イジング計算を用いた暗号解析について

山口　純平，伊豆　哲也

本稿ではイジング計算を用いた暗号解析のいくつかの事例を紹介する．具体的には，公開鍵暗号の安全性を保証する数学問題である素因数分解問題，多変数多項式の逆像問題，格子最短ベクトル問題の求解事例，および共通鍵暗号 AES の攪拌関数 (S-BOX) の解析事例を紹介する．

キーワード：イジング計算，暗号解析，素因数分解，格子最短ベクトル問題

1. はじめに

デジタル化社会の進展により，暗号技術 (cryptography) の重要性が増している[1]．暗号解析 (cryptanalysis) とは，暗号を解読するアルゴリズムを開発し，解読に必要な計算資源量を算出することであり，暗号の安全性を保証する役割を担っている．従来の暗号解析は汎用計算機（スーパーコンピュータを含むノイマン型計算機）を前提としてきたが，近年，非ノイマン型計算機の開発が急速に進んでおり，特にイジング計算機 [1] が新しい脅威の可能性として注目されている．

暗号は共通鍵暗号 (Symmetric-key Cryptosystem) と公開鍵暗号 (Public-key Cryptosystem) に大別される．共通鍵暗号は平文の暗号化と暗号文の復号に同じ秘密情報（共通鍵）を使用する暗号方式で，米国標準共通鍵暗号方式である AES (Advanced Encryption Standard) が世界中で広く使用されている．AES 暗号化/復号 では 128 ビットの平文/暗号文と 128/192/256 ビットの鍵を入力とし，128 ビットの暗号文/平文を出力する．これに対し公開鍵暗号は平文の暗号化と暗号文の復号に異なる鍵を使用し，暗号化鍵は他者にも公開する（公開鍵）一方で，復号鍵はユーザが秘密に保持する（秘密鍵）．米国標準公開鍵暗号方式である RSA (Rivest-Shamir-Adlemen) 暗号/署名や DH (Diffie-Hellman) 鍵共有などが世界中で広く使用されている．これら公開鍵暗号方式は，数学問題を利用することで実現されている．RSA 暗号/署名では素因数分解問題，DH 鍵共有では離散対数問題という数学問題が使用され，これら数学問題を解くことが難しいことが暗号の安全性を保証している．ノイマン型計算機を用いてこれらの数学問題を解く場合，問題サイズの準指数関数時間を要する．そこでこれまでの解読状況や計算機の開発予想などを勘案することで，RSA 暗号/署名では 2,048 ビット以上の問題サイズ（鍵長）が使用されている．しかし量子ゲート計算機 [2] を用いた場合，これらの数学問題は多項式関数時間で解けるため [3]，さらなる安全性を確保するために，量子ゲート計算機でも解読できない公開鍵暗号として耐量子計算機暗号 (PQC, Post Quantum Cryptosystem) の開発が始まっている．PQC を実現するために多変数多項式の逆像問題，格子における最短ベクトル問題，符号問題，同種写像問題などのさまざまな数学問題が使用されている．

本稿の目的は，イジング計算を用いた暗号解析のこれまでの事例を紹介することである．残念ながら現在のイジング計算能力には限界があるため，実社会で使用されている暗号方式そのものを解析することは難しい．そこで共通鍵暗号の場合には，暗号方式の部分的な処理（攪拌処理）を解析の対象としている．また公開鍵暗号の場合には数学問題を解析対象とし，小さなサイズの数学問題を実際に解く試みがなされている．

以下，イジング計算を用いた暗号解析の事例を具体的に紹介する．公開鍵暗号に関しては，暗号の安全性を保証する数学問題である素因数分解問題，多変数多項式の逆像問題，格子最短ベクトル問題の求解事例を紹介する[2]．また共通鍵暗号に関しては，S-BOX と呼ばれる攪拌関数の解析事例を紹介する．

2. 素因数分解問題

素因数分解問題 (IFP, Integer Factorization Problem) とは，与えられた合成数を素因数の積に分解する問題であり，数学における古典的な問題である．標準

やまぐち　じゅんぺい，いず　てつや
富士通株式会社富士通研究所 データ＆セキュリティ研究所
j-yamaguchi@fujitsu.com,
izu@fujitsu.com

