平成18年度 修士論文

LDPC符号を用いた BB84量子鍵配送プロトコル

BB84 Quantum Key Distribution Protocol Using LDPC Codes

東京大学大学院 情報理工学系研究科 電子情報学専攻

56407　大畑 真生

指導教官　松浦 幹太 助教授

平成19年 2月2日 提出

論文要旨

情報セキュリティの確保は重要な課題であるが，現在使用されている暗号に対する安全性は，古典的な計算量に基づいた安全性でしかない．例外として One-Time Pad 暗号が存在するが，One-Time Pad 暗号の鍵は無造作に作成された上で再利用不可能であり，暗号化する平文と同じだけ鍵長が必要となる．しかし，これは非実用的であり，システムを現実的に運用するためには，鍵配送の問題を解決する必要がある．この鍵配送問題の 1 つの解決策が，Bennett と Brassard によって提案された BB84 量子鍵配送プロトコル [8] である．BB84 量子鍵配送プロトコルは，量子力学の基本原理である不確定性原理に基づいた鍵共有プロトコルであり，量子力学の許す範囲での如何なる攻撃に対しても無条件安全性 [46] が保障されている．

雑音のある量子通信路において，この BB84 量子鍵配送プロトコルには，$C_1 \supset C_2$ を満たす 2 つの古典線形符号 $C_1,\ C_2$ を用いた誤り訂正及び，秘匿性増強が有用である．この符号は CSS 符号 [9] と呼ばれる代表的な量子誤り訂正符号である．$C_1,\ C_2^{\perp}$ の復号性能は盗聴者の相互情報量と関係し，実際に誤り訂正を行う C_1 は実用的な復号方法で良い復号特性を持つことが望ましい．$C_2^{\perp}$ は理論的に良い復号特性を持てば良い．しかし，このような $C_1,\ C_2$ を実際に構成するのは困難であり，BB84 プロトコルにおいては，古典処理における一番の問題点となっている．

次世代誤り訂正符号としても期待されている LDPC 符号 [19] は，実用的な復号法である sum-product 復号法にて良い復号特性を持っており，CSS 符号を構成するにあたり有用である．したがって，本論文では，良い復号特性を持つ LDPC 符号を C_1 に適用し，2 つの線形符号 $C_1,\ C_2$ を実際に構成する．

まず第 1 章で量子暗号の現状と，準備として量子情報理論の概要，関連する符号について説明する．次に第 2 章で双対符号を含む LDPC 符号構成法の提案とその手法について解説を加える [1]．第 3 章で任意の LDPC 符号を用いて構成可能な CSS 符号構成法を提案し，まず生成される符号語の解析 [2]，そして実際に様々な LDPC 符号を用いて CSS 符号構成した場合における復号性能評価実験を行った [3, 4, 5]．第 3 章で提案する構成手法は，既存研究や第 2 章の提案する符号構成法とは異なり，任意 LDPC 符号が使用可能であり，実装面に関しても柔軟に 2 つの線形符号を構成可能な手法となっている．

目次

図目次

表目次

第1章

量子暗号

1.1 情報理論的安全性

インターネットの急速な普及に伴って，新しい情報のインフラストラクチャが構築される中，情報セキュリティの確保は重要な課題である．なかでも暗号技術は，電子化された情報の秘匿性及び非改竄性の確保や，電子認証を実現する基盤技術であり，電子政府の構築においてもそのセキュリティ確保のために必要不可欠な技術とされている．

暗号技術の主な目的の一つは，ある二者（以下送信者をアリス，受信者をボブと呼ぶことにする）が安全でない通信路上で，ある攻撃者（同じく以下，イブと呼ぶことにする）に対して通信内容を解読されないようにする技術である．アリスはあらかじめ定められた規則と自分の所持している暗号化鍵によって平文（アリスが伝えたいメッセージ）を適当な形に変換し，その結果得られた暗号文（暗号化されたメッセージ）を通信路を通してボブに送信する．ここで，[38] において与えられている暗号系の定義を以下に示す．

定義 1.1. 暗号系は次のような状態を満たす5つの要素 $(P,\ C,\ K,\ E,\ D)$ から成る．

- P は平文としてとりえる有限集合．
- C は暗号文としてとりえる有限集合．
- K は鍵としてとりえる有限集合で，鍵空間と呼ばれる．
- それぞれの $K \in K$ に対して暗号化ルール $eK \in E$ とそれに対応する復号ルール $dK \in D$ が存在する．それぞれの $eK : P \to C$ と $dK : C \to P$ は全ての平文 $x \in P$ に対して $dK(eK(x)) = x$ となるような関数である．

暗号技術の安全性を評価することは大変重要であるが，現在用いられている暗号に対する安全性は，古典的な計算量的に基づいた安全性でしかない．例外として One-Time Pad 暗号がある．One-Time Pad 暗号は，情報理論的安全性が証明されている [44]．One-Time Pad 暗号で暗号文を作成するためには，平文に2進数の乱数列である鍵を加える．One-Time Pad 暗号の安全性を保証する重要な点は，鍵は無造作に作成された上で一度しか用いられてはな

らず，暗号化する平文と少なくとも同じだけ鍵長が必要である．しかし，これは非実用的であり，システムを現実的に運用するためには，鍵配送の問題を解決しなければならない．

公開鍵暗号は，鍵配送の問題に対する 1 つの解決策としてに提案された [12]．公開鍵暗号は，暗号化に用いられる公開鍵と，復号に用いられる秘密鍵が異なっている．このため，暗号化に用いられる鍵を公開でき，鍵配送が容易に行える．しかし，この公開鍵暗号の安全性は，古典的な計算問題の困難性に基づいているため，One-Time Pad 暗号の鍵配送には用いるには不適である．例えば，RSA 暗号 [41] においては，その安全性は大きな数の素因数分解の困難性に基づいているが，この安全性は十分とは言えない．なぜならば，素因数分解を古典的な計算機を用いて多項式時間で解くアルゴリズムは存在しないと広く信じられているが，その非存在性はまだ証明されてはいないためである．また，古典的な情報は容易に複製，保存が可能であり，オフラインで解読が可能となることも危険を伴う．さらに，計算機及び計算アルゴリズムの急速な進歩も軽視できない．1994 年には素因数分解アルゴリズムの改良，計算機の能力の進歩によって，同じ問題が短期間で解くことが可能となった．これは安全性の長期間保証が困難になってきていることを意味する．また，量子コンピュータの実現の可能性も考慮する必要もある．

量子コンピュータは，多数のコヒーレント状態の重ね合わせを入力として扱うことができ，これら重ね合わせの全ての状態に対して同時に演算（ユニタリ作用素）を作用させることができるため，非常に高速計算が可能である．これは大規模の並列計算機とみなせる．ただし量子コンピュータの場合，多くのプロセッサを並列に作業させる代わりに一つの量子プロセッサに状態ベクトルの全ての成分に作用する計算する．量子コンピュータを実現するためには，計算を行っている間について，系の時間発展をコントロールすることのできるコヒーレンスが必要である．これは現在のところ非常に困難な実用上の問題であると考えられているが，遅かれ早かれ技術の発展によってこのような装置の製作が可能になると信じられている．つまり，もし量子コンピュータが開発されれば，適当なアルゴリズム [45] を用いて RSA 暗号の基となっている素因数分解問題や，離散対数問題も多項式時間で解読可能となる．

