■11 群（社会情報システム）－ 7 編（金融情報システム）

5 章 IC カードに利用される暗号アルゴリズムの安全性について

（執筆者：鈴木雅貴）［2009 年 3 月 受領］

■概要■

我が国の金融業界では，IC カード型のクレジットカードやキャッシュカードが普及し始めている．IC カードは，その演算・判断・記憶の機能を活用することにより，従来の磁気ストライプ型カードに比べ，セキュリティ機能の向上が図られている．例えば，IC カードでは，カード内部のファイルへのアクセス制御を行うことによって内部データを不正なアクセスから保護するなど，磁気ストライプ型カードに比べて格段に高いセキュリティが確保されているといわれている．こうした IC カード・端末の技術的な要件や通信プロトコルを定めた標準としては，国際的なデファクト標準である EMV 仕様が広く用いられている．我が国では，銀行系カード会社で構成される日本クレジットカード協会（JCCA）が策定した「IC カード対応端末機能仕様書（JCCA 仕様）」や，全国銀行協会が策定した「全銀協 IC キャッシュカード標準仕様（全銀協仕様）」が EMV 仕様に基づいて策定されている．

EMV 仕様は，金融分野で用いられる IC カードが備えるべき共通事項を定めた標準として広く普及しているが，詳細な技術要件については，EMV 仕様の範囲内で，各利用主体が各々のニーズに基づいて独自に設定し，システムに実装することが可能である．このため，技術要件の内容やシステムへの実装の仕方によっては，セキュリティ機能の水準に差異が生じる可能性も否定できない．とりわけ暗号技術については，EMV 仕様で推奨されている暗号アルゴリズムのなかに様々な特性や強度をもつものがあり，システムに実装する場合に，暗号アルゴリズムの選択によって当該システムのセキュリティに差異が生じる可能性がある．

そこで，本章では，鈴木らの論文 [1)]を参考に，最新版の EMV 仕様である EMV 4.2[2)]について，セキュリティ機能を中心に紹介するとともに，これを実装する場合の安全性について，主として暗号アルゴリズムに着目しながら考察を行う．

【本章の構成】

5-1 節では，EMV 仕様の概要とそのセキュリティ機能を簡単に紹介する．そのうえで，5-2 節では，EMV 仕様で利用される暗号アルゴリズムの安全性について述べる．

■11 群 － 7 編 － 5 章

5-1 EMV 仕様

（執筆者：鈴木雅貴）［2009 年 3 月 受領］

5-1-1 EMV 仕様の概要

EMV 仕様は，金融分野における IC カード取引のための IC カードと端末に関するデファクト標準として，国際的に広く利用されている．EMV 仕様の策定の経緯をみると，国際クレジットカード・ブランドである Europay International，MasterCard International，Visa International の 3 社が，IC カードと端末の互換性を確保するために必要な機能要件とカードの不正使用を防ぐためのセキュリティ要件に関する検討を 1993 年に開始し，その検討結果を踏まえて 1996 年に EMV 3.0 として公開され，その後，継続的に改訂が行われてきている．

EMV 仕様は，一般的な外部端子付 IC カードの物理的・機能的条件などを規定した国際規格 ISO/IEC 7816 シリーズに準拠しつつ，金融分野向けに必要な IC カードと端末の仕様を記述している．具体的には，①IC カードと端末の物理的・電気的特性及びハードウェア・インタフェース，②IC カード・端末間でやり取りするデータ要素及びコマンド，③IC カード・端末間の処理フローなどを定義している．JCCA 仕様や全銀協仕様は，この EMV 仕様に基づいて策定されている．また，国際クレジットカード・ブランドは，EMV 仕様に準拠したかたちでより詳細な技術要件を定めており，これらをもとに，傘下のカード発行会社は各々の提供するサービスやリスク管理方針に応じて，システムへの実装を行っている（**図 5･1**）．このように，EMV 仕様は，各利用主体がより詳細な技術要件を定め，システムに実装することを可能にする拡張性を備えている．

