7　セキュリティ基盤技術
7-1　暗号の安全性評価技術の高度化

篠原直行　青野良範　林　卓也

公開鍵暗号方式は、情報社会において重要な基盤技術であり、ネットバンキングなどに実際に利用されている。公開鍵暗号方式の安全な鍵長は、解読実験の世界記録で得られた計算時間などから算出される。本稿ではプライバシー保護に適した公開鍵暗号方式であるη_Tペアリングを利用したペアリング暗号の安全性評価と、量子計算機が実用化されても安全性を確保できることが期待されている格子暗号の安全性評価について述べる。

1　まえがき

公開鍵暗号方式は、情報社会を支える基盤技術として広く使用されており、その代表的なものとしてRSA 暗号と楕円曲線暗号が挙げられる。これらの公開鍵暗号方式を含め、現在、研究が進められている公開鍵暗号方式は、数学的な問題の計算の困難性をその安全性の基盤としている。例えば、楕円曲線暗号では楕円曲線で与えられる巡回群が利用されているが、そこで定義される離散対数問題（ECDLP）が解かれると楕円曲線暗号は解読されてしまう。したがって、公開鍵暗号方式を安全に運用するためには安全な暗号パラメータ（鍵長など）を解読実験などにより見積もる必要があり、公開鍵暗号方式の安全性に関係付けられる様々な数学的な問題の研究が、世界中の研究機関において実施されている。本稿ではペアリング暗号と格子暗号の安全性評価について述べる。

現在、実用化に向けて研究が進められている公開鍵暗号方式のひとつとしてペアリング暗号が挙げられる。ペアリング暗号を利用することで、RSA 暗号や楕円曲線暗号などでは実現が困難である高機能な暗号技術が実現できる。その例として検索可能暗号について簡単に説明する。検索可能暗号ではデータとキーワードを暗号化したまま検索することができるため、サーバにデータを暗号化したまま保存することに適している。これは情報漏洩対策等に適しており、またそのためにペアリング暗号はプライバシー保護に適した暗号方式として実用化が期待されている。ペアリング暗号は、楕円曲線上の ECDLP と有限体上の離散対数問題を解く計算の困難性をその安全性の基盤としており、そこで利用される有限体の大きさはその計算の困難性を決定する重要な要素である。また、有限体の標数はペアリング暗号の安全性と暗号処理速度を決定する要素であり、本稿では高速実装の研究成果が多く報告されている標数が 3 の場合について述べる。具体的には有限体$\mathrm{GF}(3^{6\cdot 97})$上の離散対数問題を解く手法とその数値実験 [1][2] について **2** で説明する。

次に格子暗号の安全性評価について述べる。暗号方式の長期利用の観点から、量子計算機が実用化されても安全性を保障できる暗号方式の実用化が望まれている。先に述べたように現在広く利用されている公開鍵暗号方式として RSA 暗号と楕円曲線暗号があるが、これらの公開鍵暗号方式は量子計算機が実用化されると短時間に解読されてしまうことが数学的に証明されている。しかし、格子暗号は量子計算機が実用化されても短時間に解読できないため、将来にわたり安全性が確保されると期待される公開鍵暗号方式のひとつである。また、準同型暗号など様々な暗号技術を利用できることから、その実用化が期待されている。本稿では、格子暗号の安全性を評価する標準的な手法である格子攻撃の概要について述べ、格子暗号の安全性の根拠として使われる LWE 問題の評価 [3] について述べる。

2　ペアリング暗号の安全性評価

η_Tペアリングを用いたペアリング暗号では、n を素数とした有限体$\mathrm{GF}(3^{6n})$が利用される。このような標数が小さい有限体上の離散対数問題を効率よく解くアルゴリズムとして、関数体篩法（Function Field Sieve: FFS）等が知られている。本稿では、$\mathrm{GF}(3^{6\cdot 97})$上の離散対数問題に注目し、これを解くことに適した改良を施した関数体篩法とそれを用いた数値実験について述べる。

2.1　関数体篩法の概要

拡大次数nが 509 以下である$\mathrm{GF}(3^{6n})$上の離散対数問題を解くアルゴリズムとしては、2006 年に Joux と Lercier によって提案された関数体篩法（JL06-FFS）[4] が有効であることが知られている [5]。本節では有限体$\mathrm{GF}(3^{6n})$上の離散対数問題$T = g^X$の解$X = \log T$を JL06-FFS を用いて解く場合の概要を説明する。この関数体篩法は以下の 4 つの計算段階で構成される：多項式選択段階、関係探索段階、線形代数段階、個別離散対数段階。