長期的な安全性を保障する場合，保障期間内の計算機や計算アルゴリズムの進歩が明確に予想できないため，高度な暗号技術が必要となる．したがって，情報理論的安全性が保障された暗号技術が望ましい．情報理論的な安全性を保証するには，上述した One-Time Pad 暗号が有用でが，One-Time Pad 暗号は何度も言及したように，暗号化に用いる鍵の配送が問題点となる．この One-Time Pad 暗号における鍵配送問題の 1 つの解決策が，Bennett と Brassard によって提案された BB84 量子鍵配送プロトコル [8] である．量子コンピュータの実現は現在困難な状況の一方で，BB84 量子鍵配送プロトコルは欧米において製品化も進められている段階にある．この BB84 量子鍵配送プロトコルは，量子力学の基本原理である不確定性原理に基づいた鍵共有プロトコルであり，量子力学の許す範囲での如何なる攻撃に対

しても無条件安全性 [34, 46] が保障されている．この One-Time Pad 暗号と量子鍵配送 (ここでは BB84 量子鍵配送プロトコル) の組み合わせを一般に量子暗号と呼ぶ．量子暗号では既に通信距離 100km 超でも実装報告が多くの研究機関で報告されており，欧米では製品化もされている状況である．

本章では，量子暗号と量子暗号に関連する量子情報理論の概要，量子鍵配送プロトコルや誤り訂正，秘匿性増強に用いる符号について説明する．

1.2 One-Time Pad 暗号

本節では One-Time Pad 暗号について説明する．One-Time Pad 暗号は情報理論的安全性が証明されている共通鍵暗号方式である．つまり，暗号化の際に用いる鍵と，復号の際に用いる鍵が同じ鍵で行われる．

One-Time Pad 暗号は，2 元 n ビットの平文 $m \in \{0,1\}^n$ に対し，同じ 2 元 n ビットの鍵 $k \in \{0,1\}^n$ を準備し，暗号文 c を $c = m + k$　により生成する．復号は $c + k = m$ によって行われる．(演算子 $+$ は排他的論理和とする.)

1.2.1 One-Time Pad 暗号の安全性

One-Time Pad 暗号は情報理論的に安全であると述べた．本節ではその意味についての詳細な解説を行う．

一般的に，安全性とは，攻撃者イヴに与えられた環境,（すわなち，攻撃モデル）及びイヴの攻撃目標（すわなち，達成度）によって定義され，また，その安全性を実現されるための仮定により分類がなされたものである．情報理論的安全性は，そのような仮定の部分を示したものであり，攻撃モデルや達成度についての意味は何も含んでいないことに注意されたい．厳密には，One-Time Pad 暗号は，送信されている暗号文 c を観測することだけが可能な攻撃者（という攻撃モデル）に対して，c を観測せずに得られる以上の平文 m の情報を得る（という達成度）を一切許さないという安全性を，情報理論的に保証するものとなっている．

情報理論的安全性にかかわる議論においては，このような安全性の定義を情報エントロピーで表現する．One-Time Pad 暗号の安全性は，$m,\ k,\ c$ の確率変数をそれぞれ $M,\ K,\ C$ としたとき，情報エントロピーを用いて，

$$H(M|C) = H(M),\ H(M|K,\ C) = 0$$

（ここで，$H(X)$ は確率変数 X の情報エントロピーを表すものとする）と定義することが

できる．この定義から，

$$\begin{aligned} H(K) &\geq H(K|C) - H(K|M,\ C) \\ &= H(M|C) - H(M|K,\ C) \\ &= H(M|C) \\ &= H(M) \end{aligned}$$

となり，One-Time Pad 暗号においては，鍵長が平文長以上でなければならないことが導き出される．さらには鍵の再利用が不可能であり，One-Time Pad 暗号は，この秘密鍵 k を如何にして安全に配送するかが重要となる．

1.3 量子暗号の基本原理

量子暗号は暗号化の方法の一種であり，その安全性の根拠となっているのは量子力学である．現在提案されている量子暗号の装置は正確には，無造作なビット列を安全に共有するためのものであり，アリスが任意のビット列をボブに送ることは出来ないが，無造作なビット列 k を共有可能になるというものである．ただし，この装置を用いてビット列 k を共有した後に，k を鍵として古典的な暗号スキームを用いれば，暗号通信を行うことが出来るので量子暗号と呼ばれている．しかし正確には量子暗号は「鍵共有装置」である．

量子暗号という安全な通信システムは，Heisenberg の不確定性原理と量子エンタングルメントの利用によって可能になる．量子暗号では光などの量子に，アリスが鍵の情報を乗せてボブに送る．量子力学によって，単独の粒子の情報は観測によって一度乱されると元に戻すことができないことが保証されている．したがって送信途中でイヴが鍵の情報を盗聴するために，この量子の状態を観測しても，イブの盗聴行為が通信路の擾乱として明瞭になる．量子暗号はこの原理を用いて，本来の正規ユーザーであるアリス及びボブが，盗聴の有無を検知できるというものである．これは現在使われている暗号システムには不可能な技術であり，量子暗号特有の技術である．古典的な暗号方式と比べると，正規のユーザは盗聴を排除する努力をする必要はなく，盗聴を検出できるようにプロトコルを実行すればよい．付け加えると，攻撃者イヴに暗号文の内容を知られないために古典的な方式では数学的な手法を用いて計算量的安全性を仮定していたが，量子暗号では物理法則によって情報理論的安全性が保証される．

最近は雑誌や新聞などで「絶対に破れない」暗号として量子暗号が紹介されることが多い．どういう意味で「絶対」なのかというと，もし量子暗号が破れたとすると，そのことは量子論が何らかの意味で破綻していることを意味し，20 世紀以降の物理学をほぼ全域に渡る修正が生じる．量子力学は発見されてから 1 世紀近くにわたってあらゆる自然現象を解明

しつづけ，しかも現在もほころびらしいほころびは見つかっていない．また，電子，陽子，中性子を含むこの世の全ての物質は量子であり，また半導体，テレビのブラウン管，蛍光灯の動作原理から，空が青い理由や鉄が磁石になる理由にいたるまで，我々の身の周りの物理現象の殆ど全ては，量子論によって仕組みが解明されているものばかりである．そういう意味で物理学者は量子論は絶対であると信じ，量子暗号も破れないと考えられている．一方で，もし量子論にほころびが見つかれば，それは万有引力や相対論の発見に勝るとも劣らない大発見となり，そこから新しい時代の科学が始まると考えられる．したがって量子暗号はどちらにしても重要な結果を残す研究分野になる．

1.4 量子暗号の実用化に向けての動向

近年，実用化の面でも，実証実験やシステム開発が国内外で活発化してきている．量子鍵配送の技術においては，鍵の伝送距離と同時に鍵の伝送レートがシステム性能上の重要な指標である．BB84 プロトコルによる原理実験が行われて以来，長距離化・高伝送レート化に向けた量子鍵配送の研究開発が行われてい．米国ロスアラモス国立研究所で時間領域干渉計の方法により 48km の量子暗号鍵配布実験が行われ，ジュネーブ大学で InGaAs/InP 系 APD による 1550nm 帯単一光子検出器を用いて 67km, 100bps の鍵配送がなされた．新たなプロトコルやシステム方式が欧米で発案されたこともあり，こうした動きは欧米が先行してきた．しかし，近年になって日本においても伝送速度 1Mbps，伝送距離 100km という目標に向けた取り組みが行われ，三菱電機/TAO のグループが，1550nm 帯で 87km での量子暗号通信システム実験に成功した．また，既存のセキュリティと融合した統合量子暗号として実用に向けた開発も進められている．また，東芝欧州研究所では 2003 年 6 月に 101km の量子暗号鍵配布に成功した．NEC では，通信・放送機構（TAO）と科学技術振興事業団（JST）との共同で，光通信で普通に使われている商用ファイバを用いて 100km の量子暗号鍵配布に成功している．これは低ロスファイバでの 125km に相当する．