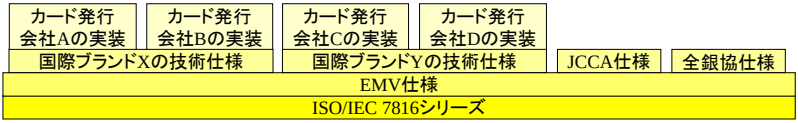


図 5・1　EMV 仕様に関連する国際標準や技術要件との関係

5-1-2 セキュリティ機能

IC カードの特徴として，偽造や不正使用を防止するため，従来の磁気ストライプ型カードに比べて高度なセキュリティ機能を備えていることがあげられる．EMV 仕様では，例えば，暗号技術を応用したセキュリティ機能として，データ認証，カード所持者認証，AC（Application Cryptogram）生成を用意している．これらのセキュリティ機能を IC カードの利用のシナリオに沿って説明すれば以下のとおりである．

まず利用者が端末にカードを提示すると，端末はカードやそのデータが真正であることを確認するために，「データ認証」を行う．次に，発行者のホスト・システムまたは端末が，カードを提示した利用者がカードの真の所持者であることを「カード所持者認証」によって確認する．これらの確認を経たうえで，取引データ（取引金額，取引日時など）やそのほかの条件に基づいて，当該取引を承認するか否かを決定する．その際，承認の判断に利用される取引データが改ざんされていないことや，当該取引のためにカードが利用されていることを

保証するために，取引ごとに「AC 生成」を行うという仕組みになっている．

これらの機能を実現する処理方法は複数あり，発行者や端末を管理する店舗などがそれぞれリスクを考慮して選択可能である．各機能の具体的な内容は以下のとおりである．

(1) データ認証

EMV 仕様では，データ認証方法として，①SDA（Static Data Authentication），②DDA（Dynamic Data Authentication），③CDA（Combined DDA / Application cryptogram generation）の 3 種類が用意されている．SDA は，カード発行時に，発行者が口座番号（PAN: Primary Account Number）・有効期限・名前などの情報（静的認証データ）とこれに対応する発行者のディジタル署名をカードに登録する．その後，カード利用時に，カードに対応する公開鍵を用いて端末がディジタル署名を検証し，カードに登録された情報が改ざんされていないことを確認する．

SDA では，ディジタル署名の検証時に，カードからつねに同じデータが送信される．これに対し，秘密の情報をもつ真正なカードだけが生成可能な取引ごとに異なるデータを検証時に利用することで，より安全性の高い認証を行う方法が DDA である．具体的には，カード発行時に，発行者がカードに固有の秘密鍵を登録する．その後，カード利用時に，端末がカードに乱数などを送ると，カードが秘密鍵を用いてディジタル署名を生成し，これを秘密鍵に対応する公開鍵とともに送り返す．端末は，このディジタル署名と公開鍵を検証することにより，カードが秘密の情報をもった真正なものであることを確認するという仕組みである．CDA は，DDA と後述する AC 生成をあわせて行うことにより，カード・端末間の通信回数を削減することを企図した認証方法である．これらの 3 種類のデータ認証において利用される公開鍵の正当性を保証するために，公開鍵証明書が用いられる．

(2) カード所持者認証

EMV 仕様では，現在，カード所持者認証の方法として，①オフライン PIN 認証，②オンライン PIN 認証，③手書き署名などが用意されている．オフライン PIN 認証では，カード発行時に，利用者（または発行者）が決めた暗証番号（PIN）をカード内に登録しておく[1]．カード利用時に，利用者が PIN パッドから入力した PIN を端末がカードに送る．カードは内部で PIN の照合を行い，その結果を端末に返すという流れである．オンライン PIN 認証では，利用者が PIN パッドに入力した PIN を用いて，端末とホスト・システムとの間で認証処理が行われるため，EMV 仕様とは別にその詳細が定められている．また，手書き署名についても，EMV 仕様のほか，カードによる照合処理について別の仕様で規定されている．

(3) AC 生成

AC は，取引データが改ざんされていないことと，当該取引のためにカードが利用されていることを検証する手段として，取引ごとに生成される．まず，あらかじめカードに登録された「マスタ鍵」とカード自身が管理する取引カウンタ（ATC: Application Transaction Counter）のデータを基に，取引ごとに「セッション鍵」が生成される．この「セッション鍵」と端末から送信されてきた取引データ（取引金額，取引日時など）を基に，AC が生成されるという仕組みである．なお，マスタ鍵は，発行者が管理する「システム鍵」からカードごとに生成される．各鍵は，システム鍵をルートとするツリー構造となっている．