[1] 暗号技術にはデータを秘匿化するための暗号，完全性を保証するための署名，エンティティを確認するための認証などが含まれるが，本稿では主に暗号について扱う．

[2] これら以外にもイジング計算を用いた離散対数問題の解法アルゴリズムが発表されているが [4]，分量の関係から本稿では触れていない．

公開鍵暗号の一つである RSA 暗号は，同じビット長の二つの素数の積である合成数を秘密鍵として使用しており，素因数分解が困難であることが暗号の安全性の根拠となっている．汎用計算機を用いた場合，最良の素因数分解法である一般数体篩法 [5] は 829 ビット合成数の分解に成功している [6]．一般数体篩法の計算量は合成数サイズの準指数関数となることから，2,048 ビット以上の合成数は事実上素因数分解は不可能であり，RSA 暗号は安全であると考えられている [7]．なお量子ゲート計算機を用いた場合，素因数分解問題は多項式時間で解ける [3] が，量子ゲート計算機を用いた最大事例は 21 の素因数分解であり [8, 9]，現状での脅威は小さい．

以下では 二つの ℓ ビット素数 p, q の積として合成数 $N = p \times q$ を定める．簡単のためそのビット長を 2ℓ ビットと仮定し，

$$N = \sum_{i=0}^{2\ell-1} N_i 2^i,\ N_i \in \{0,1\},\ N_0 = N_{2\ell-1} = 1,$$

と表すことにする．

2.1 素朴法

合成数 $N = p \times q$ に対し，関数 $\mathcal{H}(x,y) = (N - xy)^2$ は非負であり，$(x,y) = (1,N), (p,q), (q,p), (N,1)$ のときに最小値 0 をとる．この関数をハミルトニアン（0 または 1 の 2 値を取るバイナリ変数からなる多変数 2 次多項式）に変換してイジング計算を用いると，解 $(x,y) = (p,q), (q,p)$ から N の素因数分解が求められる．この素因数分解法を素朴法 (Naive Method) と呼ぶ．素朴法はシンプルでわかりやすい半面，ハミルトニアンの定数項 N^2 が大きくなってしまい，実際の数値実験には向かない．

2.2 筆算法

筆算法 (Multiplication-table Method) は，その名のとおり乗算の筆算を利用した素因数分解法である．たとえば $N = 143 = (1000\ 1111)_2$ に対し，二つの変数 $x = (1x_2x_1 1)_2$，$y = (1y_2y_1 1)_2$ の乗算の筆算は図 1 のようになる．ここで s_i は i 列目から $i+1$ 列目への繰り上がりビット，t_i は i 列目から $i+2$ 列目への繰り上がりビットを表す（筆算の最右列を 1 列目，その左の列を 2 列目，... と表記する）．このとき，求めるべき x_2, x_1, y_2, y_1 は，筆算の各列から得られる連立方程式

				1	x_2	x_1	1
			$\times)$	1	y_2	y_1	1
				1	x_2	x_1	1
			y_1	x_2y_1	x_1y_1	y_1	
		y_2	x_2y_2	x_1y_2	y_2		
	1	x_2	x_1	1			
s_7	s_6	s_5	s_4	s_3	s_2		
t_7	t_6	t_5	t_4				
1	0	0	0	1	1	1	1

図 1 2 進法表記による $N = 11 \times 13$ の筆算

$$B_1 = (x_1 + y_1) - (1 + 2s_2) = 0$$
$$B_2 = (x_2 + x_1y_1 + y_2 + s_2) - (1 + 2s_3 + 4t_4) = 0$$
$$B_3 = (1 + x_2y_1 + x_1y_2 + 1 + s_3) - (1 + 2s_4 + 4t_5) = 0$$
$$B_4 = (y_1 + x_2y_2 + x_1 + s_4 + t_4) - (2s_5 + 4t_6) = 0$$
$$B_5 = (y_2 + x_2 + s_5 + t_5) - (2s_6 + 4t_7) = 0$$
$$B_6 = (1 + s_6 + t_6) - 2s_7 = 0$$
$$B_7 = (s_7 + t_7) - 1 = 0$$

の解となる．この方程式は 14 変数の 7 連立方程式であるが，すべての変数はバイナリであり，x, y は素数であることから，本質的な解は一組に限定される．

このとき関数 $\mathcal{H}(x,y) = \sum_{i=1}^{7} B_i^2$ は非負であり，すべての B_i が 0 となるときにだけ最小値 0 をとる．よってこの関数をハミルトニアンとしたイジング計算を用いることで素因数分解が可能となる．このときハミルトニアンの係数は素朴法のハミルトニアンの係数に比べて小さくなっている．

