高速RSA暗号LSI

Fast RSA Computation Algorithm and LSI

新保　淳 SHIMBO Atsushi
野崎　華恵 NOZAKI Hanae
川村　信一 KAWAMURA Shin-ichi

公開鍵（かぎ）暗号のデファクト スタンダード（事実上の標準）であるRSA（Rivest−Shamir−Adleman）暗号に対し，ハードウェア向きの高速計算手法を開発し，LSIを設計した。この計算手法は，剰余系表現（RNS：Residue Number System）を利用していることが特徴であり，実効的な整数除算を不要とするモンゴメリ乗算手法と組み合わせている。この手法によって，RSA暗号を並列処理により実行することができ，ハードウェア内の演算装置の並列度を高めることで高速処理が可能となる。

設計したLSIは，80 MHz動作時に1,024ビットのRSA処理を約2.4 msで実行できる見通しであり，特にサーバ用のアクセラレータとして有効である。

We have developed a fast computation algorithm for the Rivest−Shamir−Adleman (RSA) cryptosystem, which is the de facto standard scheme in public key cryptosystems and digital signatures. This computation algorithm is based on the residue number system, and is a variant of the Montgomery multiplication method. By applying this algorithm, RSA encryption and decryption can be processed in a parallel manner when multiple arithmetic modules are installed in a device. Increasing the number of arithmetic modules results in higher performance.

We have designed an LSI which performs 1,024-bit RSA processing in 2.4 ms by an 80 MHz clock. This LSI is expected to be particularly useful for RSA accelerators installed in servers.

1 まえがき

公開鍵暗号及びデジタル署名の実用化が進み，その有用性は広く認識されるようになってきた。例えば，Webを利用してチケットを予約したり，書籍を購入したりする場合に，クレジットカード番号を安全にサーバに送信するために利用されるSSL（Secure Socket Layer）では，公開鍵暗号を利用した暗号通信が行われている。また，政府機関でも，2003年度までに行政手続きを電子化する電子政府(注1)の構築に向けて，2001年4月から電子署名法が施行された。電子署名法は，デジタル署名に捺印と同様の法的な意味づけを持たせることを制定したものであって，デジタル署名の利用場面はますます増加することが予想される。

こうした公開鍵暗号やデジタル署名のアルゴリズムとして，現時点でもっとも多く利用されているものはRSA方式であろう。RSA方式の実装は，1978年の同方式の発表以来，多くの機関で行われており，当社でもソフトウェア実装や，電子マネーなどに利用するICカード上へのハードウェア実装（コプロセッサ開発）を実施してきた。また，当社は，新たに並列処理に特徴のある高速なRSAチップを開発している。

ここでは，RSA方式の演算単位となる剰余付き乗算の並列処理アルゴリズムの原理と，それに適したハードウェア構成を説明する。設計したRSAチップは，80 MHzでの動作時に1,024ビットのRSA処理を約2.4 msで実行でき，現在のトップクラスの性能を実行できる見通しである。

(注1)　行政を効率化し国民負担の軽減を図るため，申請届出手続きや政府調達など，行政手続きの電子化を実現するシステム。

2 RSA方式の概要

RSA方式は多倍長整数を利用して，$C=M^e \bmod n$なる計算（べき乗剰余計算）により平文Mを暗号文Cに変換する。ここで（e, n）は，暗号化鍵である。復号も，べき指数をd（復号鍵）に変える以外は，同じべき乗剰余計算で実行される。鍵である法nや復号鍵dのサイズは，安全性の観点から1,024ビット以上が利用されることが多く，平文Mや暗号文Cのサイズも法nと同等のサイズとなる。1,024ビットもの整数のべき乗剰余計算は，現在の高性能なCPUでも10～50 ms程度の処理時間を必要とする処理である。安全性を高めるために2,048ビットの鍵長を用いる場合もあり，この場合には1,024ビット処理の6～8倍の処理時間となる傾向がある。

高性能なCPUを搭載できないICカードなどの装置や，処理の集中する可能性があるサーバ側装置では，一般にRSA方式のハードウェア実装が必要とされる。RSA方式は剰余付き乗算（$A \times B \bmod n$）の繰返しで実行されるため，実装上は剰余付き乗算をどのようにして効率的に処理するかがキーとなる。

3 高速なRSA処理アルゴリズム

剰余付き乗算の効率化には様々なアプローチがあるが，今回の開発では剰余付き乗算に対して並列処理可能な計算方式を考案し，ハードウェアの並列度を高めることで高速化していることがポイントである。

