信州大学審査学位論文

暗号方式の形式化に関する研究

2012年3月

布田　裕一

要旨

近年における情報化社会の急速な発展に伴い、インターネットを代表とする通信ネットワークを介した情報通信の重要性が高まっている。情報通信において、個人情報などをはじめとする機密情報を暗号化して送信したり、通信相手が正しい相手であるかを認証を用いて検証したりするなど、暗号技術が重要な場面が多くなってきている。

暗号技術は、共通鍵暗号や公開鍵暗号などの暗号方式を組み合わせることにより実現される。RSA 暗号は、公開鍵暗号としてデファクトスタンダードのような存在である。しかし、最近は、楕円曲線暗号や格子暗号がその処理高速性から注目されている。楕円曲線暗号は、楕円曲線と呼ばれる非特異の 3 次曲線上の点の演算を利用した暗号である。楕円曲線上の点に対し、加算が定義され、その加算の下でアーベル群を構成することが知られている。さらに、アーベル群からスカラの演算を導入し、$\mathbb{Z}$-加群を構成できる。楕円曲線暗号は、点 P のスカラ s 倍の点 sP から、スカラ s を求める楕円曲線上の離散対数問題が求解困難であることを安全性の根拠としている。

一方、格子暗号は格子上の求解困難な計算量問題を安全性の根拠としている。格子上の計算量問題で特に知られているのが、最短ベクトル問題や最近傍ベクトル問題である。最短ベクトル問題は、格子の基底が与えられたとき、格子に含まれる最短のベクトルを求める問題である。最近傍ベクトル問題は、あるベクトル (格子に含まれるとは限らない) に最も近い格子に含まれるベクトルを求める問題である。

公開鍵暗号方式やそれを他の暗号方式と組み合わせた暗号スキームと呼ばれる方式では、その安全性の根拠である計算量問題の求解困難さ以上に暗号方式や暗号スキームの破る攻撃が困難であることを証明 (安全性証明と呼ぶ) することが多くなっている。最近は、安全性証明が必須となってきており、正しく安全性証明を構成することが重要である。

暗号方式や暗号スキームの安全性証明は、近年、方式が複雑になることにより、証明の誤りが発生しやすくなっている。人手による証明は誤りが発生する場合があるため、計算機による証明の検証が必要とされている。実際、IC カードをはじめとするセキュリティ製品のセキュリティ仕様である ISO/IEC 15408 では、セキュリティレベルが高い部分では、計算機による証明が求められている。

Mizarとは、Bialystok大学(ポーランド)のA.Trybulec教授を中心としたMizar Societyによって進められている、計算機を使って数学を定式化するプロジェクトの総称である。Mizarは、厳密な数学的形式記述が可能であり、なおかつ強力な推論機能を有している。しかし、Mizarには公開鍵暗号、特に楕円曲線暗号や格子暗号で使用する数学的な定義や定理がライブラリとして含まれていない。

本論文では、楕円曲線暗号や格子暗号で使用する数学的な定義や定理の形式化を行う。楕円曲線暗号に関しては、以下の形式化を行う。

- 素体 $\mathbb{F}_p$
- 射影座標と座標の同値
- $\mathbb{F}_p$ 上の楕円曲線と楕円曲線上の点の同値類
- $\mathbb{F}_p$ 上の楕円曲線の点の個数の評価
- $\mathbb{F}_p$ 上の楕円曲線の点の演算

これらの形式化により、楕円曲線暗号の暗号化・復号化アルゴリズムや署名生成・検証アルゴリズムで使用する楕円曲線上の演算をライブラリとして使用することができる。

格子暗号に関しては、以下の $\mathbb{Z}$-加群に関する形式化を行う。

- $\mathbb{Z}$-加群の定義
- アーベル群から $\mathbb{Z}$-加群の導入
- 素数 p から生成される $\mathbb{Z}$-加群から導かれるベクトル空間
- $\mathbb{Z}$ 係数の線形結合
- 自由 $\mathbb{Z}$-加群
- 自由 $\mathbb{Z}$-加群の階数

アーベル群から $\mathbb{Z}$-加群の導入については、楕円曲線暗号でも必要であり、他の分野でも利用される重要な命題である。格子は有限階数の自由 $\mathbb{Z}$-加群にノルムを定義したものであり、自由 $\mathbb{Z}$-加群やその階数の形式化は格子暗号において必須である。

これらの楕円曲線暗号や格子暗号の数学的定義や定理を形式化することで、それぞれの暗号の演算を扱うことが可能になる。また、本研究で形式化した数学的理論を発展させることで、楕円曲線暗号や格子暗号の攻撃についても、その評価も含めて形式化が可能になる。このように、本研究は公開鍵暗号を安全性証明していく上で基本となるものである。

目 次

第1章　序論

1.1　暗号技術の重要性

近年における情報化社会の急速な発展に伴い、インターネットを代表とする通信ネットワークを介した情報通信の重要性が高まっている。情報通信において、個人情報などをはじめとする機密情報を暗号化して送信したり、通信相手が正しい相手であるかを相手認証を用いて検証したりするなど、暗号技術が重要な場面が多くなってきている。また、DVD や BD(ブルーレイディスク) や、ビデオコンテンツの配信システムであるアクトビラにおいて、商用コンテンツの不正利用を防止する目的でも暗号技術が使用されている。

暗号技術は、共通鍵暗号や公開鍵暗号などの暗号方式を組み合わせることにより実現される [6]。共通鍵暗号は、暗号化／復号化で使用する鍵が同じ (共通) である方式である。そのため、送信者と受信者の間で秘密裏に鍵を共有する必要がある。しかし、処理量が小さい利点があり、大きなデータの暗号化で使用することが多い。

それに対して、公開鍵暗号は、暗号化で使用する鍵と復号化で使用する鍵が異なる。そのため、暗号化で使用する鍵を公開することが可能になり、相手の公開した鍵を入手するだけで通信が可能になる。暗号化で使用する鍵 (公開鍵) から復号化で使用する鍵 (秘密鍵) を求めることが困難であることが、公開鍵の安全性の根拠となる。

1.2　公開鍵暗号とその安全性

公開鍵暗号は処理量が大きいため、実際に利用する際は公開鍵暗号を用い、共通鍵暗号で使用する鍵を共有して、その後の通信を共有した鍵を共通鍵暗号に適用することにより、暗号通信を実現することが多い。また、送信するデータがその送信者が確かに送信したことを示す署名や、通信相手を正しい相手であるかを認証する相手認証で、公開鍵暗号が使用される。

RSA 暗号は、WEB ブラウザなどの SSL/TLS 通信で使われるなど、公開鍵暗号としてデファクトスタンダードのような存在である。しかし、最近は、楕円曲線

暗号や格子暗号がその処理高速性から注目されている。楕円曲線暗号は、先に述べたBDにおいてはBDドライブとプレーヤ間の通信で、アクトビラにおいてはコンテンツの配信サーバと受信機器間の通信で使用されている。楕円曲線暗号は、楕円曲線と呼ばれる非特異の3次曲線上の点の演算を利用した暗号である[4]。楕円曲線上の点に対し加算が定義され、その加算の下でアーベル群を構成することが知られている。さらに、アーベル群からスカラの演算を導入し、$\mathbb{Z}$-加群を構成できる。楕円曲線暗号は、点Pのスカラs倍の点sPから、スカラsを求める楕円曲線上の離散対数問題が求解困難であることを安全性の根拠としている。このように、公開鍵暗号では求解困難な計算量問題を安全性の根拠としている。

一方、格子暗号は格子上の求解困難な計算量問題を安全性の根拠としている[7]。格子上の計算量問題で特に知られているのが、最短ベクトル問題や最近傍ベクトル問題である。最短ベクトル問題は、格子の基底が与えられたとき、格子に含まれる最短のベクトルを求める問題である。最近傍ベクトル問題は、あるベクトル(格子に含まれるとは限らない)に最も近い格子に含まれるベクトルを求める問題である。

1.3 暗号方式の安全性証明

公開鍵暗号方式やそれを他の暗号方式と組み合わせた暗号スキームと呼ばれる方式では、それらの方式が安全であることの証明をつけることが多くなってきている。この証明は具体的には、それらの方式の安全性の根拠である計算量問題の求解困難さ以上に方式を破る攻撃が困難であることの証明である。このような証明を、暗号方式や暗号スキームの安全性証明と呼ぶ。実際に証明する場合は、方式があるレベルの攻撃で破れると仮定したとき、安全性の根拠である計算量問題が解けることを証明する。最近は、安全性証明が必須となってきており、正しく安全性証明を構成することが重要性が増してきている。

暗号方式や暗号スキームの安全性証明は、近年、方式が複雑になることにより、証明の誤りが発生しやすくなっている。例えば、RSA-OAEPと呼ばれるRSA暗号を用いた暗号スキームが提案されているが、当初の方式に安全性証明の誤りがあることが判明した[2]。その後、RSA-OAEPは安全性証明を修正している。このように、人手による証明は誤りが発生する場合があるため、計算機による証明の検証(プルーフチェックと呼ぶ)が必要とされている。ICカードのセキュリティ規格であるISO/IEC 15408では、ICカードの製品のセキュリティレベルを認定する基準を策定している。このセキュリティ規格において、セキュリティレベルが高いところでは、計算機による証明(形式的証明と呼ぶ)が求められている[8]。また、最近では暗号方式、特に、暗号プロトコルにフォーカスしたセキュリティレベル

の仕様 ISO/IEC 29128 の規格化が進んできている [9]。

1.4 計算機による証明

暗号方式や暗号プロトコルの安全性証明の方法として、最近、汎用結合可能性 (Universal Composable security, UC) の理論 [1] やゲーム列による安全性証明の理論 [3] が提唱されている。

汎用結合可能性の理論では、暗号プロトコルで使用する暗号方式やハッシュ関数などを、暗号プロトコル内の部品として、それぞれで安全性証明した結果を用いて、暗号プロトコル全体の安全性証明を示す枠組みである。この枠組みでは、各部品を UC 安全という安全性を満たすことを証明 (下位レベルの安全性証明) し、これらの部品を含めた暗号プロトコル全体が UC 安全であることを証明する (上位レベルの安全性証明)[5]。Canetti と Herzog は、Dolev-Yao モデルのような記号論的アプローチにより、上位レベルの安全性証明を実施している。また、Canetti らは下位レベルの安全性証明に対し、確率的 I/O オートマトン (Probabilistic I/O Automata, PIOA) を導入し、計算機により証明する方法を提案している。しかし、証明のすべてを計算機により行うことはできておらず、確率の評価は人手により行うため、安全性証明の形式化としては、まだ不十分である。

ゲーム列による安全性証明の理論では、暗号方式や暗号プロトコルの攻撃モデルをゲーム列におけるゲーム変換として変換し、攻撃が成功する確率が十分小さいことを証明する。また、Blanchet と Pointcheval はゲーム変換を自動で行い、ゲーム列を自動生成する手法を開発した。CryptoVerif は、その手法を実装したソフトウェアである。しかし、ゲーム列による安全性証明においても、ゲーム変換の変換前のゲームと変換後のゲームの攻撃成功確率の差が十分に小さいことの証明については、人手により実施している。したがって、その意味では、ゲーム列による安全性証明も、安全性証明の形式化としては、まだ不十分である。

1.5 Mizar

Mizar[16] とは，Bialystok 大学 (ポーランド) の A.Trybulec 教授を中心とした Mizar Society によって進められている，計算機を使って数学を定式化するプロジェクトの総称である．形式化証明のツールは、他にも Isabelle などのようなものもあるが、Isabelle は数学の公理レベルからの厳密な数学的形式記述となっておらず、証明の抜けが存在する可能性も残る。それに比べ、Mizar は厳密な形式記述が可能であり、なおかつ強力な推論機能を有している。そのため、Mizar を用いて、

暗号方式や暗号プロトコルを上位レベルの安全性証明を含めて厳密に証明することが期待されている。

しかし、Mizar には公開鍵暗号、特に楕円曲線暗号や格子暗号で使用する数学的な定義や定理がライブラリとして含まれていない。したがって、楕円曲線暗号や格子暗号で使用する数学的な定義や定理をライブラリに含めて、Mizar のライブラリを充実させる必要がある。

1.6 本論文の目的

本論文では、楕円曲線暗号や格子暗号で使用する数学的な定義や定理の形式化を行う。楕円曲線暗号に関しては、以下の形式化を行う。

- 素体 $\mathbb{F}_p$
- 射影座標と座標の同値
- $\mathbb{F}_p$ 上の楕円曲線と楕円曲線上の点の同値類
- $\mathbb{F}_p$ 上の楕円曲線の点の個数の評価
- $\mathbb{F}_p$ 上の楕円曲線の点の演算

これらの形式化により、楕円曲線暗号の暗号化・復号化アルゴリズムや署名生成・検証アルゴリズムで使用する楕円曲線上の演算をライブラリとして使用することができる。

格子暗号に関しては、以下の $\mathbb{Z}$-加群に関する形式化を行う。

- $\mathbb{Z}$-加群の定義
- アーベル群から $\mathbb{Z}$-加群の導入
- 素数 p から生成される $\mathbb{Z}$-加群から導かれるベクトル空間
- $\mathbb{Z}$ 係数の線形結合
- 自由 $\mathbb{Z}$-加群
- 自由 $\mathbb{Z}$-加群の階数

アーベル群から $\mathbb{Z}$-加群の導入については、楕円曲線暗号でも必要であり、他の分野でも利用される重要な命題である。格子は、有限階数の自由 $\mathbb{Z}$-加群にノルムを定義したものであり、$\mathbb{Z}$-加群、特に自由 $\mathbb{Z}$-加群やその階数の形式化は格子暗号において必須である。

本論文の章立てについて、以下で述べる。第 2 章では、楕円曲線に関する数学的定義や定理について説明する。第 3 章では、第 2 章で説明した楕円曲線に関する数学的定義や定理の形式化について述べる。第 4 章では、$\mathbb{Z}$-加群に関する数学的定義や定理について説明する。第 5 章では、第 4 章で説明した $\mathbb{Z}$-加群に関する数学的定義や定理の形式化について述べる。第 6 章では、結論を述べる。

第2章　楕円曲線に関する数学的定義・定理

この章では、楕円曲線についての必要性や数学的定義・定理を説明する。

2.1　楕円曲線の必要性

楕円曲線は、数学的には非特異3次曲線であり、種数1の曲線である。楕円曲線上の点に対し、加法の演算が定義でき、その演算の下で楕円曲線の点集合はアーベル群となることが知られている。また、楕円曲線は解析学的には楕円関数から導かれるものであり、2次元の格子とも関連を持つ。これらをはじめとして楕円曲線は、数論、代数、解析、位相幾何、代数幾何というほとんどの数学分野に関連し、数学理論の中で極めて重要な位置を占める数学的概念である。

楕円曲線の応用として、符号理論では楕円符号という代数幾何符号が挙げられる。また、暗号理論では重要な楕円曲線暗号で使用されていることが有名である[4]。楕円曲線暗号では、上で述べた楕円曲線のアーベル群からスカラの演算を導入し、$\mathbb{Z}$-加群を構成し、そこでのスカラ倍演算を使用する。楕円曲線暗号は、楕円曲線上の点Pのスカラs倍の点sPから、スカラsを求める楕円曲線上の離散対数問題が求解困難であることを安全性の根拠としている。楕円曲線の離散対数問題は、準指数関数時間アルゴリズムが発見されていないため、楕円曲線暗号は短い鍵長であっても、安全である暗号が構成できる。そのため、鍵などを格納するメモリサイズや楕円曲線暗号の処理時間を小さくでき、実装コストを軽減できる。それらの利点から、先に述べたBDプレーヤやアクトビラの受信機器であるTVなどで楕円曲線暗号が適用されている。