$N = 143$ に対する筆算法から得られる関係式 $B_1 = (x_1 + y_1) - (1 + 2s_2)$ に着目すると，x_1, y_1 はバイナリ変数であることからその合計値は 2 以下であるため，$1 + 2s_2$ も 2 以下でなければならず，結果として $s_2 = 0$ となることが（イジング計算なしで）わかる．同様の計算により，連立方程式から変数値が自明な変数を消去することを変数消去 (variable elimination) と呼ぶ．変数消去により，最終的なハミルトニアンの変数の個数・係数の大きさが抑制できるため，現実的な効果はとても大きい．$N = 143$ の場合，変数消去後に得られる連立方程式は $x_1 + y_1 = 1$, $x_2 + y_2 = 1$, $x_1y_2 + x_2y_1 = 1$ の三つに簡約化され，ハミルトニアンの変数の個数・係数の大きさが抑制できる．しかし，平均的には数個程度の変数削減しかできないため，合成数 N が大きくなった場合での効果は限定的である．

表 1　イジング計算を用いた素因数分解結果

分解年	計算デバイス	N	$\|N\|$	#V	#Q	文献
2008 年	NMR	21	5	3		[11]
2011 年	NMR	143	8	4		[12]
2016 年	NMR	551	10	3		[13]
2016 年	D-Wave 2X	200099	18	72	897	[14]
2017 年	NMR	291311	19	3		[15]
2018 年	D-Wave 2000Q	8137	13	43	535	[16]
2019 年	Digital Annealer	541000303	30	154		[10]
2019 年	qbsolv	1005973	20	89		[17]
2020 年	Digital Annealer	4049874797	32	310		[18]

2.3　部分的生成法

筆算法が用いるハミルトニアンはすべての（意味のある）関係式 $B_1, \ldots, B_{2\ell-1}$ の二乗和であることから，変数の個数や係数が大きくなってしまう．そこで清水ら [10] は，一部の関係式だけからハミルトニアンを生成することで，変数の個数や係数の大きさを抑制する方法を考案した（部分的生成法）．イジング計算機においては，変数の個数が小さなハミルトニアンの方が最小値の求まる確率が高いことから，部分的生成法が用いる関係式の個数と素因数分解の成功確率はトレードオフの関係となる．使用する関係式の選択法については提案論文 [10] を参照されたい．なお部分的生成法によるハミルトニアンの最小値は，正しい素因数分解のほかに，N の素因数分解に対応しない解を導く可能性がある．しかしイジング計算による素因数分解の場合，素因数分解が得られたかどうかの判定は容易であるため，現実的な問題が生じない．

2.4　実験結果

イジング計算による素因数分解実験の結果を表 1 にまとめる．ここで N は素因数分解におけるターゲット合成数，$|N|$ はそのビット長を表す．また #V はハミルトニアンの変数の個数，#Q は使用した（量子）ビット数を表す．

2008 年にイジング計算を用いた最初の素因数分解実験が報告され，素朴法により $N = 21$ の分解に成功した [11]．ただし実験環境の制約から，合成数 $N = 21$ に対するハミルトニアン $\mathcal{H}(x, y) = (21 - (2x_1 + 1)(4y_2 + 2y_1 + 1))^2$ を用いることで必要な量子ビットの個数を 3 個に抑制し，イジングモデルとは異なるモデル上で計算を実現した．しかし素朴法はハミルトニアンの係数が大きくなりがちなため，これ以降は筆算法が使用されている．

現時点での記録は，伊豆ら [18] が 2020 年に報告した $N = 4049874797$（32 ビット）の素因数分解であり，富士通が開発したイジング計算機 Digital Annealer（第二世代）が使用された．

3.　多変数多項式の逆像問題

多変数多項式の逆像問題 (MPP, Multivariate Polynomial Problem) とは，多変数多項式の連立方程式の求解問題のことで，一般には n 個の変数 $x_1, x_2, \ldots, x_n$ に関する m 個の多変数多項式

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ &\vdots \\ f_m(x_1, x_2, \ldots, x_n) &= 0 \end{aligned}$$

の解 $(x_1, x_2, \ldots, x_n) = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$ を求める問題である．暗号で利用する場合，方程式は適切な有限体 k 上で考え，多項式を 2 次に限定する場合が多い．一般に MPP は NP 完全であるため，この問題を利用した暗号は量子ゲート計算機を用いても解読が困難であると予想されている．

九州大学は MPP 問題の現実的な難しさを評価するため，Fukuoka MQ Challenge というチャレンジ問題を公開している [19]．Fukuoka MQ Challenge では (k, n, m) の組み合わせによって六つの問題群が定められ，type I, II, III の三つの問題群は $m = 2n$, type IV, V, VI の三つの問題群は $n \sim 1.5m$ となっている．

