図表 3 参照）。

② Kusuda and Matsumoto [1997]

DES の全数探索法に対する安全性に関して、1996 年から 2001 年における計算機性能の向上を一定の前提のもとで推計したうえで、一定時間内に全数探索法を成功させるために必要な資金（専用ハードウエア調達のためのコスト）を試算した。全数探索法実施の時間として、6 つのバリエーション（1 時間、4 時間、1 日、1 週間、1 ヵ月、3 ヵ月）を考慮した。

図表 3　1995 年時点の全数探索法にかかる資金と時間の関係

ハードウエア調達の資金	ハードウエアの種類	全数探索法にかかる時間	
		鍵長 40 ビット	鍵長 56 ビット（DES）
400 ドル	FPGA	5 時間	38 年
1 万ドル	FPGA	12 分	18 ヵ月
30 万ドル	FPGA／ASIC	24 秒／0.18 秒	19 日／3 時間
1,000 万ドル	FPGA／ASIC	0.7 秒／0.005 秒	13 時間／6 分
3 億ドル	ASIC	0.0002 秒	12 秒

資料：Blaze *et al.* [1996] の Table 1 を引用

14 専用ハードウエアとして、FPGA（Field Programable Gate Array）を利用したハードウエアの場合と ASIC（Application Specific Integrated Circuit）を利用したハードウエアを想定している。FPGA は回路構成をソフトウエアによって書換えが可能な集積回路であり、ASIC は、特定用途のために設計・製造され、機能が固定されている集積回路である。

③ Lenstra and Verheul [2001]

DES の安全性に対する信頼が極めて高かった時期を 1982 年と特定したうえで、全数探索法に対して同時点の DES と同程度の安全性を達成するために必要な鍵のサイズを試算した。具体的には、「CPU の処理速度が 18 ヵ月で 2 倍になる」というムーアの法則を前提に、ハードウエアの調達に必要な資金と時間が 1982 年時点の試算（資金は 5,000 万ドル、時間は約 2 日）と等価となる鍵のサイズを算出した[15]。

④ 情報処理推進機構［2004］

Lenstra and Verheul [2001] と同様に、専用ハードウエアによる全数探索法に対して十分な安全性を確保する鍵のサイズ（の下限）を評価した。国際半導体技術ロードマップによる専用ハードウエアの性能向上を前提に、専用ハードウエア調達の資金に関する 4 つの想定[16]のもとで攻撃実行に必要な時間が 1,000 年となるような鍵のサイズを試算した。例えば、専用ハードウエア調達の資金が 1,000 万ドルの場合、2009 年時点で安全な鍵のサイズの下限を 96 ビットと試算している。

(3) 暗号ユーザーの観点からの安全性評価の方向性

イ．どのような評価結果が活用しやすいか

上記の①～④の評価結果は基本的にはムーアの法則による計算機能力の向上を前提としているものの、評価方法の詳細は異なっており、評価結果の意味を適切に理解するためには評価方法の詳細を精査する必要がある。ここでは、攻撃に必要な資金と時間への翻訳のアイデアについて検討することに主眼を置き、評価方法の細かい技術的特徴には立ち入らないこととする。

まず、①の Blaze *et al.* [1996]、②の Kusuda and Matsumoto [1997]、④の情報処理推進機構［2004］は「解読が成功するケース」に着目し、③の Lenstra and Verheul [2001] は「十分な安全性を確保するケース」に着目しているといえる。攻撃実行のタイミングにおける資金と時間に注目するという観点からは、①②④の「解読が成功するケース」に着目した評価の方が活用しやすいと考えられる。

また、①の Blaze *et al.* [1996] の評価では、攻撃に必要な資金として 5 つのケースを前提としており、これらのいずれかに対応する攻撃者を想定する場合には利用可能である。また、②の Kusuda and Matsumoto [1997] の評価も同様である。一方、1 つのケース（5,000 万ドルによる資金）のみを取り上げている③の Lenstra and Verheul [2001] の評価は、同ケースに合致する攻撃者を想定している暗号ユーザーにとっては利用可能であるものの、その他の暗号ユーザーにとっては別途検討が必要となる。

15 本試算では、米国の GDP 成長率の予測値を参照し、攻撃に用いられる資金（1982 年時点で 5,000 万ドル）の名目値が「10 年間で 2 倍になる」との前提のもとで資金の時系列的な増加を考慮している。

16 中規模（1,000 万ドル）、大規模（100 億ドル）、超大規模（米国 GDP の 4%〈国防予算にほぼ匹敵〉）、限界規模（世界の年間 GDP）の 4 つを想定している。