1 カード発行後に，利用者が端末などで PIN の変更を可能とする運用もある．

■11 群 － 7 編 － 5 章

5-2 EMV 仕様で利用される暗号アルゴリズムの安全性

（執筆者：鈴木雅貴）［2009 年 3 月 受領］

EMV 仕様では，各セキュリティ機能を提供するうえで推奨される暗号アルゴリズムを列挙している．これらの暗号アルゴリズムについては，定期的な見直しが行われているものの，最初の EMV 仕様が策定・公開されてから既に 10 年以上が経過しており，現在の暗号技術の水準からみれば，必ずしも安全とは言い切れないものも含まれている．これは，EMV 仕様の問題というよりも，むしろ EMV 仕様のもとで各システムに応じた暗号アルゴリズムを選択する際のシステム設計上・運用上の問題であり，そうした問題を意識しておくことが EMV 仕様を利用していくうえで重要であると考えられる．以下では，EMV 仕様で推奨されている暗号アルゴリズムなどに着目して，その安全性について若干の検討を行う．

5-2-1 公開鍵暗号方式に対して想定される攻撃

(1) RSA 署名と 2 種類の攻撃

EMV 仕様では，データ認証において公開鍵暗号方式の暗号アルゴリズムである RSA 署名が使用されている．すなわち，公開鍵の正当性を保証するための公開鍵証明書，SDA における静的認証データに対するディジタル署名，DDA における乱数などに対するディジタル署名に RSA 署名が用いられている．仮に，RSA 署名を偽造可能な攻撃者が存在した場合には，公開鍵証明書や，静的認証データ，乱数などに対する署名の偽造が行えるため，データ認証において，真正であると誤認させることができる IC カードの作製が可能になると考えられる．

RSA 署名は，署名生成に用いる秘密鍵と署名検証に用いる公開鍵が異なり，一方の鍵から他方の鍵を求めることが計算量的に困難であるため，公開鍵を公開することができる．RSA 署名における鍵生成は，①二つの大きな素数 p と q を選び，これらの積 $n = p \cdot q$ を計算する，②p-1 と q-1 の最小公倍数 L を算出する，③L と互いに素な自然数 e を選ぶ，④$e \cdot d = 1 \pmod{L}$ を満たす d を求めるという手順で行われる．この結果，得られた d を秘密鍵，(e, n)を公開鍵とする．署名生成者はメッセージ m と自分の秘密鍵 d を用いて $s = m^d \bmod n$ を計算し，m に対するディジタル署名 s を生成する．ディジタル署名 s とメッセージ m を受け取った署名検証者は，署名生成者の公開鍵(e, n)を用いて $m = s^e \bmod n$ が成立するかどうかを検証することによって，①ディジタル署名 s の署名生成者が公開鍵(e, n)に対応する秘密鍵の持主であることと，②m が改ざんされていないことを確認する．

RSA 署名に対する署名の偽造方法は，(a)秘密鍵を求めることなく署名を偽造する攻撃（署名偽造攻撃）と，(b)RSA 署名が安全性の根拠としている素因数分解問題[2]を解くことで，公開鍵から秘密鍵を求め，署名を偽造する攻撃（素因数分解攻撃）に分類できる．それぞれの攻撃についてみていく[3]．

(a) 署名偽造攻撃

上述のとおりに署名を生成した場合，「(mod n において）メッセージ m_1 とメッセージ m_2 の積で表されるメッセージ $m_1 \cdot m_2$ に対する署名と，m_1 に対する署名と m_2 に対する署名の積

2 素因数分解問題：自然数 n が与えられたとき，$n = p \cdot q$ を満たす相異なる素数 p, q を求めるという問題．
3 RSA 署名に関する研究動向としては，宇根 [3]，齋藤 [4]が参考になる．