下山ら [20] はイジング計算機を用いた MPP に対する 3 種類の解法 LSL1, LSL2, LSL3 を提案した．LSL1 とその改良である LSL2, LSL3 は素因数分解における筆算法を有限体 $k = GF(p)$ に拡張した解法と

なっている．これらの解法では，まず有限体上の各式 f_i を $\mathbb{Z}$ 上の式 F_i にもち上げ

$$F_i(x_1, x_2, \ldots, x_n, y_i) = f_i(x_1, x_2, \ldots, x_n) + py_i = 0,$$

次に，各整数変数 $x_1, \ldots, x_n, y_1, \ldots, y_m$ を筆算法と同様にバイナリ変数に変換し，$\mathcal{H} = \sum_{i=1}^{m} F_i^2$ をハミルトニアンとしたイジング計算を用いることで MPP を求解する．また下山ら [20] は Fukuoka MQ Challenge のいくつかの問題に対し，LSL1, LSL2, LSL3 を適用した場合のハミルトニアンの変数の個数と方程式の個数を算出しているが，イジング計算機を用いた数値実験には至っていない．

4. 最短ベクトル問題

自然数 n に対して，1 次独立な整数ベクトルの組 $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Z}^n$ で生成される集合

$$L(\mathbf{B}) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \in \mathbb{Z}^n \ ; \ x_1, \ldots, x_n \in \mathbb{Z} \right\}$$

を $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{Z}^{n \times n}$ を基底とする n 次元の格子と呼ぶ．最短ベクトル問題 (SVP, Shortest Vector Problem) とは，格子の基底 $\mathbf{B}$ が与えられたとき，$L(\mathbf{B})$ に含まれる非零な最短ベクトルを求める問題，つまり，$\mathrm{argmin}_{\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}} \|\mathbf{Bx}\|^2$ を計算する問題である．理想的な量子ゲート計算機の存在を仮定しても SVP を多項式時間で解くアルゴリズムは現在では発見されていないため，SVP の求解困難性を利用した格子暗号は耐量子計算機暗号であると期待されている．

Darmstadt SVP Challenge [21] で SVP の計算量評価が行われている．厳密には SVP ではなく，最短ベクトルの見積もり長の 1.05 倍以下の長さの格子ベクトルを見つけた場合は求解成功となる近似 SVP の計算量評価が行われている．汎用計算機を用いた計算では，現在では $n = 180$ 次元の近似 SVP の求解が最高記録となっている．一方で，Digital Annealer（第二世代）を用いた解読では，最大で $n = 45$ の SVP の求解に成功している [22]．他の暗号問題と異なり SVP は，近似の問題が存在することや近似解でも価値がある（近似解があればさらなる近似解を得やすい）ため，イジング計算機との相性が良い問題であるといえる．

イジング計算機を用いた SVP 求解法として，主に分割探索法 [23]，近似 Enum 法 [24]，疑似マルチスピンフリップ [25] の三つが知られている．以下では分割探索法と疑似マルチスピンフリップの概略を説明し，Darmstadt SVP Challenge を用いたこれらの比較と現在の記録について紹介する．

4.1 分割探索法

格子の基底 $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{Z}^{n \times n}$ が与えられたとき，分割探索法は以下のようにして最短ベクトルを計算する．まず，格子の対称性を利用して SVP の探索範囲を次の n 個に分割する．

$$X_i = \left\{ (x_1, \ldots, x_n)^{\mathrm{T}} \in \mathbb{Z}^n : x_i \geq 1, x_{i+1} = \ldots = x_n = 0 \right\}.$$

次に，各範囲 X_i に対応するハミルトニアン $\mathcal{H}_i$ を以下の方法で生成し，イジング計算機で $\mathcal{H}_i$ の最小解 $\mathbf{x}_i \in X_i$ を計算する．このとき，$\|\mathbf{Bx}_k\|$ が最小となる k $(1 \leq k \leq n)$ に対して $\mathbf{Bx}_k$ が最短ベクトルとなる．

ハミルトニアン $\mathcal{H}_i$ は以下のように生成する．まず，範囲 X_i 内のベクトルをバイナリ変数ベクトルに encoding [26] する．具体的には $\mathbf{x} = (x_1, \ldots, x_n)^{\mathrm{T}} \in X_i$ とすると，$1 \leq j \leq i$ に対して

$$x_j = \sum_{\ell=1}^{m_j} a_{j,\ell} x_{j,\ell} + a_{j,m_j+1} \ (a_{j,\ell} \in \mathbb{Z}, x_{j,\ell} \in \{0,1\})$$