こうしたことから、①②④のように**評価対象の「攻撃に必要な資金と時間」のバリエーションが多いほど、さまざまな暗号ユーザーが利用可能であり汎用性の観点で有用である。**

これに加えて、②③④のように**暗号ユーザーがどのタイミングで暗号アルゴリズムの使用を開始するかという点で、将来攻撃が開始される可能性も考慮し、攻撃開始のタイミングが将来の場合の評価結果も示されることが望ましい。**例えば、2年後に運行開始予定のシステムにおいて暗号アルゴリズムを利用したいという暗号ユーザーの場合、現時点を基準とした評価よりも、2年後を攻撃開始のタイミングに設定した評価結果が有用である。

ロ．検討のアイデア

以上の考察を踏まえたうえで、先行き見通しという観点で④の情報処理推進機構［2004］を参照しつつ、資金と時間への翻訳方法を検討する。情報処理推進機構［2004］においては、国際半導体技術ロードマップを参照して将来の専用ハードウエアの性能向上の見通しを試算しており、将来の性能向上を見積もる際に、「鍵の探索を行うためのICチップが単位時間および単位価格当たり何個の鍵を検索可能か」を「価格性能」として算出している点が注目される[17]。価格性能、攻撃者の資金、鍵の探索の時間の積を計算すると、次のように当該攻撃者が探索可能な鍵の個数が得られる。

$$[\text{探索可能な鍵の数}] = [\text{価格性能}] \times [\text{攻撃者の資金}] \times [\text{鍵の探索の時間}].$$

これを変形すると、以下の関係が得られる。

$$[\text{探索可能な鍵の数}]/[\text{価格性能}] = [\text{攻撃者の資金}] \times [\text{鍵の探索の時間}].$$

ここで、「探索可能な鍵の数」に「攻撃対象の暗号アルゴリズムの鍵の候補数」を代入し、「価格性能」に情報処理推進機構［2004］において算出されている値（脚注17を参照）を代入すれば、「攻撃者の資金」と「鍵の探索の時間」の関係が明らかになる。

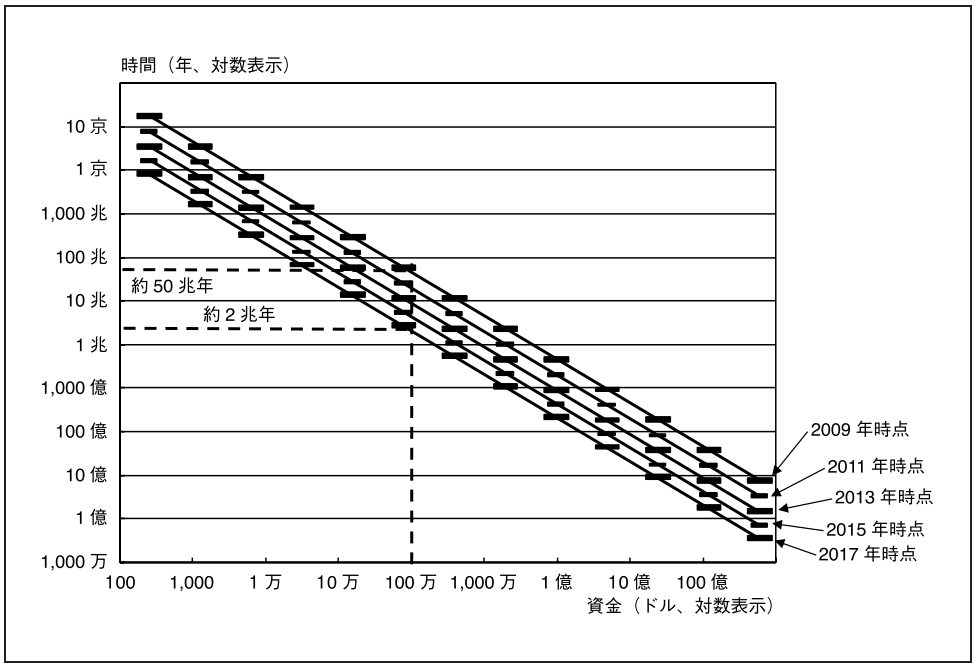
一例として、攻撃開始のタイミングを2009年から2年ごとに2017年まで5つ設定するほか、資金を250（$= 50 \times 5$）ドル、1,250（$= 50 \times 5^2$）ドル、6,250（$= 50 \times 5^3$）