量子暗号鍵配布システムにおける最大伝送可能距離は，現時点において検出器雑音により制限されていることから，検出器の高性能化が重要な課題となってきた．最近は通信波長帯である 1550nm 帯において単一光子検出器の研究開発が精力的に行われ，誤り検出確率の低減，検出効率の向上等の成果が報告されている．また，従来のゲート動作に対し，Active-quench 方式の検出器が報告されている．最近では，欧米ベンチャーにより商用化も始まっている．

1.5 量子ビット

量子情報理論では，量子ビットの概念を扱うため，その説明を行う．

量子ビットとは 2 準位の任意の量子系のことを表す．2 状態の基底を $|0\rangle,\ |1\rangle$ で定義する．これらを用いて古典的な 1 ビットの情報の送信や保存が可能である．量子系ではさらに一般的にその重ね合わせ

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,\ \alpha,\ \beta \in \mathbb{C},\ |\alpha|^2 + |\beta|^2 = 1$$

が存在する．換言すれば，一般に量子ビットの状態は，2 次元複素ベクトル空間の単位ベクトルである．

次に観測について説明する．量子状態と古典的状態の最大の違いは観測を行ったときに現れる．量子論の仮定から，観測は一般にエルミート行列（エルミート演算子）で表され，観測量は常にその固有値になる．また，系が状態 $|\psi\rangle$ にあるとき，固有状態 $|\phi\rangle$ が観測される確率 P は

$$P = |\langle\phi|\psi\rangle|^2$$

で与えられる．

以下キュービット=2 準位系のみを考える．エルミート演算子の例として,

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

を考えると， A の固有状態は,

$$|\psi_\pm\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ \pm 1 \end{bmatrix}$$

であり，その固有値は ± 1 である．従って演算子 A に相当する観測で得られる観測量は ± 1 となる．固有状態でない一般的な $|\psi\rangle$ を観測した場合にも ± 1 のいずれかが必ず得られ，その確率 $P_\pm$ は状態の内積の二乗として得られる．

$$P_\pm = |\langle\psi|\psi_\pm\rangle|^2 = \frac{1}{2} \pm \Re(\alpha\beta)$$

また，観測後の状態は観測量に対応した状態 $|\psi_\pm\rangle$ に遷移する．

1.6 量子情報の距離尺度

古典情報の距離速度においては，ハミング距離によって定量的に測定可能である．量子情報では，トレース距離と忠実度が広く用いられているが，ここでは忠実度に関して説明する．

確率分布 $p_x,\ q_x$ における忠実度は次式で定義される．

$$F(p_x,\ q_x)\ :=\ \sum_x \sqrt{p_x q_x}$$

忠実度はトレース距離とは非常に異なる確率分布間の距離を測る方法である．また，状態 ρ と σ の忠実度は，

$$F(\rho,\ \sigma)\ :=\ \mathrm{tr}\sqrt{\rho^{1/2}\sigma\rho^{1/2}}$$

で定義される．これより，純粋状態 $|\psi\rangle$ と任意状態 σ の忠実度は，

$$\begin{aligned} F(|\psi\rangle,\ \sigma) &= \mathrm{tr}\sqrt{\langle\psi|\sigma|\psi\rangle|\psi\rangle\langle\psi|} \\ &= \sqrt{\langle\psi|\sigma|\psi\rangle} \end{aligned}$$

となる．

1.7 CSS 符号

本節では，量子誤り訂正符号である CSS 符号について説明する．CSS 符号は，t 個までの誤りを訂正することができる 2 つの古典線形符号から構成され，その古典誤り訂正能力を利用して t 個までのビット反転誤り及び t 個までの位相反転誤りを同時に訂正する量子誤り訂正符号である．以下，CSS 符号の構成法について詳しく説明する．

いま，$[n,\ k_1]$ 線形符号 C_1 及び $[n,\ k_2]$ 線形符号 C_2 が，次の 2 つの条件を満たしているとする．

(i) $C_1 \supset C_2$ を満たす．
(ii) $C_1,\ C_2^{\perp}$ はともに t 個まで誤りを訂正が可能．

このとき， C_1 及び C_2 を用いて，$[n,\ k_1 - k_2]$ 量子誤り訂正符号を以下のように構成できる．$\boldsymbol{x}$ を C_1 の任意の符号語とする．このとき量子状態 $|\boldsymbol{x} + C_2\rangle$ を

$$|\boldsymbol{x} + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{\boldsymbol{y}\in C_2} |\boldsymbol{x} + \boldsymbol{y}\rangle$$

により定義する．（演算子 $+$ は排他的論理和としている．）また，量子状態 $|\boldsymbol{x} + C_2\rangle$ は，次の性質を満たすことに注意しておく．

(a) $\boldsymbol{x} + \boldsymbol{x}' \in C_2$ ならば $|\boldsymbol{x} + C_2\rangle = |\boldsymbol{x}' + C_2\rangle$
(b) $\boldsymbol{x} + \boldsymbol{x}' \notin C_2$ ならば $\langle \boldsymbol{x} + C_2 | \boldsymbol{x}' + C_2\rangle = 0$

CSS 符号は任意の $\boldsymbol{x} \in C_1$ に対して量子状態 $|\boldsymbol{x} + C_2\rangle$ により張られる空間として定義される．上述の 2 つの性質より，CSS 符号の次元は， $|C_1|/|C_2| = 2^{k_1 - k_2}$ であり，CSS 符号が $[n,\ k_1 - k_2]$ 量子誤り訂正符号であることがわかる．

次に，実際に t 個までのビット反転誤り及び t 個までの位相反転誤りを同時に訂正可能であることを示す．そこで，$\boldsymbol{x} \in C_1$ に対応する量子状態

$$|\boldsymbol{x} + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{\boldsymbol{y} \in C_2} |\boldsymbol{x} + \boldsymbol{y}\rangle$$

が，ビット反転誤り $\boldsymbol{e}_1$ 及び $\boldsymbol{e}_2$ により

$$\frac{1}{\sqrt{|C_2|}} \sum_{\boldsymbol{y} \in C_2} (-1)^{(\boldsymbol{x}+\boldsymbol{y})\cdot\boldsymbol{e}_2} |\boldsymbol{x} + \boldsymbol{y} + \boldsymbol{e}_1\rangle$$

と変化した状況を考えることにする．まず，ビット反転誤りを検出するために，線形符号 C_1 のシンドロームを格納するのに十分なビット数を有する補助量子ビットを用意し，はじめ全てのビットを 0 にセットしておく．この合成系 $|\boldsymbol{x} + \boldsymbol{y} + \boldsymbol{e}_1\rangle|0\rangle$ に対して，C_1 のパリティ検査行列 H_1 を可逆に作用させると，

$$|\boldsymbol{x} + \boldsymbol{y} + \boldsymbol{e}_1\rangle|H_1(\boldsymbol{x} + \boldsymbol{y} + \boldsymbol{e}_1)\rangle = |\boldsymbol{x} + \boldsymbol{y} + \boldsymbol{e}_1\rangle|H_1\boldsymbol{e}_1\rangle$$