多項式選択段階： まず、$k \in \{1,2,3,6\}$を選び、Adleman によって提案された 8 つの条件 [6] を満たす二変数多項式$H(x,y) \in \mathrm{GF}(3^k)[x,y]$を決定する。ただし、与えられた整数$d_H$に対して$\deg_y H = d_H$とする。さらに与えられた自然数$d_m$に対して、次数が$d_m$である多項式$m \in \mathrm{GF}(3^k)[x]$をランダムに生成し、以下の条件を満たすモニックで既約な多項式$f \in \mathrm{GF}(3^k)[x]$を計算する：

$$H(x,m) \equiv 0 (\mathrm{mod}\, f), \deg f = 6n/k.$$

このとき有限体$\mathrm{GF}(3^{6n})$は$\mathrm{GF}(3^k)[x]/(f)$で表現され、$\mathrm{GF}(3^k)[x,y]/(H)$から$\mathrm{GF}(3^k)[x]/(f)$への全準同型写像$\xi$で$\xi(y) = m$を満たすものが存在する。次に、与えられた自然数$B$に対して 2 つの因子基底$F_R(B), F_A(B)$を以下のように定める：

$$F_R(B) = \{\rho \in \mathrm{GF}(3^k)[x]: \deg \rho \le B, \rho \text{ is monic and irreducible}\},$$

$$F_A(B) = \{< \rho, y - t > \in Div(GF(3^k)[x,y] / (H)): \rho \in F_R(B),\ H(x,t) \equiv 0 (\mathrm{mod}\ \rho)\}.$$

ただし、$Div(\mathrm{GF}(3^k)[x,y] / (H))$は$\mathrm{GF}(3^k)[x,y] / (H)$の因子群とし、$\langle \rho, y - t \rangle$を$\rho$と$y - t$で生成される因子とする。

このように、多項式選択段階では関数体篩法の初期値設定を行う。この計算時間は無視できるほど小さい。

関係探索段階： 与えられた 2 つの自然数R, Sに対して、以下の条件を満たす多項式の組、$(r,s) \in (\mathrm{GF}(3^k)[x])^2$を十分な個数ほど求める：

$$\deg r \le \mathrm{R}, \deg s \le \mathrm{S}, \gcd(r,s) = 1, \quad (1)$$

$$rm + s = \prod_{\rho_i \in F_R(B)} {\rho_j}^{a_i}, \quad (2)$$

$$(-r)^{d_H} H(x, -s/r) = \prod_{\langle \rho_j, y - t_j \rangle \in F_A(B)} {\rho_j}^{b_j}. \quad (3)$$

ただし、a_i, b_jは非負整数とする。また、$\mathrm{GF}(3^k)(x)[y]/(H)$の類数を$h$とし、$h$は$(3^{6n} - 1)/(3^k - 1)$と互いに素とする。このとき (1) から (3) を満たす(r,s)に対して次の合同式が成り立つ：

$$\sum_{\rho_i \in F_R(B)} a_i \log \rho_i \equiv \sum_{\langle \rho_j, y - t_j \rangle \in F_A(B)} b_j \log \sigma_j \ (\mathrm{mod}\ (3^{6n} - 1)/(3^k - 1)). \quad (4)$$

ただし、$\sigma_i = \xi(t_j)^{1/h}, \langle t_j \rangle = h\langle \rho_j, y - t_j \rangle$とする。合同式 (4) は relation とよばれる。

線形代数段階： 関係探索段階で求めた十分な個数の relation から線形方程式が与えられる。線形代数段階ではこの線型方程式を Lanczos 法などで解き、因子基底の元の離散対数を得る：

$$\log \rho_1, \ldots, \log \rho_{\#F_R(B)}, \log \sigma_1, \ldots, \log \sigma_{\#F_A(B)}.$$

個別離散対数段階： この段階では与えられている離散対数問題の解$\log T$と因子基底の元の離散対数を関係付ける。すなわち下記を満たす整数A_i, B_jを、special-Q descent 等を用いて求める：

$$\log T \equiv \sum_{\rho_i \in F_R(B)} A_i \log \rho_i + \sum_{\langle \rho_j, y - t_j \rangle \in F_A(B)} B_j \log \sigma_j \ (\mathrm{mod}\ (3^{6n} - 1)/(3^k - 1)).$$