が等しい」という性質（乗法性）がある．このため，攻撃者が署名を偽造したいメッセージを x，署名生成者から正当な署名が得られたメッセージを m とすると，x と m の積 $x \cdot m \pmod n$ に対する署名から x の署名を偽造する攻撃（積攻撃）が可能であることが知られている．

こうした攻撃への耐性をもたせるために，メッセージそのものではなく，パディング，ハッシュ値，乱数などを用いてメッセージを変換したデータ（署名変換データ）を RSA 署名への入力とした署名方式が提案されるようになった．しかし，経験に頼ってメッセージの変換方法が設計された署名方式は，署名偽造が可能であることが指摘されているものが多いほか，指摘されていないものについても今後署名偽造攻撃が提案されないとはいい切れない．例えば，EMV 4.0[5]が採用していた ISO/IEC 9796-2:1997 の署名方式についても，積攻撃を高度化した攻撃によって署名偽造が可能であることがコロンら [6]によって指摘されている．

次に，RSA 署名の代表的な実装形態の一つである PKCS#1 v1.5 署名方式について，近年ブライヘンバッハーによって新たに指摘された署名偽造攻撃 [7]に突いて見ていく．PKCS#1 v1.5 署名方式は，長年利用され続けても素因数分解問題を解くよりも効率的な攻撃が発見されていないという実績があり，セキュリティ機能をもつ電子メール用のプロトコル（S/MIME）などに広く利用されている．しかし，この署名方式において，公開指数 e が小さく（$e = 3$），検証処理が不適切な場合に，署名を偽造できる可能性があることが示された [7]．

本署名方式は，ヘッダやパディングなどからなるビット列とメッセージのハッシュ値を連結して生成したビット列を署名変換データとしている．通常，検証時には，署名変換データの長さをチェックし，ハッシュ値が最後にあることを確認する必要がある．この攻撃では，何らかの理由により不適切な実装が行われ，署名変換データの左端から各ビットを検査し，ハッシュ値を抽出したところで，ハッシュ値の検査に移り，ハッシュ値が署名変換データの最後にあるのか（つまり，ハッシュ値より右に余分なデータが存在しないか）を検査しない場合において，署名を偽造できるケースがあることを指摘している．具体的には，署名変換データのパディングを短くし，その分をハッシュ値の右側に任意の値を取れる余分なデータ（garbage）として加えた署名変換データを用いても，garbage をチェックされることがないことから，garbage をうまく設定することで署名の偽造が行える可能性があることを示している．公開指数が $e = 3$ の場合には，署名を偽造したいメッセージを平均 3 回程度修正すれば，攻撃者は都合の良いメッセージを得ることができることが報告されている．

EMV 4.2 においても，公開指数 e として $e = 3$（または，$2^{16} + 1$）を用いている．また，EMV 4.2 では，公開鍵証明書の検証手順において，パディングの値と長さを検査することは規定されていない．そのため，パディングを検査していない実装も少なくないと考えられる．この場合，攻撃者が故意に鍵長の短い公開鍵を発行者の公開鍵として設定することで，パディングのビット列を長く設定し，任意の値をパディングに設定できる可能性がある．

EMV 仕様の公開鍵証明書における署名変換データは，ブライヘンバッハーの攻撃が対象とする署名変換データとは若干異なるが，本攻撃のアイデアを利用した類似の攻撃を受けることがないか確認しておく必要がある．仮に，こうした攻撃が考えられる場合，実装において検証時にパディング領域の検査を行うなどの対策を講じていくことが一案であろう．

なお，1993 年に，暗号アルゴリズムが安全性の根拠としている数学的問題を解く難しさと，暗号アルゴリズムの安全性が等価であること，言い換えると，署名偽造攻撃が存在しないことを数学的に証明可能とする安全性の考え方（証明可能安全性）が提案された [8]．この考え

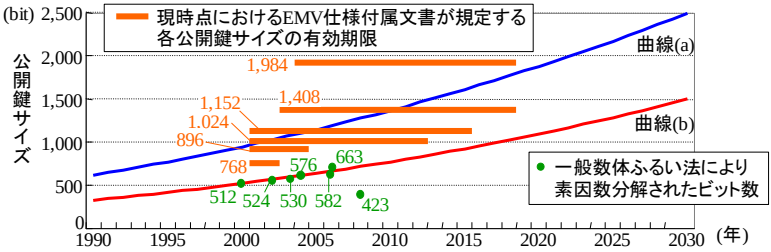
方に基づく RSA 署名である RSA-PSS 署名方式が提案されており，ISO/IEC 14888-2 として国際標準化されている．