を得る．この作用により，上の量子状態は，

$$\frac{1}{\sqrt{|C_2|}} \sum_{\boldsymbol{y} \in C_2} (-1)^{(\boldsymbol{x}+\boldsymbol{y})\cdot\boldsymbol{e}_2} |\boldsymbol{x} + \boldsymbol{y} + \boldsymbol{e}_1\rangle|H_1\boldsymbol{e}_1\rangle$$

となる．ここで，$|H_1\boldsymbol{e}_1\rangle$ を観測し，その後それを廃棄すれば，もとの状態

$$\frac{1}{\sqrt{|C_2|}} \sum_{\boldsymbol{y} \in C_2} (-1)^{(\boldsymbol{x}+\boldsymbol{y})\cdot\boldsymbol{e}_2} |\boldsymbol{x} + \boldsymbol{y} + \boldsymbol{e}_1\rangle$$

を得る．誤り $\boldsymbol{e}_1$ が t 個以内であれば，シンドローム $H_1\boldsymbol{e}_1$ から $\boldsymbol{e}_1$ が推定できるので，結局ビット反転誤りを訂正した状態

$$\frac{1}{\sqrt{|C_2|}} \sum_{\boldsymbol{y} \in C_2} (-1)^{(\boldsymbol{x}+\boldsymbol{y})\cdot\boldsymbol{e}_2} |\boldsymbol{x} + \boldsymbol{y}\rangle$$

を得ることができる．次に位相反転誤りを検出するために，各量子ビットにアダマール変換

$$\boldsymbol{x} \to \frac{1}{\sqrt{2^n}} \sum_{\boldsymbol{z}} (-1)^{\boldsymbol{x}\cdot\boldsymbol{z}} |\boldsymbol{z}\rangle$$

を作用させてやると，量子状態は，

$$\frac{1}{\sqrt{|C_2|2^n}} \sum_{\boldsymbol{z}} \sum_{\boldsymbol{y} \in C_2} (-1)^{(\boldsymbol{x}+\boldsymbol{y})\cdot(\boldsymbol{e}_2+\boldsymbol{z})} |\boldsymbol{z}\rangle$$

となる．ここで，$\boldsymbol{z}' = \boldsymbol{z} + \boldsymbol{e}_2$ を導入すると，

$$\frac{1}{\sqrt{|C_2|2^n}} \sum_{\boldsymbol{z}'} \sum_{\boldsymbol{y} \in C_2} (-1)^{(\boldsymbol{x}+\boldsymbol{y})\cdot\boldsymbol{z}'} |\boldsymbol{z}' + \boldsymbol{e}_2\rangle$$

となる．

(i) $\boldsymbol{z}' \in C_2^{\perp}$ において $\sum_{\boldsymbol{y}\in C_2}(-1)^{\boldsymbol{y}\cdot\boldsymbol{z}'} = |C_2|$
(ii) $\boldsymbol{z}' \notin C_2^{\perp}$ において $\sum_{\boldsymbol{y}\in C_2}(-1)^{\boldsymbol{y}\cdot\boldsymbol{z}'} = 0$

より，

$$\frac{1}{\sqrt{2^n/|C_2|}}\sum_{\boldsymbol{z}'\in C_2^{\perp}}(-1)^{\boldsymbol{x}\cdot\boldsymbol{z}'}|\boldsymbol{z}'+\boldsymbol{e}_2\rangle$$

と変形できる．これは誤り $\boldsymbol{e}_2$ によるビット判定誤りと同じ形としている．したがって，ビット反転誤りのときと同様に，線形符号 C_2 のパリティ検査行列 H_2 を作用させることによって，シンドローム $H_2\boldsymbol{e}_2$ を得る．そして，誤り $\boldsymbol{e}_2$ が t 個以内であれば，シンドローム $H_2\boldsymbol{e}_2$ から $\boldsymbol{e}_2$ が推定できるので，位相反転誤りを訂正した状態

$$\frac{1}{\sqrt{2^n/|C_2|}}\sum_{\boldsymbol{z}'\in C_2^{\perp}}(-1)^{\boldsymbol{x}\cdot\boldsymbol{z}'}|\boldsymbol{z}'\rangle$$

を得ることができる．この量子状態の各量子ビットに，再度アダマール変換を作用させてやれば，元の状態

$$\frac{1}{\sqrt{|C_2|}}\sum_{\boldsymbol{y}\in C_2}|\boldsymbol{x}+\boldsymbol{y}\rangle$$

が復元される．

次節で説明する BB84 量子鍵配送プロトコルは，この CSS 符号を用いた誤り訂正，秘匿性増強により，雑音下においても無条件安全性を保証している．また，古典的な付加処理で実現可能である点も大きい．

1.8 BB84 量子鍵配送プロトコル

BB84 量子鍵配送プロトコルでは，アリスは古典ビット $a \in \{0,\ 1\}$ を 2 準位系（量子ビット）に乗せて通信するが，その際基底を 2 通りに変化させる．まず任意の基底 $|0\rangle, |1\rangle$ をとり，これを基底 0 でのビット $a = 0,\ 1$ とし，基底 1 でのビットを $|+\rangle, |-\rangle$ で定義する．ただし，$|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$, $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$ である．基底を $b = 0, 1$ と表現すると，これらの状態は，

$$|\psi\rangle = \cos\vartheta|0\rangle + \sin\vartheta|1\rangle,\ \vartheta = \frac{\pi}{2}b + \frac{\pi}{4}a$$

と表現できる．ボブはこの $|\psi\rangle$ を基底 0,1 のどちらかに選んで観測し，観測結果を a' とする．このとき観測結果において

(a) 同じ基底を選んだ場合には結果は当然 $a = a'$ となり，ビットが共有できる．
(b) 一方で，$i, j \in \{0, 1\}$ に対して $|\langle i|j'\rangle|^2 = \frac{1}{2}$ が成り立つので，ボブがアリスと異なる基底で観測した場合には，a' は a とは依存しないランダムビットとなる．状態 $|\psi\rangle$ もそれに従って無造作に遷移する．

(c) イヴが $|\psi\rangle$ が盗聴した場合も同様なことが言える．イヴがアリスと異なる基底を選択した場合には，状態 $|\psi\rangle$ を破壊してしまう．これが通信路の擾乱として明瞭になるため，盗聴の検知が可能となる．

次に BB84 量子鍵配送プロトコルの手順について説明する．ただし，この BB84 量子鍵配送プロトコルを実行する際に用いられる装置は理想的な装置でなければならない．よってアリスとボブは理想的な装置を用いてプロトコルを実行すると仮定する．

(1) アリスは $(4+\delta)n$ 個のランダムビットを作成する．
(2) ランダムビット記号列 b に従って，ランダムビット記号列 b が 0 のときは状態を $|0\rangle, |1\rangle$ の基底から，ランダムビット記号列 b が 1 のときは，状態を $|+\rangle, |-\rangle$ 基底から各ビット毎に選択して，量子ビットを作成する．
(3) アリスは作成した量子ビットを量子通信路にてボブに送信する．
(4) ボブは量子ビットを受信し，そのことを古典通信路にてアリスに伝える．そして各量子ビットを $|0\rangle, |1\rangle$，または $|+\rangle, |-\rangle$ の基底をランダムに選択して測定する．
(5) アリスは b を古典通信路にて送信する．
(6) アリスとボブは，ボブが b でない基底で測定したビットを廃棄する．高い確率で少なくとも $2n$ ビットが残る．そうでなければプロトコルを中止する．アリスはランダムに $2n$ ビットから検査ビットとして用いる n ビットを選択し，それを古典通信路にて送信する．
(7) アリスとボブは検査ビットを古典通信路にて比較する．もし一定のビット数以上不一致ならばプロトコルを中止する．アリスには n ビット記号列 $\boldsymbol{x}$ が，ボブには雑音や盗聴の影響で n ビット記号列 $\boldsymbol{x}+\boldsymbol{e}$ が残る．
(8) アリスはランダムな $\boldsymbol{u} \in C_1$ を選択する．
(9) アリスは $\boldsymbol{x}+\boldsymbol{u}$ を古典通信路にて送信する．ボブは $\boldsymbol{u}+\boldsymbol{e}$ を得て，符号 C_1 により誤り訂正を行い $\hat{\boldsymbol{u}}$ を得る．
(10) アリスは剰余類 $\boldsymbol{u}+C_2$，ボブは剰余類 $\hat{\boldsymbol{u}}+C_2$ を鍵として得る．