とする．ここで m_j は整数変数 x_j をバイナリ変数で表したときの変数の個数である．各 $1 \leq j \leq i$ に対して，$\sum_{k=1}^{j-1} m_k + 1$ 番目から $\sum_{k=1}^{j} m_k$ 番目に $a_{j,1}, \ldots, a_{j,m_j}$ を並べ，$m \coloneqq \sum_{k=1}^{i} m_k + 1$ 番目に a_{j,m_j+1} を置いたベクトルを

$$\mathbf{a}_j = (0, \ldots, 0, a_{j,1}, \ldots, a_{j,m_j}, 0, \ldots, 0, a_{j,m_j+1})^{\mathrm{T}} \in \mathbb{Z}^m$$

とし，バイナリ変数を順番に並べたベクトルを

$$\mathbf{t} = (x_{1,1}, \ldots, x_{1,m_1}, \ldots, x_{i,1}, \ldots, x_{i,m_i}, 1)^{\mathrm{T}} \in \{0,1\}^m$$

とする．このとき $\mathbf{A}_i = (\mathbf{a}_1, ..., \mathbf{a}_n)^{\mathrm{T}} \in \mathbb{Z}^{n \times m}$ とすると（ただし $\mathbf{a}_{i+1} = \cdots = \mathbf{a}_n = \mathbf{0}$），整数変数ベクトルの encoding は $\mathbf{x} = \mathbf{A}_i \mathbf{t}$ で表せる．以上より，範囲 X_i のハミルトニアン $\mathcal{H}_i = \|\mathbf{Bx}\|^2 = \|\mathbf{BA}_i \mathbf{t}\|^2$ を得る．イジング計算機を用いて最小値を与える $\mathbf{t}$ を計算することで，X_i 内の最小解 $\mathbf{A}_i \mathbf{t} \in X_i$ を得る．

具体例を紹介する．基底 $\mathbf{b}_1 = (5,6)^{\mathrm{T}}, \mathbf{b}_2 = (3,2)^{\mathrm{T}}$ に対する SVP の解は $\mathbf{x} = (-1,2)^{\mathrm{T}}$ である．たとえば範囲 X_2 に対して

$$\mathbf{A}_2 = \begin{pmatrix} 1 & 2 & -1 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 1 \end{pmatrix} \in \mathbb{Z}^{2 \times 7}$$

としたとき，ハミルトニアン $\mathcal{H}_{par} = \|(\mathbf{b}_1, \mathbf{b}_2)\mathbf{A}_2 \mathbf{t}\|^2$ の解は $\mathbf{t} = (0,0,1,0,1,0,1)^{\mathrm{T}} \in \{0,1\}^7$ である．

4.2 疑似マルチスピンフリップ

分割探索法のハミルトニアン $\mathcal{H}_i$ において変数 $x_{j,\ell}$

が 0 から 1 に更新されると，整数変数 x_j に $a_{j,\ell}$ が加算される．これは，更新前の目的関数の値が表す格子ベクトルが $\mathbf{v}$ であったとすると，更新後は $\mathbf{v}+a_{j,\ell}\mathbf{b}_j$ となることを意味する．このように SVP のハミルトニアンのイジング計算では，「現在の状態からある一つの基底ベクトルの整数倍を加算する」ことを繰り返して最短ベクトルを計算している．一方で疑似マルチスピンフリップとは，一つの変数のフリップ（バイナリ変数が 0 から 1 または 1 から 0 に更新されること）で複数個の変数がフリップした効果をもつ項をハミルトニアンに追加する技術である．これにより局所解を脱出しやすくなり，結果としてイジング計算が高速化されることが期待される．

疑似マルチスピンフリップの仕組みと効果を 4.1 節の具体例を用いて説明する．分割探索法のハミルトニアン $\mathcal{H}_{par}$ において，$\mathbf{t}=(0,0,0,0,0,0,1)^{\mathrm{T}}$ は局所解である．一方で，以下のように疑似マルチスピンフリップを用いてハミルトニアン $\mathcal{H}_{mul}$ を生成することで上記の $\mathbf{t}$ を局所解ではなくすることができる．具体的には

$$\mathbf{A}_2=\begin{pmatrix} 1 & 2 & -1 & -2 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 2 & 1 & 1 & 1 \end{pmatrix}\in\mathbb{Z}^{2\times 9}$$