(b)　素因数分解攻撃

素因数分解問題を解く困難さは，合成数 *n* の大きさ（公開鍵のサイズ）に依存しており，現在知られている素因数分解アルゴリズムを利用した素因数分解を実際に行うことで，どの程度の大きさの合成数であれば現実的な時間で素因数分解可能であるかによって評価されている．最新（2005 年 5 月）の実験結果によると，663 ビットの合成数（公開鍵）を実際に素因数分解できたことが報告されている．

素因数分解可能な合成数の大きさについては，様々な予測が行われている．**図 5･2** の曲線(b)は，実際に素因数分解された最大の値を出発点とし，解読技術は現在のままで，コンピュータの性能向上がムーアの法則[4]に従うことを仮定した場合に素因数分解が可能と考えられる公開鍵サイズを示している [9]．**図 5･2** の曲線(a)は，過去の経験則に基づき，コンピュータの性能だけでなく素因数分解アルゴリズムの効率も 1.5 年で 2 倍になるとの仮定のもとに，1982 年当時の DES と同程度の安全性をもつと考えられる公開鍵サイズを予測している [10]．

EMVCo では，認証機関の公開鍵サイズに関する見直しを毎年行っている．**図 5･2** において，EMV 仕様付属文書（Bulletins Notices）が規定する各公開鍵サイズの有効期限と曲線(a)(b)を比較すると，EMV 仕様は，曲線(b)より(a)に近いセキュリティ・マージンを確保したうえで設定しようとしていることがわかる．



（備考）公開鍵サイズの有効期限については，EMVCo のウェブサイト（http://www.emvco.com/）で公表されている仕様付属文書を参照した．

図 5･2　RSA の公開鍵サイズに関する予測

5-2-2 共通鍵暗号方式に対して想定される攻撃

(1)　EMV 仕様で利用される共通鍵暗号

EMV 仕様では，発行者のシステム鍵から各 IC カードのマスタ鍵を生成する段階（マスタ鍵生成）と，各 IC カードがマスタ鍵から取引ごとにセッション鍵を生成する段階（セッション鍵生成）において，2-key トリプル DES を用いた例が示されている．また，取引時に生成する AC には，2-key トリプル DES をベースにした MAC 生成方式と，DES をベースにした MAC 生成方式が推奨されている．AC 生成に至るプロセスでは，システム鍵，マスタ鍵，セ

4 ムーアの法則：半導体の集積密度が 1.5～2 年で倍増するという予測．これに基づき，一定金額で購入可能なコンピュータの処理速度が 1.5～2 年で 2 倍になることを予測している．

ッション鍵という，上位鍵から下位鍵の生成の連鎖を経ている．このため，AC あるいは下位鍵から上位鍵を特定できるような場合は，こうした生成の連鎖を経て，最終的に別の AC が生成される可能性がある．これにより，端末から送られてきた取引データに対する AC を偽造することや，悪意のある端末管理者が取引データを改ざんし（例えば，取引金額を増やすなど），それに対する AC を偽造することなどが考えられる．