17 価格性能は、チップ集積度、クロック周波数、鍵探索回路規模、鍵探索クロック数（1回の暗号化を行う際のクロック数）、ICチップ単価から算出されている。具体的には、以下の式によって算出されている。

$$[\text{価格性能}] = [\text{チップ集積度}] \times [\text{クロック周波数}]/([\text{鍵探索回路規模}] \times [\text{ラウンド数}] \times [\text{IC チップ単価}]).$$

これらのうち、鍵探索回路規模はトランジスター5万個相当、ラウンド数は10、ICチップ単価は50ドルに設定されており、チップ集積度とクロック周波数は国際半導体技術ロードマップの予測値が利用されている。

図表 4　128 ビットの鍵の全数探索に必要な資金と時間の関係



ドルという具合に増加させる場合[18]、鍵のサイズが 128 ビットの AES における鍵の全数探索に必要な時間を計算すると図表 4 のとおりである。攻撃の資金を約 100 万ドル（約 1 億円）とした場合、2009 年時点で全数探索に約 50 兆年を要し、2017 年時点では約 2 兆年を要するとの見通しが示される。

(4) 考察

本節では、重視すべき安全性評価結果を明確にした後の次のステップとして、各安全性評価結果において示される「攻撃実行に必要な計算量」を「資金と時間」に翻訳することの有用性を説明した。そのうえで、共通鍵暗号における全数探索法に焦点を当てて、資金と時間の関係を明らかにする方法の一例を検討した[19]。今後、同様のアイデアに基づく検討が他の攻撃においても進められることが望まれる。例えば、共通鍵暗号における既知平文・暗号文ペアを利用した攻撃の場合には、本節で検討した計算機能力の試算に加えて、通信回線の帯域の将来見通し等を考慮しつつ、攻撃者が入手可能な既知平文・暗号文ペア数を見積もることが必要となる。このよ

18 ここでは、IC チップの単価が 50 ドルと仮定していることから、攻撃の資金を 50 ドルの倍数として表している。図表 4 は攻撃の資金を 13 通りに設定して攻撃の時間を計算しており、理解のしやすさからそれらの点を実線で結んでいる。厳密には、鍵探索専用ハードウエアを構成する IC チップの単価を 50 ドルに設定しており、50 ドルごとに攻撃の時間が低下する「階段」型のグラフで表現されることになる。

19 公開鍵暗号（RSA）の場合における素因数分解実行に必要な資金と時間に関しては、補論を参照されたい。

うに、検討内容はより複雑になるとみられることから、暗号ユーザー側から暗号研究の専門家にこうした評価のアップデートを依頼するなど、専門家の支援を得ながら検討を進めることが有用である。

また、こうした検討結果を利用する際には、**同検討結果があくまで当該検討時点の技術条件に基づくものであり、今後の技術環境の変化次第で検討結果が変化することを認識しておくことが重要である。**暗号アルゴリズム導入後、想定よりも急速に計算機性能が向上した場合や、新しい強力な攻撃方法が提案された場合には、当初の検討結果よりも安全性低下が急速に進む可能性がある。**学界の動向を適切にフォローし、当初検討した資金と時間の関係を適宜アップデートすることが必要である。**安全性評価の状態が状態4になった場合、当該暗号アルゴリズムの移行を検討することが優先的に必要になると考えられるが、そうした場合であっても、移行にどの程度の時間的余裕があるかに関する見通しを得ることが重要となる。そうした時間的余裕を明確にする際に、資金と時間への翻訳が有用であると考えられる。

5. おわりに

金融機関をはじめとする暗号ユーザーが暗号アルゴリズムを利用するうえで、それらの安全性評価結果を適切に解釈し、リスク管理に活用することが求められる。そのためには、まず、**これまでに学界において提案されているさまざまなタイプの攻撃とその安全性評価結果を、個別のアプリケーションや安全性評価の状態に応じて取捨選択することが必要である。また、安全性評価結果をリスク管理に利用する際に、攻撃実行に必要な計算量をそのための資金と時間に翻訳することが求められる。**

これらの課題に関して、本稿では、まず、インターネット・バンキングを例にどのようなタイプの攻撃を相対的に重視すべきかについて検討を行った。さらに、暗号アルゴリズムにおける安全性評価の状態を4つに分類したうえで、各状態において留意すべき安全性評価結果や安全性低下の進み方をDESの事例に基づいて説明した。特に、問題点を指摘するような安全性評価結果が発表されていない場合であっても、当該暗号アルゴリズムの安全性は日々低下していることを十分認識しておくことが重要であることを説明した。また、攻撃実行に必要な計算量を資金と時間に翻訳する方法の一例を検討した。

