第4回 シリーズFRAM

FRAM搭載商品のセキュリティ設計

セキュリティ向けメモリには，パラメタの保存・更新，高速化実装，アプリケーション間のファイアウォール，耐タンパ対応などさまざまな要求があります。本稿では，暗号システムにおけるFRAMの利用事例と，FRAMを用いたセキュリティシステム構築についてご紹介します。

はじめに

近年，ネットワーク上でオープンに通信や電子商取引を行う音楽配信や映像配信，電子情報販売などの各種サービスは，生活をより効率的で豊かにするIT技術として期待されています。これらのサービスを快適に享受できるために不可欠な基盤技術は，暗号技術を使ったセキュリティの確立です。ネットワークを前提としたシステムに結合されるメディアは，どの段階で接続されても，このセキュリティ要求に対応できる機能を搭載する必要があります。

当社では，セキュリティ機能を搭載したICカードやセキュアデバイスにFRAMを適用した設計・開発を行っています。FRAMは，高速書込み・低消費電力・実質無制限の書換え回数・バイト書換え可能という４つの特長を持った不揮発メモリです。このメモリを利用したデバイスでは，鍵の記憶を行うだけの単なる不揮発メモリの用途から，暗号アルゴリズムの高速化を担う機能としての応用まで進展を見せています。

本稿では，暗号システムにおけるFRAMの利用事例と，FRAMを用いた暗号とセキュリティシステムへの取組みについてご紹介します*1~*4。

暗号の分類とアルゴリズム

暗号アルゴリズムには，大別して共通鍵暗号と公開鍵暗号があります。共通鍵暗号は秘密鍵暗号とも呼ばれ，暗号用の鍵と復号用の鍵で共通の鍵を用いる暗号です（**図１**）。この種類の暗号の歴史は古く，米国商務省連邦標準局において1977年に定められたDES（Data Encryption Standard）が，事実上の世界標準として使われ今に至ります。現在は，64ビット鍵長のシングルDESは安全性を確保できないため，トリプルDESの利用が多くなっています。今後は，DESの後継であるAES（Advanced Encryption Standard）のアルゴリズムがRijndaelに決まったため，これに移行していくものと考えられています。

一方，公開鍵暗号は，暗号用の鍵と復号用の鍵が異なる暗号です。暗号通信をするには，送信者が公開鍵を持ち，受信者はプライベート鍵（復号鍵）を持ちます。公開鍵からプライベート鍵を導き出すためには莫大な計算量が必要なため，実用上不可能であるように設計されています。公開鍵暗号では，受信者の公開鍵を知っていれば，送受信者間でプライベート鍵を共有しなくても暗号通信が行えるので，不特定多数との通信に向いているといわれます。主にデジタル署名に適した暗号として広まり，RSA暗号や楕円曲線暗号は，このグループの代表といえます。

図２に署名生成・署名確認の関係を示します。

RSA暗号では安全性の点で1,024ビット鍵長が使われますが，同じ安全性で鍵長が160ビットと短い楕円曲線暗号方式が今後は有望視されています。当社では，特に回路規模が小さく実装でき，安全性の高い２の拡大体をICカード向けやセキュアデバイスに採用しています。

図１　暗号化・復号（秘密鍵暗号）

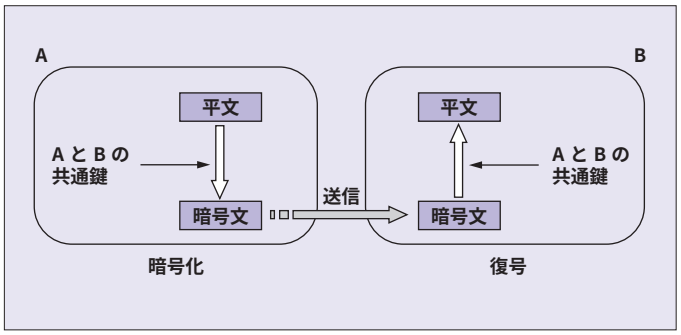
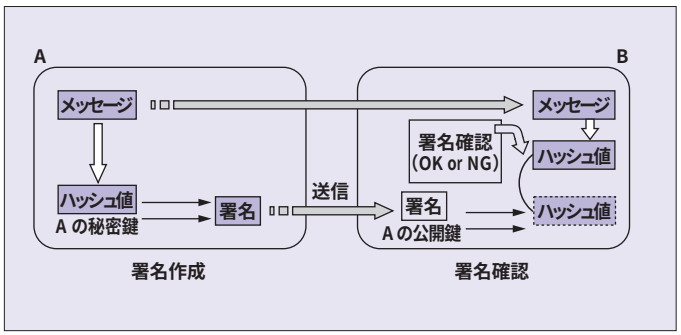