以上の操作により，アリスとボブは鍵を共有する．この一連の手順において，C_1 により誤り訂正を行い，C_2 により秘匿性増強を行っている．この秘匿性増強により，イヴの情報量を抑えることが可能となる．また，誤り訂正を行ったあとの状態において，$\boldsymbol{u} \neq \hat{\boldsymbol{u}}$ の場合でも，$\hat{\boldsymbol{u}}-\boldsymbol{u} \in C_2$ となれば良く，実際には $C_1, C_2^{\perp}$ の復号性能に関わらず $C_1/C_2, C_2^{\perp}/C_1^{\perp}$ が復号性能が良ければかまわない．この剰余類は，例えば C_1/C_2 であれば，C_1 の符号語を C_2 におけるシンドロームを計算し，同じシンドロームを持つ符号語を持つ符号語を C_1/C_2 において同じ剰余類とする．また，C_1 と C_2 の差分のパリティ検査行列を用いて鍵は生成可能である．

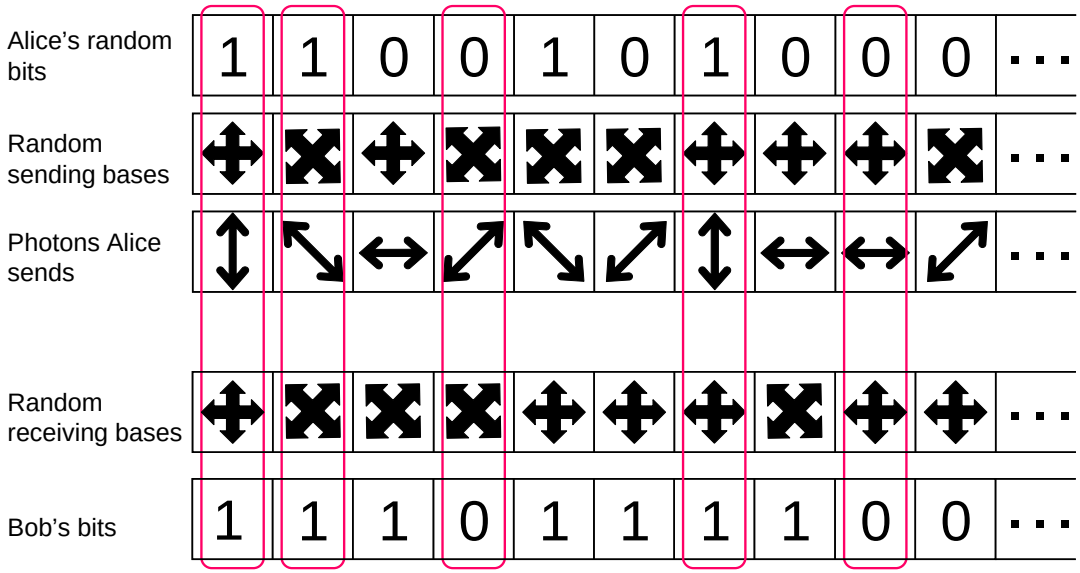

図 1.1: BB84 プロトコル

$u_0 = 0 = v_0$	v_1	v_2	$\cdots$	$v_{2^{k_2}-1}$
u_1	$u_1 + v_1$	$u_1 + v_2$	$\cdots$	$u_1 + v_{2^{k_2}-1}$
u_2	$u_2 + v_1$	$u_2 + v_2$	$\cdots$	$u_2 + v_{2^{k_2}-1}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
$u_{2^{k_1}-k_2-1}$	$u_{2^{k_1}-k_2-1} + v_1$	$u_{2^{k_1}-k_2-1} + v_2$	$\cdots$	$u_{2^{k_1}-k_2-1} + v_{2^{k_2}-1}$

表 1.1: 剰余類の例

1.9 単一光子

無条件安全性が保障されている BB84 量子鍵配送プロトコルを実行する際には単一光子を用いる．ごく最近まで，十分に高い確率で，単一光子を生成することはできなかった．光パルス中の光子の平均数を調整することは可能だったが，単一光子を生成する確率を高めようとすると，同時に光子が生成しない，あるいは，2 個の光子が生成してしまう確率も増えてしまう．量子暗号の実験では，光子の平均数が 0.1，つまり，10 個中 9 個のパルスが一つも光子を持たないようにすることによって，複数の光子が存在する確率を小さくしている．しかし，そのようにしてもなお，5 %のパルスは複数の光子を含むことになる．複数の光子を含むパルスは検出されない可能性があり，さらに，そもそもアリスとボブはどのパルスが複数の光子を含むか分からないので，安全性のリスクを回避するには鍵の長さを短くしなければならない．

Michler らは，マイクロディスクに埋め込まれた量子ドットから，単一の光子を得ること

に成功した [35]．さらにこの論文の著者の 1 人によって，これに関連する構造が，量子コンピューターの有力な候補として提案されている [22]．マイクロディスクはレーザーパルスを照射され，量子ドットの周りの GaAs 配列中の電子が励起される．電子は，正に帯電したホールとともに量子ドット中に捕らえられ，励起が起こる．電子とホールは励起子と呼ばれる混合系を形成し，やがて多くは光子の放出によって再結合する．もし複数の励起子が量子ドットを占拠していれば，それらの相互作用によって全ての初期の再結合は直前のものと違う周波数で起きる．モノクロメーターを用いて特定の量子ドットの周波数を分離すれば，この構造において一つのレーザーパルスに単一の光子を高確率で実現できる．

これより数週間早く，Lounis と Moerner[27] は全く違った単一光子生成法，すなわち，単一の分子（terrylene）を低濃度で 1μm の薄さの固体のフレークに埋め込む方法を発表した．この方法では，パンプレーザーで terrylene 分子を励起し，高エネルギーの励起状態にする．この状態はすぐに低エネルギー状態になり，0.5 ナノ秒で単一光子を放出する．この周波数において 2 個の光子が放出されるには，35 ナノ秒のパンプレーザーの持続期間の間に低エネルギー状態が崩壊しなければならない．これが起きる確率は 1200 分の 1 よりも小さい．Michler らの方式が低温状態を必要にするのに対しこの方式は常温で可能である．一方，この実験では，背景となる素材から放出される同じ周波数の光子が混ざってしまうために，単一光子パルスの割合は理論どおりにならない．よってどちらの方式がより実際的かは明らかではない．