(2) 具体的な攻撃とそれへの対策

以下では，マスタ鍵生成に使用した上位鍵を特定する攻撃と，EMV 仕様が推奨する MAC 生成方式に対してセッション鍵を求める攻撃について考察する．

(a) マスタ鍵生成方式に対する攻撃

最新の EMV 仕様では，各 IC カードのマスタ鍵の生成方法として，2-key トリプル DES と発行者のシステム鍵によりユーザの口座番号（PAN）などを暗号化したビット列を用いるという方法を例示している [2]．2-key トリプル DES における平文と暗号文は，マスタ鍵生成における PAN などの入力と，出力であるマスタ鍵にそれぞれ対応している．この入出力のペアが何らかの方法で入手できたとすると，2-key トリプル DES に対する既知平文攻撃が可能となる．この既知平文攻撃については，攻撃者が N 組の既知平文を入手した場合，2 の（$120 - \log_2 N$）乗のオーダーの計算量で鍵を特定できる方法が知られている [10]．この方法は，例えば，2^{80} 程度の計算能力をもつ攻撃者を仮定した場合，2^{40} 程度の平文と暗号文のペアが入手できれば，システム全体で利用する秘密鍵が解読できてしまうことを意味している．実際には，2^{40} 程度の平文と暗号文のペアを入手することは困難と考えられるが，より効率的な 2-key トリプル DES の解読法が発見される，あるいは，PAN などとマスタ鍵のペアが大量に入手できる条件が成立した場合には，システム全体の安全性が著しく低下するおそれがある．2-key トリプル DES を利用するのではなく，AES や Camellia などのより安全な暗号アルゴリズムを採用した場合は，上記の攻撃を計算量的に防ぐことができる．システムの性能要件や運用要件などの許容範囲内であるならば，より安全な方式に移行することが望ましいといえる．

(b) メッセージ認証子生成方式に対する攻撃

EMV 4.2 では，AC 生成に利用するメッセージ認証子生成方式として，2 種類のアルゴリズムを推奨している（**図 5・3**）．第 1 は，2-key トリプル DES を用いた CBC-MAC（TDES-MAC）である．第 2 は，DES を用いた CBC-MAC の最後に DES の処理を 2 回加えることにより，最終処理が 2-key トリプル DES の形式を備えているアルゴリズムである（擬似 TDES-MAC と呼ぶ）[5]．この方式は，IC カードの計算能力が低い場合に配慮して用意された方式と考えられる．

擬似 TDES-MAC に対しては，ある条件のもとで，DES と同程度の計算量で鍵の特定が可能となる攻撃が考えられる．擬似 TDES-MAC のメッセージ認証子の長さが 64 ビットであることから，ランダムに選んだ 2^{32} 個のメッセージに対するメッセージ認証子を入手すると，メッセージ認証子の値が同一となる異なるメッセージのペアを得ることができる[6]．更に，このときのメッセージ認証子と当該ペアを用いて DES の鍵の全数探索（計算量 2^{56}）を 2 回行

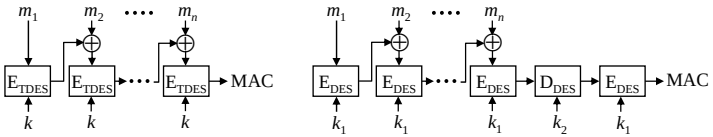
5 擬似 TDES-MAC は，ISO/IEC 9797-1 で規定されている「アルゴリズム 3」をベースとし，ブロック暗号として DES を適用した方式である．

6 バースディ・パラドックスの原理に基づいている．ランダムに 23 人集めると 1/2 以上の確率で同じ誕生日の人が存在することから，バースディ・パラドックスの原理と呼ばれている．

うことで，暗号鍵 k_1, k_2 を推定することができる．

この攻撃により，擬似 TDES-MAC は 112 ビットの鍵を利用しているにもかかわらず，2^{32} 個のメッセージとメッセージ認証子のペアを集められるならば，DES と同程度の計算量，すなわち，2^{57} (= $2^{32} + 2^{56} + 2^{56}$)で 112 ビットの鍵を推定可能である．EMV 仕様が推奨している擬似 TDES-MAC は，一定条件のもとでは DES と同程度の安全性しかもたないことが分かる．

擬似 TDES-MAC への攻撃に必要となるメッセージとメッセージ認証子のペアは，AC 生成における取引データと AC のペアに対応する．仮に，セッション鍵がつねに固定であり，かつ，IC カードに何度も取引データに対する AC を生成させられる場合には，攻撃に必要な数のペアを攻撃者が集められる可能性があるため，実装の際には注意が必要である．ただし，TDES-MAC や，AES や Camellia などの 128 ビット・ブロック暗号を用いた CBC-MAC などを利用すれば，この攻撃自体を計算量的に防ぐことができるため，本質的には擬似 TDES-MAC を利用しないことが望ましいといえるであろう．



