FUN-ECM プロジェクト
FUN-ECM project

A グループ (A)　Search Group (A)

1016228　青山和弘　Kazuhiro Aoyama

1　背景

情報化が進んだ現代において，暗号化技術は至るところで利用されている．情報保護，セキュリティのために利用される暗号化は時代とともに進化を続けているが，それに伴って攻撃の方法も常に進化してきた．進化し多様化する攻撃方法に対して，強く安全である暗号化技術を開発し，その安全性を評価するには，まず開発者側が暗号の解読方法を熟知していなければならない．そこで，本プロジェクトではその暗号解読方法の一つである楕円曲線法 (ECM：Elliptic Curve Method) について学び，RSA 暗号と呼ばれる暗号化技術の安全性について検討する．

RSA 暗号とは現在も公開鍵暗号に使用されている技術の一つで，2 つの素数を掛け合わせて作られた長大桁合成数の素因数分解が困難であることを根拠に安全性が保障されている暗号である．実際にやってみると分かるが，例えば 190 桁の長大桁数を素因数分解しようとすると非常に長い時間を要する．楕円曲線法は，与えられた曲線の点が無限遠点になることにより合成数の因数である素数を見つけることができる方法である．これを利用すると RSA 暗号を解読することができるが，ただ行うだけではやはり非常に長い時間と大きな計算量を要する．

前年度までのプロジェクト活動で，楕円曲線法によって素因数分解を行うプログラムが完成していた．そこで，今年度のプロジェクト活動ではそのプログラムを高速化すること，または別のプログラムを作成することによって，前年度までに作成されたプログラム以上の速度で素因数分解を行うことを目標とした．また，楕円曲線法の知識を一般 (特に本校への進学を考えている高校生) に普及したいと考えた．そこでもう 1 つの目標として，本プロジェクトが数年前から運営している広報用の Web サイトを刷新し，かつ学んだ知識を利用した楕円曲線法の学習用コンテンツを新たに盛り込むことを掲げた．

2　課題の設定と到達目標

まずメンバー全員で楕円曲線法について基礎学習を行い，その後前項で述べた本プロジェクトの 2 つの目標に沿って活動を行うために，次の 2 つの班に分かれた．

- FORTRAN によって素因数分解プログラムを作成する高速化班
- Web サイトを刷新する広報班

前期はこの 2 班体制で活動を行っていたが，中間発表会後に高速化班が行おうとしていた手法があまり効果的でないとわかったため，後期からは次の 3 班体制で活動を行った．

- 広報班
- 前年度までに作成されたプログラムの利便性を向上させるシステム班
- FPGA に楕円曲線法を実装するハードウェア班

また，作成したプログラムの性能確認のために，STUDIO KAMADA という Web サイトで公開されている長大桁の合成数を対象に素因数分解を行った．これに成功すると，STUDIO KAMADA[1] 上に本プロジェクトの名前が記載される．

以下に各班の課題を述べる．

高速化班

前年度までに C 言語によって作成された素因数分解プログラムと同様の動作をするプログラムを，スーパーコンピュータなどの開発に使用される言語である FORTRAN によって作成することで高

速化を図ろうとし，FORTRAN の学習，及び C と FORTRAN の速度比較を課題とした．

広報班

Web サイトを刷新するにあたり，従来のものの何が問題点なのかを洗い出すことを前期の課題とした．その後，HTML，PHP，CSS，JavaScript の学習を行い実際に新しい Web サイトの構築を行った．

システム班

前年度までに作成されたプログラムの利便性という観点から，次の問題点が挙げられた．

- 1 度の実行で 1 つの合成数についてしか処理できない
- プログラムの動作・終了を確認するために，その都度 SSH でワークステーションに接続し，プロセスの動作状況を確認しなければならない

この 2 点を課題とし，UI の向上を目標として素因数分解プログラムを内包したシステムの構築を行うことにした．また，それと並行して前年度までに作成されたプログラムを運用し，素因数の発見を試みた．

ハードウェア班

今年度のプロジェクト予算で購入していた FPGA に，楕円曲線法を実装して素因数分解を行いたいと考えた．FPGA の特徴として，並列化可能なプログラムやアルゴリズムを実装すると高速な処理が期待できるという点がある．楕円曲線法の手順のうちに，最も処理時間がかかる点のスカラー倍算というステップがあり，これは並列化が可能であった．そこで，このスカラー倍算を行うステップを FPGA 上に実装することを課題とし，実際に素因数分解を行うこと，及び昨年度までに作成されたプログラム以上の速度で素因数分解を行うことを目標とした