1.10 盗聴者の持ちうる相互情報量

本節では，盗聴者の持ちうる相互情報量に関して議論する．アリスとボブがエンタングル状態 m ビット共有を試みるとき，忠実度 $F := |\langle\psi|\psi'\rangle|^2$ に対し $F > 1-\delta$ を満たす δ において，

$$\begin{aligned} I_{Eve} &< -(1-\delta)\log_2(1-\delta) - \delta\log_2\frac{\delta}{2^{2m}-1} \\ &< -(1-\delta)\log_2(1-\delta) - \delta\log_2\delta + 2m\delta \\ &= h(\delta) + 2m\delta \end{aligned}$$

がイヴの相互情報量の上界として与えられる [25]．ただし，$h(\cdot)$ は 2 元エントロピー関数である．BB84 量子鍵配送プロトコルに換言すれば，δ は復号誤り率に対応する．つまり，復号誤り率 P_E において，

$$\begin{aligned} I_{Eve} &< -(1-P_E)\log_2(1-P_E) - P_E\log_2\frac{P_E}{2^{2m}-1} \\ &< -(1-P_E)\log_2(1-P_E) - P_E\log_2 P_E + 2mP_E \\ &= h(P_E) + 2mP_E \end{aligned}$$

m は鍵長，つまり CSS 符号の符号長と符号化率が決定されれば定数となるとこから，C_1/C_2, $C_2^{\perp}/C_1^{\perp}$ の復号誤り率 P_E によりイブの相互情報量が抑制可能である．具体的に言えば，復号誤り率 P_E が $1/2m$ 以下になれば，イブの相互情報量は約 1 ビット以下と十分に小さい．

1.11 LDPC 符号

上述した CSS 符号を構成する古典線形符号として，本研究では良い復号特性を持つ LDPC 符号を用いて実際に C_1, C_2 を構成する．

LDPC 符号が Gallager によって提案された当時は，低速・低雑音・符号長も短い領域では，Reed-Solomon 符号などに比べ，目立った優位点が無く，符号長が大きいと計算量が莫大になるなどの理由ですぐに忘れ去られてしまった．その後，1993 年に開発されたターボ符号が繰り返し処理を使うことで高い性能を上げたことから，繰り返し処理を使う誤り訂正符号の研究が盛んになった．その結果，1996 年に 2 組のグループが独立に Gallager の LDPC 符号と同等なものを開発し，LDPC 符号は事実上再発見された．Galleger が提案してから再発見されるまでの間も，LDPC 符号に着目した研究者は存在したが，活発に研究が再開されたのは，やはり再発見以後になる．

再発見以後は，2001 年にターボ符号を超えて Shannon 限界に迫ることが示されてから，急速に支持を広げてきている．そして，衛星デジタル・テレビ放送の「DVB—S2」の採用を皮切りに，IEEE 標準規格に採用された 10G ビット/秒イーサネット，また，無線 LAN や無線インターネット，長距離光通信，携帯電話機へのコンテンツ配信，記憶装置（HDD）の信号処理回路などでも採用が有力になっている．ここにきて LDPC 符号が注目された背景には，それぞれの分野でこれまでの改善手法が軒並み限界に近づいていることがあげられれる．特に，通信分野では，従来の手法の延長では，更なる高速化が難しくなり，通信インタフェースの高速化が頭打ちになったことで，LDPC 符号のような高性能な誤り訂正符号は，この限界を超える手段になりうる．このため，次世代誤り訂正符号として期待されている．

LDPC 符号とその復号法である sum-product 復号法の組み合わせは極めて強力である．例えば，乱数に基づいて構成された符号長 1 万，符号化率 0.5 の正則 LDPC 符号により，ビット誤り率 10^{-5} において白色ガウス通信路の Shannon 限界から 1.3dB という復号特性が容易に得られる．さらに，うまくデザインされた非正則 LDPC 符号はより優れたビット誤り率特性を発揮することが知られている [39]．このように LDPC 符号と sum-product 復号法により，無記憶通信路における符号化問題は肯定的解決への道筋が開けてきた．

LDPC 符号の特長として，様々な符号長，符号化率の符号を容易に構成できる柔軟性が挙げられる．従来の誤り訂正符号では，符号の種類によって構成可能なパラメータが限られている場合が多い．他にもブロック誤り率特性が良いこと，ターボ符号の復号特性に観測されるエラーフロア現象がほとんど生じないことも LDPC 符号の利点である．

LDPC 符号の復号は，sum-product 復号法により行う．復号に要する時間計算量は符号長について線形時間である．また，この復号アルゴリズムは本質的に並列アルゴリズムであるため並列分散型ハードウェア実装に適している．

1.11.1 LDPC 符号の定義

まず最初に正則 LDPC 符号を定義する [19]．M 行 N 列のパリティ検査行列 H のどの列のハミング重み (列重みと呼ぶ) も j であり，どの行のハミング重み (行重みと呼ぶ) も k であるとする．さらに $j << M$ であるとき，パリティ検査行列 H により定義される符号 C を正則 LDPC 符号と呼ぶ．等式 $jN = kM$ から，$j/k = M/N$ を得る．従って符号 C の符号化率 r は，$r >= 1 - j/k$ となる．不等式となるのは，H のランクが必ずしも M であるとは限らないためである．LDPC 符号を構成するときには，列重み j は符号長 N に依らない定数ととることが一般的である (正則 LDPC 符号の場合，列重み j の値として 3 が選ばれることが多い)．その場合，検査行列内の 1 の数は $O(N)$ となる．例えば，i.i.d.2 元乱数列により無造作にパリティ検査行列を生成したとすると，そのパリティ検査行列内の 1 の個数は $O(N^2)$ となる．それに比べると 1 の数が $O(N)$ である正則 LDPC 符号のパリティ検査行列は非常に疎な行列である．

各列各行の重みが一定でない LDPC 符号を非正則 LDPC 符号と呼ぶ．良い列重み分布，行重み分布 (次数分布と呼ばれる) を持つ非正則 LDPC 符号は正則 LDPC 符号よりも優れた復号ビット誤り率特性を与えることが知られている [39]．

LDPC 符号はパリティ検査行列と等価なものとして，Tanner グラフ [47] と呼ばれる疎な 2 部グラフが存在する．

1.11.2 Tanner グラフ

疎な 2 部グラフである Tanner グラフについて説明する．Tanner グラフは，線形符号を定義する検査行列から構成される 2 部グラフである．具体例として，2 元線形符号のパリティ検査行列が

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

の時，この検査行列 H に対応する Tanner グラフは図 1.2 となる．

検査行列 H の行，列がそれぞれグラフのチェックノード，変数ノードに対応しており，H の $i+1$ 行 $j+1$ 列の要素を h_{ij} とすると，h_{ij} が 1 の時は変数ノード v_j とチェックノード c_i