とし，ハミルトニアン $\mathcal{H}_{mul}=\|(\mathbf{b}_1,\mathbf{b}_2)\mathbf{A}_2\mathbf{t}\|^2$ を生成する．このとき，上記 $\mathbf{A}_2$ の第 8 と 9 列が疑似マルチスピンフリップの項を表す．実際，バイナリ変数ベクトルが $\mathbf{t}=(0,0,0,0,0,0,1,0,0)^{\mathrm{T}}$ から $(0,0,0,0,0,0,1,0,1)^{\mathrm{T}}$ に変化（第 9 成分が 0 から 1 にフリップ）したとき，目的関数が表す格子ベクトルは $\mathbf{v}=\mathbf{b}_2$ から $\mathbf{v}=-\mathbf{b}_1+2\mathbf{b}_2$ に変化する．その変化量は $-\mathbf{b}_1+\mathbf{b}_2$ であり，一つの変数フリップで従来の 2 個分の変数フリップを実現できていることがわかる．さらに変化後の $\mathbf{v}$ は最短ベクトルであり，局所解が解消されていることもわかる．

上記具体例のように，疑似マルチスピンフリップのハミルトニアンは行列 $\mathbf{A}_i$ にマルチフリップの項を追加することで生成できる．ただし，追加した項の数だけ変数が増加する．たとえば，係数が 1 または -1 の疑似 2 スピンフリップの項（$\pm\mathbf{b}_j\pm\mathbf{b}_k$ の形）をすべて追加する場合，その総数 $N=2(n-1)^2$ だけ変数の数が増加する．これを考慮すると，Digital Annealer（第二世代）では疑似マルチスピンフリップの適用は $n=40$ 程度の SVP が限界である（SVP の場合は 4,096 変数以下のハミルトニアンしか扱えないため）．ただし，より大きなイジング計算機の開発により将来的にはより大きな次元の SVP にも適用が可能となる．

4.3 イジング計算機を用いた SVP 求解の状況

SVP の求解効率は，アルゴリズムの選び方（分割探索法または近似 Enum 法），疑似マルチスピンフリップの有無，整数変数のバイナリ変数への encoding 方法に依存することがわかっている [22, 25]．文献 [25] の Digital Annealer（第二世代）を用いた実験において，疑似マルチスピンフリップの実施ありとなしとでは，実施ありの方が平均して 25 倍高速に 40 次元 SVP を求解できることがわかっている．このことから，疑似マルチスピンフリップは実施した方が効率的であるといえる．ただし，アルゴリズムと encoding 方法の最適な選び方はまだ未解決であり，今後の課題となっている．また，分割探索法，疑似マルチスピンフリップなしの組み合わせでの 40 と 45 次元 SVP の求解時間はそれぞれ 664 秒，13,750 秒である [22]．一方で疑似マルチスピンフリップありの場合，40 次元を超える SVP は現在の計算機の規模の問題により解けない（入力できない）ため実験を行っていない．今後，より大規模な計算機が開発されたときに実施する予定である．

5. 共通鍵暗号 AES の安全性解析

AES は平文を 16 バイト（128 ビット）ごとのブロックに分解し，ブロックごとに暗号化する．まず，各ブロックの 16 バイトを 1 バイトずつ 4 行 4 列の配列の形に並べる．これを状態配列と呼ぶ．次に，状態配列に対する四つの演算 (SubBytes, ShiftRows, MixColumns, AddRoundKey) を 1 セットとし（ラウンドと呼ぶ），このラウンドを指定の回数繰り返して暗号化する．繰り返し回数は選択した鍵長によって異なる．

AES を含むブロック暗号に有力な解読法として差分解読法 [27] や線形解読法 [28] がある．これらは，AES の四つの中で唯一非線形な演算である SubBytes によって生じるデータの偏りを利用して暗号化に使用した鍵を特定する解析方法である．差分解読法では，ある二つの平文ブロック X_1, X_1' に対して，第 r 回目のラウンド開始前のブロックの値 X_r, X_r' の差分 $\Delta X_r = X_r \oplus X_r'$ に注目する．差分 ΔX_r を状態配列で表したとき，ある i 行 j 列の成分が 0（つまり差分なし）であれば第 r ラウンドの i 行 j 列の S-BOX は inactive であるといい，0 でなければ（つまり差分があれば）active であるという．第 1 から第 r ラウンドまでの active な S-BOX の総数を active S-BOX 数と呼ぶ．差分解読法の計算量を評価するうえで，各 r において active S-BOX 数の下界と下界を与える active S-BOX の組み合わせ（解

と呼ぶ）を求めることが重要になる [29, 30].