2.2　問題設定と FFS のパラメータ設定

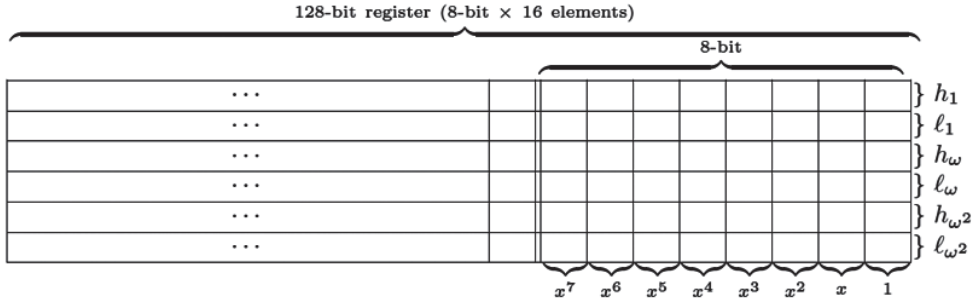
ペアリング暗号の安全性評価の観点から、本稿では$\mathrm{GF}(3^{6\cdot 97})$の乗法部分群であって、位数が 151-bit 素数P_{151}であるものについて議論する。またこのような離散対数問題を JL06-FFS をベースとする関数体篩法で解く際のパラメータの初期値は [7] で報告されている値を導入した。ただしkの値については、我々の実装において$k = 3,6$としてそれぞれ計算実験を行い、その結果から適していると考えられる$k = 3$を選択した。結果として **2.1** で述べたパラメータについて以下のように設定した：

$$(k, d_H, d_m, B, R, \mathrm{S}) = (3,6,33,6,6,6).$$

2.3　関数体篩法の改良

ここでは、$\mathrm{GF}(3^{6\cdot 97})$の離散対数問題を解くために施した、関数体篩法の改良について述べる。本稿で扱っていない改良やその詳細については文献 [1][2] を参照いただきたい。

篩処理の SIMD 実装： ある(r,s)が式 (2)、(3) を満たすかをチェックするためには、素朴に考えると多項式の分解が必要となる。しかし、多項式の分解は計算コストが高いため、篩処理という前処理を行い、その後に残った(r,s)のみから与えられる多項式の素因子分解を行うことで計算量を削減する。式 (2) を例にすると、$rm + s$がρ_jで割り切れるとき、$rm + s \equiv 0 (\mathrm{mod}\, \rho_j)$となるから、ある$r$について$s_0 \equiv -rm (\mathrm{mod}\, \rho_j)$とすると、$s = s_0 + \kappa\rho_j\ (\kappa \in \mathrm{GF}(3^3)[x])$は全て$rm + s \equiv 0 (\mathrm{mod}\, \rho_j)$を満たす。この事実を利用して、十分な個数の$\rho_j \in F_R(B)$に対して上記の計算を行い、$rm + s$が$\rho_j$で割り切れるか否かの情報を集めることで、$rm + s$を直接分解することなく、$(r,s)$に対して式 (2) が成り立つか否かを低コストでチェックすることができる。式 (3) についても同様のことが行える。



Note: an element in $GF(3^3) \cong GF(3)[\omega]/(\omega^3 - \omega - 1)$ is represented using 6-bit $(h_1, \ell_1, h_\omega, \ell_\omega, h_{\omega^2}, \ell_{\omega^2}) \in GF(2)^6$.