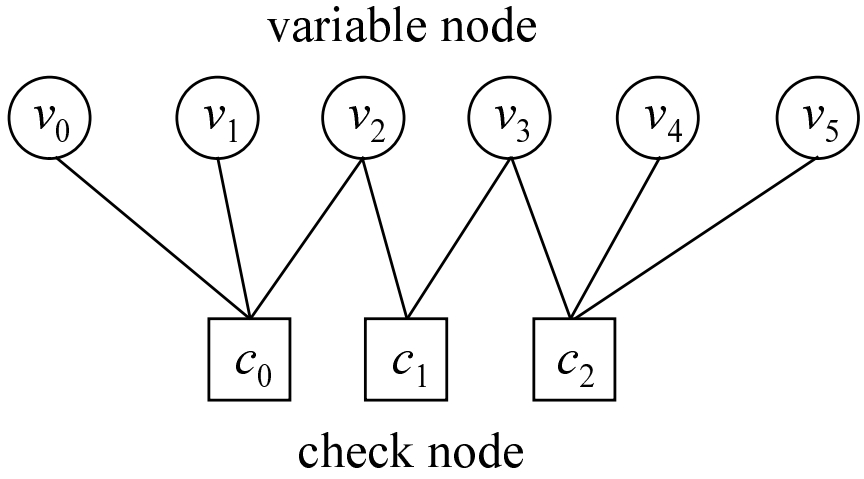

図 1.2: Tanner グラフの例

を枝で連結し，0 の時は連結しない．これにより，疎なパリティ検査行列を疎な Tanner グラフで表現でき，符号解析をグラフ理論で行うことも可能である．

1.12 sum-product 復号法

LDPC 符号は sum-product 復号法を用いた復号により，Shannon 限界に迫る性能を示している．sum-product アルゴリズムは，復号アルゴリズムとしてだけでなく，ディジタル通信，信号処理，人工知能などの分野において様々な形で利用されている．例えば，隠れマルコフモデルに対する前向き後ろ向きアルゴリズム（BCJR アルゴリズム），Vitebi アルゴリズム，ターボ復号法，カルマンフィルタなどが挙げられる．このように，様々な一見異なるアルゴリズムが同じアルゴリズム原理に基づいているということが明らかになってきたのは比較的最近のことである．

Gallager は彼の博士論文の中で，sum-product 復号法を LDPC 符号の復号法として完全な形で記述している．Wiberg は，Tanner グラフに状態ノードを追加することを考案した [52]．このアイディアにより，線形符号の最簡トレリス，畳み込み符号のトレリスに基づく BCHR アルゴリズムと Viterbi アルゴリズムが sum-product アルゴリズムのインスタンスであることが明確になった．また，同時に彼は，sum-product アルゴリズムの抽象化を行った．

Aji と McEliece は，Wiberg のアイディアを発展させるとともに，sum-product アルゴリズムが，因子分解される目的関数の計算を分配則に基づいて効率的に行うアルゴリズムであることを明らかにした [6]．Kscishang と Frey はファクターグラフ [24] を導入し，目的関数の因子分解をグラフ化することにより sum-product アルゴリズムが見通しよく理解できることを

示した．彼らはさらに，基礎となるグラフィカルモデルを変えることにより，sum-product アルゴリズム原理から FFT など数多くの既知アルゴリズムが自然に導かれることを明らかにした．この sum-product 復号法について説明する．

H を 2 元 $M \times N$ の検査行列とし，集合 $A(m),\ B(n)$ を次のように定義する．

$$A(m) := \{j : h_{ij} = 1\}$$

$$B(n) := \{i : h_{ij} = 1\}$$

このとき，sum-product 復号法は次のようなステップで実行される．

Step1: $h_{ij} = 1$ を満たす全ての組 $(i,\ j)$ に対し，事前対数比 $\beta_{ij} = 0$ とする．また，ループ変数を $l = 1$ とし，ループ最大回数を $l_{\max}$ とする．

Step2: $i = 1,\ 2,\ \cdots,\ M$ の順に，$h_{ij} = 1$ を満たす全ての組 $(i,\ j)$ に対して

$$\alpha_{ij} = \left(\prod_{n \in A(m) \backslash j} \mathrm{sign}(\lambda_n + \beta_{in}) \right) \times f \left(\sum_{n \in A(m) \backslash j} f(|\lambda_n + \beta_{in}|) \right)$$

を計算することで外部値対数比 α_{ij} を求める．

また，それぞれ

$$\mathrm{sign}(x) := \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$

$$f(x) := \ln \frac{\exp(x) + 1}{\exp(x) - 1}$$

$$\lambda_n := \ln \frac{P(y_n | x_n = 0)}{P(y_n | x_n = 1)}$$

と定義されるものとする．

Step3: $i = 1,\ 2,\ \cdots,\ N$ の順に，$h_{ij} = 1$ を満たす全ての組 $(i,\ j)$ に対して

$$\beta_{ij} = \sum_{m \in b(n) \backslash i} \alpha_{mj}$$

を計算する．

Step4: $j \in [1,\ N]$ において，

$$\hat{u_j} = \begin{cases} 0, & \mathrm{sign}(\lambda_j + \sum_{m \in b(n) \backslash i} \alpha_{mj}) = 1 \\ 1, & \mathrm{sign}(\lambda_j + \sum_{m \in b(n) \backslash i} \alpha_{mj}) = -1 \end{cases}$$

を計算する．

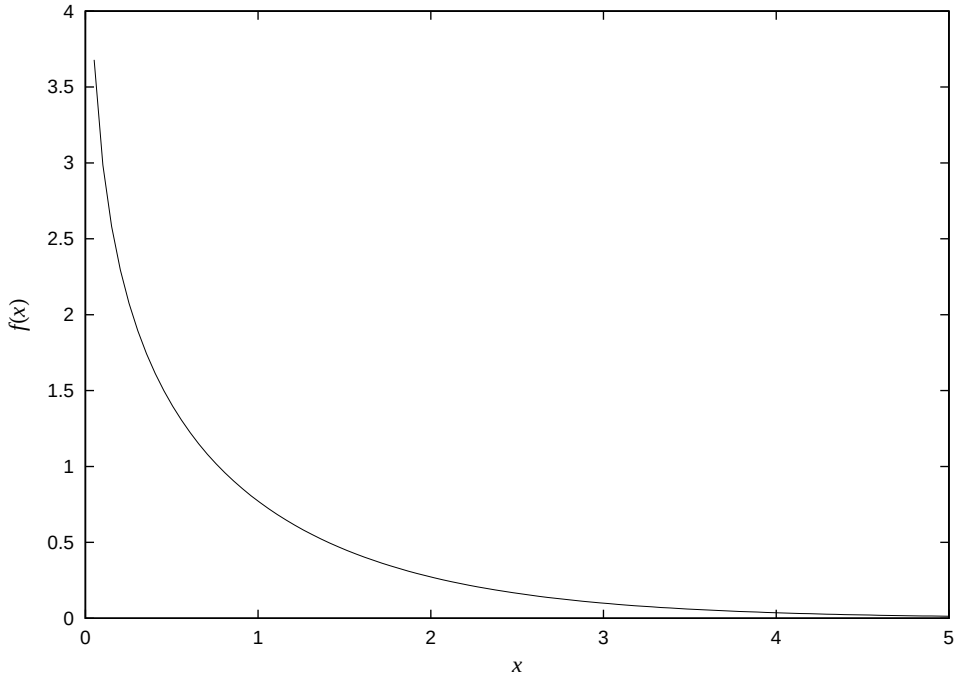

図 1.3: Gallager の $f(x)$ 関数

Step5: 符号語になっているかどうかを検査する．$(\hat{u_1},\ \hat{u_2},\ \cdots,\ \hat{u_N})H^T = \mathbf{0}$ なら $(\hat{u_1},\ \hat{u_2},\ \cdots, \hat{u_N})$ を推定語として出力し，アルゴリズムを終了する．

Step6: $l \leq l_{\max}$ なら $l = l + 1$ として Step2 へ戻り，$l > l_{\max}$ なら $(\hat{u_1},\ \hat{u_2},\ \cdots,\ \hat{u_N})$ を推定語として出力し，アルゴリズムを終了する．

sum-product 復号法は，Tanner グラフ中にサイクルが存在しない場合は正確な最大事後確率復号法となるが，Tanner グラフ中にサイクルが存在する場合には正確な最大事後確率復号法復号にはならない．特に長さ 4 などの短いサイクル（図 1.4）は，復号性能に大きく影響することが知られている．