図２　署名生成・署名確認（公開鍵暗号）



セキュリティ向けメモリの課題

暗号セキュリティシステムに求められるメモリの機能は，次のように類別されます。

●パラメタの保存と更新

最近のインターネットにおける運用サービスでは，利用者が既存サービスとは独立した新しいサービスを受けたり，既存サービスをやめる場合などには，高速通信を用いたアプリケーションソフトの消去・再ローディングによりそれらが実施されます。その際には，運営サイトごとに暗号やパラメタ・鍵を使い分けなければなりません。これは，今後のICカードや携帯電話に実装されるSIMカードにおいても必須の要件になると考えられます。限定されたリソースしかないICカードでは，パラメタ管理・変更も重要な要件になっています。

●高速化実装

共通鍵暗号や公開鍵暗号においてデファクトスタンダードとなっている暗号処理の実装は，鍵長やパラメタサイズを制限できないため，アルゴリズム上必要な計算量が固定化されます。そのため高速化には，ソフトウェア記述をプロセッサ言語（アセンブラ言語）で記述し，実行プログラムのサイクル数を削減するソフトウェア手法や，演算処理部のハードマクロ化を行い，ハードマクロとソフトウェアで速度を向上させる方法が一般的に行われます。チップサイズにゆとりのある場合は，すべてをハードマクロ化するのが一番効果的な方法ですが，ICカードなどリソースに制限のある環境では限界があります。そのため，メモリ上に事前計算したテーブルを作っておき，これを検索して結果を得ることで，演算器の実装を軽くする手法が用いられることがあります。この場合テーブルは，不揮発メモリ上に生成します。アプリケーションごとに変更が行われることを考慮すると，書込み速度が速いメモリを選ぶことが重要になります。