図 1　篩処理の SIMD 表現

パラメータ $(B,R,S) = (6,6,6)$ より、篩処理で扱う $\mathrm{GF}(3^3)[x]$ の多項式の次数は 6 以下である。また、$-rm \pmod{\rho_j}$ の計算 $\mathrm{GF}(3^3)[x]$ の多項式に対する x 倍算後に逐次 $\bmod \rho_j$ を計算することで計算できる。よって、篩処理においては 7 次の $\mathrm{GF}(3^3)[x]$ の多項式が表現できれば十分である。このような、小さな対象をたくさん処理する場合には、Single instruction Multiple Data（SIMD）による処理が適している。篩処理で扱うデータ表現を図 1 に示す。この表現では、$\mathrm{GF}(3)$ は 2 ビット $(h,l) \in \mathrm{GF}(2)^2$ で表現され、その演算は最小で 6 回のビット演算で記述できる [8]。また、$\mathrm{GF}(3^3) \cong \mathrm{GF}(3)[\omega]/(\omega^3 - \omega - 1)$ で表現され、$\mathrm{GF}(3^3)[x]$ の x 倍算は左ビットシフト、x 除算は右ビットシフトで記述できる。よってこの表現であれば、最大 16 個の $\mathrm{GF}(3^3)[x]$ 多項式を一度に処理できる。

Frobenius 写像による変数削減と Montgomery 乗算： 線形代数段階で扱う、Lanczos 法などの線形方程式の解法アルゴリズムは、変数の個数 N について $O(N^\epsilon)(2 < \epsilon \leq 3)$ 回の乗算剰余演算を必要とする。このため、変数を削減することで計算量を削減することができる。変数削減の手法として、Frobenius 写像による変数削減がある [4][5]。例えば、因子基底の元 ρ_i Frobenius 写像 ϕ により、別の因子基底の元 ρ_j に写るとすると、$\rho_j = \rho_i^{3^{97^2}} = \phi(\rho_i)$ となることから、

$$\log \rho_j \equiv 3^{97^2} \log(\rho_i) \pmod{P_{151}}$$

となり、$\log \rho_j$ を線形方程式の変数から取り除くことができる。これにより変数の削減が可能となるが、係数が $3^{97^2} \pmod{P_{151}}$ 倍されて P_{151} 程度に大きくなる。（そもそも、合同式 (4) の各係数は、式 (2)、(3) の指数部分であたるため、高々数十程度の大きさである。）このため、特に乗算後の剰余演算の計算量が大きくなってしまう。この計算量増大を抑えるために、Montgomery 乗算を用いた。

素数 P_{151} と互いに素である整数として $A = 2^k$（k は通常、CPU のワード長を選ぶ）を用意し、各係数を以下のような写像

$$\mathbb{Z}_{P_{151}} \to A\mathbb{Z}_{P_{151}}$$
$$a \mapsto Aa \pmod{P_{151}}$$

で $A\,\mathbb{Z}_{P_{151}}$ に写し、Montgomery 乗算

$$Aa \otimes Ab = A(ab) \pmod{P_{151}}$$

により、乗算剰余演算を行う。Montgomery 乗算では剰余算は乗算とビットシフトに置き換えられるため、実質的に剰余算の計算を必要とせずに乗算剰余演算を実現できる。

2.4　実験結果

計算実験の結果を段階ごとに説明する。

関係探索段階： 篩処理等により得られた relation は全部で 187,602,242 個（使われた因子基底 134,697,663 個）である。そのうち 33,786,299 個が free relation と呼ばれる、計算を必要とせずに得られる自明な relation である。これらは 212 CPU コアを用いて 53.1 日間で計算でき、2011 年 5 月 14 日に計算を開始し、2011 年 9 月 9 日に終了、実時間 118 日（計画停電やコード改良による停止を含む）を要した。

線形代数段階： 式の数 187,602,242、変数の数 134,697,663 の線形方程式について、まず Frobenius 写像で変数削減を行い、変数の数が 45,059,572 に削減された。その後、前処理として filtering 処理を行い、式の個数が 6,141,443、変数の個数が 6,121,440 に削減された。この線形方程式を並列 Lanczos 法により解いた。これらは 252 CPU コアを用いて 80.1 日間で計算でき、2012 年 1 月 16 日に計算開始、2012 年 4 月 14 日に終了、実時間 90 日を要した。

個別離散対数段階： 与えられた離散対数問題の解と因子基底の元の離散対数の関係式を得るために、168 CPU コアを使用して計算を行い、2012 年 2 月 3 日に計算を開始し、2012 年 2 月 28 日に終了し、実時間で 15 日を必要とした。この段階の計算は線形代数段階とは独立に計算できるため、線形代数段階計算中に、別のサーバで計算を行った。

線形代数段階の計算終了の後、2012 年 4 月 24 日に与えられた離散対数問題の解の計算とその検証が終了

した。すなわち位数がP_{151}である、GF($3^{6\cdot97}$)の部分群における離散対数の計算に成功した。実際の解やその確認用スクリプトについては、文献 [2] を参照いただきたい。