Mouha et al. [30] は下界とその解を混合整数線形計画法 (MILP) を用いて計算する手法を提案している．具体的には，第 r ラウンドの各 i 行 j 列の S-BOX が active であれば 1, inactive であれば 0 を表すバイナリ変数 $x_{r,i,j}$ を導入し，不等式制約の下で目的関数 $\mathcal{H}_r = \sum_{k=1}^{r} \sum_{i,j} x_{k,i,j}$ を最小化するバイナリ変数の組を計算する．しかし，汎用的な MILP ソルバでは，解が複数ある場合は複数回の定式化を必要とするなど，それらを効率的に求めることには不向きである．そこで，平野ら [31] は Mouha らによって与えられた不等式制約をハミルトニアンに定式化し，イジング計算を用いて下界と解を計算する方法を与えた．不等式制約のハミルトニアン定式化においては，Lucas のナップサック型ハミルトニアン生成法を応用するなど，変数の増加を抑える工夫を行っている．イジング計算であれば，その確率的性質により 1 回の定式化で複数個の解を見つけることが可能となる．

Digital Annealer（第二世代）を用いた実験では，鍵長が 256 ビットの AES の $r = 2$ までの active S-BOX 数の下界と解の計算に成功している．特に $r = 2$ においては，その解を 3 通り計算することに成功した．これらの結果を用いて実際に差分解読法の計算量を評価することは今後の課題である．

6. おわりに

本稿ではイジング計算を用いた暗号解析のいくつかの事例を紹介した．公開鍵暗号の場合，ほとんどのアプローチは汎用計算機用の解法アルゴリズムをイジング計算機に置き換えた場合が多く，正しい脅威判定のためには，イジング計算機の特性を利用した新しい解法のアルゴリズムが必要であろう．しかし共通鍵暗号の場合，未知の active S-BOX の組み合わせを見つけることに成功していることから，汎用計算機よりも解析の多様性を向上できている．

謝辞　本稿を作成するにあたり，筑波大学の國廣昇教授から貴重なコメントを頂戴しました．ありがとうございました．

参考文献

[1] 田村泰孝，神田浩一，片山健太朗，“デジタルアニーラのアーキテクチャと将来展望，” オペレーションズ・リサーチ：経営の科学，**67**，pp. 320–326，2022.
[2] 渡邊靖志，『入門講義　量子コンピュータ』，講談社，2021.
[3] P. W. Shor, “Algorithms for quantum computation: Discrete logarithms and factoring,” In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, 1994.
[4] M. Wroński, “Practical solving of discrete logarithm problem over prime fields using quantum annealing,” *IACR Cryptology ePrint Archive*, Report 2021/527, 2021.
[5] A. K. Lenstra and H. W. Lenstra Jr, *The Development of the Number Field Sieve*, Springer, 1993.
[6] F. Boudot, P. Gaudry, A. Guillevic, N. Heninger, E. Thomé and P. Zimmermann, “Factorization of RSA-250,” https://sympa.inria.fr/sympa/arc/cado-nfs/2020-02/msg00001.html（2022 年 4 月 26 日閲覧）
[7] CRYPTREC, CRYPTREC Report 2018 暗号技術評価委員会報告，https://www.cryptrec.go.jp/report/cryptrec-rp-2000-2018.pdf （2022 年 2 月 4 日閲覧）
[8] E. Martín-López, A. Laing, T. Lawson, R. Alvarez, X. Q. Zhou and J. L. O’brien, “Experimental realization of Shor’s quantum factoring algorithm using qubit recycling,” *Nature Photonics*, **6**, pp. 773–776, 2012.
[9] M. Amico, Z. H. Saleem and M. Kumph, “Experimental study of Shor’s factoring algorithm using the IBM Q experience,” *Physical Review* A, **100**, 012305, 2019.
[10] 清水俊也，伊豆哲也，篠原直行，盛合志帆，國廣昇，“アニーリング計算による素因数分解について，” *SCIS 2019*, 2B4-3, 2019.
[11] X. Peng, Z. Liao, N. Xu, G. Qin, X. Zhou, D. Sutter and J. Du, “Quantum adiabatic algorithm for factorization and its experimental implementation,” *Physical Review Letters*, PRL 101, 220405, 2008.
[12] N. Xu, J. Zhu, D. Lu, X. Zhou, X. Peng and J. Du, “Quantum factorization of 143 on a dipolar-coupling nuclear magnetic resonance system,” *arXiv:quant-ph*, 1111.3726v1, 2011.
[13] S. Pal, S. Morit, V. S. Anjusha, A. Kumar and T. S. Mahesh, “Hybrid scheme for factorization: Factoring 551 using a 3-qubit NMR quantum adiabatic processor,” *arXiv:quant-ph*, 1611.00998v2, 2016.
[14] R. Drid and H. Alghassi, “Prime factorization using quantum annealing and computational algebraic geometry,” *arXiv*, 1604.057962v2, 2016.
[15] Z. Li, N. S. Dattani, X. Chen, X. Liu, H. Wang, R. Tanburn, H. Chen, X. Peng and J. Du, “High-fidelity adiabatic quantum computation using the intrinsic hamiltonian of a spin system: Application to the experimental factorization of 291311,” *arXiv quant-ph*, 1706.08061v1, 2017.
[16] S. Jiang, K. A. Britt, A. J. McCaskey, T. S. Humble and S. Kais, ”Quantum annealing for prime factorization,” *arXiv quant-ph*, 1804.02733v2, 2018.
[17] W. C. Peng, B. N. Wang, F. Hu, Y. J. Wang, X. Fang, X. Y. Chen and C. Wang, “Factoring larger integers with fewer qubits via quantum annealing with optimized parameters,” *Science China Physics, Mechanics & Astronomy*, **62**, 060311, 2019.
[18] 伊豆哲也，清水俊也，篠原直行，盛合志帆，國廣昇，“アニーリング計算による素因数分解について（その 2），” *SCIS 2020*, 4B2-1, 2020.
[19] Fukuoka MQ Challenge, https://www.mqchallenge.org/ （2022 年 2 月 4 日閲覧）
[20] 下山武司，大堀龍一，清水俊也，山口純平，“アニーリングを用いた多変数多項式暗号解析，” *SCIS 2019*, 2B4-5,