今後、本稿で示したような翻訳方法のアイデアをベースに具体的な検討が学界において進められることが期待される。また、安全性評価に関する検討をより効果的に行ううえで、暗号ユーザーは、暗号の研究者や技術者に自らの問題意識を伝え、認識を共有しておくことが重要である。こうした取組みが今後進展し、暗号ユーザーがより適切かつ効率的に学界における安全性評価結果をリスク管理に活用することができるようになることを期待したい。

参考文献

宇根正志・神田雅透、「暗号アルゴリズムにおける 2010 年問題について」、『金融研究』第 25 巻別冊第 1 号、日本銀行金融研究所、2006 年、31～71 頁

神田雅透・山岸篤弘、「暗号世代交代についての暗号学会とビジネスサイドのギャップをどう埋めるか～SSL サーバの暗号設定の現状からの考察～」、『2009 年暗号と情報セキュリティシンポジウム予稿集』4E2-4、電子情報通信学会、2009 年

齊藤真弓、「RSA 署名方式の安全性を巡る研究動向について」、『金融研究』第 21 巻別冊第 1 号、日本銀行金融研究所、2002 年、285～324 頁

情報処理推進機構、『電子政府行政事業化事業　将来の暗号技術に関する安全性要件調査　調査報告書』、情報処理推進機構、2004 年 2 月

情報通信研究機構・情報処理推進機構、『CRYPTREC Report 2006』、情報通信研究機構・情報処理推進機構、2007 年 3 月、13～18 および 35～41 頁

情報処理推進機構セキュリティセンター、『APOP（エーポップ）方式におけるセキュリティ上の弱点（脆弱性）の注意喚起』、情報処理推進機構、2007 年 4 月

Biham, Eli, and Adi Shamir, "Differential Cryptanalysis of DES-Like Cryptosystems," *Journal of Cryptology*, 4 (1), 1991, pp. 3–72.

Blaze, Matt, Whitfield Diffie, Ronald L. Rivest, Bruce Schneier, Tsutomu Shimomura, Eric Thompson, and Michael Wiener, "Minimal Key Length for Symmetric Ciphers to Provide Adequate Commercial Security," *A Report by an Ad Hoc Group of Cryptographers and Computer Scientists*, January, 1996.

Bleichenbacher, Daniel, "Chosen Ciphertext Attacks against Protocols Based on the RSA Encryption Standard PKCS #1," *Proceedings of CRYPTO '98*, LNCS 1462, Springer-Verlag, 1998, pp. 1–12.

Coppersmith, Don, Matthew Franklin, Jacques Pararin, and Michael Reiter, "Low-Exponent RSA with Related Messages," *Proceedings of EUROCRYPT '96*, LNCS 1070, Springer-Verlag, 1996, pp. 1–9.

den Boer, Bert, and Antoon Bosselaers, "Collisions for the Compression Function of MD5," *Proceedings of EUROCRYPT '93*, LNCS 773, Springer-Verlag, 1994, pp. 293–304.

Dobbertin, Hans, "Cryptanalysis of MD5 Compress," *Rump Session Talks at EUROCRYPT '96*, 1996

Electronic Freedom Frontier, *Cracking DES*, O'Reilly & Associates, 1998.

European Network of Excellence in Cryptology, *ECRYPT Yearly Report on Algorithms and Keysizes (2007–2008)*, IST-2002-507932 ECRYPT, July, 2008.

Güneysu, Tim, Timo Kasper, Martin Novotný, Christof Paar, and Andy Rupp, "Cryptanalysis with COPACOBANA," *IEEE Transactions on Computers*, 57 (11), IEEE, November, 2008, pp. 1498–1513.

Kusuda, Koji, and Tsutomu Matsumoto, "A Strength Evaluation of the Data Encryption Standard," IMES Discussion Paper No. 97-E-4, Institute for Monetary and Economic Studies, Bank of Japan, 1997.

Lenstra, Arjen K., and Eric R. Verheul, "Selecting Cryptographic Key Sizes," *Journal of Cryptology,* 14 (4), Springer-Verlag, 2001, pp. 255–293.

———, Xiaoyun Wang, and Benne de Weger, "Colliding X.509 Certificates," *Cryptography ePrint Archive* 2005/067, IACR, 2005.

Matsui, Mitsuru, "Linear Cryptanalysis Method for DES Cipher," *Proceedings of EUROCRYPT '93,* LNCS 765, Springer-Verlag, 1993, pp. 368–397.

Molnar, David, Marc Stevens, Arjen Lenstra, Benne de Weger, Alexander Sotirov, Jacob Appelbaum, and Dag Arne Osvik, "MD5 Considered Harmful Today: Creating a Rogue CA Certificate," *Presentation at 25th Chaos Communication Congress,* 25C3, December, 2008.