3 格子暗号の安全性評価

格子暗号はその安全性の根拠として、ある条件をもつ格子点の探索問題をおいている。例えば、ゼロ点以外で最もノルムの小さい格子点を探索する最短ベクトル問題、与えられた空間内の点に最も近い格子点を発見する最近ベクトル問題、その亜種の Learning With Errors 問題（以下 LWE 問題と呼称）等がある。

暗号を実社会で運用するためには、鍵長などのパラメータを適切に設定することが不可欠であり、そのためにはどの程度のパラメータを持った暗号方式がどの程度の時間で解けてしまうのかを知ることが必要である。実社会での使用に耐える暗号パラメータを設定するためには、その暗号パラメータを持つ暗号方式が現実的な時間及び機材では解読不可能であることを示さなければならないため、実際に暗号を解く以外にも解読シミュレーションを用いて暗号の強度評価を行う必要がある。本節では、格子点探索問題の代表的な解法である格子攻撃の概説と、その具体例として LWE 問題の評価について述べる。

3.1 格子攻撃の概要

格子暗号を解読するため、与えられた暗号学的問題（例えば公開鍵と暗号文のペアから平文を復元する問題）を、格子$L=(\boldsymbol{b}_1,\dots,\boldsymbol{b}_n)$上のある条件をみたす点を探索する問題に変換する。この格子に対して、格子基底簡約アルゴリズムを適用した後、格子点探索アルゴリズムによって目的の点$\boldsymbol{v}$を見つけ、そこから秘匿情報を復元する。

前半の格子基底簡約ステップでは、LLL アルゴリズムや BKZ アルゴリズム等の格子基底簡約アルゴリズムが、単体あるいはそれらを組み合わせた形で用いられる。格子基底簡約により後半の格子点探索の計算時間が削減されるため、格子基底簡約ステップと格子点探索ステップの間には計算時間のトレードオフ関係がある。

格子点探索アルゴリズム：最も単純な例として、格子の最短ベクトルを探索する ENUM アルゴリズム [9] について述べる。まず、入力の格子基底$(\boldsymbol{b}_1,\dots,\boldsymbol{b}_n)$に対してその Gram-Schmidt 基底$(\boldsymbol{b}_1^*,\dots,\boldsymbol{b}_n^*)$を計算し、以下の探索木を深さ優先探索によって探索する。

- 各ノードは格子点がラベル付されており、根ノードにはゼロベクトルが対応する。
- 深さkのノードには、$\boldsymbol{v}=\sum_{i=n-k+1}^{n}a_i\boldsymbol{b}_i$（$\forall i, a_i\in\mathbf{Z}$）の形のベクトルが対応し、その子ノードは全てベクトル$\boldsymbol{v}+a_{n-k}\boldsymbol{b}_{n-k}$（$a_{n-k}\in\mathbf{Z}$）が対応する。

この性質により木の深さはnとなり、末端の葉ノードと全ての格子点は 1 対 1 に対応することになる。格子点の個数は無限個であるため、実際の探索では深さkのノードを、対応するベクトル$\boldsymbol{v}$の射影長$|\pi_{n-k+1}(\boldsymbol{v})|$が基準値$c$よりも大きい場合に枝狩りをすることで探索範囲を限定している。ここで、cは探索半径と呼ばれるパラメータであり、上記格子点探索アルゴリズムにより、長さc以下の格子点がすべて列挙できることが保証されている。

このアルゴリズムを最短ベクトルの発見に用いる場合、$c=\min(|\boldsymbol{b}_1|,\dots,|\boldsymbol{b}_n|)$とする。探索によって複数のベクトルが発見された場合にはそれらの中で一番短い非ゼロベクトルを出力し、何も見つからなかった場合には、基底ベクトルの中で一番短いものが格子の最短ベクトルであることが判明する。

格子点探索アルゴリズムの計算量：上記探索において、探索ノード数は以下の式で高精度に予測可能であることが知られている [10]。

$$\mathrm{N}=\sum_{i=1}^{n}\frac{V_i(c)}{\prod_{j=n-i+1}^{n}|\boldsymbol{b}_i^*|} \tag{5}$$