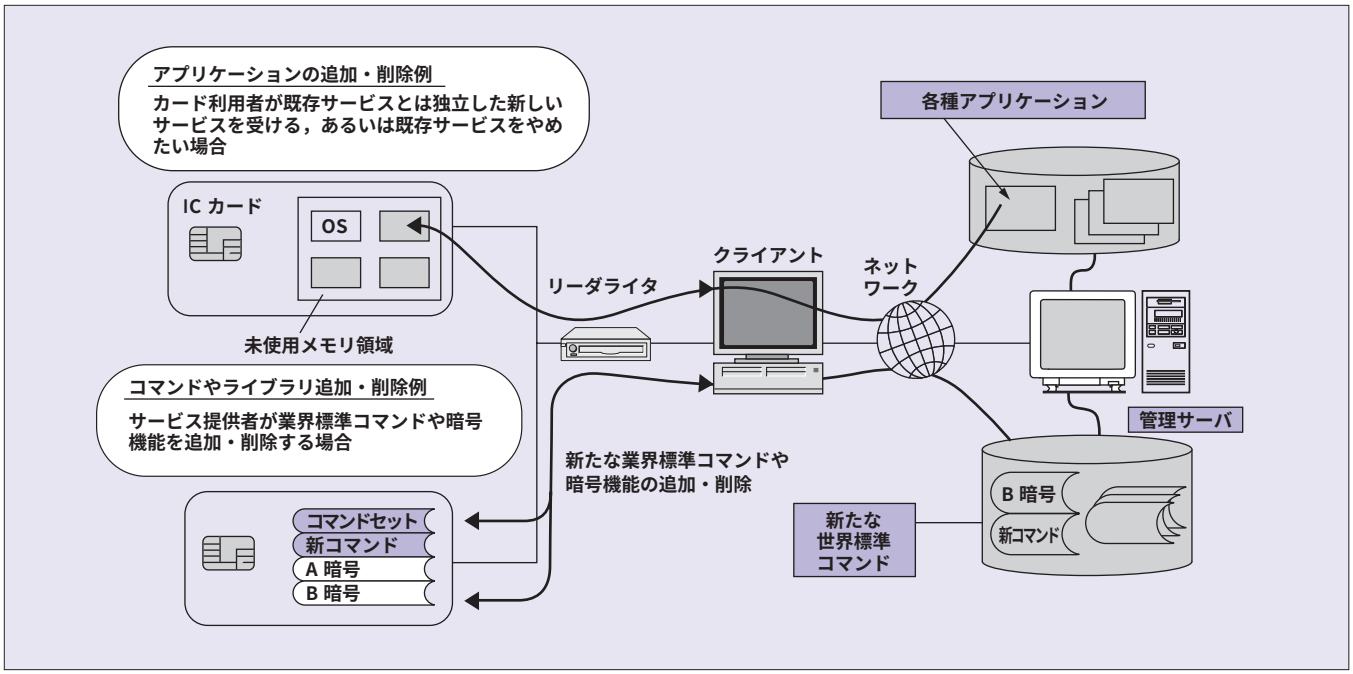
●ファイアウォール

マルチアプリケーションを前提とした場合に，あるアプリケーションから，別のアプリケーションの格納領域（プログラムないしデータ）の参照や変更は，許可なく行えないように管理する手段が必要になります。ソフトウェアではVM（Virtual Machine）を実装し，OSの介入がなくてはアプリケーションを実行できないようにすることで，ハードウェアによる領域分離を可能とする実装が求められます（**図3**）。

●耐タンパ対応機能

耐タンパ対応とは，悪意を持った攻撃に対し，セキュリティを確保する対応のことです。回路・方式・プロセスの観点から，攻撃への対処技術が分かれます。想定される攻撃は，大別して侵入攻撃（Invasive attacks）と非侵入攻撃（Non-invasive attacks）に分けられます。前者は，IC表面に直接アクセスすることにより回路の盗視・操作を行い，カードの耐タンパ特性を侵害したり破壊したりする攻撃です。これには，薬液処理・マイクロプロービング・収束イオンビーム・電子ビーム・レイアウト復元などの工程が用いられます。後者は，ICそのものに直接関与しないで行われる攻撃です。ソフトウェアのプロトコル・暗号のアルゴリズムの弱点を見つけて悪用したり，サプライ電流の変動・漏洩電流のシグナルを解析してプロテクトされた情報にアクセスしたり（電流解析法），外的なストレス下でデバイスを操作し誤動作を誘導すること（グリッジアタック）によって，セキュリティ鍵の効力を失効させるという手段が考えられています。これらは，比較的簡単な設備と短時間の解析で実行される恐れがあります。このためメモリには，これらの攻撃に対応できる機能が求められています。

図3　マルチアプリケーションスマートカード機能



FRAMを利用したセキュリティ構造の実装

●FRAMとほかの不揮発メモリの比較

表1に不揮発メモリとして提供されているEEPROM，フラッシュメモリ，FRAMの特性比較を示します。FRAMは，従来の不揮発メモリであるEEPROMに比べて，書込みサイクル10万倍，書換え回数10万倍などの向上が図られています。

表2にセキュリティ機能の特性比較を示します。セキュリティの課題を比較すると，EEPROMは，バイトアクセスできるものの書込みサイクルや書込み回数の制限の点で変更容易性が低くなっています。また，高速化実装にも不向きです。フラッシュメモリは，セクタ書換えであるためファイアウォールは実現しやすい反面，書込み時間に難点があります。また，耐タンパ構造に対しても未知数です。FRAMは，書込みサイクルがほかのメモリの10万倍以上速いことと，書換え回数が向上していることで，パラメタの変更容易性，アプリケーションダウンロードの優位性，高速化実装の点で最適なメモリといえます。