楕円曲線は、暗号理論で利用できることが判明してから、関連する様々な定理や計算アルゴリズムが発見されている。特に、楕円曲線暗号の安全性を評価するために、様々な攻撃アルゴリズムが提案されており、それらの提案により、楕円曲線の数学理論が発展している。本論文ではさらに数学理論や暗号理論が発展する礎にする目的で、Mizarにおける、楕円曲線の定義や関連する定理などの数学理論を形式化していく。

2.2 楕円曲線の定義

平面上の点に対し、以下のアフィン座標と射影座標が定義される。

定義 2.1 (座標) 平面上に存在する点 $P=(x,y)$ に対し、(x,y) を P の**アフィン座標**と呼ぶ。さらに、$P=[X,Y,Z], x=X/Z, y=Y/Z$ としたとき、$[X,Y,Z]$ を P の**射影座標**と呼ぶ。ここで除算が存在するために、厳密には、x,y,X,Y,Z はある体 K の元である。また、射影座標は厳密には $[X,Y,Z] \in K^3 \backslash \{[0,0,0]]\}$ を満たし、X,Y,Z のすべての座標が同時に 0 となることはない。

アフィン座標で点 P,Q が等しいことを、射影座標でもいうために、以下の射影座標の同値関係を導入する。

定義 2.2 (射影座標の同値) 射影平面上の 2点 $P=[X_P,Y_P,Z_P], Q=[X_Q,Y_Q,Z_Q]$ が**同値**であるとは、以下の式を満たす $c \in K \backslash \{0\}$ が存在することをいう。

$$\begin{aligned} X_Q &= cX_P \\ Y_Q &= cY_P \\ Z_Q &= cZ_P \end{aligned} \tag{2.1}$$

楕円曲線は、非特異の 3 次曲線で定義されるが、非特異 3 次曲線は以下で定義されるワイヤーシュトラスの標準形と双有理同値となるため、本論文では以下のように楕円曲線を定義する。

定義 2.3 (楕円曲線 (アフィン座標)) ワイヤーシュトラスの標準形

$$y^2 = x^3 + ax + b \tag{2.2}$$

を定義方程式に持ち、非特異である 3次曲線を**楕円曲線**と呼ぶ。楕円曲線が非特異であることと、x^3+ax+b の判別式 δ が

$$\delta = 4a^3 + 27b^2 \neq 0 \tag{2.3}$$

であることは同値である。一般に体 K 上で定義された楕円曲線とは $a,b \in K$ である楕円曲線のことをいう。楕円曲線の **K-有理点**とは、x,y 座標が K に属する点であり、その点と後で述べる無限遠点 O を含めた集合を $E(K)$ で表す。

上記の定義はアフィン座標に対する楕円曲線の定義である。射影座標に対する楕円曲線の定義は以下のとおりである。

定義 2.4 (楕円曲線 (射影座標))

$$Y^2Z = X^3 + aXZ^2 + bZ^3 \tag{2.4}$$

を定義方程式に持ち、非特異である曲線を**楕円曲線**と呼ぶ。楕円曲線が非特異であることと、$X^3 + aX + b$の判別式δが

$$\delta = 4a^3 + 27b^2 \neq 0 \tag{2.5}$$

であることは同値である。一般に体K上で定義された楕円曲線とは$a, b \in K$である楕円曲線のことをいう。楕円曲線の**K-有理点**とはX, Y, Z座標がKに属する点であり、その点全体を$E(K)$で表す。

上記で定義した楕円曲線は、a, bがどのような値であっても、その楕円曲線上の点$O = [0, 1, 0]$(無限遠点と呼ぶ) が存在する。この元は、後で述べる楕円曲線の点の演算 (加算) における単位元 (零元) となっている。なお、無限遠点はアフィン座標で表せない点であるため、アフィン座標では座標値では表さず、単にOと表す。

2.3 有限素体

楕円曲線で扱う体K(定義体と呼ぶ) は様々なものを利用できるが、楕円曲線暗号においては、大きな素数pに対する有限体$\mathbb{F}_p (p > 3)$や、$\mathbb{F}_2$の拡大体$\mathbb{F}_{2^n}$ (nは2以上の整数) を扱うことが多い。本論文では、この中の体$\mathbb{F}_p$を扱う。$\mathbb{F}_p$は有限であり、以下で定義する素体であるため、**有限素体**と呼ばれる。

定義 2.5 (素体) 自分自身以外に部分体を持たない体を、**素体**と呼ぶ。

素体は有限であれば$\mathbb{F}_p$ (pは素数)、無限であれば有理数体$\mathbb{Q}$のみ存在することが知られている。本論文では、特に以下の定理を形式化する。

定理 2.1 素数pに対し、$\mathbb{F}_p$は素体である。

この定理は、$\mathbb{F}_p$の部分体に含まれる$1 \bmod p$から$\mathbb{F}_p$の加算を用いて、$\mathbb{F}_p$のすべての元を生成できることを用いて証明する。

2.4 楕円曲線の$\mathbb{F}_p$-有理点の個数

楕円曲線の定義体を$\mathbb{F}_p$とするとき、楕円曲線上の$\mathbb{F}_p$-有理点はアフィン座標で示す場合、$P=(x_P,y_P)$ $(x_P,y_P\in\mathbb{F}_p)$で書ける。この他に楕円曲線上の点として、無限遠点$O=[0,1,0]$が存在する。

$x_P\in\mathbb{F}_p$をx座標とする点は、楕円曲線の方程式$y^2=x^3+ax+b$より、$x_P^3+ax_P+b$の平方根として持つyの個数分存在する。すなわち、$x_P^3+ax_P+b$が0でない場合、$x_P^3+ax_P+b$を平方根として持つyが存在する場合、$-y$も平方根となるため、x_Pをx座標とする点は2つ存在し、存在しない場合はそのような点は存在しない。また、$x_P^3+ax_P+b$が0の場合は、$y=0$であるため、x_Pをx座標とする点は1つ存在する。

一方、sの平方根の存在に関し、以下の平方剰余記号が定義される。

定義 2.6 (平方剰余記号) $s\in\mathbb{F}_p$に対して、**平方剰余記号** $(\frac{s}{p})$を以下のように定義する。

$$\left(\frac{s}{p}\right)=\begin{cases}1 & (s\text{の平方根が存在する場合})\\ 0 & (s=0\text{の場合})\\ -1 & (s\text{の平方根が存在しない場合})\end{cases}\tag{2.6}$$

したがって、$x_P^3+ax_P+b$の平方根の数、すなわち、x_Pをx座標とする点の個数は、平方剰余記号を使って以下のように計算できる。

定理 2.2

$$x_P\text{を}x\text{座標とする点の個数}=\left(\frac{x_P^3+ax_P+b}{p}\right)+1\tag{2.7}$$

以上より、$\mathbb{F}_p$を定義体とする楕円曲線Eの$\mathbb{F}_p$-有理点の個数($\#E(\mathbb{F}_p)$で表記)は、以下のように計算できる。

定理 2.3 $\mathbb{F}_p$を定義体とする楕円曲線Eの$\mathbb{F}_p$-有理点の個数$\#E(\mathbb{F}_p)$は、以下を満たす。

$$\#E(\mathbb{F}_p)=p+1+\sum_{x_P\in\mathbb{F}_p}\left(\frac{x_P^3+ax_P+b}{p}\right)\tag{2.8}$$

右辺第2項の1は、無限演算の点を示している。平方剰余記号を効率的に計算できるアルゴリズムが存在するため、pが小さい範囲では、定理2.3を用いて、楕円曲線の$\mathbb{F}_p$-有理点の個数$\#E(\mathbb{F}_p)$を計算することが多い。なお、楕円曲線の$\mathbb{F}_p$-有理点の個数$\#E(\mathbb{F}_p)$は、集合$E(\mathbb{F}_p)$の濃度とも呼ぶ。

2.5 楕円曲線の点の演算

楕円曲線の K-有理点の集合 $E(K)$ は、以下の演算 $+$ により、アーベル群の構造をもつ。

定義 2.7 (演算 $+$(アフィン座標)) 楕円曲線 $y^2 = x^3 + ax + b$ 上の点 $P = (x_P, y_P)$, $Q = (x_Q, y_Q) \neq -P$, $R = P + Q = (x_R, y_R)$ に対し、

$$\begin{aligned} x_R &= -x_P - x_Q + \lambda^2 \\ y_R &= -\lambda(x_R - x_P) - y_P \end{aligned} \tag{2.9}$$

で演算 $+$ を定義する。ただし、

$$\lambda = \begin{cases} \dfrac{y_Q - y_P}{x_Q - x_P} & P \neq Q \\ \dfrac{3x_P^2 + a}{2y_P} & P = Q \end{cases} \tag{2.10}$$

である。零元を $O = [0, 1, 0]$, P の逆元を $-P = (x_P, -y_P)$ とする。

アフィン座標における演算の上記定義の場合、$P + (-P)$ が厳密には定義されておらず、$P + (-P) = O$ と別に定義が必要である。それに対し、射影座標における以下の定義の場合、$P + (-P)$ も定義されている。なお、$-P = [X_P, -Y_P, Z_P]$ である。

定義 2.8 (演算 $+$(射影座標)) 楕円曲線 $Y^2Z = X^3 + aXZ^2 + bZ^3$ 上の点 $P = [X_P, Y_P, Z_P]$, $Q = [X_Q, Y_Q, Z_Q]$, $R = P + Q = [X_R, Y_R, Z_R]$ に対し、以下のように演算 $+$ を定義する。

Case 1. $P = O$ の場合、$R = Q$

Case 2. $P \neq O$ かつ、$Q = O$ の場合、$R = P$

Case 3. $P \neq O$, $Q \neq O$ かつ、$Q \neq P$ の場合、

$$X_R = vA \tag{2.11}$$

$$Y_R = u(v^2 X_P Z_Q - A) - v^3 Y_P Z_Q \tag{2.12}$$

$$Z_R = v^3 Z_P Z_Q \tag{2.13}$$

ここで, $u = Y_Q Z_P - Y_P Z_Q$, $v = X_Q Z_P - X_P Z_Q$, $A = u^2 Z_P Z_Q - v^3 - 2v^2 X_P Z_Q$ である。

Case 4. $P \neq O$, $Q \neq O$ かつ、$Q = P$ の場合、

$$X_R = 2hs \tag{2.14}$$
$$Y_R = w(4B - h) - 8Y_P^2 s^2 \tag{2.15}$$
$$Z_R = 8s^3 \tag{2.16}$$

ここで, $w = aZ_P^2 + 3X_P^2$, $s = Y_P Z_P$, $B = X_P Y_P s$, $h = w^2 - 8B$ である.

上記の定義の場合、射影座標の定義 (定義 2.1) から $[X_R, Y_R, Z_R] \neq [0, 0, 0]$ を満たす必要がある。Case 1, 2 の場合は当然満たすことは言えるが、Case 3 の場合においても必ず満たすことが証明可能である。Case 4 の場合においては、判別式 $\delta \neq 0$ である条件下で必ず満たすことが証明可能である。

第3章　楕円曲線の形式化

この章では、第2章で述べた楕円曲線に関連する定義・定理をMizarにおいて形式化する。以下ではMizarで形式化した定義・定理を数学の通常の定義・定理と区別するため、**定義** [Mizar形式化]、**定理** [Mizar形式化] として表す。なお、以下では形式化の主要な流れを辿ることを目的に説明しており、実際には他の命題・定理の形式化も必要であることを注意しておく。

3.1　素体

ここでは、2.3節で述べた有限素体をMizarにおいて形式化する。Mizarのライブラリでは、部分体の定義がないため、素体を定義する前にまず、部分体を定義する。

定義 [Mizar形式化] **3.1 (部分体)**

```
definition
  let K be Field;
  mode Subfield of K -> Field means

  the carrier of it c= the carrier of K
  & the addF of it = (the addF of K) || the carrier of it
  & the multF of it = (the multF of K) || the carrier of it
  & 1.it = 1.K & 0.it = 0.K;
end;
```

`the carrier of it` とは、部分体を構成する集合 (台集合と呼ぶ) を示し、`(the addF of K) || the carrier of it` 及び `(the multF of K) || the carrier of it` は、体 K の加算及び乗算を部分体の集合に限定するという意味である。また、`1.it`, `0.it` はそれぞれ、部分体の乗算の単位元及び加算の単位元 (零元) を示している。上記の定義は、K の部分体を、

- 構成する台集合が K の台集合の部分集合

- 乗算及び加算が K の上記部分集合に限定したもの
- 乗算と加算の単位元が同じ

で定義する、という意味である。
次に素体の定義を形式化する。

定義 [Mizar 形式化] 3.2 (素体)

```
definition let IT be Field;
  attr IT is prime means
  K1 is strict Subfield of IT implies K1 = IT;
end;
```

この定義は、定義 2.5 をそのまま書き下したものである。すなわち、部分体 $K1$ は元の体そのものという意味であり、これは自分自身以外に部分体が存在しないということと同じである。

以下で、定理 2.1 を形式化する。$\mathbb{F}_p$ が素体であることを証明も含めると以下のようになる。GF(p) は $\mathbb{F}_p$ を示している。

定理 [Mizar 形式化] 3.1 (素体 $\mathbb{F}_p$)

```
theorem
  for p be Prime holds GF(p) is prime
  proof
    set K = GF(p);
P0: p > 1 by INT_2:def 5;
    now let K1 be strict Subfield of K;
      set C = the carrier of K;
      set C1 = the carrier of K1;
      set n1 = p-1;
      reconsider n1 as Element of NAT by P0,NAT_1:20;
  A1: for x st x in K holds x in K1
      proof
    A5: for n be Element of NAT st n in Segm(p) holds n in C1
        proof
          defpred P[Nat] means $1 in C1;
          0 in Segm(p) by NAT_1:45;
          then 0.K = 0 by FUNCT_7:def 1;
          then 0.K1 = 0 by PFDef2;
          then A2: P[0];
```

```
  A3: now let n be Element of NAT such that B0: 0<=n & n<n1;
        assume B1: P[n];
    B2: 1.K1 = 1.K by PFDef2
        .= 1 by P0,INT_3:24;
        then B3: [1,n] in [:C1,C1:] by B1,ZFMISC_1:106;
    B4: the addF of K1 = (the addF of K) || C1 by PFDef2;
    B6: 1+n < n1+1 by B0,XREAL_1:8;
        n < n1+1 by B0,NAT_1:13;
        then B7: 1 in Segm(p) & n in Segm(p) by P0,NAT_1:45;
        (the addF of K1).(1,n) = (addint(p)).(1,n)
        by B3,B4,FUNCT_1:72
        .= (1+n) mod p by B7,GR_CY_1:def 5
        .= 1+n by B6,INT_3:10;
        hence P[n+1] by B1,B2,BINOP_1:29;
      end;
  A4: for n being Element of NAT st 0 <= n & n <= n1 holds
      P[n] from INT_1:sch 7(A2,A3);
      thus for n be Element of NAT st n in Segm(p) holds P[n]
      proof
        let n be Element of NAT such that
B0: n in Segm(p);
        0 <= n & n < n1+1 by B0,ALGSEQ_1:10;
        then 0 <= n & n <= n1 by NAT_1:13;
        hence P[n] by A4;
      end;
    end;
    thus for x st x in K holds x in K1
    proof
      let x be set such that B0: x in K;
      x in C by B0,STRUCT_0:def 5;
      then x in C1 by A5;
      hence x in K1 by STRUCT_0:def 5;
    end;
  end;
  K is strict Subfield of K by PF1;
  then K is strict Subfield of K1 by A1,PF8;
```

```
      hence K1 = K by PF5;
    end;
    then K1 is strict Subfield of K implies K1 = K;
    hence thesis by PFDef3;
  end;
end;
```