3.1 剰余系表現

多倍長整数の表現としては2進数表現が使われる場合が多いが，ここではRNSを用いている。RNS表現は，複数の互いに素な整数の組（基底と呼ぶ）を用意し，整数xをこの基底の各成分での剰余値の組で表現する手法である。基底の全成分の積よりも小さい正整数は，中国人剰余定理（CRT：Chinese Remainder Theorem）によりRNS表現によって一意に表現できることが保証される。また，RNS表現された整数xを2進数表現に変換することやその逆もできる。

RNS表現では二つの整数の加算，減算，乗算が成分ごとに独立に行えるという特長がある。すなわち，RNS表現された2数の加算，減算，乗算は各基底成分を法とした成分単位の剰余付きの加算，減算，乗算で計算できる。基底成分をワードサイズとしてワードサイズの整数演算器を複数実装すれば，RNS表現での加算，減算，乗算を高速に処理できる。2進数表現ではキャリーの伝播（でんぱ）が生ずるため，各ワードで独立に処理を行うだけでは正当な結果が得られないことを考えると，演算の独立性がRNS表現の利点である。

一方で，RNS表現の欠点としては，除算や2数の大小比較が困難なことが挙げられる。RSA暗号に代表される公開鍵暗号では法nでの剰余計算が必要であり，これには除算を利用するのが単純な方法であるが，RNS表現ではこの除算を回避する手法が必要となる。

基底a（$a=\{a_1, a_2, \cdots, a_n\}$）を用いた整数xのRNS表現とRNS表現された整数x，yに対する乗算を図1に示す。

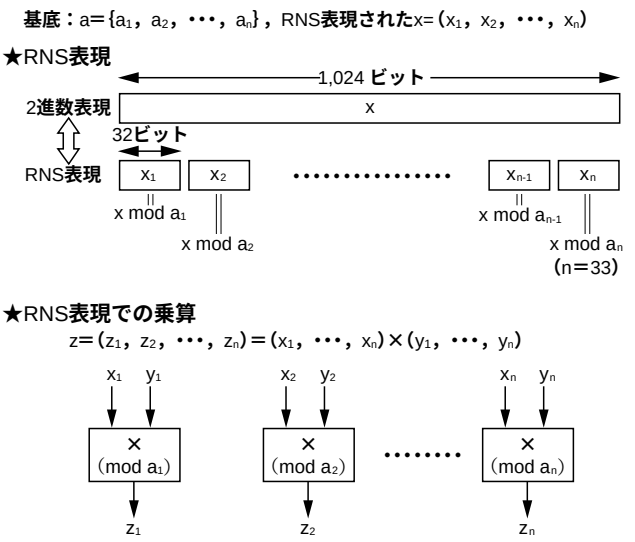

図1．RNS表現と演算法　RNS表現では，基底での剰余値の組を用いて多倍長整数を表す。乗算は，各成分ごとに基底での剰余付き乗算を行う。
Residue number system and arithmetic

3.2 モンゴメリ乗算

モンゴメリ乗算は，除算処理を利用せずに剰余付き乗算を計算する手法である。通常，多倍長整数での四則演算においてもっとも処理時間が掛かるのが除算であるため，モンゴメリ乗算は2進数表現を利用したRSA暗号の実装でも適用される場合が多い。

2進数表現でのモンゴメリ乗算の原理は，図2に示すように，最初に対象の2数の積を求めたうえで，法の倍数を加算し，その結果に対してビットシフトを行う（あるいは所定の部分だけ切り出す）ことで適切なサイズに縮小する手法である。法の倍数を加算しても剰余値としては値が保存される性質を利用している。あらかじめ縮小するビットサイズに相当する定数Rをモンゴメリ乗算の対象である2数に掛けておけば，ビットシフトによりR^{-1}を掛けたことになり，結果として所望の剰余付き乗算結果に定数Rを掛けた値が得られる。したがって，この結果を再びモンゴメリ乗算の入力に利用することができ，また，最後に定数R^{-1}を掛ける処理を行えば求める結果が得られる。