だたし、$V_i(c)$は半径cのi次元球の体積である。この式から計算量は Gram-Schmidt 基底長（$|\boldsymbol{b}_1^*|,\dots,|\boldsymbol{b}_n^*|$）のみによって予測できることがわかり、形から、計算量を削減するためには後半のインデックス$i=n,n-1,\dots$に対する$|\boldsymbol{b}_i^*|$を大きくする必要があることがわかる。格子基底の性質から、格子の体積$\prod_{i=1}^{n}|\boldsymbol{b}_i^*|$は一定であるため、後半の$|\boldsymbol{b}_i^*|$を大きくすることは前半の$|\boldsymbol{b}_i^*|$を小さくすることに対応する。

格子基底簡約アルゴリズム：格子基底簡約を行うことで、格子点探索の計算量 (5) が削減できることが知られている。例として、格子暗号評価で頻繁に使われる BKZ アルゴリズムを紹介する。

アルゴリズムの入力は基底L以外にブロックサイズパラメータβがある。次元βの部分射影格子$L_{[i:i+\beta-1]}:=\pi_i(\boldsymbol{b}_i,\dots,\boldsymbol{b}_{i+\beta-1})$において最短ベクトル$\boldsymbol{v}$を求め、$\boldsymbol{b}_i$と置き換える操作を$i=1,2,\dots$と逐次的に行うことで、少しずつ基底の質を上げて探索計算量 (5) を削減できる。途中で$i+\beta-1$が格子の次元数nを超える場合にはnで読みかえるものとする。サブルーチンとして ENUM アルゴリズムを呼んでいるため、比較的正確な計算量予測が可能であることも特徴である。

ENUM サブルーチンによって発見されたベクトル

は $|\pi_i(\boldsymbol{v})| \le |\pi_i(b_i)| = |\boldsymbol{b}_i^*|$ を満たす。インデックス i における新たな $|\boldsymbol{b}_i^*|$ は以前よりも小さくなる可能性があり、その分後ろの $|\boldsymbol{b}_i^*|$ が大きくなることがわかるため、基底簡約が進む。

以上の操作を $i = 1,2,\dots,n-1$ に対して行った後、再び $i = 1$ に戻る。全ての i に対して $\boldsymbol{b}_i^*$ が $L_{[i:i+\beta-1]}$ の最短ベクトルとなったとき、これ以上の更新が無くなりアルゴリズムは停止する。

以上が Schnorr と Euchner[11] により提案された BKZ アルゴリズムの概要であるが、実際の問題に合わせて様々な高速化手法が提案されている。例えば、格子の短いベクトルを求める目的であれば $\mathbf{b}_1$ がある程度短くなった時点で計算を打ち切る手法 [12] や、格子点探索アルゴリズムの探索半径を $|\mathbf{b}_i^*|$ ではなく Gaussian-Heuristic と呼ばれる、最短ベクトル長の期待値とすることで計算量を下げる手法 [13] などがある。

3.2　格子攻撃のシミュレーション

格子暗号の解読時間評価のため、格子基底簡約アルゴリズムと格子点探索アルゴリズム双方のシミュレーションを行う必要がある。前述のように、格子点探索の計算量は格子基底簡約アルゴリズムの出力の Gram-Schmidt 基底長 $(|b_1^*|,\dots,|b_n^*|)$ から予測することが可能であるため、格子基底簡約アルゴリズムについて計算時間と出力の Gram-Schmidt 長のシミュレーションを行えば目的を果たせる。つまり、BKZ アルゴリズムのパラメータ調整により格子暗号解読の最短計算時間をシミュレート可能であり、この予測が暗号の安全性評価となる。

出力される $(|\boldsymbol{b}_1^*|,\dots,|\boldsymbol{b}_n^*|)$ のシミュレーションは文献 [13] の BKZ シミュレーターによって行う。以下、$|\boldsymbol{b}_i^*|$ のシミュレーション値を ℓ_i と書き、BKZ アルゴリズムの各インデックス i におけるシミュレーションを以下のように行う。ENUM サブルーチンによって発見される、射影部分格子 $L_{[i:i+\beta-1]}$ の最短ベクトルの長さは格子の Gaussian-Heuristic

$$\mathrm{GH}\left(L_{[i:i+\beta-1]}\right) \coloneqq V_\beta(1)^{-\frac{1}{\beta}} \cdot \mathrm{vol}\left(L_{[i:i+\beta-1]}\right)^{\frac{1}{\beta}}$$