2019.
[21] T. U. Darmstadt SVP Challenge, http://www.latticechallenge.org/svp-challenge/ （2022 年 2 月 4 日閲覧）
[22] J. Yamaguchi, T. Shimizu, K. Furukawa, R. Ohori, T. Shimoyama, A. Mandal, H. Montgomery, A. Roy and T. Ohwa, “Annealing-based algorithm for solving CVP and SVP,” *Journal of the Operations Research Society of Japan (JORSJ)*, 2022 (in press).
[23] 山口純平，Avradip Mandal，Hart Montgomery，Arnab Roy，清水俊也，大堀龍一，下山武司，“アニーリングを用いた格子問題の求解，”*SCIS 2019*, 2B4-4, 2019.
[24] 山口純平，清水俊也，古川和快，“格子 enumeration に基づく SVP のハミルトニアンの構成と解読実験，”*SCIS 2020*, 4B1-5, 2020.
[25] 山口純平，大輪拓也，古川和快，“アニーリング計算機を用いた最短ベクトル問題の求解―疑似マルチスピンフリップを用いたハミルトニアン生成―”，電子情報通信学会技術研究報告，**120**, *ISEC2020*–51, pp. 58–65, 2021.
[26] G. Rosenberg, P. Haghnegahdar, P. Goddard, P. Carr, K. Wu and M. L. De Prado, “Solving the optimal trading trajectory problem using a quantum annealer,” *IEEE Journal of Selected Topics in Signal Processing*, **10**, pp. 1053–1060, 2016.
[27] E. Biham and A. Shamir, “Differential cryptanalysis of DES-like cryptosystems,” *Journal of CRYPTOLOGY*, **4**, pp. 3–72, 1991.
[28] M. Matsui, “Linear cryptanalysis method for DES cipher,” In *Proceedings of Workshop on the Theory and Application of of Cryptographic Techniques*, pp. 386–397, 1993.
[29] J. Saemon and V. Rijmen, “The wide trail design strategy,” In *Proceedings of IMA International Conference on Cryptography and Coding*, pp. 222–238, 2001.
[30] N. Mouha, Q. Wang, D. Gu and B. Preneel, “Differential and linear cryptanalysis using mixed-integer linear programming,” In *Proceedings of International Conference on Information Security and Cryptology*, pp. 57–76, 2011.
[31] 平野遥，垣本修吾，米山一樹，山口純平，“アニーリング計算を用いた AES の差分特性探索に向けて，”電子情報通信学会技術研究報告，**119**, *ISEC2019*–105, pp. 127–133, 2020.