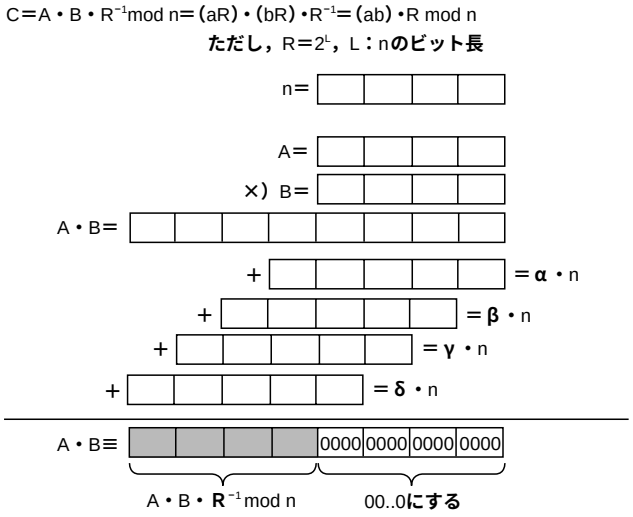

図2．モンゴメリ乗算　モンゴメリ乗算は，定数Rが掛けられた表現で剰余付き乗算を実行する。結果の下位側がすべて0になるようなnの倍数を求めて加算する。法サイズに縮小する演算が，乗算とビットシフトで構成できる。
Montgomery multiplication

モンゴメリ乗算をRNS表現された整数に対して適用する場合には，多倍長整数を表現するために用意した基底aとは別の基底bを用意し，基底bの成分の積Bをモンゴメリ定

数Rとして利用するというアイデアが提案されている[1]。この場合，基底aでの演算と基底bでの演算を組み合わせて処理を進めることになる。この手法（RNSモンゴメリ乗算と呼ぶことにする）により除算を用いずに剰余付き乗算を行える。RNSモンゴメリ乗算アルゴリズムを図3に示す。図では基底aによりRNS表現された整数xを，$<x>_a$と表記している。RNSモンゴメリ乗算では，一方の基底で表現された整数を他方の基底での表現に変換する基底変換処理（図3のstep4と8）が必要となる。基底変換がRNSモンゴメリ乗算の中でもっとも計算コストを要する処理であって，基底aやbの成分数をnとしたときに，n^2回のワードサイズでの乗算を必要とする。

基底変換処理は，成分単位に独立に計算できる処理と全成分での計算結果に依存した処理とに分けられるが，われわれは後者の処理を従来よりも軽量化する改良を行っており，この改良手法を利用すると基底変換処理の大半は成分単位に独立に処理が行える[2]。また，基底変換処理以外のRNSモンゴメリ乗算の処理は加算と乗算であるため，RNSモンゴメリ乗算のほとんどが成分単位の並列処理で実行できる。RNSモンゴメリ乗算を繰り返し適用することでRSA暗号の実装ができる。

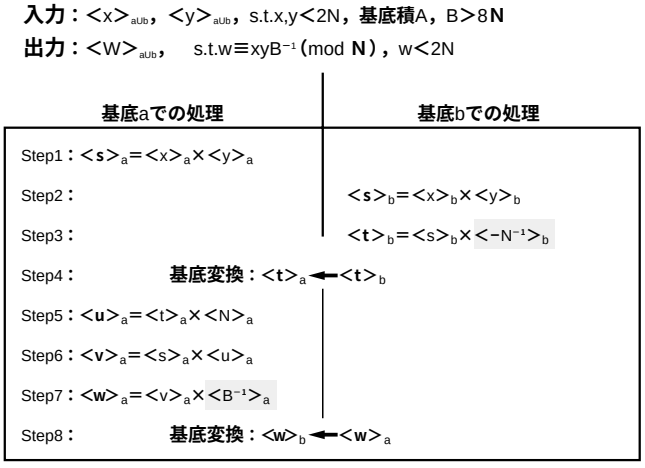

図3．RNSモンゴメリ乗算　　RNS表現での乗算，加算を用いてx，yに対するモンゴメリ乗算の結果wを求める。Step4と8で基底変換が行われる。Step3と7での乗数は事前に計算しておく。
RNS Montgomery multiplication

4 ハードウェア構成

RNSモンゴメリ乗算の実行に適したハードウェア構成を図4に示す。ワード単位での剰余付きの積和演算を行う演算ユニットと基底拡張時の全成分に依存した処理を行う補正回路，除算器，外部インタフェース回路から構成される。除算器はRNSモンゴメリ乗算に利用する初期パラメータの計算など，補助的に用いられ，処理のほとんどは演算ユニットが並列に動作して行われる。すべての演算ユニットは，リング状に結線されており，互いにデータを交換できる。

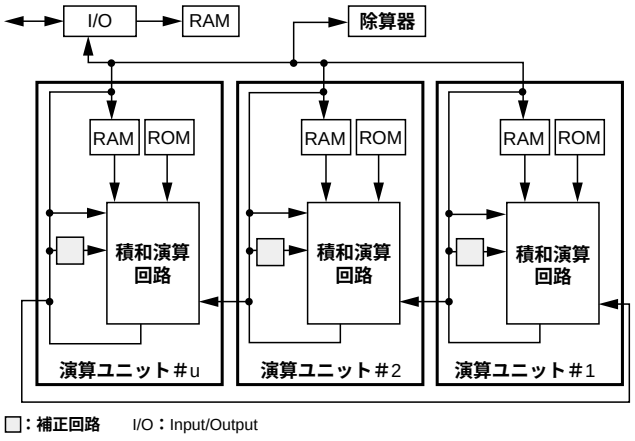

図4．RSA LSIの構成　　u個の演算ユニットを備えた構成になっている。RNSモンゴメリ乗算を並列処理で効率よく計算できる。
Architecture of RSA LSI