によってシミュレーションできることが知られている。いま、射影部分格子の体積のシミュレーション値

$$\mathrm{vol}\left(L_{[i:i+\beta-1]}\right) = \prod_{j=i}^{i+\beta-1} \ell_j$$

を代入することで新たな $|\boldsymbol{b}_i^*|$ のシミュレーション値を計算することができる。これを ℓ_{i-new} と書く。格子基底の性質から、基底の更新後にも ℓ_i の積は変化しないため、ℓ_{i+1} を新たに $\ell_i \cdot \frac{\ell_i}{\ell_{i-new}}$ で置き換えた後、次のインデックスに進む。

以上の操作を通常の BKZ アルゴリズムと同様に、$i = 1,\dots,n-1$ に対して適切な回数繰り返すことで、ブロックサイズ β の BKZ アルゴリズムの出力基底のシミュレーションが可能となる。計算時間は ENUM アルゴリズムの計算量の累計となる。

3.3　LWE 問題に対する格子攻撃のシミュレーション

LWE 問題はパラメータ $(n, m, q, \boldsymbol{s})$ に対して、ランダム行列 $A \in \mathbf{Z}_q^{n\times m}$ と、以下の関係式で計算されたベクトル $\mathbf{b}$ が与えられたとき、計算に使われたベクトル $\mathbf{e}$ と $\mathbf{s}$ を求める問題である。

$$\mathbf{b} = A\mathbf{s} + \mathbf{e} \pmod q \qquad (6)$$

ただし、問題作成時に $\mathbf{s}$ は $\boldsymbol{Z}_q^n$ からランダムに、ベクトル $\mathbf{e}$ の各成分は分散値 $\boldsymbol{s}^2$ の離散的ガウス分布から一様独立にサンプリングするものとする。

問題のインスタンス $(A, \mathbf{b})$ から格子点探索問題への変換は以下の手順で行う。関係式 (6) よりある整数ベクトル $\mathbf{w}$ を用いて

$$\begin{aligned}\mathbf{b} &= A\mathbf{s} + \mathbf{e} + q\mathbf{w}\\ &= [A \quad qI]\begin{bmatrix}\mathbf{s}\\ \mathbf{w}\end{bmatrix} + \mathbf{e}\end{aligned}$$

と書くことができる。よって、上式内の行列で指定される格子内の、$\mathbf{b}$ の最近ベクトルを求めればそこから問題の解を導出できることがわかる。格子攻撃の計算時間を評価するため、前述の手法を用いて格子基底簡約後の $(|\boldsymbol{b}_1^*|,\dots,|\boldsymbol{b}_n^*|)$ をシミュレートし、さらに、格子点探索の時間を求める。

3.4　格子点探索手法の改良

我々は LWE 問題の評価 [3] を行うため、**3.1** で述べた格子点探索アルゴリズムを以下のように改良した。深さ k における生存条件、つまりこの条件を満たさない場合に枝狩りを行う条件を

$$L_k \le |\pi_{n-k+1}(\boldsymbol{v} - \boldsymbol{b})| \le R_k \qquad (7)$$

のように上下から抑える形とし、L_k 及び R_k を $\mathbf{e}$ の各

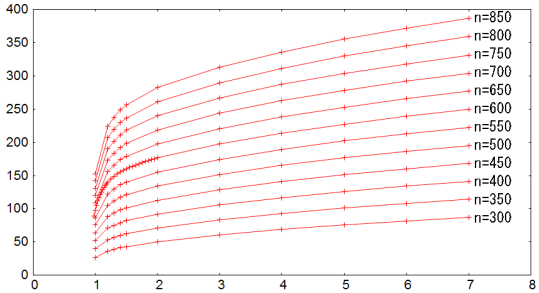


図 2　q=32749 と様々な s (横軸) に対する bit-security=$\log_2$ (計算時間 [秒])