●ICカードに見るセキュリティ構造

図4にFRAM制御のセキュリティ構成を示します。マルチアプリケーションに対応したスマートカード「MB94R202/R211」は，非侵入攻撃への対策としてこの構成を持っています。本製品は，FRAMのアクセスコントロールに加え，FRAM領域のセクタ分けを行い，セキュリティレジスタの設定に従って，セクタごとにリード/ライトのプロテクト設定が可能です。この設定に違反したアクセスが発生すると，アクセス可否信号によりシステムバス上の例外割込み信号が発生し，プロセッサに通知される機構も実装されています。

また，電圧の降下も検出し，システムに通知する一方で，FRAMへアクセス中のデータの保証も行っています。

図5に，侵入攻撃への対処を目的としたレイアウトを，スマートカード向けIC「MB94R202」のチップ写真に示します。このように，プロセッサコア部やロジックコントローラ，暗号マクロのリソースを混在して配線することで，スクランブルが掛かった状況になっているため，LSI表面からの観察では配線接続を特定することができません。

さらに実際のプロセス技術では，配線層間膜の平坦化や多層配線・ダミー配線，メタルカバー膜を用いることで，表面や表面から次下層のパターンとの配線接続を観察することも困難にしています。

またFRAMマクロ構成は，プロセッサの論理アドレスから遊離した物理アドレス配置を有しています。論理アドレス上は連続した32ビット長のデータでも物理的には点在するため，侵入攻撃の手法によってその位置を探し出し，かつビットごとに読出すことはほとんど不可能となっています。

表1　EEPROM，フラッシュメモリ，FRAMの特性比較

	EEPROM	フラッシュメモリ	FRAM
メモリタイプ	不揮発性/10年	不揮発性/10年	不揮発性/10年
データ書換え単位	バイト	セクタ	バイト
読出しサイクル	200ns	100ns	100ns
書込みサイクル	10ms	1s以上	100ns
書換え回数	10万回	10万回	100億回
内部書込み電圧	14V	14V	3V

表2　セキュリティ機能の特性比較

	EEPROM	フラッシュメモリ	FRAM
鍵・パラメタの保存	○	○	○
鍵・パラメタの変更容易性	×	○	○
高速化実装	×	×	○
ファイアウォール	○	○	○
耐タンパ構造	実績あり	実績なし	実績あり

図4　FRAM制御のセキュリティ構成

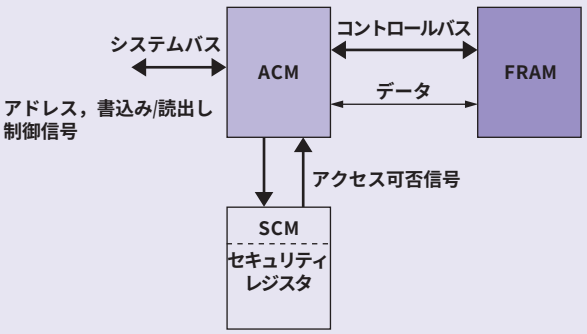
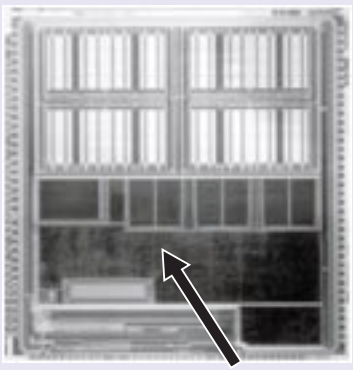


図5　MB94R202のチップレイアウト



●カード向け暗号の速度性能

表３に開発した各種暗号アルゴリズムに対する処理速度とコードサイズを示します。

共通鍵暗号であるDESやAESはすべてハードマクロで実現されています。DESの場合は，トリプルDESで処理速度が低下しています。これは，１つのハードマクロを３回呼び出すので，ソフトウェア処理時間がかかるためです。今後，利用されるAESでは，256ビット長までハードマクロで実現できているため，13.35Mbpsと高速になっています。