この証明の基本構成は以下のとおりである。

1. K = GF(p) の部分体K1を導入し、任意のKの元xがK1に含まれることを証明。
   A1: for x st x in K holds x in K1 以下の証明がそれに対応する。

2. 上記より、KがK1の部分体であり、元々K1はKの部分体であることからK1 = Kを証明。
   hence K1 = K by PF5 がそれに対応する。

上記の1は、帰納法を用いて証明している。具体的には、

(1) defpred P[Nat] means $1 in C1からthen A2: P[0] の部分で0がK1に含まれること

(2) assume B1: P[n] でnがK1に含まれると仮定するとき、hence P[n+1] でn+1がK1に含まれること

(3) (1), (2) より、帰納法を用いて、0からp-1までのすべての元がK1に含まれること

を証明している。K = GF(p) の元は0からp-1であるため、Kのすべての元をK1が含むことになる。上記 (2) は、K1が部分体であることから1.K = 1をK1が含むことを利用している。具体的には、K1の加算がK1で閉じていることから、1とnが含まれていれば、1+nもK1に含むことを示している。これは、(the addF of K1).(1,n) = ... 以下の数式変形の部分で形式化している。

3.2 平方剰余記号

ここでは、定義2.6で述べた平方剰余記号と2次方程式 $b^2 = a$ の解の個数との関係の命題を形式化する。まず、$a \bmod p$ に対する平方剰余、平方非剰余を以下のように形式化する。

定義 [Mizar 形式化] 3.3 (平方剰余)

```
definition let p, a;
  attr a is quadratic_residue means
  a <> 0 & ex x being Element of GF(p) st x|^2 = a;
  attr a is not_quadratic_residue means
  a <> 0 & not ex x being Element of GF(p) st x|^2 = a;
end;
```

すなわち、$a \bmod p \neq 0$ であり、かつ $x^2 = a$ となる x が存在する場合を**平方剰余**、存在しない場合を**平方非剰余**と定義する。次に平方剰余記号を以下のように形式化する。

定義 [Mizar 形式化] 3.4 (平方剰余記号)

```
definition let p, a;
  func Lege_p(a) -> Integer equals
  0 if a = 0,
  1 if a is quadratic_residue
  otherwise -1;
end;
```

Lege_p(a) が定義 2.6 における平方剰余記号 $(\frac{a}{p})$ を示している。この平方剰余記号を利用して、2 次方程式 $b^2 = a$ の解 b の個数に対し、以下が成り立つ。

定理 [Mizar 形式化] 3.2 (2 次方程式の解の個数)

```
theorem
  2 < p implies card({b : b|^2 = a}) = 1 + Lege_p(a);
```

これは、平方剰余記号 $(\frac{a}{p})$ に 1 を足すことで、$b^2 = a$ の解の個数になるという命題である。

3.3 楕円曲線

ここでは、射影座標とそれを用いた楕円曲線の定義を形式化する。定義 2.1 の射影座標を以下のように形式化する。

定義 [Mizar 形式化] 3.5 (射影座標)

```
definition
  let K be Field;
  func ProjCo(K) -> non empty Subset of
```

```
  [:the carrier of K, the carrier of K, the carrier of K:] equals
  [:the carrier of K, the carrier of K, the carrier of K:]
  \ {[0.K,0.K,0.K]};
end;
```

定義2.2の射影座標の同値関係を以下のように形式化する。以下では、体Kが$\mathbb{F}_p$の場合で形式化している。

定義 [Mizar形式化] 3.6 (射影座標の同値)

```
definition
  let p be Prime;
  let P,Q be Element of ProjCo(GF(p));
  pred P _EQ_ Q means
    ex a be Element of GF(p) st a <> 0.GF(p)
    & P‘1 = a*Q‘1 & P‘2 = a*Q‘2 & P‘3 = a*Q‘3;
  reflexivity;
  symmetry;
end;
```

P‘1, Q‘1がX_P, X_Q, P‘2, Q‘2がY_P, Y_Q, P‘3, Q‘3がZ_P, Z_Qをそれぞれ示している。

さらに、楕円曲線の点の集合を形式化する準備として、判別式δ (以下ではDisc)と楕円曲線の定義方程式EC_WEqProjCoを以下のように形式化する。

定義 [Mizar形式化] 3.7 (楕円曲線の判別式)

```
definition
  let p be Prime;
  let a, b be Element of GF(p);
  func Disc(a,b,p) -> Element of GF(p) means
  for g4, g27 be Element of GF(p) st g4 = 4 mod p &
  g27 = 27 mod p
  holds it = g4*a|^3 + g27*b|^2;
end;
```

定義 [Mizar形式化] 3.8 (楕円曲線の定義方程式)

```
definition
  let p be Prime;
  let a, b be Element of GF(p);
```

```
  func EC_WEqProjCo(a,b,p) -> Function of
  [:the carrier of GF(p), the carrier of GF(p),
  the carrier of GF(p):], GF(p) means
  for P be Element of [:the carrier of GF(p),
  the carrier of GF(p), the carrier of GF(p):] holds
  it. P = ((P‘2) |^2)*(P‘3)-((P‘1) |^3 +a*(P‘1)*(P‘3) |^2
  +b*(P‘3) |^3);
end;
```

楕円曲線の$\mathbb{F}_p$-有理点の集合は、定義方程式を用いて、以下のように形式化する。

定義 [Mizar 形式化] 3.9 (楕円曲線の$\mathbb{F}_p$-有理点の集合)

```
definition
  let p be Prime;
  let a, b be Element of GF(p);
  func EC_SetProjCo(a,b,p) -> non empty Subset of ProjCo(GF(p))
  equals {P where P is Element of ProjCo(GF(p)) :
  EC_WEqProjCo(a,b,p).P = 0.GF(p)};
end;
```

上記は、楕円曲線の点の集合を定義方程式が0となる点の集合で定義している。

3.4 楕円曲線の$\mathbb{F}_p$-有理点の個数

ここでは、定理2.3を形式化する。

まず、楕円曲線の任意の$\mathbb{F}_p$-有理点が$O=[0,1,0]$または$P=[X,Y,1]$の形の点のどちらか射影座標として同値になることを以下のように形式化する。

定理 [Mizar 形式化] 3.3 (楕円曲線の点の同値)

```
theorem
  for p be Prime, a, b be Element of GF(p), x be set st
  p > 3 & Disc(a,b,p) <> 0.GF(p)
  & x in Class (R_EllCur(a,b,p)) holds
  ( ex P be Element of ProjCo(GF(p)) st P in EC_SetProjCo(a,b,p)
  & P=[0,1,0]
  & x = Class(R_EllCur(a,b,p),P) ) or
  ex P be Element of ProjCo(GF(p)), X,Y be Element of GF(p)
  st P in EC_SetProjCo(a,b,p) & P=[X,Y,1]
  & x = Class(R_EllCur(a,b,p),P);
```

Class(R_EllCur(a,b,p),P) は、定義 [Mizar 形式化]3.6 で形式化した同値関係_EQ_ の P を含む同値類を示している。また、Class(R_EllCur(a,b,p)) は同値類全体、すなわちすべての $\mathbb{F}_p$-有理点の集合を示している。この定理は、

> 楕円曲線の $\mathbb{F}_p$-有理点である x は、P = [0,1,0] を含む同値類に含まれるか、P = [X, Y ,1] を含む同値類に含まれる

という意味である。定理 [Mizar 形式化]3.3 を利用して、以下の定理が形式化できる。

定理 [Mizar 形式化] 3.4 (楕円曲線の点の同値類)

```
theorem
  for p be Prime, a, b be Element of GF(p) st
  p > 3 & Disc(a,b,p) <> 0.GF(p) holds
  Class (R_EllCur(a,b,p)) = {Class(R_EllCur(a,b,p),[0,1,0])}
  \/ {Class(R_EllCur(a,b,p),P)
  where P is Element of ProjCo(GF(p)):
  P in EC_SetProjCo(a,b,p) & ex X,Y be Element of GF(p)
  st P=[X,Y,1]};
```

上記定理は、楕円曲線の $\mathbb{F}_p$-有理点の同値類全体は、P = [0,1,0] を含む同値類と、P = [X, Y ,1] を含む同値類の和集合であることを示している。

次に、定理 2.2 を以下のように形式化する。

定理 [Mizar 形式化] 3.5 (X 座標と $\mathbb{F}_p$-有理点の個数)

```
theorem
  for p be Prime, a, b, d be Element of GF(p) st
  p > 3 & Disc(a,b,p) <> 0.GF(p) holds
  card ({Class(R_EllCur(a,b,p),[d,Y,1]) where Y is Element of GF(p)
  : [d,Y,1] in EC_SetProjCo(a,b,p) }) = 1 + Lege_p(d|^3 + a*d + b);
```

上記定理は、X 座標が d であり、Z 座標が 1、すなわち、x 座標が d の点の個数が 1 + Lege_p(d|^3 + a*d + b) であることを示している。この定理の証明では、定理 [Mizar 形式化]3.2 を用いる。

以下で点の個数を数え上げるため、それぞれの同値類が異なることを示す。まずは、$O = [0,1,0]$ と $P = [X,Y,1]$ のそれぞれの同値類が交わらないことを以下のように形式化する。

定理 [Mizar 形式化] 3.6 (同値類の関係 1)

```
theorem
  for p be Prime, a, b be Element of GF(p),
```

```
F1,F2 be set st p > 3 & Disc(a,b,p) <> 0.GF(p)
& F1 = {Class(R_EllCur(a,b,p),[0,1,0])} &
F2 = {Class(R_EllCur(a,b,p),P) where P is
Element of ProjCo(GF(p)): P in EC_SetProjCo(a,b,p) &
ex X,Y be Element of GF(p) st P=[X,Y,1]} holds F1 misses F2;
```

さらに、X 座標が異なる $[X, Y, 1]$ の形をした点の 2 つの同値類が交わらないことを以下のように形式化する。

定理 [Mizar 形式化] 3.7 (同値類の関係 2)

```
theorem
  for p be Prime,
  a, b,d1,Y1,d2,Y2 be Element of GF(p)
  st p > 3 & Disc(a,b,p) <> 0.GF(p)
  & [d1,Y1,1] in EC_SetProjCo(a,b,p)
  & [d2,Y2,1] in EC_SetProjCo(a,b,p) holds
  Class(R_EllCur(a,b,p),[d1,Y1,1]) =
  Class(R_EllCur(a,b,p),[d2,Y2,1]) iff d1=d2 & Y1=Y2;
```

上記定理の意味は、X 座標が d1 の [d1, Y1, 1] の同値類と X 座標が d2 の [d2, Y2, 1] の同値類が等しいことと、d1 = d2 & Y1 = Y2 が等価であることを示している。この対偶を取ると、d1 と d2 が等しくないとき、[d1, Y1, 1] の同値類と [d2, Y2, 1] の同値類が交わらないことが言える。

以上で、定理 2.3 の形式化の準備が整った。定理 2.3 は以下のように形式化する。

定理 [Mizar 形式化] 3.8 (楕円曲線の $\mathbb{F}_p$-有理点の個数)

```
theorem
  for p be Prime, a, b be Element of GF(p)
  st p > 3 & Disc(a,b,p) <> 0.GF(p)
  ex F be FinSequence of INT st len F = p &
  (for n be Nat st n in Seg p ex d be Element of GF(p)
  st d=n-1 & F.n = Lege_p(d|^3 + a*d + b)) &
  card(Class(R_EllCur(a,b,p))) = 1 + p + Sum(F);
```

3.5 楕円曲線の点の演算

ここでは、2.5 節で述べた楕円曲線の点の演算を形式化する。

まず、P の逆元 $-P = [X_P, -Y_P, Z_P]$ を以下のように形式化する。

定義 [Mizar 形式化] 3.10 (楕円曲線の点の逆元)

```
definition
  let p be 5_or_greater Prime;
  let z be Element of EC_WParam p;
  func compell_ProjCo(z,p) ->
  Function of EC_SetProjCo(z‘1,z‘2,p), EC_SetProjCo(z‘1,z‘2,p)
  means
  for P be Element of EC_SetProjCo(z‘1,z‘2,p)
  holds it.P = [P‘1,-P‘2,P‘3];
end;
```

上記の compell_ProjCo(z,p).P が $-P$ を示している。5_or_greater Prime とは、5 以上の (3 より大きな) 素数を示す。EC_WParam p とは、p に対し、判別式 $\delta = 4a^3 + 27b^2 \neq 0$ を満たすパラメータ a, b である。上記では、z‘1 が a、z‘2 が b を示している。

なお、$P = -P \neq O$ であれば、$[X_P, Y_P, Z_P] = [X_P, -Y_P, Z_P]$ であるため、$Y_P = -Y_P$ 及び $p > 3$ より、$Y_P = 0$ であることが導かれる。このことは以下のように形式化する。

定理 [Mizar 形式化] 3.9 (楕円曲線の 2 等分点)

```
theorem
  for p be 5_or_greater Prime, z be Element of EC_WParam p,
  P be Element of EC_SetProjCo(z‘1,z‘2,p) st P‘3 <> 0 holds
  P _EQ_ compell_ProjCo(z,p).P iff P‘2 = 0;
```

$P = -P$ より、$2P = O$ であるため、このような P は 2 等分点ともいう。

定義 2.8 における楕円曲線の加算 $+$ は以下のように形式化する。

定義 [Mizar 形式化] 3.11 (楕円曲線の演算 $+$)

```
definition
  let p be 5_or_greater Prime,
      z be Element of EC_WParam p;

  func addell_ProjCo(z,p) -> Function of
  [:EC_SetProjCo(z‘1,z‘2,p),EC_SetProjCo(z‘1,z‘2,p):],
  EC_SetProjCo(z‘1,z‘2,p) means
  for P, Q, O being Element of EC_SetProjCo(z‘1,z‘2,p) st
  O = [0,1,0]
```

```
  holds
  (P _EQ_ 0 implies it.(P,Q) = Q) &
  ((Q _EQ_ 0 & not P _EQ_ 0) implies it.(P,Q) = P) &
  ((not P _EQ_ 0 & not Q _EQ_ 0 & not P _EQ_ Q) implies
  for g2, gf1, gf2, gf3 being Element of GF(p)
  st g2 = 2 mod p & gf1 = Q‘2*P‘3 - P‘2*Q‘3 &
  gf2 = Q‘1*P‘3 - P‘1*Q‘3 &
  gf3 = (gf1 |^2)*P‘3*Q‘3 - (gf2 |^3) - g2*(gf2 |^2)*P‘1*Q‘3
  holds it.(P,Q) = [gf2*gf3,
  gf1 * ((gf2 |^2)*P‘1*Q‘3-gf3) - (gf2 |^3)*P‘2*Q‘3,
  (gf2 |^3)*P‘3*Q‘3]) &
  ((not P _EQ_ 0 & not Q _EQ_ 0 & P _EQ_ Q) implies
  for g2, g3, g4, g8, gf1, gf2, gf3, gf4 being Element of GF(p)
  st g2 = 2 mod p & g3 = 3 mod p & g4 = 4 mod p &
  g8 = 8 mod p & gf1 = (z‘1)*(P‘3 |^2) + g3*(P‘1 |^2) &
  gf2 = P‘2*P‘3 &
  gf3 = P‘1*P‘2*gf2 &
  gf4 = (gf1 |^2) - g8*gf3
  holds it.(P,Q) = [g2*gf4*gf2,
  gf1*(g4 * gf3-gf4) - g8*(P‘2 |^2)*(gf2 |^2),
  g8*(gf2 |^3)]);
end;
```