3 課題解決のプロセスとその結果

3.1 基礎学習

まず楕円曲線法を理解するため基礎学習を行った．有限体，Euclid の互除法，楕円曲線の定義方程式，楕円曲線上での加算・逆元・スカラー倍算，無限遠点などについてメンバー全員で学んだ．その後，メンバーの希望をもとに班分けを行い課題解決に向けて活動を開始した．

3.2 高速化班

高速化班には FORTRAN を扱えるメンバーが 1 名しかいなかったので，最初に班員全員で参考書などをもとに FORTRAN の基礎学習を行った．また，FORTRAN で書くことで高速化が可能か検証するために，モンテカルロ法やライプニッツ公式を使用して円周率を求めるプログラムを C 言語と FORTRAN で作成し，実行速度の比較を行った．これにより FORTRAN の方が早いという結果が得られたため開発を進行させようとしたが，いくつか問題が浮上した．

1 つ目は，C 言語と FORTRAN の間には通常の使用方法では演算性能に大きな差がないと近年の研究結果で言及されていることが分かったことである．前述の比較方法のみでは，楕円曲線法で素因数分解を行うプログラムを作成したときに C 言語より FORTRAN の方が高速に実行できるとは判断しきれない．

2 つ目は学習コストと開発期間の問題である．前期が終了した時点で速度比較までしか行えなかったことを考えると，今年度中の完成は難しいと判断した．来年度以降に引き継ぐことも考えたが，今年度と同じように FORTRAN の学習から始め開発に移るサイクルでは開発期間を十分に設けることができずプログラムを完成させるのに大きな時間がかかってしまう．

以上の理由から FORTRAN でプログラムを作成することを断念し，高速化班のメンバーは後期からシステム班とハードウェア班に分かれて活動することになった．

3.3 広報班

Web サイトを刷新するにあたり，前期の活動では従来の Web サイトの問題点について検討を行った結果，次の 2 点が挙げられた．

1 つ目はデザイン性があまり良くない点である．楕円曲線法を広めるためにも，親しみやすく操作しやすいシンプルなデザインの Web サイトにし，様々な人に見てもらいたいと考えた．

2 つ目は，実際に掲載されている情報量よりも Web サイトを閲覧した際の情報量が多く見えてしまう点である．これを解決するために，Adobe Illustrator を利用して従来の Web サイトの構造化 (Web サイトのカテゴリ階層構造を一目で確認できるよう図に起こす作業) を行うことで，スマートでコンパクトな Web サイトを構築するための分析を行った．また，前期には他にレイア

ウトの試案やロゴの制作，カテゴリ分け，プレビューの作成を行った．

後期には実際に Web サイトの構築を開始し，それと並行して使用する言語の学習を行った．基本である HTML をはじめ，シンプルなスタイルにするための CSS，任意の楕円曲線のグラフを表示するための JavaScript，後述するシステム班と連携して素因数分解プログラムの実行結果を取得・表示するための PHP の記述方法について学んだ．

Web ページを作成しながら，実際にそれがどう見えるか確認を行った．作成するページはカテゴリごとに広報班のメンバーで分担したため，フォントサイズが異なっていたりレイアウトが崩れていたりしていることがあった．そのため，内容の繋がりに応じてマージンの統一を行ったり，HTML5 以降で使用可能な header タグ，footer タグを使用してグローバルナビゲーション及びフッターの部分を全ページについて統一した．

ページの作成が完了するとファイルを統合し，レンタル Web サーバの XREA にアップロードした．XREA にはデータベースと PHP が利用可能なフリープランがある．システム班と連携し，素因数分解プログラムの実行結果をデータベースに格納して，それを取得し表示するページを作成するのに最適だと判断したためこれを採用した．

成果発表会の時点で，ファイルの統合と全体の微修正まで完了した．

3.4 システム班

保守運用の観点から昨年度までに作成されたプログラムに大きな影響を与えないようにしつつ，前述の課題を解決するために，次の要素で構成されるシステムを構築した．

- ユーザとシステム間のインタフェースとなるフロントエンド：GUI 上で合成数，推測される素因数の桁数，処理結果を保存するファイル名を入力し，これを CSV 出力してワークステーションに転送するもの．
- ワークステーション上で，プログラムに引数 (素因数分解の対象) を渡し，かつ終了時にその実行結果を外部 Web サーバ (XREA) に転送するシェルスクリプト：フロントエンドが出力した CSV ファイルから 1 行 (合成数，推測される素因数の桁数，ファイル名) を読み出し，それを引数としてプログラムに渡し実行させる．プログラムが終了すると XREA にその結果をポストして，CSV ファイルから次の 1 行を読み出し次の実行を開始する．
- シェルスクリプトが転送したデータを取得してプログラムの実行結果を表示する Web ページ：シェルスクリプトがポストした json ファイル形式の実行結果を PHP で取得し表示する．
- 前年度までに作成された素因数分解プログラム：合成数，楕円曲線法によって素因数分解を行うときに用いるパラメータ B1・B2，実行結果の出力ファイル名を引数として受け取り素因数分解を行う．