公開鍵暗号である楕円曲線暗号（２の拡大体）では，**図６**に示すように，スカラ倍算を行う部分をFRAM上でテーブル展開し，検索することで高速化を可能にしています。テーブルの計算は，運営サイトより与えられるパラメタに基づき，事前に1度だけ計算され設定されます。また，署名生成・確認でFR30プロセッサによるコードサイズは，11Kバイトで実現されています。RSA暗号は素数計算になるため，専用の32ビット剰余演算器を保有しています。これにより，署名確認で147.2msの高速処理を実現しています。

今　後

今後さらにIT化が進み，社会基盤の基礎技術として暗号とセキュリティは必須のものになると考えられます。とくにICカードは，マルチアプリケーションに対応したOSも拡充し，利便性は著しく向上すると考えられます。しかし反面，パソコンで発生しているウイルス問題などのように，新しい攻撃が登場する恐れもあります。当社は今後とも，セキュリティにおける新たな問題に対しても積極的に取り組み，安全かつ高速で，信頼性の高いFRAMを搭載したICカードやセキュリティデバイスを提供していく予定です。 ■

FRAM事業部 商品設計部
笛木俊介

【参考文献】

＊１：加藤辰也ほか:セキュリティとサービス向上をかなえるICカード．FUJITSU，Vol.52，No.6，p.525-530（2001）．

＊２：富士通：FRAMスマートカードセキュリティ．富士通，2001.

＊３：鳥居直哉ほか：楕円曲線暗号．FUJITSU，Vol.50，No.4，p.197-201（1999）．

＊４：岡本栄治：暗号理論入門．共立出版，1993.

表３　暗号処理速度とコードサイズ（13.56MHz時）

暗号アルゴリズム	コードサイズ	処理速度	
2べきECC	11Kバイト	署名生成＊4	49.88ms
		署名確認＊4	129.82ms
RSA	7Kバイト＊3	署名生成＊5＊6	210.8ms
		署名確認＊5＊7	147.2ms
DES＊1	——	シングルDES ECB（ハードマクロ）	43.39Mbps
		シングルDES ECB	1.046Mbps
		トリプルDES CBC	465kbps
AES＊2	——	Round処理（鍵長128ビット）	18.46Mbps
		Round処理（鍵長192ビット）	15.50Mbps
		Round処理（鍵長256ビット）	13.35Mbps

＊1：シングルDES ECB各モードでは，ハードマクロを利用したソフト処理を施している。
＊2：ハードマクロのみ
＊3：べき乗剰余演算のみのサイズ
＊4：鍵長163ビット
＊5：鍵長1,024ビット
＊6：CRT法を使用
＊7：公開指数e=65537

図６　テーブルを用いた固定点のスカラ倍算

固定点 P の倍数をあらかじめ計算した倍数テーブルを，FRAM に格納する。

上位からスカラ係数 K を与えるだけで，KP を計算する。
KP の計算は，テーブル上の値を効率良く使用しながら，加算・2 倍算を組み合わせる。

固定点には，計算頻度の高い点を選択する。

●スカラ倍算計算例

1.A ＝（2^{11}P ＋ 2^{7}P ＋ 2^{3}P）
2.A ＝ A ＋ A
3.A ＝ A ＋（2^{10}P ＋ 2^{6}P ＋ 2^{2}P）
4.A ＝ A ＋ A
5.A ＝ A ＋（2^{9}P ＋ 2^{5}P ＋ 2P）
・・・

●テーブルの例

FRAM 領域
P
2P
2P ＋ P
・・・
2^{9}P ＋ 2^{5}P ＋ 2P
・・・
2^{10}P ＋ 2^{6}P ＋ 2^{2}P
・・・
2^{11}P ＋ 2^{7}P ＋ 2^{3}P
・・・
2^{11}P ＋ 2^{10}P ＋ ・・・ ＋ 2P ＋ P