成分がガウス分布であるという性質を用いて改良した。この枝狩りにおける、格子点探索の計算量は

$$N=\sum_{i=1}^{n}\frac{\mathrm{Vol}C_k}{\prod_{j=n-i+1}^{n}|b_i^*|}$$

で評価される。ここで、C_k は探索範囲 (7) から定められる k 次元物体

$$\{(x_1,\ldots,x_k):\ L_k \le x_1^2+\cdots+x_k^2 \le R_k\}$$

である。アルゴリズムの詳細は文献 [3] を参照いただきたい。図 2 に代表的なパラメータに対するビットセキュリティのグラフを掲載する。

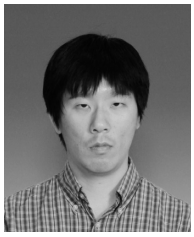
4　まとめと今後の展望

ペアリング暗号の安全性の基盤となっている有限体上の離散対数問題を効率よく解く手法を提案し、それを用いて実装及び実験を行うことで有限体 $\mathrm{GF}(3^{6\cdot97})$ 上の離散対数問題を解くことに成功した。また、格子暗号については既存研究では顧みられなかったガウス分布の詳細な性質を用いることで、格子点探索を改良し、攻撃手法の高速化につながった。これらの成果は、安全な暗号パラメータの見積もりに利用される。本研究室では引き続き離散対数問題や格子に関する問題を解く計算手法の改良に取り組み、さらに暗号の安全性評価に関する他の数学的な問題についても研究開発を続けていく。

【参考文献】

1 T. Hayashi, T. Shimoyama, N. Shinohara, and T. Takagi, “Breaking Pairing-Based Cryptosystems Using ηT Pairing over GF(3^{97}),” ASIACRYPT 2012, LNCS 7658, pp.43–60, 2012.
2 T. Hayashi, T. Shimoyama, N. Shinohara, and T. Takagi, “Breaking Pairing-Based Cryptosystems Using ηT Pairing over GF(3^{97}),” IACR Cryptology ePrint Archive 2012:345, 2012.
3 Y. Aono, X. Boyen, L. T. Phong, and L.Wang, “Key-private proxy re-encryption under LWE,” INDOCRYPT 2013, LNCS 8250, pp.1–18, 2013.
4 A. Joux and R. Lercier, “The function field sieve in the medium prime case,” EUROCRYPT 2006, LNCS 4004, pp.254–270, 2006.
5 T. Hayashi, N. Shinohara, L. Wang, S. Matsuo, M. Shirase, and T. Takagi, “Solving a 676-bit discrete logarithm problem in GF(3^{6n}),” PKC 2010, LNCS 6056, pp.351–367, 2010.
6 L. M. Adleman, “The Function Field Sieve,” ANTS-I, LNCS 877, pp.108–121, 1994.
7 N. Shinohara, T. Shimoyama, T. Hayashi, and T. Takagi, “Key length estimation of pairing-based cryptosystems using ηT pairing over GF(3^n),” IEICE Transactions, vol.97-A, no.1, pp.236–244, 2014.
8 Y. Kawahara, K. Aoki, and T. Takagi, “Faster Implementation of eta-T Pairing over GF(3^m) Using Minimum Number of Logical Instructions for GF(3)-Addition,” Pairing 2008, LNCS 5209, pp.282–296, 2008.
9 R. Kannan, “Improved algorithms for integer programming and related lattice Problems,” STOC 1983, pp.193–206, 1983.
10 N. Gama, P. Q. Nguyen, and O. Regev, “Lattice enumeration using extreme pruning”, EUROCRYPT 2010, LNCS, vol.6110, pp.257–278, 2010.
11 C. P. Schnorr and M. Euchner, “Lattice basis reduction: improved practical algorithms and solving subset sum problems,” Math. Program., vol.66, no.1–3, pp.181–199,1994.
12 G. Hanrot, X. Pujol, and D. Stehle,“Analyzing blockwise lattice algorithms using dynamical systems,” CRYPTO 2011, LNCS, vol.6841, pp.447–464, 2011.
13 Y. Chen and P. Q. Nguyen, “BKZ 2.0: Better lattice security estimates,” ASIACRYPT 2011, LNCS, vol.7073, pp.1–20, 2011.

篠原直行　（しのはら　なおゆき）
サイバーセキュリティ研究所
セキュリティ基盤研究室
主任研究員
博士 (数理学)
暗号解析、計算機整数論



青野良範　（あおの　よしのり）
サイバーセキュリティ研究所
セキュリティ基盤研究室
研究員
博士 (理学)
暗号解析、格子理論

林　卓也　（はやし　たくや）
サイバーセキュリティ研究所
セキュリティ基盤研究室
研究員
博士 (機能数理学)
暗号解析、高速実装