National Institute of Standards and Technology, "Recommendation for Key Management—Part 1: General (Revised)," NIST Special Publication 800-57, March, 2007, pp. 61–71.

Sasaki, Yu, and Kazumaro Aoki, "Finding Preimages in Full MD5 Faster than Exhaustive Search," *Proceedings of EUROCRYPT 2009,* LNCS 5479, Springer-Verlag, 2009, pp. 134–152.

———, Go Yamamoto, and Kazumaro Aoki, "Practical Password Recovery on an MD5 Challenge and Response," *Cryptography ePrint Archive,* 2007/101, IACR, 2007.

Shimoyama, Takeshi, and Toshinobu Kaneko, "Quadratic Relation of S-box and Its Application to the Linear Attack of Full Round DES," *Proceedings of CRYPTO '98,* LNCS 1462, Springer-Verlag, 1998, pp. 200–211.

Wang, Xiaoyun, Dengguo Feng, Xuejia Lai, and Hongbo Yu, "Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD," *Cryptography ePrint Archive,* 2004/199, IACR, 2004.

———, and Hongbo Yu, "How to Break MD5 and Other Hash Functions," *Proceedings of EUROCRYPT 2005,* LNCS 3494, Springer-Verlag, 2005, pp. 19–35.

補論．MD5 と RSA における安全性評価結果の分析

（1）MD5 の事例

イ．安全性評価の流れ

MD5 は、任意の長さの入力に対して 128 ビットのデータをハッシュ値として出力するハッシュ関数であり、1991 年に提案された。MD5 は SSL のハッシュ関数として幅広く利用されており、わが国のインターネット・バンキング・サービスにおけるサーバーの認証等にも広く利用されている（神田・山岸［2009］）。これまでに知られている MD5 の攻撃結果等は次のとおりである。

- 1991 年：MD5 が提案された。
- 1993 年：MD5 を構成する関数（圧縮関数と呼ばれる）において安全性の問題が発見された（der Boer and Bosselaers [1994]）。
- 1996 年：MD5 の仕様の一部を変更したハッシュ関数の衝突が、汎用 CPU 搭載の計算機によって約 10 時間で発見された（Dobbertin [1996]）。
- 2004 年：MD5 の衝突が PC レベルの計算機資源によって約 1 時間で発見された（Wang, Feng, Lai, and Yu [2004]、Wang and Yu [2005]）。
- 2005 年：MD5 を利用して作成された公開鍵証明書（X.509 準拠）の偽造の一例が示された（Lenstra, Wang, and de Weger [2005]）。
- 2007 年：MD5 を利用した電子メール受信時のパスワード認証方式 APOP（Authenticated Post Office Protocol）において、同パスワードが漏洩する可能性が指摘された（情報処理推進機構セキュリティセンター［2007］、Sasaki, Yamamoto, and Aoki [2007]）。
- 2008 年：MD5 を利用した一部の SSL 証明書が現実的な計算機資源と時間によって偽造可能である旨が発表された（Molnar *et al.* [2008]）。
- 2009 年：ある特定の出力に対する MD5 の入力を探索可能であることが発表された（Sasaki and Aoki [2009]）。

これらの評価結果をベースに状態 1～4 がどの期間に対応するかを考えると、以下のとおり整理することができる。

【状態 1】1991 年（MD5 の提案時）～1992 年頃
【状態 2】1993 年頃（MD5 を構成する関数の問題判明）～2003 年頃
【状態 3】2004 年頃（MD5 の衝突発見）
【状態 4】2005 年頃（X.509 等における脅威判明）以降

ロ．状態１（暗号アルゴリズムの欠陥を用いた攻撃方法の提案はない）

1991 年に MD5 が提案されてから 1992 年頃までは、MD5 およびその構成要素に関して安全性上の問題点は報告されていなかった。この時期においては、計算機能力の向上による攻撃の実行可能性に留意することが必要であった。

ハ．状態２（「学術的に解読された」状態に至っていない）

1993 年に、MD5 の圧縮関数において衝突が発見されるという問題点が示された。また、1996 年には、MD5 の仕様の一部に変更を加えたハッシュ関数に対して効率的に衝突を探索する攻撃が示された。MD5 自身については衝突が発見されたわけではないが、計算機能力の向上に加えて、攻撃方法の向上による攻撃の実行可能性に留意する必要があった。

また、実際のシステムにおいてはオリジナルの MD5 の仕様に変更を加えて使用するケースも想定される。例えば、MD5 への入力の一部となる初期値を仕様とは異なる値に変更したうえで使用するといったケースである。こうした場合には、運用上の弱点からの攻撃の実行可能性にも留意する必要がある。