上記では、定義2.8における Case 1 を

```
(P _EQ_ 0 implies it.(P,Q) = Q)
```

で表す。Case 2 を

```
((Q _EQ_ 0 & not P _EQ_ 0) implies it.(P,Q) = P)
```

で表す。Case 3 を

```
((not P _EQ_ 0 & not Q _EQ_ 0 & not P _EQ_ Q) implies
 for g2, gf1, gf2, gf3 being Element of GF(p)
 st g2 = 2 mod p & gf1 = Q‘2*P‘3 - P‘2*Q‘3 &
 gf2 = Q‘1*P‘3 - P‘1*Q‘3 &
 gf3 = (gf1 |^2)*P‘3*Q‘3 - (gf2 |^3) - g2*(gf2 |^2)*P‘1*Q‘3
 holds it.(P,Q) = [gf2*gf3,
```

```
gf1 * ((gf2 |^2)*P‘1*Q‘3-gf3) - (gf2 |^3)*P‘2*Q‘3,
(gf2 |^3)*P‘3*Q‘3])
```

で表す。ここで、gf1 は定義 2.8 の u, gf2 は v, gf3 は A をそれぞれ示している。Case 4 を

```
((not P _EQ_ 0 & not Q _EQ_ 0 & P _EQ_ Q) implies
 for g2, g3, g4, g8, gf1, gf2, gf3, gf4 being Element of GF(p)
 st g2 = 2 mod p & g3 = 3 mod p & g4 = 4 mod p &
 g8 = 8 mod p & gf1 = (z‘1)*(P‘3 |^2) + g3*(P‘1 |^2) &
 gf2 = P‘2*P‘3 &
 gf3 = P‘1*P‘2*gf2 &
 gf4 = (gf1 |^2) - g8*gf3
 holds it.(P,Q) = [g2*gf4*gf2,
 gf1*(g4 * gf3-gf4) - g8*(P‘2 |^2)*(gf2 |^2),
 g8*(gf2 |^3)])
```

で表す。ここで、gf1 は定義 2.8 の w, gf2 は s, gf3 は B, gf4 は h をそれぞれ示している。

上記では、演算した結果 addell_ProjCo(z,p).(P,Q)(以下では R (通常の数式で示す場合は $R = P+Q$) とする) が楕円曲線の $\mathbb{F}_p$ の集合 EC_SetProjCo(z‘1,z‘2,p) に含まれる必要がある。このことは、

1. R が ProjCo(GF(p)) に含まれる。すなわち、addell_ProjCo(z,p).(P,Q) が [0, 0, 0] でない。

2. R が EC_WEqProjCo(z‘1,z‘2,p).R = 0.GF(p) を満たす。

を示している。

上記の 1 については、2.5 節で述べたように Case 1, 2 の場合は自明に満たすことが言える。Case 3 の場合は、以下のようになる。

- $P = -Q$(P _EQ_ compell_ProjCo(z,p).Q) のとき:
  定義 2.8 より、$v = 0$ となるが $u \neq 0$ であるため、Y_R が 0 にならない。

- $P \neq -Q$(not P _EQ_ compell_ProjCo(z,p).Q) のとき:
  定義 2.8 より、$v \neq 0$ であるため、Z_R が 0 にならない。

Case 4 の場合は、以下のようになる。

- $Y_P = 0$(すなわち、Pが2等分点)のとき:
  後で述べる定理より$w \neq 0$であるため、$X_R = Z_R = 0$であるが、$Y_R \neq 0$である(すなわち、$R = O$である)。

- $Y_P \neq 0$のとき:
  $s \neq 0$であるため、$Z_R \neq 0$である。

$Y_P = 0$の場合に対する定理を以下で述べる。

定理 [Mizar 形式化] 3.10 (楕円曲線の判別式と2等分点)

```
theorem
  for p be 5_or_greater Prime, z be Element of EC_WParam p,
  g3 be Element of GF(p), P be Element of EC_SetProjCo(z‘1,z‘2,p)
  st g3 = 3 mod p & P‘2 = 0 & P‘3 <> 0 holds
  (z‘1)*(P‘3 |^2) + g3*(P‘1 |^2) <> 0;
```

この定理の意味は、$Y_P = 0$である$P \neq O$の点に対し、$w \neq 0$が成り立つということである。この定理は、楕円曲線の判別式$\delta \neq 0$であることを用いて証明する[1]。

上記の2(`R`が`EC_WEqProjCo(z‘1,z‘2,p).R = 0.GF(p)`を満たす)は、複雑な式変形をしていくことで証明が可能である。以下でその式変形で必要な関係式を形式化していく。

以下では、まず、アフィン座標において、楕円曲線の加算及び2倍算の公式を導く。さらに、その際に出現する関係式を示す。Mizarでは、その関係式が楕円曲線の加算及び2倍算で成り立つことを証明している。

楕円曲線の点$P, Q(P \neq Q)$の加算$R = P + Q$は、図3.1のように、PとQを結ぶ直線が楕円曲線と交わる点を$-R$とし、そのy座標を負の値に反転させた点をRとして得る。

P, Qを通る直線は、以下の方程式で与えられる。

$$\begin{aligned} y &= \lambda x + \nu \\ \lambda &= \frac{y_Q - y_P}{x_Q - x_P} \\ \nu &= y_P - x_P \times \frac{y_Q - y_P}{x_Q - x_P} \end{aligned} \tag{3.1}$$

したがって、上記直線と楕円曲線の交点は、以下の式を満たす。

$$(\lambda x + \nu)^2 = x^3 + ax + b \tag{3.2}$$

[1]楕円曲線の判別式は、2等分点が非特異でない、すなわち、2等分点で接線が引けることを判定するための式である。2等分点以外の楕円曲線上の点は、判別式に依らず、非特異になることが知られている。したがって、判別式が0でなければ、楕円曲線上のすべての点が非特異になる。定理 [Mizar 形式化]3.10 は、2等分点で接線が引けることを示している。

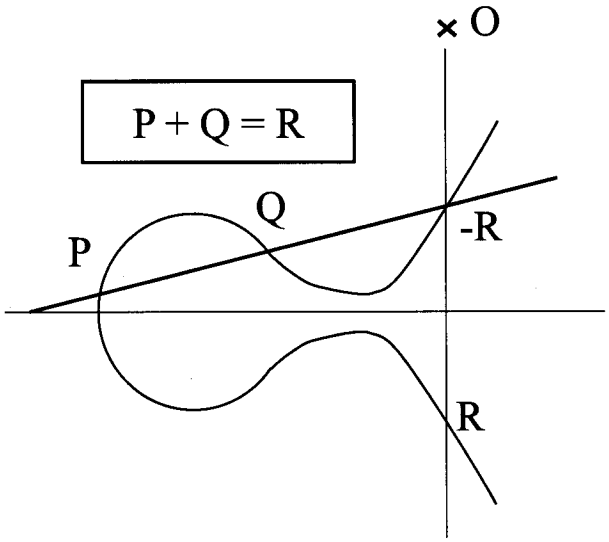

図 3.1: 楕円曲線の加算

この式を変形すると、

$$x^3 - \lambda^2 x^2 + (a - 2\lambda\nu)x + (b + \nu^2) = 0 \tag{3.3}$$

上式は、$x = x_P, x = x_Q, x = x_R$ のそれぞれで成り立つため、上式左辺の x の多項式は x_P, x_Q, x_R を根として持つ。したがって、

$$\begin{aligned} & x^3 - \lambda^2 x^2 + (a - 2\lambda\nu)x + (b + \nu^2) \\ = \ & (x - x_P)(x - x_Q)(x - x_R) \end{aligned} \tag{3.4}$$

上式の両辺を x の多項式として係数を比較すると、x^3 の係数はどちらも 1 であるため、2, 1, 0 次の係数に対し比較すると、

$$-\lambda^2 = -(x_P + x_Q + x_R) \tag{3.5}$$

$$a - 2\lambda\nu = x_P x_Q + x_Q x_R + x_R x_P \tag{3.6}$$

$$b + \nu^2 = -x_P x_Q x_R \tag{3.7}$$

が成り立つことがわかる。

また、直線を $-R$ が通るので、式 3.1 より、

$$-y_R = \frac{y_Q - y_P}{x_Q - x_P} x_R + \left(y_P - x_P \times \frac{y_Q - y_P}{x_Q - x_P} \right) \tag{3.8}$$

である。

逆に、定義2.7の演算＋の公式において、x_R及びy_Rが式3.8, 3.5, 3.6及び3.7を満たせば、Rが楕円曲線上の点であることを示せる。このことに着目して、Mizarで以下の定理を形式化した。なお、Mizarによる形式化ではアフィン座標ではなく、射影座標において式3.8, 3.5, 3.6及び3.7が成り立つことを形式化した。

式3.8については、以下のように形式化した。

定理 [Mizar形式化] 3.11 (楕円曲線の加算に関する関係式1)

```
theorem
  for p be 5_or_greater Prime, z be Element of EC_WParam p,
  g2, gf1, gf2, gf3 be Element of GF(p),
  P, Q be Element of EC_SetProjCo(z‘1,z‘2,p),
  R be Element of
  [:the carrier of GF(p), the carrier of GF(p),
    the carrier of GF(p):]
  st g2 = 2 mod p & gf1 = Q‘2*P‘3 - P‘2*Q‘3 &
     gf2 = Q‘1*P‘3 - P‘1*Q‘3 &
     gf3 = (gf1 |^2)*P‘3*Q‘3 - (gf2 |^3)
           - g2*(gf2 |^2)*P‘1*Q‘3 &
     R = [gf2*gf3, gf1*((gf2 |^2)*P‘1*Q‘3-gf3)
          -(gf2 |^3)*P‘2*Q‘3, (gf2 |^3)*P‘3*Q‘3]
  holds gf2*P‘3*R‘2
     = -(gf1*(R‘1*P‘3-P‘1*R‘3)+gf2*P‘2*R‘3);
```

式3.5については、以下のように形式化した。

定理 [Mizar形式化] 3.12 (楕円曲線の加算に関する関係式2)

```
theorem
  for p be 5_or_greater Prime, z be Element of EC_WParam p,
  g2, gf1, gf2, gf3 be Element of GF(p),
  P, Q be Element of EC_SetProjCo(z‘1,z‘2,p),
  R be Element of
  [:the carrier of GF(p), the carrier of GF(p),
    the carrier of GF(p):]
  st g2 = 2 mod p & gf1 = Q‘2*P‘3 - P‘2*Q‘3 &
     gf2 = Q‘1*P‘3 - P‘1*Q‘3 &
     gf3 = (gf1 |^2)*P‘3*Q‘3 - (gf2 |^3)
           - g2*(gf2 |^2)*P‘1*Q‘3 &
```

```
    R = [gf2*gf3, gf1 * ((gf2 |^2)*P‘1*Q‘3-gf3)
         -(gf2 |^3)*P‘2*Q‘3, (gf2 |^3)*P‘3*Q‘3]
  holds -(gf2 |^2)*(P‘3*Q‘3*R‘1
    + P‘3*Q‘1*R‘3+P‘1*Q‘3*R‘3)
    + P‘3*Q‘3*R‘3*(gf1 |^2) = 0.GF(p);
```

式3.6については、以下のように形式化した。

定理 [Mizar 形式化] 3.13 (楕円曲線の加算に関する関係式 3)

```
theorem
  for p be 5_or_greater Prime, z be Element of EC_WParam p,
  g2, gf1, gf2, gf3 be Element of GF(p),
  P, Q be Element of EC_SetProjCo(z‘1,z‘2,p),
  R be Element of
  [:the carrier of GF(p), the carrier of GF(p),
    the carrier of GF(p):]
  st g2 = 2 mod p & gf1 = Q‘2*P‘3 - P‘2*Q‘3 &
     gf2 = Q‘1*P‘3 - P‘1*Q‘3 &
     gf3 = (gf1 |^2)*P‘3*Q‘3 - (gf2 |^3)
           - g2*(gf2 |^2)*P‘1*Q‘3 &
     R = [gf2*gf3, gf1 * ((gf2 |^2)*P‘1*Q‘3-gf3)
          -(gf2 |^3)*P‘2*Q‘3, (gf2 |^3)*P‘3*Q‘3]
  holds z‘1*(gf2 |^2)*P‘3*Q‘3*R‘3 =
    (gf2 |^2)*(P‘1*Q‘1*R‘3+P‘3*Q‘1*R‘1
    + P‘1*Q‘3*R‘1)
    + g2*gf1*Q‘3*R‘3*(gf2*P‘2 - gf1*P‘1);
```

式3.7については、以下のように形式化した。

定理 [Mizar 形式化] 3.14 (楕円曲線の加算に関する関係式 4)

```
theorem
  for p be 5_or_greater Prime, z be Element of EC_WParam p,
  g2, gf1, gf2, gf3 be Element of GF(p),
  P, Q be Element of EC_SetProjCo(z‘1,z‘2,p),
  R be Element of
  [:the carrier of GF(p), the carrier of GF(p),
    the carrier of GF(p):]
  st g2 = 2 mod p & gf1 = Q‘2*P‘3 - P‘2*Q‘3 &
```

```
    gf2 = Q‘1*P‘3 - P‘1*Q‘3 &
    gf3 = (gf1 |^2)*P‘3*Q‘3 - (gf2 |^3)
          - g2*(gf2 |^2)*P‘1*Q‘3 &
    R = [gf2*gf3, gf1 * ((gf2 |^2)*P‘1*Q‘3-gf3)
         -(gf2 |^3)*P‘2*Q‘3, (gf2 |^3)*P‘3*Q‘3]
  holds z‘2*(gf2 |^2)*(P‘3 |^2)*Q‘3*R‘3
    = -(gf2 |^2)*P‘3*P‘1*Q‘1*R‘1
      + ((gf2*P‘2 - gf1*P‘1) |^2)*Q‘3*R‘3;
```

定理 [Mizar 形式化]3.11, 3.12, 3.13, 3.14 を用いて、以下の関係式を形式化した。

定理 [Mizar 形式化] 3.15 (楕円曲線の加算に関する関係式 5)

```
theorem
  for p be 5_or_greater Prime, z be Element of EC_WParam p,
  g2, gf1, gf2, gf3 be Element of GF(p),
  P, Q be Element of EC_SetProjCo(z‘1,z‘2,p),
  R be Element of
  [:the carrier of GF(p), the carrier of GF(p),
    the carrier of GF(p):]
  st g2 = 2 mod p & gf1 = Q‘2*P‘3 - P‘2*Q‘3 &
     gf2 = Q‘1*P‘3 - P‘1*Q‘3 &
     gf3 = (gf1 |^2)*P‘3*Q‘3 - (gf2 |^3)
           - g2*(gf2 |^2)*P‘1*Q‘3 &
     R = [gf2*gf3, gf1 * ((gf2 |^2)*P‘1*Q‘3-gf3)
          -(gf2 |^3)*P‘2*Q‘3, (gf2 |^3)*P‘3*Q‘3]
  holds (gf2 |^2)*(P‘3 |^2)*Q‘3*((R‘2 |^2)*R‘3
    - ((R‘1 |^3) + z‘1*R‘1*(R‘3 |^2) + z‘2*(R‘3 |^3)))
    = 0.GF(p);
```