演算ユニットは，典型的には32ビットの乗算器と，その結果を累積していく加算器と，累積加算された結果を32ビットの基底成分で剰余計算する剰余算器で構成される。基底成分は$2^{32}-\delta$（δは10ビット以下）なる形式にすることができ，基底成分での剰余計算は，2^{32}がδに合同な性質を利用して乗算器をベースに構成できる。演算ユニットは，このほかに設計時に計算したパラメータを記憶したROMと，中間結果を保持するRAMを備えている。ROMのデータは，基底に依存したパラメータであり，それを利用する演算ユニットのROMに保持される。すなわち，ROMの内容は演算ユニットごとに異なる。

RNSモンゴメリ乗算では，基底成分ごとに同じ手順で処理を進めることができるので，演算ユニットごとに別の制御回路を持つのではなく，ハードウェア全体で一つの制御回路を共用している。補正回路もハードウェア全体で一つに共用することができる。なお，図3の構成では演算ユニットごとに補正回路を備えているが，典型的なRSAのサイズでは補正回路は10ビットの加算器で構成でき，小規模である。したがって，補正回路を共用しなくても全体のハードウェア規模に及ぼす影響は少ない。

以上のハードウェア構成により次に挙げる特長が得られる。

⑴ スケーラビリティがある　　演算ユニット数を増やすに伴い，比例的に処理性能が向上する。同時に，ハードウェア規模も増加することを考慮したうえで求める性能のチップを設計できる。

⑵ モジュラリティと規則性が高い　　複数の演算ユニットは，ROMに書かれるデータ以外は同一構成である。

また，すべての演算ユニットを同時に同じ手順で制御できる。

5 試作LSI

0.25 μmのCMOS（相補型金属酸化膜半導体）プロセスを用いてLSIを試作している。機能としては，RSA方式の基本関数であるべき乗剰余計算を実行でき，更に，RSA方式でのモジュラスが二つの素数p，qの積である性質を利用して，法pとqでの演算に分解してべき乗剰余計算を行う手順（CRT利用手順と呼ぶことにする）にも対応している。動作周波数80 MHzの場合に，1,024ビットのRSA処理を4.2 ms（CRT利用手順では2.4 ms），2,048ビットのRSA処理を29.2 ms（CRT利用手順では8.9 ms）でそれぞれ実行できる見通しである。演算ユニット数11で，ゲート規模は約280 kゲートである。

6 あとがき

RNSモンゴメリ乗算を実装したモジュール構成と，並列処理を特徴としたRSAチップについて述べた。試作しているチップは，現時点で商用RSAチップの中ではトップクラスの処理性能を持っている。高性能を引き出すために，演算ユニット数の並列度を高めているため回路規模が大きいが，高性能を要求されるサーバ装置のRSAアクセラレータとしての利用が見込まれる。

文　献

(1) Posch, K.C., et al. "Modulo Reduction in Residue Number Systems". IEEE Tr. Parallel and Distributed System. **6**, 5, 1995, p.449－454.
(2) Kawamura, S., et al. "Cox-Rower Architecture for Fast Montgomery Multiplication". Advances in Cryptology EUROCRYPT2000, Springer-Verlag. 2000, p.523－538.

新保　淳　SHIMBO Atsushi
研究開発センター　コンピュータ・ネットワークラボラトリー研究主務。暗号技術及び暗号応用システムの研究・開発に従事。電子情報通信学会，情報処理学会会員。
Computer & Network Systems Lab.

野崎　華恵　NOZAKI Hanae, D.Sc.
研究開発センター　コンピュータ・ネットワークラボラトリー研究主務，理博。暗号及び情報セキュリティ技術の研究・開発に従事。電子情報通信学会会員。
Computer & Network Systems Lab.

川村　信一　KAWAMURA Shin-ichi, D.Eng.
研究開発センター　コンピュータ・ネットワークラボラトリー主任研究員，工博。暗号及びセキュリティ技術の研究・開発に従事。電子情報通信学会，情報処理学会，IEEE，IACR，SITA会員。
Computer & Network Systems Lab.