なお、1996 年の攻撃方法の提案を行ったドバーティン（Dobbertin）は、「MD5 の衝突は近いうちに発見され、多くのアプリケーションにおいて MD5 が利用できなくなる日はそう遠くない」と発言しているが、MD5 の衝突発見が発表されるまでに約 8 年を要した。その背景には、1997 年に NIST が AES の選定を発表し、AES の候補アルゴリズムの開発や評価が学界の主要な関心事になった結果、MD5 等のハッシュ関数の研究が下火になっていたという事情があるとみられる。

ニ．状態３（「学術的に解読された」状態に至っている）

2004 年に、MD5 の衝突を効率的に探索する攻撃がワン（Wang）らによって示された。MD5 を利用しているシステムのうち、衝突を見つけることが困難であることを利用したシステム（例えばデジタル署名に MD5 を利用している場合）への攻撃方法もいくつか明らかとなり、標準化機関や各国の組織において MD5 の利用に関する注意喚起が始まった。このように、計算機能力や攻撃方法の向上に加え、運用上の弱点からの攻撃の実行可能性に留意する必要があった。

ホ．状態４（実システムへの攻撃方法が提案されている）

2005 年以降、MD5 への攻撃方法の改良が進み、MD5 を利用するさまざまなアプリケーションやシステムの脆弱性が明らかにされた。2005 年には、MD5 を利用した公開鍵証明書（X.509 準拠）の偽造の一例が論文によって示されたほか、2008 年には、約 200 台のプレイステーション 3 によって約 1 日で一部の SSL 証明書が偽造可能であるとの口頭発表が行われた。これらの攻撃方法が適用可能な証明書は一部に限定されるとの見方もあるが、ベリサイン社が 2008 年の口頭発表を受けて 2009 年初に MD5 の発行停止を発表するなど、証明書発行サービス業者側における対応の

動きもみられている。また、2007 年には、多くの市販のメール・ソフトウエアに組み込まれている電子メール受信用のパスワード認証方式 APOP（パスワードの暗号化に MD5 を利用）において、MD5 の衝突探索によって同パスワードを解読する現実的な攻撃方法が提案され、情報処理推進機構セキュリティセンターは 2007 年 4 月に注意喚起の発表を行った。最近では、MD5 によって暗号化されたパスワードをオンラインで復元するというインターネット上のサイト（http://passcracking.com/ 等）が登場しているほか、2009 年には、MD5 の原像を効率的に探索する攻撃も発表されており、MD5 の安全性低下に一層の拍車がかかっているのが実情である。

これら一連の流れからもわかるように、本状態においては、衝突探索に関する計算機能力がもはや問題でなくなっており、運用上の弱点からの攻撃の実行可能性に留意することが必須となっているといえる。

（2）RSA の事例

イ．安全性評価の流れ

RSA は、1977 年にリベスト、シャミア、エイドルマンの 3 人によって提案された公開鍵暗号であり、データを暗号化して秘匿したり、真正性を保証したりする目的として一般的に利用されている。RSA は、使用する公開鍵が本当に正しい相手のものであるかを確認するための公開鍵基盤（Public Key Infrastructure）において広く一般的に利用されており、わが国のインターネット・バンキング・サービスにおけるサーバーの認証等にも広く利用されている。

RSA の具体的な方式としてさまざまな提案が行われており、攻撃方法についても多種多様である。前述のインターネット・バンキングの例に焦点を当て、相互認証用の公開鍵暗号（デジタル署名）として利用される場合がある RSASSA-PKCS-v1_5 という署名方式（RSA 社の仕様書 PKCS #1 において記述されている方式）と、セッション鍵の配送用の公開鍵暗号（守秘）として利用される場合がある RSAES-PKCS-v1_5 という守秘方式（PKCS #1 において記述されている方式）を検討対象とする。これらの方式は 1991 年に提案されているが、同年以降に提案された主な攻撃方法は次のとおりである。

- 1996 年：レンストラらによって、RSA 社の素因数分解コンテスト RSA Factoring Challenge において素因数分解の対象であった 10 進数で 130 桁（430 ビット）の合成数が「一般数体ふるい法」と呼ばれる手法で分解された。
- 1996 年：Coppersmith, Franklin, Pararin, and Reiter [1996] によって、サイズの小さい公開指数[20]を選択すると、暗号文の間に既知の多項式で表される関連性が存在する場合、平文が解読される可能性が示された。ただし、本攻撃は、