定理 [Mizar 形式化]3.15 を変形することにより、R が EC_WEqProjCo(z‘1,z‘2,p).R = 0.GF(p) を満たすことを示せる。

楕円曲線の点 P の 2 倍算 $R = 2P$ は、図 3.2 のように、P を通る接線が楕円曲線と交わる点を $-R$ とし、その y 座標を負の値に反転させた点を R として得る。

P を通る接線は、以下の方程式で与えられる。

$$\begin{aligned} y &= \lambda x + \nu \\ \lambda &= \frac{3x_P^2 + a}{2y_P} \\ \nu &= y_P - x_P \times \frac{3x_P^2 + a}{2y_P} \end{aligned} \tag{3.9}$$

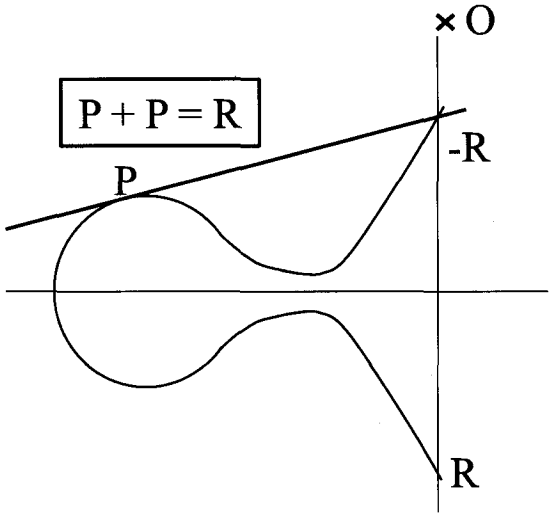

図 3.2: 楕円曲線の 2 倍算

したがって、上記直線と楕円曲線の交点は、以下の式を満たす。

$$(\lambda x + \nu)^2 = x^3 + ax + b \tag{3.10}$$

この式を変形すると、

$$x^3 - \lambda^2 x^2 + (a - 2\lambda\nu)x + (b + \nu^2) = 0 \tag{3.11}$$

上式は、$x = x_P, x = x_Q, x = x_R$ のそれぞれで成り立つため、上式左辺の x の多項式は x_P, x_R を根 (x_P は 2 重根) として持つ。したがって、

$$\begin{aligned} & x^3 - \lambda^2 x^2 + (a - 2\lambda\nu)x + (b + \nu^2) \\ = \ & (x - x_P)^2 (x - x_R) \end{aligned} \tag{3.12}$$

上式の両辺を x の多項式として係数を比較すると、x^3 の係数はどちらも 1 であるため、2, 1, 0 次の係数に対し比較すると、

$$-\lambda^2 = -(2x_P + x_R) \tag{3.13}$$

$$a - 2\lambda\nu = x_P^2 + 2x_P x_R \tag{3.14}$$

$$b + \nu^2 = -x_P^2 x_R \tag{3.15}$$

が成り立つことがわかる。

また、直線を $-R$ が通るので、式3.9より、

$$-y_R = \frac{y_Q - y_P}{x_Q - x_P} x_R + \left(y_P - x_P \times \frac{y_Q - y_P}{x_Q - x_P} \right) \tag{3.16}$$

である。

逆に、定義2.7の演算 $+$ の公式において、x_R 及び y_R が式3.16, 3.13, 3.14及び3.15を満たせば、R が楕円曲線上の点であることを示せる。このことに着目して、Mizarで以下の定理を形式化した。なお、加算のときと同様に、Mizarによる形式化ではアフィン座標ではなく、射影座標において式3.16, 3.13, 3.14及び3.15が成り立つことを形式化した。

式3.16については、以下のように形式化した。

定理 [Mizar形式化] 3.16 (楕円曲線の2倍算に関する関係式1)

```
theorem
  for p be 5_or_greater Prime, z be Element of EC_WParam p,
  g2, g3, g4, g8, gf1, gf2, gf3, gf4 be Element of GF(p),
  P be Element of EC_SetProjCo(z‘1,z‘2,p),
  R be Element of
  [:the carrier of GF(p), the carrier of GF(p),
    the carrier of GF(p):]
  st g2 = 2 mod p & g3 = 3 mod p & g4 = 4 mod p & g8 = 8 mod p &
     gf1 = z‘1*(P‘3 |^2) + g3*(P‘1 |^2) &
     gf2 = P‘2*P‘3 &
     gf3 = P‘1*P‘2*gf2 & gf4 = (gf1 |^2) - g8*gf3 &
     R = [g2*gf4*gf2, gf1*(g4*gf3-gf4)-g8*(P‘2 |^2)*(gf2 |^2),
          g8*(gf2 |^3)]
  holds g2*gf2*P‘3*R‘2
    = -(gf1*(P‘3*R‘1 - P‘1*R‘3)
      + g2*gf2*P‘2*R‘3);
```

式3.13については、以下のように形式化した。

定理 [Mizar形式化] 3.17 (楕円曲線の2倍算に関する関係式2)

```
theorem
  for p be 5_or_greater Prime, z be Element of EC_WParam p,
  g2, g3, g4, g8, gf1, gf2, gf3, gf4 be Element of GF(p),
  P be Element of EC_SetProjCo(z‘1,z‘2,p),
  R be Element of
```

```
  [:the carrier of GF(p), the carrier of GF(p),
    the carrier of GF(p):]
  st g2 = 2 mod p & g3 = 3 mod p & g4 = 4 mod p & g8 = 8 mod p &
     gf1 = z‘1*(P‘3 |^2) + g3*(P‘1 |^2) &
     gf2 = P‘2*P‘3 &
     gf3 = P‘1*P‘2*gf2 & gf4 = (gf1 |^2) - g8*gf3 &
     R = [g2*gf4*gf2, gf1*(g4*gf3-gf4)-g8*(P‘2 |^2)*(gf2 |^2),
          g8*(gf2 |^3)]
  holds g4*(gf2 |^2)*P‘3*R‘1
    = R‘3*((gf1 |^2)*P‘3 - g8*(gf2 |^2)*P‘1);
```

式3.14については、以下のように形式化した。

定理 [Mizar 形式化] 3.18 (楕円曲線の 2 倍算に関する関係式 3)

```
theorem
  for p be 5_or_greater Prime, z be Element of EC_WParam p,
  g2, g3, g4, g8, gf1, gf2, gf3, gf4 be Element of GF(p),
  P be Element of EC_SetProjCo(z‘1,z‘2,p),
  R be Element of
  [:the carrier of GF(p), the carrier of GF(p),
    the carrier of GF(p):]
  st g2 = 2 mod p & g3 = 3 mod p & g4 = 4 mod p & g8 = 8 mod p &
     gf1 = z‘1*(P‘3 |^2) + g3*(P‘1 |^2) &
     gf2 = P‘2*P‘3 &
     gf3 = P‘1*P‘2*gf2 & gf4 = (gf1 |^2) - g8*gf3 &
     R = [g2*gf4*gf2, gf1*(g4*gf3-gf4)-g8*(P‘2 |^2)*(gf2 |^2),
          g8*(gf2 |^3)]
  holds g2*(gf2 |^2)*(P‘3 |^2)*(z‘1*R‘3)
    = gf1*P‘3*R‘3*(g2*gf2*P‘2-gf1*P‘1)
      + (gf2 |^2)*(g4*P‘1*P‘3*R‘1
                   + g2*(P‘1 |^2)*R‘3);
```

式3.15については、以下のように形式化した。

定理 [Mizar 形式化] 3.19 (楕円曲線の 2 倍算に関する関係式 4)

```
theorem
  for p be 5_or_greater Prime, z be Element of EC_WParam p,
  g2, g3, g4, g8, gf1, gf2, gf3, gf4 be Element of GF(p),
```

```
P be Element of EC_SetProjCo(z‘1,z‘2,p),
R be Element of
[:the carrier of GF(p), the carrier of GF(p),
  the carrier of GF(p):]
st g2 = 2 mod p & g3 = 3 mod p & g4 = 4 mod p & g8 = 8 mod p &
   gf1 = z‘1*(P‘3 |^2) + g3*(P‘1 |^2) &
   gf2 = P‘2*P‘3 &
   gf3 = P‘1*P‘2*gf2 & gf4 = (gf1 |^2) - g8*gf3 &
   R = [g2*gf4*gf2, gf1*(g4*gf3-gf4)-g8*(P‘2 |^2)*(gf2 |^2),
        g8*(gf2 |^3)]
holds g4*(gf2 |^2)*(P‘3 |^2)*(z‘2*R‘3)
  = R‘3*((g2*gf2*P‘2-gf1*P‘1) |^2)
    - g4*(gf2 |^2)*(P‘1 |^2)*R‘1;
```

定理 [Mizar 形式化]3.16, 3.17, 3.18, 3.19 を用いて、以下の関係式を形式化した。

定理 [Mizar 形式化] 3.20 (楕円曲線の 2 倍算に関する関係式 5)

```
theorem
  for p be 5_or_greater Prime, z be Element of EC_WParam p,
  g2, g3, g4, g8, gf1, gf2, gf3, gf4 be Element of GF(p),
  P be Element of EC_SetProjCo(z‘1,z‘2,p),
  R be Element of
  [:the carrier of GF(p), the carrier of GF(p),
    the carrier of GF(p):]
  st g2 = 2 mod p & g3 = 3 mod p & g4 = 4 mod p & g8 = 8 mod p &
     gf1 = z‘1*(P‘3 |^2) + g3*(P‘1 |^2) &
     gf2 = P‘2*P‘3 &
     gf3 = P‘1*P‘2*gf2 & gf4 = (gf1 |^2) - g8*gf3 &
     R = [g2*gf4*gf2, gf1*(g4*gf3-gf4)-g8*(P‘2 |^2)*(gf2 |^2),
          g8*(gf2 |^3)]
  holds g4*(gf2 |^2)*(P‘3 |^2)*((R‘2 |^2)*R‘3
    - ((R‘1 |^3) + z‘1*R‘1*(R‘3 |^2) + z‘2*(R‘3 |^3)))
    = 0.GF(p);
```

定理 [Mizar 形式化]3.20 を変形することにより、R が EC_WEqProjCo(z‘1,z‘2,p).R = 0.GF(p) を満たすことを示せる。

第4章　$\mathbb{Z}$-加群に関する数学的定義・定理

この章では、$\mathbb{Z}$-加群についての必要性や数学的定義・定理を説明する。

4.1　格子と$\mathbb{Z}$-加群

格子は、ベクトル空間上の離散点の集合である。例えば、図4.1のような規則正しく並んだ点(ベクトル)の集合である。この図は2次元の格子の例である。図のように、ベクトルv_1とv_2は格子に含まれ、$v_3 = v_1 + v_2$もこの格子に含まれている。すなわち、格子に含まれる任意のベクトルの和は、格子に含まれる。格子の任意のベクトルは、互いに一次独立なベクトルの和になっているため、格子は後で述べる$\mathbb{Z}$-加群の構造を持っている。格子に含まれる一次独立なベクトルの数を次元(もしくは階数)、一次独立なベクトル全体を基底と呼ぶ。格子の基底は、1通りとは限らない。例えば、図4.1のように(v_1, v_2)は格子の基底であるが、(v_3, v_2)も基底である。

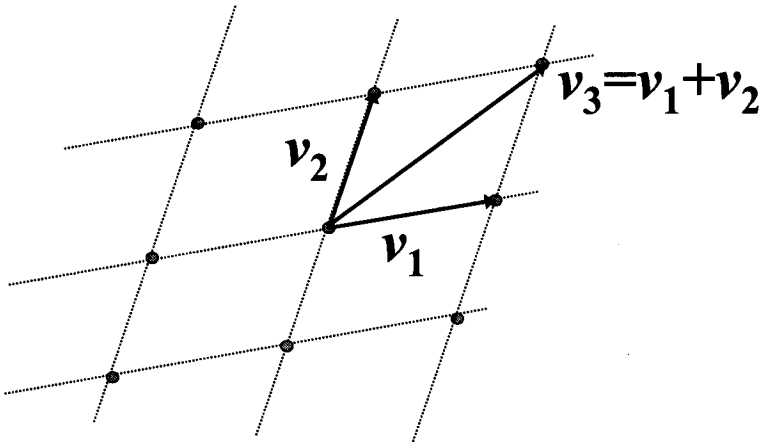

図 4.1: 格子の例

格子に関して、求解困難な計算量問題として、以下の2つがよく知られている。

1. 最短ベクトル問題

2. 最近傍ベクトル問題

これらは、一般の格子に対し、次元が大きければ計算が困難 (具体的には、NP 困難) であることが知られている。以下でそれぞれの問題を説明する。

定義 4.1 (最短ベクトル問題 (Shortest Vector Problem, SVP)) 格子の基底を与えたとき、格子の最短ベクトルを見つける問題を**最短ベクトル問題 (Shortest Vector Problem, SVP)** という。

最短ベクトルは、ベクトルのノルム (大きさ) が最小のベクトルである。格子の基底に最短ベクトルが含まれている場合は、簡単な問題であるが、上記で述べたように、基底は1通りではなく、複数通り存在する。次元が小さい場合は、基底の取りうる数は少ないが、次元が大きくなるにつれて、爆発的に多くなる。したがって、次元が大きいところでは、SVP を解くことが困難になる。LLL(Lenstra, Lenstra and Lovász) アルゴリズム [10] は、基底からノルムが小さなベクトルを出力するアルゴリズムであり、この問題を解くために、最も効率的なアルゴリズであることが知られている。しかし、次元が大きくなると LLL アルゴリズムは最短ベクトルではなく、それよりかなり大きなノルムのベクトルを出力するため、SVP を解くことができなくなる。

定義 4.2 (最近傍ベクトル問題 (Closest Vector Problem, CVP)) 格子の基底と、格子を含むベクトル空間に属するベクトルを与えたとき、そのベクトルに最も近い格子に含まれるベクトルを見つける問題を**最近傍ベクトル問題 (Closest Vector Problem, CVP)**] という。

格子の応用先として、暗号や誤り訂正符号がある。

格子を用いた暗号は、上記のような格子上の求解困難な計算量問題を安全性の根拠としている。Ajtai-Dwork 暗号 [11]、GGH 暗号 [12] や NTRU 暗号 [13] が格子暗号として知られている。また、Gentry により提唱された新しいタイプの暗号である完全準同型暗号 [15] では、Learning with Error(LWE) 問題という格子上の問題を安全性の根拠としている。LWE 問題は、最近傍ベクトル問題に類似した問題であり、格子のベクトルの分布を知ることにより、格子に含まれるベクトルと誤りベクトルとの和である、誤り付きのベクトルから元の格子のベクトルを求める問題である [14]。

格子のベクトルの要素のそれぞれを mod2 で還元すると、誤り訂正符号となる。誤り訂正符号では、最近傍ベクトル問題を解きやすいような工夫をすることで、誤りが含まれるベクトル (符号語) から正しい符号語を求めている。

以上のように、格子は暗号や誤り訂正符号へ応用される有用な数学理論である。それらの応用に関して形式化するためにも、格子の形式化は必要である。本論文では、格子を形式化するために必須となる $\mathbb{Z}$-加群の形式化をする。

4.2 R-加群

$\mathbb{Z}$-加群は、一般の環 R に対する R-加群において、$R = \mathbb{Z}$ に限定したものである。ここでは、R-加群について説明する。

定義 4.3 (R-加群) R を環、V をアーベル群とするとき、任意の $a \in R$ と $v \in V$ に対し、積 $av \in V$ が定義され、以下の条件を満たすとき、V を R-加群であるという。