（備考）$m_1, m_2, \ldots, m_n$ はメッセージ，k, k_1, k_2 は暗号鍵，E_{TDES}，E_{DES} はそれぞれ 2-key トリプル DES と DES の暗号化関数，D_{DES} は DES の復号関数．

図 5・3　TDES-MAC と擬似 TDES-MAC

本章では，EMV 仕様を実装する場合の安全性について，主として暗号アルゴリズムの観点から考察してきた．暗号アルゴリズムの安全性については，いわゆる 2010 年問題のような鍵長の観点に加え，署名変換データやメッセージ認証子の生成方法などについても考慮する必要があることがわかる．暗号アルゴリズムの安全性低下などの理由から，システムにいったん実装した暗号アルゴリズムを別のものに移行する場合には，そのシステム規模が大きければ，多大なコスト負担を要することとなる．こうしたコスト負担を可能な限り回避するという意味において，システムに実装する暗号アルゴリズムの選択はシステム投資判断の重要な要素である．暗号アルゴリズムの選択に当たっては，当該システムの使用期間を考慮しつつ，使用期間満了までの安全性が確認されている暗号アルゴリズムのなかから，十分な安全性を有するものを選択することが望ましいといえる．

もっとも，システム構築段階で適切な暗号アルゴリズムを選択したとしても，新たな攻撃技術の発見やコンピュータの性能向上により，システムで使用している暗号アルゴリズムの安全性が低下することもありうる．このような場合，安全な暗号アルゴリズムに移行することが本来望ましいが，既存システムとの相互運用性の確保や実装技術上の問題から直ちに移行することが難しい状況も考えられる．そうした場合には，安全性の低下した暗号アルゴリズムへの攻撃の前提条件を充足しないように，システムの運用に制約を設けることが次善の策として考えられよう．例えば，攻撃に必要な暗号文と平文のペアが集まらないよう鍵を定

期的に更新する，特定のパラメータの値を避けて利用するといった運用上の方策が考えられる．これらの方策を講じる場合には，具体的な攻撃を想定した適切な対応が必要となるため，暗号技術の専門的知見を十分に活用することが重要となろう．

■参考文献

1) 鈴木雅貴・神田雅透，“IC カードに利用される暗号アルゴリズムの安全性について：EMV 仕様の実装上の問題点を中心に，”金融研究，vol.26，no.s1，pp.31-51，日本銀行金融研究所，2007.
2) EMVCo，“EMV Integrated Circuite Card Specification for Payment Systems: Book 2 – Security and Key Management Version 4.2，”EMVCo，2008.
3) 宇根正志，“RSA 署名に対する新しい攻撃法の提案について－Coron-Naccache-Stern の攻撃法，”金融研究，vol.18，no.s1，pp.51-84，日本銀行金融研究所，1999.
4) 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，vol.21，no.s1，pp.285-324，日本銀行金融研究所，2002.
5) EMVCo, “EMV2000 Integrated Circuite Card Specification for Payment Systems: Book 2 – Security and Key Management,” EMVCo, 2000.
6) J.S. Coron, D. Naccache and P. Stern, “On the Security of RSA Padding,” Proc. CRYPTO ’99, LNCS, Springer-Verlag, vol.1666, pp.1-18, 1999.
7) D. Bleichenbacher, “Forging some RSA signatures with pencil and paper,” CRYPTO 2006 Rump Schedule, 2006.
8) M. Bellare and P. Rogaway, “Random Oracles are Practical: A Paradigm for Designing Efficient Protocols,” Proc. of 1st ACM Conf. on Computer and Communications Security, ACM Press, pp.62-73, 1993.
9) R. Brent, “Recent progress and prospects for integer factorization algorithms,” Proc. COCOON 2000, LNCS, Springer-Verlag, vol.1858, pp.3-20, 2000.
10) A.K. Lenstra and E.R. Verheul, “Selecting Cryptographic Key Sizes,” J. Cryptology, vol.14, no.4, pp.255-293, 2001.
11) P.C. van Oorshot and M.J. Wiener, “A known plaintext attack on two-key triple encryption,” Advances in Cryptology – Proceedings of EUROCRYPT ’90, Springer-Verlag, LNCS, vol.473, pp.318-325, 1990.