20 RSA の公開鍵は剰余演算の法（N で表される）とベキ乗に用いられる指数（e で表される）によって構成され、公開指数はベキ乗に用いられる指数 e を意味する。

平文にパディングを追加するなどの加工を暗号化前に何も施さない方式（「教科書的 RSA」と呼ぶ）を対象としている[21]。

- 1998 年：Bleichenbacher [1998] によって、攻撃者が自分で選択した暗号文を利用することで別の暗号文を解読するという攻撃が RSAES-PKCS-v1_5 に対して提案された。SSL 3.0 を利用したクライアントとサーバー間の通信において、サーバーに対して暗号文を送信し、その暗号文に対するサーバーの反応を手掛かりにして別の暗号文を解読するという方法が示された。
- 2004 年：Wang, Feng, Lai, and Yu [2004] により、RSASSA-PKCS1-v1_5 のハッシュ関数として広く利用されている MD5 に衝突が発見された。
- 2005 年：クラインユング（Kleinjung）らによって、10 進数 200 桁（663 ビット）の特定の合成数が一般数体ふるい法を用いて分解された。

RSA は公開鍵を素因数分解することで秘密鍵が解読されてしまうため、その安全性は素因数分解アルゴリズムの効率に関係している。RSASSA-PKCS1-v1_5 の場合、ハッシュ関数を用いて平文のハッシュ値を計算するため、デジタル署名としての安全性はハッシュ関数の安全性にも影響される。例えば、ハッシュ関数として MD5 を利用する場合は、補論(1)において整理した MD5 の安全性評価の状態に従うことになる。以下では、ハッシュ関数に脆弱性が存在しない場合を前提に、各方式の安全性評価結果を整理する。状態 1～4 のいずれに対応するかを考えると、以下のとおりである。

RSAES-PKCS1-v1_5（守秘方式）の場合：

【状態 1】1991 年（RSAES-PKCS1-v1_5 提案時）～1996 年頃
【状態 2】1996 年（Coppersmith による攻撃方法の提案）～1997 年頃
【状態 3】（なし）
【状態 4】1998 年（Bleichenbacher による攻撃方法の提案）以降

RSASSA-PKCS1-v1_5（署名方式）の場合：

【状態 1】1991 年（RSASSA-PKCS1-v1_5 提案時）以降

ロ．状態 1（暗号アルゴリズムの欠陥を用いた攻撃方法の提案はない）

MD5 や SHA-1 の場合にみられるような使用するハッシュ関数の脆弱性（2004～2005 年）を無視すれば、RSASSA-PKCS1-v1_5（署名方式）の安全性を脅かすほどの攻撃方法は現在まで発見されていない。RSAES-PKCS1-v1_5（守秘方式）においては、1991 年から 1995 年頃までは特段新しい攻撃方法が提案されていなかった。このように、計算機能力の向上による攻撃の実行可能性に留意する必要があった。

21 教科書的 RSA やそれを単純に拡張した方式には、積攻撃等に類する乗法性を利用した攻撃（齊藤［2002］）を受ける可能性があるため、一般に利用される暗号方式としては適さない。

ハ．状態 2（「学術的に解読された」状態には至っていない）

1996 年に Coppersmith, Franklin, Pararin, and Reiter [1996] によって新たな攻撃が提案され、具体的なアプリケーションへの適用可能性は示されていないものの、RSAES-PKCS1-v1_5 に対しても理論上適用可能と推定される。このように、計算機能力の向上に加え、攻撃方法の向上による攻撃の実行可能性に留意する必要があった。

ニ．状態 4（実システムへの攻撃方法が提案されている）

1998 年に、Bleichenbacher [1998] によって、RSAES-PKCS1-v1_5 をクライアント・サーバー間での暗号通信等において一定の形態で実装した場合、攻撃者が巧みに選択した暗号文をサーバーに送信してサーバーの反応を観察することで別の暗号文を効率的に解読するという方法が提案された。こうした実装上の処理のタイミングやバグによって本攻撃方法が実現可能になることを適切な運用によって防止することが必要である。このように、RSAES-PKCS1-v1_5 に関しては、1998 年頃から現在に至るまで状態 4 となっており、状態 3 の時期はほとんど存在しなかったといえる。計算機能力および攻撃方法の向上に加えて、運用上の弱点による攻撃の実行可能性に留意することが必須となっているといえる。

ホ．素因数分解のこれまでの記録

RSA の安全性の源泉となっている素因数分解については、1990 年代後半以降の計算機性能の向上によって、一般数体ふるい法による具体的な素因数分解実験が活発に行われて一般に公表されるようになった（図表 A-1 参照）。