(1) 【V 側の分配法則】任意の $a \in R,\ v, w \in V$ に対し、$a(v+w) = av + aw$

(2) 【R 側の分配法則】任意の $a, b \in R,\ v \in V$ に対し、$(a+b)v = av + bv$

(3) 【R 側の結合法則】任意の $a, b \in R,\ v \in V$ に対し、$(ab)v = a(bv)$

(4) 【積の単位元の存在】任意の $v \in V$ に対し、$1v = v$

次に、R-部分加群の定義を述べる。

定義 4.4 (R-部分加群) V を R-加群とし、W を V の部分群とするとき、W が V の R-部分加群であるとは、任意の $a \in R,\ w \in W$ に対して、積 aw が W に属するときをいう。

ここで、aw が W に属することは、積が W で閉じていることを示している。

次に、R-加群 V の部分加群 W による剰余加群を以下のように定義する。

定義 4.5 (剰余加群) V を R-加群 、W をその部分加群をとする。アーベル群としての剰余群 V/W に対し、任意の $a \in R$ と V/W の元 $w+W$ の積を $a(w+W) = aw+W$ と定義すると、V/W は R-加群になる。この R-加群 V/W を、V の部分加群 W による剰余加群という。

4.3 $\mathbb{Z}$-加群

ここでは、環Rが整数環$\mathbb{Z}$であるときに、アーベル群Vに対して自然な$\mathbb{Z}$-加群の構造が入ることを述べる。

定理 4.1 (アーベル群に入る$\mathbb{Z}$-加群の構造) Vをアーベル群とするとき、任意の$i \in \mathbb{Z}$と$x \in V$に対し、$i > 0$のとき、

$$ix = \overbrace{x + \cdots + x}^{i \text{ times}} \tag{4.1}$$

$i = 0$のとき、$ix = 0$、$i < 0$のとき、$ix = -((-i)x)$とおき、これをスカラ積と呼ぶ。このスカラ積によりVは$\mathbb{Z}$-加群をなす。

また、アーベル群Vに入る$\mathbb{Z}$-加群の構造は上で述べたスカラ積のみに限られる。

Wをアーベル群Vの部分群とする時、$i \in \mathbb{Z}$と$w \in W$に対して、iwはWの元であるので、WはVの$\mathbb{Z}$-部分加群となり、また剰余群V/Wも$\mathbb{Z}$-(剰余)加群となることを注意しておく。

以下では、素数pに対して、Vの部分群$pV = \{px | x \in W\}$を考える。上で述べた注意より、pVはVの$\mathbb{Z}$-部分加群であり、剰余群V/pVには剰余加群としての$\mathbb{Z}$-加群の構造が入る。任意の$x + pV \in V/pV$に対して$px \in pV$より、剰余加群V/pVの元として$p(x + pV) = 0$である。このことより以下が成り立つ。

定理 4.2 (pから生成される$\mathbb{Z}$-加群から導かれるベクトル空間) 剰余加群V/pVは、有限体$\mathbb{F}_p$上のベクトル空間の構造を持つ。

4.4 線形独立

定義 4.6 (線形結合) 自然数n, $\mathbb{Z}$-加群Vの元$v_1, v_2, \cdots, v_n$, 整数 (スカラ) $a_1, a_2, \cdots, a_n$に対し、

$$a_1v_1 + a_2v_2 + \cdots + a_nv_n \tag{4.2}$$

を$v_1, v_2, \cdots, v_n$の ($a_1, a_2, \cdots, a_n$を係数とする) **線形結合**と呼ぶ。

定義 4.7 (線形独立) $\mathbb{Z}$-加群の元$v_1, v_2, \cdots, v_n$に対し、以下の式を満たす係数$(a_1, a_2, \cdots, a_n)$がすべての0の場合のみであるとき、**線形独立**という。

$$\sum_{i=1}^{n} a_iv_i = 0 \tag{4.3}$$

4.5 自由加群

ここでは、自由加群の数学的定義や定理について説明する。

自由加群を説明するために、基底の定義をする必要があるため、まず、基底の定義について述べる。基底は $\mathbb{Z}$-加群を生成する集合であるため、最初に部分集合により生成される部分加群について、以下のように定義する。

定義 4.8 (部分集合で生成される部分加群) $\mathbb{Z}$-加群 V の部分集合 A に属する元の線形結合全体

$$\langle A\rangle = \left\{ \sum_{\lambda} a_\lambda v_\lambda | a_\lambda \in \mathbb{Z}, v_\lambda \in A \right\} \tag{4.4}$$

で表す。このとき、$\langle A\rangle$ は V の部分加群をなす。この部分加群を A **で生成される部分加群**という。

この定義と定義 4.7 を用いて、基底を以下のように定義する。

定義 4.9 (基底) $\mathbb{Z}$-加群 V の部分集合 I が、以下の条件を満たすとき、I を V の**基底**という。

(1) I は線形独立

(2) V は I で生成される

基底の定義を用いて自由加群は、以下のように定義する。

定義 4.10 (自由加群) $\mathbb{Z}$-加群 V が基底をもつとき、V は**自由加群**という。

4.6 自由加群の階数

自由加群が有限集合から生成されるとき、階数が定義できる。ここでは、階数の定義について説明する。

定義 4.11 (有限階数) $\mathbb{Z}$-加群 V が自由加群であり、その基底が有限集合であるとき、V を**有限階数**であるという。

格子の基底が一つとは限らないように、自由加群の基底も一つとは限らない。しかし、有限階数であるとき、基底の元の個数 (基底の濃度) は、以下の定理よりどの基底を取っても同じ値になることがいえる。

定理 4.3 (基底の濃度の一意性) $\mathbb{Z}$-加群 V が自由加群であるとき、その基底の濃度は基底の取り方によらず同じ値になる。

上記の基底の濃度の一意性より、階数を以下のように定義する。

定義 4.12 (階数) $\mathbb{Z}$-加群 V が自由加群であるとき、その基底の濃度を**階数**という。

第5章　$\mathbb{Z}$-加群の形式化

この章では、第4章で述べた$\mathbb{Z}$-加群に関連する定義・定理をMizarにおいて形式化する。以下ではMizarで形式化した定義・定理を数学の通常の定義・定理と区別するため、第3章と同様に**定義** [Mizar形式化]、**定理** [Mizar形式化] として表す。なお、以下では形式化の主要な流れを辿ることを目的に説明しており、実際には他の命題・定理の形式化も必要であることを注意しておく。

5.1　$\mathbb{Z}$-加群

ここでは、4.3節で述べた$\mathbb{Z}$-加群をMizarにおいて形式化する。Mizarのライブラリでは、台集合に演算を定義した代数構造を、はじめに定義する。さらに、そこに定義4.3で述べたR-加群を$R = \mathbb{Z}$に限定した$\mathbb{Z}$-加群の条件を付加することで、$\mathbb{Z}$-加群を定義する。

まず、$\mathbb{Z}$-加群の代数構造である$\mathbb{Z}$-加群構造を以下のように定義する。

定義 [Mizar形式化] **5.1** ($\mathbb{Z}$-**加群構造**)

```
definition
  struct (addLoopStr) Z_ModuleStruct
    (# carrier -> set,
         ZeroF -> Element of the carrier,
          addF -> BinOp of the carrier,
          Mult -> Function of [:INT, the carrier :], the carrier #);
end;
```

carrier, ZeroF, addFはそれぞれ、$\mathbb{Z}$-加群構造の台集合、零元、加算演算を示し、Multはスカラ積の演算を示している。Vを上記$\mathbb{Z}$-加群構造とするとき、Mizarライブラリでは`the carrier of V`と書けばVの台集合、`the ZeroF of V`もしくは`0.V`と書けばVの零元を示す。また、Vに属する元v, wに対し、`(the addF of V).(v, w)`もしくは`v+w`と書けばvとwの加算結果を示す。

零元0.Vや加算v+wは、代数構造Z_ModuleStructの下位構造であるaddLoopStrで定義されているため、特に定義をしなくとも使用することが可能である。しかし、スカラ積については定義されていないため、以下で定義する。

定義 [Mizar 形式化] 5.2 (スカラ積の記号)

```
definition
  let V,v;
  let a be integer number;
  func a * v -> Element of V equals
  (the Mult of V).(a,v);
end;
```

この定義により、整数aとVの元vに対し、a*vがaとvのスカラ積を示すことになる。

次に、$\mathbb{Z}$-加群の条件について形式化する。

定義 [Mizar 形式化] 5.3 ($\mathbb{Z}$-加群の条件)

```
definition
  let IT be non empty Z_ModuleStruct;

  attr IT is vector-distributive means
  for a for v, w being VECTOR of IT holds
  a * (v + w) = a * v + a * w;

  attr IT is scalar-distributive means
  for a, b for v being VECTOR of IT holds
  (a + b) * v = a * v + b * v;

  attr IT is scalar-associative means
  for a, b for v being VECTOR of IT holds
  (a * b) * v = a * (b * v);

  attr IT is scalar-unital means
  for v being VECTOR of IT holds 1 * v = v;
end;
```

vector-distributiveが定義4.3の条件(1)、scalar-distributiveが(2)、scalar-associativeが(3)、scalar-unitalが(4)を示している。

以上より、$\mathbb{Z}$-加群は以下のように定義する。

定義 [Mizar 形式化] 5.4 (ℤ-加群)

```
definition
  mode Z_Module is Abelian add-associative right_zeroed
   right_complementable scalar-distributive vector-distributive
   scalar-associative scalar-unital non empty Z_ModuleStruct;
end;
```

上記定義は、定義 4.3 の条件 (1)〜(4) を満たす ℤ-加群構造を ℤ-加群と定義している。ここで、Abelian, add-associative, right_zeroed, right_complementable はアーベル群の条件である。したがって、V が定義 4.3 のようにアーベル群であれば、満たしている。

定義 4.4 で $R = \mathbb{Z}$ に限定したときの ℤ-部分加群は以下のように形式化する。

定義 [Mizar 形式化] 5.5 (ℤ-部分加群)

```
definition
  let V;
  mode Submodule of V -> Z_Module means
  the carrier of it c= the carrier of V & 0.it = 0.V
  & the addF of it = (the addF of V) || the carrier of it
  & the Mult of it = (the Mult of V) | [:INT, the carrier of it:];
end;
```

ここで、the carrier of it c= the carrier of V, 0.it = 0.V 及び the addF of it = (the addF of V || the carrier of it) は、it が V の部分群であることを示している。the Mult of it = (the Mult of V) | [:INT, the carrier of it:] は、スカラ積が it で閉じていることを示している。

5.2 アーベル群から ℤ-加群の導入

定理 4.1 で説明したようにアーベル群に自然なスカラ積を定義することで ℤ-加群を導くことができる。本節では、その定理について形式化する。

まず、定理 4.1 で定義したスカラ積を以下のように定義する。

定義 [Mizar 形式化] 5.6 (アーベル群の自然なスカラ積)

```
definition
  let AG be non empty addLoopStr;
  func Int-mult-left(AG) -> Function of
  [:INT,the carrier of AG:], the carrier of AG means
```

```
  for i being Element of INT, a being Element of AG holds
  (i >= 0 implies it.(i,a) = (Nat-mult-left(AG)).(i,a)) &
  (i < 0 implies it.(i,a) = (Nat-mult-left(AG)).(-i,-a));
end;
```

ここで、Nat-mult-left(AG) は、アーベル群 AG における自然数のスカラのスカラ積を示している。これは、Mizar では BINOM:def 3 で定義されている。

アーベル群が上記定義のスカラ積を導入して、$\mathbb{Z}$-加群になるためには、定義 4.3 のすべての条件を満たす必要がある。以下で、それぞれの条件を満たすことを示す定理の形式化をする。

まず、定義 4.3 の (1) に対応するベクトルの分配法則については、以下のように形式化する。

定理 [Mizar 形式化] 5.1 (ベクトルの分配法則)

```
theorem
  for R being Abelian right_zeroed add-associative
  right_complementable non empty addLoopStr,
  a, b being Element of R, i being Element of INT
  holds (Int-mult-left(R)).(i,a+b)
  = (Int-mult-left(R)).(i,a) + (Int-mult-left(R)).(i,b);
```

(2) に対応するスカラの分配法則については、以下のように形式化する。

定理 [Mizar 形式化] 5.2 (スカラの分配法則)

```
theorem
  for R being Abelian right_zeroed add-associative
  right_complementable non empty addLoopStr,
  a being Element of R, i, j being Element of INT
  holds (Int-mult-left(R)).(i+j,a)
  = (Int-mult-left(R)).(i,a) + (Int-mult-left(R)).(j,a);
```

(3) に対応するスカラの結合法則については、以下のように形式化する。

定理 [Mizar 形式化] 5.3 (スカラの結合法則)

```
theorem
  for R being Abelian right_zeroed add-associative
  right_complementable non empty addLoopStr,
  a being Element of R, i, j being Element of INT holds
  (Int-mult-left(R)).(i*j,a)
  = (Int-mult-left(R)).(i,(Int-mult-left(R)).(j,a));
```

最後に、(4) に対応するスカラ積の単位元の存在に付いては、以下のように形式化する。

定理 [Mizar 形式化] 5.4 (スカラ積の単位元の存在)

```
theorem
  for R being right_zeroed non empty addLoopStr, a being Element of R,
  i be Element of INT st i = 1
  holds (Int-mult-left(R)).(i,a) = a;
```

これらの定理 [Mizar 形式化]5.1 から 5.4 を用いて、アーベル群と定義 [Mizar 形式化]5.6 から $\mathbb{Z}$-加群を導くことができる。そのことを示す定理は、以下のように形式化する。

定理 [Mizar 形式化] 5.5 (アーベル群から $\mathbb{Z}$-加群の導入)

```
theorem
  for AG be non empty Abelian add-associative right_zeroed
  right_complementable addLoopStr holds
  Z_ModuleStruct(# the carrier of AG, the ZeroF of AG, the addF of AG,
  Int-mult-left(AG) #) is Z_Module;
```

5.3 素数 p から生成される $\mathbb{Z}$-加群から導かれるベクトル空間

$\mathbb{Z}$-加群 V の剰余加群 V/W に関する定義 4.5 を形式化する。まず、剰余加群の台集合を以下のように定義する。

定義 [Mizar 形式化] 5.7 ($\mathbb{Z}$-加群の剰余加群の台集合)

```
definition
  let V be Z_Module;
  let W be Submodule of V;
  func CosetSet(V,W) ->non empty Subset-Family of V equals
  {A where A is Coset of W: not contradiction};
end;
```

剰余加群 V/W のアーベル群としての零元を、以下のように形式化する。

定義 [Mizar 形式化] 5.8 ($\mathbb{Z}$-加群の剰余加群の零元)

```
definition
```

```
  let V be Z_Module;
  let W be Submodule of V;
  func zeroCoset(V,W) -> Element of CosetSet(V,W) equals
  the carrier of W;
end;
```

剰余加群 V/W のアーベル群としての加算を、以下のように形式化する。

定義 [Mizar 形式化] 5.9 ($\mathbb{Z}$-加群の剰余加群の加算)

```
definition
  let V be Z_Module;
  let W be Submodule of V;
  func addCoset(V,W) -> BinOp of CosetSet(V,W) means
  for A,B be Element of CosetSet(V,W)
  for a,b be VECTOR of V st A = a + W & B = b + W holds
  it.(A,B) = (a+b)+W;
end;
```