また，昨年度までに作成されたプログラムを運用し STUDIO KAMADA で公開されていた合成数に対し素因数分解を行った結果，今年度は以下の 3 件について素因数を発見することができた．なお，ここで $7(4)_{188}1$ という表記は，7444…(4 が 188 個)…4441 を示す．

- $7(4)_{188}1$
- $7(4)_{189}9$
- $9(2)_{188}3$

3.5 ハードウェア班

楕円曲線法の手順は次の 3 ステップである．

(1) 任意の点 P(x, y) を決定し，その点を含む楕円曲線 $y^2 = x^3 + ax + b$ を 1 つ決定する
(2) 十分大きな値 B を指定し，点 Q=BP を求める
(3) 点 Q の x 座標の分母と，素因数分解対象である合成数 N との最大公約数をとり，これが 1 以外の値ならばそれが N の素因数である

前述した点をスカラー倍算するステップとはこの (2) のことである．これについて FPGA 上に実装することを目標とした．回路の作成を分担するのは作業効率が悪いと考えたこと，及び他にも行うべき作業があったことから班内で役割分担を行った．

1 名は，(2) で点 P にスカラー倍する B をスカラー値に分解しリスト化するプログラムの作成を行った．B には例えば 4300 万の階乗などが選ばれるが，これをそのまま求めてから点 P にスカラー倍するのでは無駄のある計算となってしまう．よって次の式で B を求めその構成要素をリスト化し，それを点 P に繰り返し計算する

ようにすれば無駄のない計算で BP を求めることができるうえ，(2) のステップを並列化することができるので FPGA の特性の恩恵を受けることができる．

$$B = \prod_{prime:4300\text{ 万以下の素数}} prime^{e}$$

$$\because e = \lfloor \log prime 4300\text{ 万} \rfloor$$

また，他には今年度購入した FPGA や対応している NiosII の英版マニュアルの日本語訳を行った．

もう 1 名は，VHDL の基礎学習を行ったあと，(2) のステップを実装するためにはどのような機能が回路に必要かを検討した．STUDIO KAMADA で公開されている合成数は平均 190 桁であるため，2 進数に変換すると 640bit となる．よって，128bit のデータ (これを 1 ワードとする) を扱えるレジスタを 128 個内蔵した，128 ワード・レジスタファイルを用意すれば，合成数とそれを素因数分解する過程で出現した数を格納しきれると考えた．また，レジスタファイルの他に加算器や乗算器，桁上がりの状態を保持しておくキャリーレジスタ，選択用のマルチプレクサなどスカラー倍算を行ううえで必要なコンポーネント (論理ブロック) をあげ，その配置と配線を試案し，次の回路を構成した．

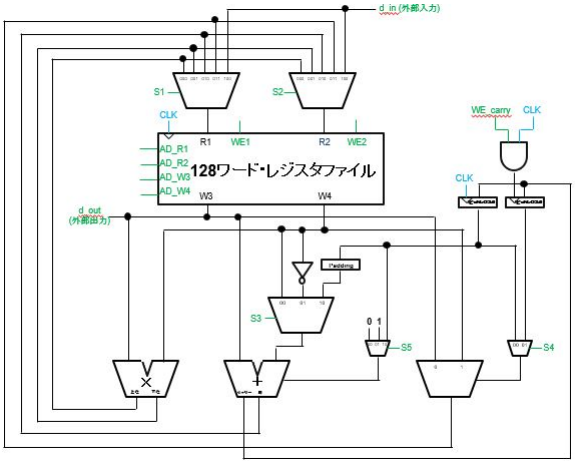

図 1　設計した回路

4　今後の課題

以下が来年度以降の課題として挙げられる．

- 新しい Web サイトに，高校生を対象とした楕円曲線法の学習コンテンツを盛り込む
- ハードウェア班が作成した回路について，回路のコンパイルを完了させ，動作確認を行うこと，及び制御信号を組み合わせた命令セットを格納しておく命令メモリを実装すること

参考文献

[1] STUDIO KAMADA. http://stdkmd.com/, （最終アクセス 2018 年 1 月 9 日）