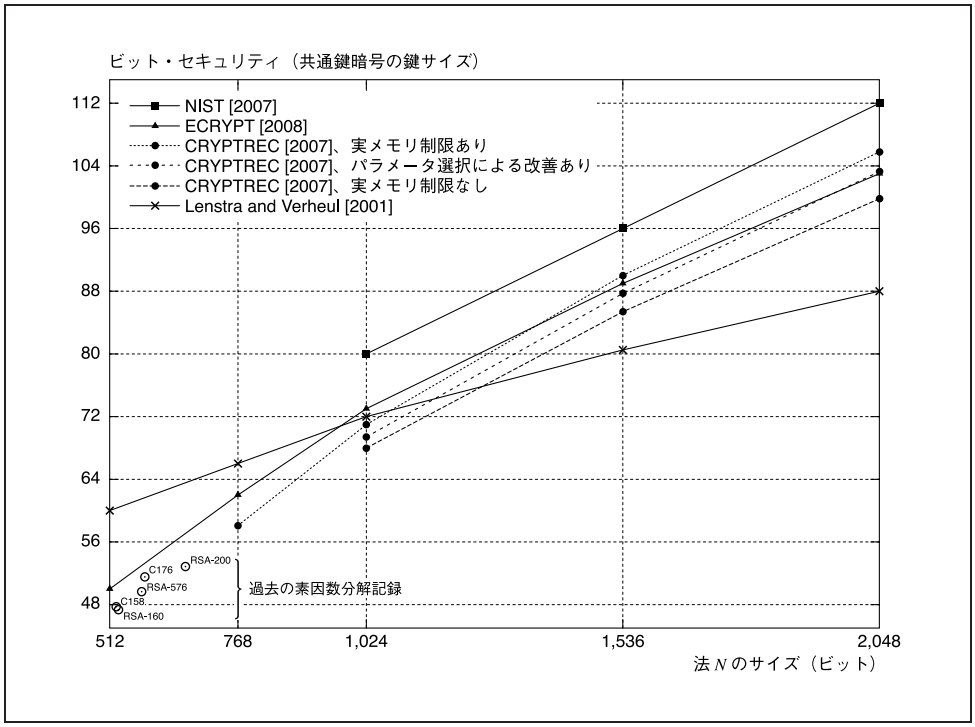
RSA の法のサイズをどのように設定すれば安全と考えられるかについては、「素因数分解の実行に必要な資金と時間」に基づいて検討することが考えられる。その際に、一般数体ふるい法の実験結果等から先行きの予測を行った（図表 A-2 参照）うえで、法 N のサイズ（図表 A-2 の横軸）から共通鍵暗号の鍵のサイズ（「ビット・

図表 A-1　一般数体ふるい法による近年の素因数分解の記録

合成数	法 N のサイズ	分解が完了した時期
RSA-200	200 桁（663 ビット）	2005 年 9 月
RSA-640	193 桁（640 ビット）	2005 年 11 月
$11^{281}+1$ の約数	176 桁（582 ビット）	2005 年 4 月
RSA-576	174 桁（576 ビット）	2003 年 12 月
$2^{1826}+1$ の約数	164 桁（545 ビット）	2003 年 12 月
RSA-160	160 桁（530 ビット）	2004 年 4 月
$2^{953}+1$	158 桁（524 ビット）	2002 年 1 月
RSA-155	155 桁（512 ビット）	1999 年 8 月
RSA-140	140 桁（463 ビット）	1999 年 2 月
RSA-130	130 桁（430 ビット）	1996 年 3 月

資料：http://www.crypto-world.com/FactorWorld.html

図表 A-2　一般数体ふるい法での計算量において等価な法 N のサイズと共通鍵暗号の鍵サイズの推定



備考：図表中の破線は、一般数体ふるい法に関する異なる実装方法によって処理効率に違いが生じたことを意味する。詳しくは CRYPTREC [2007] を参照されたい。

セキュリティ」と呼ばれる、図表 A-2 の縦軸）への「変換」を行い、3 節において検討した計算量を資金と時間に翻訳する方法を用いて「素因数分解の実行に必要な資金と時間」を試算することが考えられる。

図表 A-2 から、例えば、法 N のサイズが 1,024 ビットの場合は共通鍵暗号の鍵サイズの約 70 ビット前後であると推定される。逆に、共通鍵暗号の鍵サイズで 100 ビット相当の安全性を確保するには、法 N のサイズとして約 2,048 ビットは確保しなければならない。2,048 ビットよりも大きい法 N のサイズがどの程度の安全性を与えているかは ECRYPT [2008] において試算結果が示されている。