剰余加群 V/W の積を以下のように形式化する。

定義 [Mizar 形式化] 5.10 ($\mathbb{Z}$-加群の剰余加群の積)

```
definition
  let V be Z_Module;
  let W be Submodule of V;
  func lmultCoset(V,W) -> Function of [:INT, CosetSet(V,W):],
  CosetSet(V,W) means
  for z be Element of INT, A be Element of CosetSet(V,W)
  for a be VECTOR of V st A = a+W holds it.(z,A) = z*a + W;
end;
```

$\mathbb{Z}$-加群 V の剰余加群 V/W は、これらの台集合、零元、加算及び積を用いて、以下のように形式化する。

定義 [Mizar 形式化] 5.11 ($\mathbb{Z}$-加群の剰余加群)

```
definition
  let V be Z_Module;
  let W be Submodule of V;
  func Z_ModuleQuot(V,W) -> strict Z_Module means
  the carrier of it = CosetSet(V,W) &
```

```
  the addF of it = addCoset(V,W) &
  0.it = zeroCoset(V,W) & the Mult of it = lmultCoset(V,W);
end;
```

次に、$\mathbb{Z}$-加群 V を a 倍 (a は整数) した元全体 aV が V の部分加群をなすことを示す。まず、aV の集合を以下のように形式化する。

定義 [Mizar 形式化] 5.12 (aV の台集合)

```
definition
  let V be Z_Module;
  let a be integer number;
  func a * V -> non empty Subset of V equals
  {a * v where v is Element of V : not contradiction};
end;
```

aV の零元を以下のように形式化する。

定義 [Mizar 形式化] 5.13 (aV の零元)

```
definition
  let V be Z_Module;
  let a be integer number;
  func Zero_(a,V) -> Element of  a*V
  equals
  0.V;
end;
```

零元は、V と同じ 0.V となる。

aV の加算を以下のように形式化する。

定義 [Mizar 形式化] 5.14 (aV の加算)

```
definition
  let V be Z_Module;
  let a be integer number;
  func  Add_(a,V) -> Function of [:a*V,a*V :], a*V
  equals
  (the addF of V) | [:a*V,a*V:];
end;
```

この定義の形式化において実際は、加算が aV で閉じていることも証明している。

aV の積を以下のように形式化する。

定義 [Mizar 形式化] 5.15 (aV の積)

```
definition
  let V be Z_Module;
  let a be integer number;
  func Mult_(a,V) -> Function of [:INT,a*V :], a*V
  equals
  (the Mult of V) | [:INT,a*V:];
end;
```

この定義の形式化において実際は、積が aV で閉じていることも証明している。

これらの台集合、零元、加算及び積を用いて、($\mathbb{Z}$-) 部分加群 aV は以下のように形式化する。

定義 [Mizar 形式化] 5.16 (部分加群 aV)

```
definition
  let V be Z_Module;
  let a be integer number;
  func a (*) V -> Submodule of V equals
  Z_ModuleStruct (# a * V, Zero_(a,V), Add_(a,V), Mult_(a,V) #);
end;
```

ここで、aV の V の部分加群としての構造を `a (*) V`、その台集合を `a * V` で表している。

a を素数 p とするとき、剰余加群 V/pV(Mizar では、`Z_ModuleQuot(V, p(*)V)` と表記) の積は以下のような特性を満たす。

定理 [Mizar 形式化] 5.6 (剰余加群 V/pV の積の特性)

```
theorem
  for p, i be Integer, V, X be Z_Module, W be Submodule of V,
  x be VECTOR of X st p <> 0 & W = p (*) V &
  X = Z_ModuleQuot(V, W) holds i * x = (i mod p) * x;
```

上記特性により、$\mathbb{F}_p$ の元 a と剰余加群 V/pV の元との積を以下のように再定義できる。

定義 [Mizar 形式化] 5.17 (V/pV の積)

```
definition
  let p be Prime;
  let V be Z_Module;
```

```
  func Mult_Mod_pV(V,p) -> Function of
  [:the carrier of GF(p), the carrier of Z_ModuleQuot(V,p(*)V):],
  the carrier of Z_ModuleQuot(V,p(*)V) means
  for a being Element of GF(p), i being Integer,
  x being Element of Z_ModuleQuot(V,p(*)V) st a = i mod p
  holds it.(a,x) = (i mod p) * x;
end;
```

上記の積を用いることで、剰余加群 V/pV は $\mathbb{Z}$-加群の構造だけでなく、$\mathbb{F}_p$ 上のベクトル空間の構造も持つことがいえる。以下でその定理を形式化する。この定理は、定理 4.2 に対応している。

定理 [Mizar 形式化] 5.7 (剰余加群 V/pV とベクトル空間)

```
theorem
  for p be Prime, V be Z_Module holds
  VectSpStr (# the carrier of Z_ModuleQuot(V,p(*)V),
  the addF of Z_ModuleQuot(V,p(*)V),
  the ZeroF of Z_ModuleQuot(V,p(*)V), Mult_Mod_pV(V,p) #)
  is VectSp of GF(p);
```

剰余加群 V/pV に対応するベクトル空間を、以下のように定義する。

定義 [Mizar 形式化] 5.18 (剰余加群 V/pV に対応するベクトル空間)

```
definition
  let p be Prime, V be Z_Module;
  func Z_MQ_VectSp(V,p) -> VectSp of GF(p) equals
  VectSpStr (# the carrier of Z_ModuleQuot(V,p(*)V),
  the addF of Z_ModuleQuot(V,p(*)V), the ZeroF of Z_ModuleQuot(V,p(*)V),
  Mult_Mod_pV(V,p) #);
end;
```

さらに、V の元から上記ベクトル空間の元へ変換関数を、以下のように定義する。

定義 [Mizar 形式化] 5.19 ($\mathbb{Z}$-加群の元からベクトル空間の元への変換)

```
definition
  let p be Prime, V be Z_Module, v be VECTOR of V;
  func ZMtoMQV(V,p,v) -> Vector of Z_MQ_VectSp(V,p) equals
  v + p(*)V;
end;
```

5.4 $\mathbb{Z}$ 係数の線形結合

定義4.6の線形結合を以下で形式化する。まず、線形結合で使用する係数$a_1, a_2, \cdots, a_n$であることを示すモードを以下のように定義する。

定義 [Mizar 形式化] 5.20 (線形結合のモード (係数の集合))

```
definition
  let V be non empty ZeroStr;
  mode Z_Linear_Combination of V ->
  Element of Funcs(the carrier of V, INT)
  means
  ex T being finite Subset of V st for v being Element of V
  st not v in T holds it.v = 0;
end;
```

このモードを満たす係数が使用する$\mathbb{Z}$-加群の元$v_1, v_2, \cdots, v_n$の集合を以下のように定義する。

定義 [Mizar 形式化] 5.21 (線形結合の係数の台集合)

```
definition
  let V be non empty addLoopStr, L be Z_Linear_Combination of V;
  func Carrier(L) -> finite Subset of V equals
  {v where v is Element of V : L.v <> 0};
end;
```

線形結合は、$a_1v_1, a_2v_2, \cdots, a_nv_n$の和である。Mizarでは和を形式化するために、まず、和を取りたいそれぞれのa_1v_1, $a_2v_2, \cdots,$ やa_nv_nを配列として、以下のように定義する。

定義 [Mizar 形式化] 5.22 (線形結合の係数から生成される配列)

```
definition
  let V;
  let F;
  let f;
  func f (#) F -> FinSequence of the carrier of V means
  len it = len F & for i st i in dom it holds
  it.i = f.(F/.i) * F/.i;
end;
```

次に線形結合は、これらの配列の和であると、以下のように定義する。

定義 [Mizar 形式化] 5.23 (線形結合)

```
definition
  let V;
  let L;
  func Sum(L) -> Element of V means
  ex F st F is one-to-one & rng F = Carrier(L) & it = Sum(L (#) F);
end;
```

この線形結合の定義を用いて、定義 4.7 は以下のように形式化する。

定義 [Mizar 形式化] 5.24 (線形独立)

```
definition
  let V;
  let A;
  attr A is linearly-independent means
  for l st Sum(l) = 0.V holds Carrier(l) = {};
end;
```

5.5 自由加群

ここでは、4.5 節で述べた自由加群を形式化する。
定義 4.8 は、以下のように形式化する。

定義 [Mizar 形式化] 5.25 (部分集合から生成される部分加群)

```
definition
  let V;
  let A;
  func Lin(A) -> strict Submodule of V means
  the carrier of it = {Sum(l) : not contradiction};
end;
```

ここで、Lin(A) は定義 4.8 における $\langle A \rangle$ を示している。
次に定義 4.10 を形式化する。

定義 [Mizar 形式化] 5.26 (自由加群)

```
definition
  let IT be Z_Module;
  attr IT is free means
```

```
  ex A being Subset of IT
  st A is linearly-independent & Lin(A) = the Z_ModuleStruct of IT;
end;
```

ここで、上記の定義中の A は IT の基底である。
定義 4.9 は、以下のように形式化する。

定義 [Mizar 形式化] 5.27 (自由加群の基底)

```
definition
  let V be free Z_Module;
  mode Basis of V -> Subset of V means
  it is linearly-independent & Lin (it) = the Z_ModuleStruct of V;
end;
```

5.6 自由加群の階数

ここでは、4.6 節で述べた自由加群の階数を形式化する。
定義 4.11 は、以下のように形式化する。

定義 [Mizar 形式化] 5.28 (有限階数)

```
definition
  let IT be free Z_Module;
  attr IT is finite-rank means
  ex A being finite Subset of IT st A is Basis of IT;
end;
```

次に基底の濃度の一意性を形式化していく。一意性は、$\mathbb{Z}$-加群 V の基底と、剰余加群 V/pV の基底が全単射で対応することを利用する。このとき、剰余加群 V/pZ は定理 [Mizar 形式化] 5.7 で形式化したように $\mathbb{F}_p$ 上のベクトル空間の構造を持つことがいえる。剰余加群 V/pV の基底は、それに対応する $\mathbb{F}_p$ 上のベクトル空間の基底であり、その濃度の一意性は Mizar で既に以下のように形式化されている [28]。

定理 [Mizar 形式化] 5.8 (体 K 上ベクトル空間の基底濃度の一意性)

```
theorem :: VECTSP_9:22
  V is finite-dimensional implies for A, B being Basis of V holds
  card A = card B;
```

ここで、V は一般の体 K に対する K 上ベクトル空間 (本論文中の V とは異なる) であり、A, B は V の基底である。この定理は、A の濃度 card A と B の濃度 card B

が等しいことを示しており、どの基底であっても基底の濃度は同じ (一意) であることを示している。

まず、自由 $\mathbb{Z}$-加群 V の元 v に対応する剰余加群 V/pV の元が存在することを以下のように形式化する。

定理 [Mizar 形式化] 5.9 (剰余加群への元の変換)

```
theorem
  for p being Prime, V being Z_Module,
  ZQ being VectSp of GF(p), vq being Vector of ZQ
  st ZQ = Z_MQ_VectSp(V,p)
  holds ex v being VECTOR of V st vq = ZMtoMQV(V,p,v);
```

上記の v に対応する V/pV の元における剰余加群 V/pV の線形結合の係数と等しい、元 v における基底 I に対する線形結合の係数 `l.v`(すなわち、$\sum_{v \in I} l_v v$ の係数 $l_v (v \in I)$) が存在することを、以下のように形式化する。

定理 [Mizar 形式化] 5.10 (自由加群と剰余加群の線形結合の係数対応)

```
theorem
  for p being Prime, V being free Z_Module, I being Basis of V,
  lq being Linear_Combination of Z_MQ_VectSp(V,p)
  holds ex l being Z_Linear_Combination of I
  st for v being VECTOR of V st v in I holds
  l.v = lq.(ZMtoMQV(V,p,v))
```

自由 $\mathbb{Z}$-加群 V の線形結合と定理 [Mizar 形式化] 5.10 で導いた V/pV の線形結合が対応することを、以下のように形式化する。

定理 [Mizar 形式化] 5.11 (自由加群と剰余加群の線形結合の対応)

```
theorem
  for p being Prime, V being free Z_Module, I being Basis of V,
  l being Z_Linear_Combination of I,
  IQ being Subset of Z_MQ_VectSp(V,p),
  lq being Linear_Combination of IQ
  st IQ = {ZMtoMQV(V,p,u) where u is VECTOR of V : u in I} &
  (for v being VECTOR of V st v in I holds
  l.v = lq.(ZMtoMQV(V,p,v)) )
  holds Sum(lq). = ZMtoMQV(V,p,Sum(l));
```

以上で、自由 $\mathbb{Z}$-加群 V の基底 I に対応する剰余加群の集合が存在し、それを用いた線形結合が V の線形結合と対応することが示せた。

次に、自由$\mathbb{Z}$-加群Vの基底Iとそれに対応する剰余加群の集合Xが全単射写像で対応することを以下のように形式化する。

定理 [Mizar 形式化] 5.12 (自由加群の基底と対応する剰余加群の集合)

```
theorem
  for p being Prime, V being free Z_Module, I being Basis of V,
  X be non empty Subset of Z_MQ_VectSp(V,p)
  st I is non empty
  & X = {ZMtoMQV(V,p,u) where u is VECTOR of V : u in I}
  holds
  ex F be Function of X,  the carrier of V
  st
  (for u be  VECTOR of V  st u in I
  holds F.(ZMtoMQV(V,p,u)) = u)
  & F is one-to-one & dom F = X & rng F = I;
```

ここで、Fが全単射写像であることを、F is one-to-one & dom F = X & rng F = Iで示している。

定理 [Mizar 形式化]5.12 を用いて、自由$\mathbb{Z}$-加群Vの基底の濃度と、それに対応する剰余加群V/pVの集合の濃度が等しいことを、以下のように形式化する。

定理 [Mizar 形式化] 5.13 (自由加群の基底と対応する剰余加群の集合の濃度)

```
theorem
  for p being Prime, V being free Z_Module, I being Basis of V
  holds
  card ( {ZMtoMQV(V,p,u) where u is VECTOR of V : u in I} )
  = card(I);
```

さらに、以下で上記で述べた剰余加群V/pVの集合IQが基底となることを示していく。定義 4.9 の条件 (1) を満たすことを、以下のように形式化する。

定理 [Mizar 形式化] 5.14 (自由加群の基底と対応する剰余加群の集合の特性 1)

```
theorem
  for p being Prime, V being free Z_Module, I being Basis of V,
  ZQ being VectSp of GF(p), IQ being Subset of ZQ
  st ZQ = Z_MQ_VectSp(V,p) &
  IQ = {ZMtoMQV(V,p,u) where u is VECTOR of V : u in I}
  holds IQ is linearly-independent;
```

定義 4.9 の条件 (2) を満たすことを、以下のように形式化する。

定理 [Mizar 形式化] 5.15 (自由加群の基底と対応する剰余加群の集合の特性 2)

```
theorem
  for p being Prime, V being free Z_Module, I being Basis of V,
  IQ being Subset of Z_MQ_VectSp(V,p),
  l be Z_Linear_Combination of I
  st
  IQ = {ZMtoMQV(V,p,u) where u is VECTOR of V : u in I}
  holds
  ZMtoMQV(V,p,Sum(l)) in Lin(IQ);
```

定理 [Mizar 形式化]5.14, 5.15 を用いて、剰余加群の集合 IQ が剰余加群 V/pV の基底であることを、以下のように形式化する。

定理 [Mizar 形式化] 5.16 (自由加群の基底と対応する剰余加群の基底)

```
theorem ThQuotBasis4:
  for p being Prime, V being free Z_Module, I being Basis of V,
  IQ being Subset of Z_MQ_VectSp(V,p)
  st
  IQ = {ZMtoMQV(V,p,u) where u is VECTOR of V : u in I}
  holds IQ is Basis of Z_MQ_VectSp(V,p);
```

定理 [Mizar 形式化]5.13, 5.16 と、Mizar で既に形式化されている定理 [Mizar 形式化]5.8 を用いて、自由 $\mathbb{Z}$-加群 V の基底の濃度の一意性を以下のように形式化する。

定理 [Mizar 形式化] 5.17 (基底の濃度の一意性)

```
theorem
  for V be finite-rank free Z_Module
  holds for A, B being Basis of V holds
  card A = card B;
```

上記定理より、基底の濃度、すなわち、階数が一意であることが示せたので、改めて自由 $\mathbb{Z}$-加群 V の階数の定義を、以下のように形式化する。

定義 [Mizar 形式化] 5.29 (自由加群の階数)

```
definition
  let V being finite-rank free Z_Module;
  func rank(V) -> Nat means
  for I being Basis of V holds it = card I;
end;
```

なお、本論文では自由$\mathbb{Z}$-加群Vの階数の一意性を、剰余加群V/pVを$\mathbb{F}_p$上ベクトル空間として見たときの次元の一意性から示したが、Vを基にして作られる$\mathbb{Q}$上ベクトル空間$\mathbb{Q}\otimes_{\mathbb{Z}}V$の次元の一意性からも示すことが可能である。今後は、Vの階数と$\mathbb{Q}\otimes_{\mathbb{Z}}V$の次元が等しいことについても形式化する予定である。

第6章　結論

6.1　本論文の結果

本論文では、楕円曲線暗号や格子暗号で使用する数学的な定義や定理の形式化を行った。楕円曲線暗号に関しては、以下の形式化を行った。

- 素体 $\mathbb{F}_p$
- 射影座標と座標の同値
- $\mathbb{F}_p$ 上の楕円曲線と楕円曲線上の点の同値類
- $\mathbb{F}_p$ 上の楕円曲線の $\mathbb{F}_p$-有理点の個数の評価
- $\mathbb{F}_p$ 上の楕円曲線の点の演算

これらの形式化により、楕円曲線暗号の暗号化・復号化アルゴリズムや署名生成・検証アルゴリズムで使用する楕円曲線上の演算をライブラリとして使用することができる。

格子暗号に関しては、以下の形式化を行った。

- $\mathbb{Z}$-加群の定義
- アーベル群から $\mathbb{Z}$-加群の導入
- 素数 p から生成される $\mathbb{Z}$-加群から導かれるベクトル空間
- $\mathbb{Z}$ 係数の線形結合
- 自由 $\mathbb{Z}$-加群
- 自由 $\mathbb{Z}$-加群の階数

アーベル群から $\mathbb{Z}$-加群の導入については、楕円曲線暗号でも必要であり、他の分野でも利用される重要な命題である。格子は、有限階数の自由 $\mathbb{Z}$-加群にノルムを定義したものであり、$\mathbb{Z}$-加群、特に自由 $\mathbb{Z}$-加群やその階数の形式化は格子暗号において必須である。

これらの楕円曲線暗号や格子暗号の数学的定義や定理を形式化することで、それぞれの暗号の演算を扱うことが可能になる。

6.2 Mizar における代数関連の形式化に関する今後の研究の展開

楕円曲線暗号においては、今後は楕円曲線の $\mathbb{F}_p$-有理点がアーベル群をなすことを形式化していきたい。その後は、以下のような形式化の展開を考えている。

(E1) $\mathbb{F}_p$ 上楕円曲線の同型と j-不変量

(E2) 無限素体 $\mathbb{Q}$ 及び $\mathbb{Q}$ 上楕円曲線

(E3) $\mathbb{F}_p$ の拡大体 $\mathbb{F}_{p^n}$ 及び $\mathbb{F}_{p^n}$ 上の楕円曲線

(E4) 虚数 α を用いた $\mathbb{Q}$ の拡大体 $\mathbb{Q}(\alpha)$ 及び $\mathbb{Q}(\alpha)$ 上楕円曲線

(E5) 虚数乗法を持つ楕円曲線を用いた $\mathbb{F}_p$ 上楕円曲線の $\mathbb{F}_p$-有理点の個数の評価

(E6) $\mathbb{F}_p$ 上楕円曲線の点のスカラ倍演算アルゴリズム

(E7) $\mathbb{F}_p$ 上楕円曲線を用いた楕円曲線 DH 鍵共有方式

これらは、互いの形式化の結果を用いて形式化を進める関係にある。図 6.1 は、その関係を示している。この図において、矢印の始端の形式化の結果を、終端が使用することを示している。例えば、(E1) の形式化の結果を (E6) が使用する。

格子においては、$\mathbb{Z}$-加群に対し、ノルムを定義し、格子暗号の方式の形式化をしていきたい。その後は、以下のような形式化の展開を考えている。

(L1) 格子の基底を用いた近傍ベクトル計算アルゴリズム

(L2) ノルム縮約アルゴリズム (LLL(Lenstra-Lenstra-Lovasz) アルゴリズム)[10]

(L3) GGH(Goldreich-Goldwasser-Halevi) 暗号 [12]

(L4) ルート格子、その基本ルート及び基本ルートの個数の評価

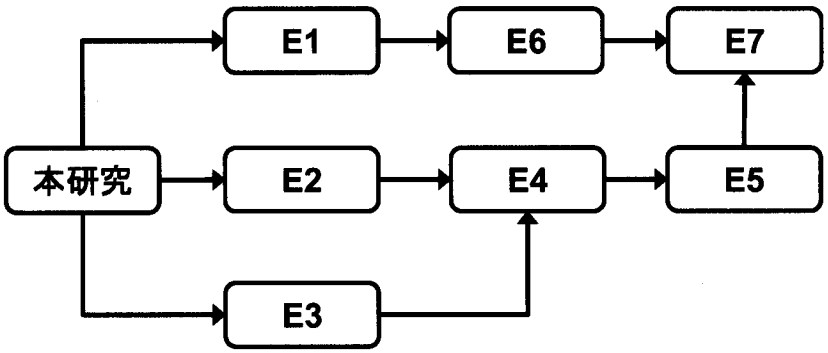

図 6.1: 楕円曲線の形式化の展開

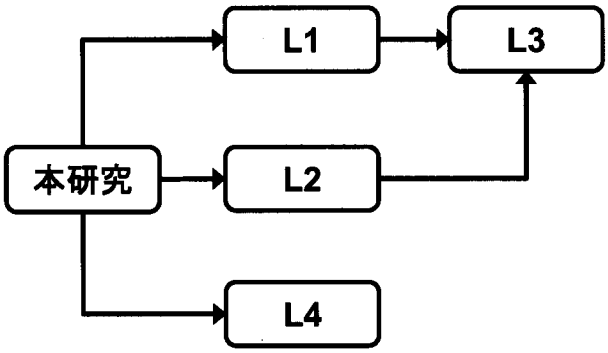

図 6.2: 格子の形式化の展開

これらは、互いの形式化の結果を用いて形式化を進める関係にある。図6.2は、その関係を示している。

(L1), (L2), (L3) は暗号に関する形式化である。(L4) は、符号化変調へ格子を応用する際に使われるものであり、効率のよい符号化変調が可能な格子であるルート格子に関するものである。

6.3 暗号の安全性証明に関する今後の研究の展開

本研究で形式化した数学的理論を展開させることで、楕円曲線暗号や格子暗号の攻撃についても、その評価も含めて形式化が可能になる。これらの形式化を進めていくことで、楕円曲線暗号や格子暗号の方式自体を形式化していくことが可能になる。しかし、楕円曲線暗号や格子暗号の安全性証明をしていくためには、計算アルゴリズムの動作検証や計算量評価、確率や乱数などの分布の評価が必要である。また、1.4節で述べたような汎用結合可能性の理論やゲーム列による安全性証明の理論の形式化も必要である。

汎用結合可能性の理論においては、安全性証明を効率よく実施するために、暗号方式やハッシュ関数などの各部品の安全性証明をMizarで実施し、暗号プロトコル全体の安全性証明をDolev-Yaoモデルのような記号論的アプローチで実施するハイブリッドの安全性証明方法も考慮が必要と考えている。また、ゲーム列による安全性証明においても、ゲーム変換の変換前後のゲームの攻撃成功確率の差が十分に小さいことをMizarで、ゲーム列の自動生成をCryptoVerifで行うようなハイブリッドの安全性証明方法が必要になる。このように、Mizarを使用する箇所を、厳密な証明が必要な部分に限定して、Mizarをより効果的に使用する安全性証明を考えていきたい。

謝辞

本論文を執筆するにあたり、主指導教官として熱心にご指導を賜り、貴重なご議論をいただきました信州大学工学部情報工学科 教授 師玉 康成先生に深謝いたします。信州大学大学院工学系研究科情報工学専攻 助教 岡崎 裕之先生には、本学の博士課程に入学する機会を与えていただき、また、本研究を進めるにあたり、日頃より有益なご討論ならびにご助言をいただきました。ここに心より感謝いたします。本研究を進めるにあたり、ご協力いただきました東京理科大学理工学部電気電子情報工学科 助教 荒井 研一先生、信州大学大学院工学系研究科 水島 大地さんに感謝いたします。

関東学院大学工学部 教授 長尾 孝一先生には、本学の博士課程への入学を勧めていただくとともに、多くのご助言を与えていただき感謝いたします。本論文を執筆するにあたり、信州大学工学部情報工学科 教授 和崎 克己先生、准教授 カワモト・ポーリン・ナオミ先生、准教授 宮尾 秀俊先生には、副査として貴重なコメントをいただきました。ここに感謝いたします。

筆者が京都工芸繊維大学工芸学部電子情報工学科在籍時より、研究の進め方のご指導を賜りました、大阪学院大学 教授、京都工芸繊維大学 名誉教授 笠原 正雄先生に感謝の意を表します。九州大学大学院 数理学研究院 教授 金子 昌信先生には、筆者が京都工芸繊維大学在籍時に、楕円曲線や数論アルゴリズムについてお教えいただき、また、楕円曲線に関する研究のご指導をいただきました。ここに心より感謝いたします。北陸先端科学技術大学院大学 情報科学研究科 教授 宮地 充子先生には、会社入社以来、楕円曲線に関する研究のご指導をいただきました。ここに感謝いたします。大阪電通大学 金融経済学部 准教授 境 隆一先生には、研究のご指導をいただきました。ここに感謝いたします。

パナソニック株式会社 R&D 知的財産権センター 大森 基司さんには、会社に入社以来、研究の進め方のご指導をいただきました。ここに感謝の意を表します。筆者が会社における業務と両立して、本論文を作成し得たのは、パナソニック株式会社 デジタル・ネットワーク開発センター CRM クラウド開発室 CRM クラウド第2チーム 松崎 なつめさんのご協力によるところです。ここに感謝いたします。また、CRM クラウド第2チームのチームメンバーの方々に感謝いたします。

最後になりましたが、日頃より支援いただいた父、母に感謝いたします。

参考文献

[1] R.Canetti, "Universal composable security: A new paradigm for cryptographic protocols", Cryptology ePrint Archive, Report 2000/067, 2000. *Available at* http://eprint.iacr.org/2000/067.

[2] V.Shoup, "OAEP reconsidered", CRYPTO 2001, Lecture Notes in Computer Science, Vol. 2139, Springer-Verlag, pp.239–259, 2001

[3] V.Shoup, "Sequences of Games: A Tool for Taming Complexity in Security Proofs", Cryptology ePrint Archive, Report 2004/332, 2004, *Available at* http://eprint.iacr.org/2004/332.

[4] I.Blake, G.Seroussi and N.Smart, "Elliptic Curves in Cryptography", London Mathematical Society Lecture Note Series, No. 265, Cambridge University Press, 1999

[5] 岡本 龍明, 真鍋 義文, "汎用的結合可能性による暗号システムの安全性証明", 電子情報通信学会論文誌, Vol. J92-D, No. 5, pp.587–595, 2009.

[6] 岡本 龍明, 山本 博資, "現代暗号", 産業図書, 1997.

[7] D.Micciancio and S.Goldwasser, "Complexity of Lattice Problems: A Cryptographic Perspective", The International Series in Engineering and Computer Science, Springer-Verlag, 2002

[8] ISO/IEC 15408-3: 2008, "Information technology – Security techniques – Evaluation criteria for IT security – Part 3: Security assurance components"

[9] S.Matsuo, K.Miyazaki, A.Otsuka and D.Basin, "How to Evaluate the Security of Real-life Cryptographic Protocols? The cases of ISO/IEC 29128 and CRYPTREC", SCIS 2010, 1C1-1, 2010.

[10] A.K.Lenstra, H.W.Lenstra, Jr. and L. Lovász, "Factoring Polynomials with Rational Coefficients", Math. Ann. 261, pp.515–534, 1982.

[11] M.Ajtai and C.Dwork, "A public-key cryptosystem with worst-case/average-case equivalence", STOC'97, ACM, pp.284–293, 1997.

[12] O.Goldreich, S.Goldwasser and S.Halevi, "Public-key cryptosystems from lattice reduction problems", CRYPTO'97, Springer-Verlag, pp.112–131, 1997.

[13] J.Hoffstein, J.Pipher and J.H.Silverman, "NTRU: A Ring-Based Public Key Cryptosystem", Algorithmic Number Theory (ANTS III), LNCS 1423, Springer-Verlag, pp.267–288, 1998.

[14] O.Regev, "On lattices, learning with errors, random linear codes, and cryptography", STOC'05, ACM, pp.84–93, 2005.

[15] C.Gentry, "Fully Homomorphic Encryption Using Ideal Lattices", STOC'09, ACM, pp.169–178, 2009.

[16] Mizar Proof Checker : *Available at* http://mizar.org/.

[17] C.Schwarzweller, "The Binomial Theorem for Algebraic Structures", Formalized Mathematics, vol.9, no.3, pp.559–564, 2001.

[18] G.Bancerek, "Cardinal Numbers", Formalized Mathematics, vol.1, no.2, pp.377–382, 1990.

[19] K.Raczkowski and P.Sadowski, "Equivalence Relations and Classes of Abstraction", Formalized Mathematics, vol.1, no.3, pp.441–444, 1990.

[20] J.Trybulec, "Integers", Formalized Mathematics, vol.1, no.3, pp.501–505, 1990.

[21] C.Schwarzweller, "The Ring of Integers, Euclidean Rings and Modulo Integers", Formalized Mathematics, vol.8, no.1, pp.29–34, 1999.

[22] A.Trybulec, "Tuples, Projections and Cartesian Products", Formalized Mathematics, vol.1, no.1, pp.97–105, 1990.

[23] G.Bancerek, "The Fundamental Properties of Natural Numbers", Formalized Mathematics, vol.1, no.1, pp.41–46, 1990.

[24] W.A.Trybulec, "Vectors in Real Linear Space", Formalized Mathematics, vol.1, no.2, pp.291–296, 1990.

[25] E.Kusak, W.Leonczuk, and M.Muzalewski, "Abelian Groups, Fields and Vector Spaces", Formalized Mathematics, vol.1, no.2, pp.335–342, 1990.

[26] M.Muzalewski, "Construction of Rings and Left-, Right-, and Bi-Modules over a Ring", Formalized Mathematics, vol.2, no.1, pp.3–11, 1991.

[27] W.A.Trybulec, "Subspaces and Cosets of Subspaces in Vector Space", Formalized Mathematics, vol.1, no.5, pp.865–870, 1990.

[28] M.W.Zynel, "The Steinitz Theorem and the Dimension of a Vector Space", Formalized Mathematics, vol.5, no.3, pp.429–438, 1996.