また，2 元対称通信路では，

$$\lambda_n = \begin{cases} \ln(1-p)/p, & y_n = 0 \\ \ln p/(1-p), & y_n = 1 \end{cases}$$

となる．

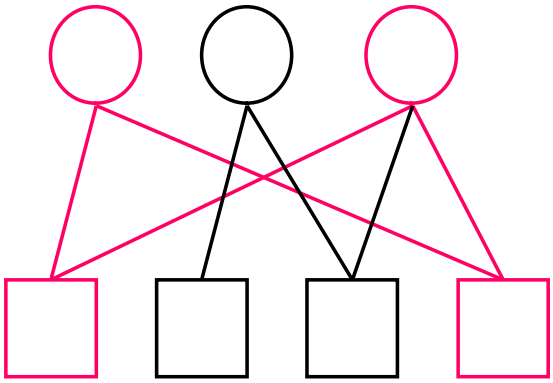

図 1.4: 長さ 4 のサイクルの例

1.13 次数分布

Tanner グラフは 2 つの多項式 $\lambda(x)$，$\rho(x)$ によって作られる．多項式が

$$f(x) = \sum_{i\geq 2} f_i x^{i-1} \quad (0 \leq f_i \leq 1,\ \forall i \geq 2) \tag{1.1}$$

$$f(1) = \sum_{i\geq 2} f_i = 1 \tag{1.2}$$

を満たしているとき，この多項式を次数分布と呼ぶ．変数ノードの次数分布を $\lambda(x)$，チェックノードの次数分布を $\rho(x)$ として定義する．

枝が i 本出ているノードのことを次数 i のノード，と呼び，f_i は枝の総数に対する次数 i のノードの割合を表す．

この次数分布により，最適な非正則 LDPC 符号を表現する．Sannon 限界（図 1.5）に対し，現在行重み 2〜100 を用いて構成された非正則 LDPC 符号の二元対称通信路における復号性能は表 1.2 となっている [7].

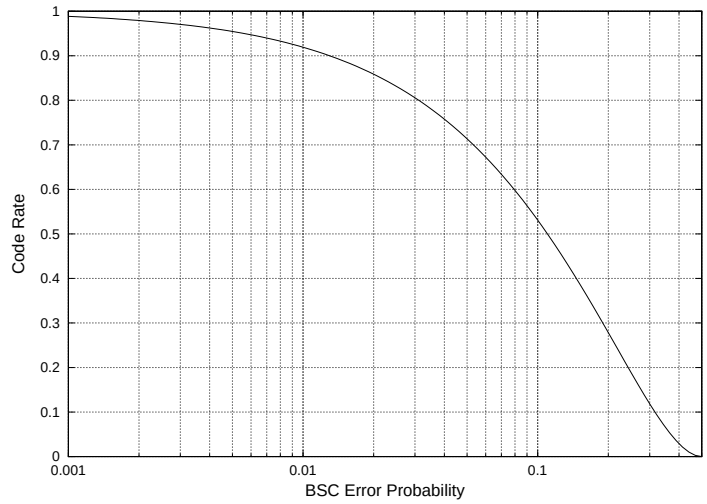

図 1.5: Shannon 限界

符号化率	平均行重み	(通信路容量 - 符号化率)/通信路容量
0.05	3.50	1.72173%
0.06	3.65	1.13250%
0.07	3.75	1.96752%
0.1	4.00	1.12614%
0.15	4.50	1.86714%
0.2	5.00	1.62356%
0.25	5.20	1.84981%
0.35	6.50	1.74238%
0.4	8.00	12.36440%
0.45	15.20	20.14620%
0.5	10.00	0.01681%
0.6	8.00	3.12008%
0.63	10.81	0.05641%
0.65	14.00	1.62836%
0.68	15.50	0.08023%
0.7	17.70	0.14172%
0.72	17.50	0.29916%
0.75	24.00	0.41712%
0.8	35.00	2.54405%
0.85	40.00	0.99696%

表 1.2: 二元対称通信路における LDPC 符号の復号性能

第2章

双対符号を含むLDPC符号構成

本章では実際に $C_1,\ C_2$ を構成する．現在実用的な復号方法で良い復号特性と持つ符号としてLDPC符号がある．このLDPC符号の1つであるArray-type LDPC符号を用いて，双対符号を含む符号 C_1 を構成する．双対符号とは，元の $[n,\ k]$ 線形符号に対する直交補空間であり，双対符号もまた $[n,\ n-k]$ 線形符号となる．

まず，構成する際に使用するLDPC符号について説明する．

2.1 Array-type LDPC符号

素数 p に対し，サイズ $p \times p$ の巡回置換行列を

$$\alpha := \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

とすると，[13] のLDPC符号のパリティ検査行列 H_{RS} は

$$H_{RS} = \begin{bmatrix} I & I & I & \cdots & I \\ I & \alpha & \alpha^2 & \cdots & \alpha^{k-1} \\ I & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(k-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I & \alpha^{j-1} & \alpha^{2(j-1)} & \cdots & \alpha^{(j-1)(k-1)} \end{bmatrix} \tag{2.1}$$

となる．I は $p \times p$ の単位行列，j, k は $3 \leq j, k \leq p$ とする．H_{RS} は行重みが k，列重みが j となる．H_{RS} により定義される符号の最小距離は $j+1$ 以上であり，Tannerグラフ中の最も短いサイクル（以後内径とする.）は6である [18]．このArray-type LDPC符号は正則LDPC符号となり，様々な符号長，符号化率において良い復号性能を示している [13].

2.2 双対符号を含むLDPC符号

双対符号を含むLDPC符号を用いて C_1 を構成する．双対符号を含むLDPC符号は [31] でも検討されているが，以下に提案する構成法は別手法で構成していることを注意しておく．

C_1 が双対符号を含むので，$C_2 = C_1{}^{\perp}$ とすれば $C_1 \supseteq C_2$ となる．このとき $C_1 = C_2{}^{\perp}$ であるから，C_1, $C_2{}^{\perp}$ の復号性能は等しくなる．双対符号を含む符号 C_1 についてもう少し具体的に述べると，パリティ検査行列 H の各行ベクトルが符号語となれば良く，H の任意の行ベクトル $\boldsymbol{h}_{r_1}$，$\boldsymbol{h}_{r_2}(1 \leq r_1,\ r_2 \leq m,\ r_1 \neq r_2)$ を比較して，両ベクトルともに列成分が 1 となる列数が偶数個であれば良い．ここで，列成分がともに 1 となる列数が 2 列以上存在する場合，Tanner グラフに長さ 4 のサイクルが出来ることを注意しておく．3 節で述べたように，長さ 4 のサイクルは復号性能に悪影響を与える．つまり $C_1 \supset C_2$ を満たし，長さ 4 のサイクルの数をなるべく少なくするには，列成分がともに 1 となる列数を 0 または 2 列になるようにパリティ検査行列を構成すれば良い．

ここで最小距離と列重みの関係について考察する．列重みが 2 以下でこのようにパリティ検査行列を構成すると，パリティ検査行列 H の列ベクトルにおいて $\boldsymbol{h}_{c_1} = \boldsymbol{h}_{c_2}(1 \leq c_1,\ c_2 \leq n,\ c_1 \neq c_2)$ となる c_1, c_2 が存在し，重み 2 の符号語が出来る．つまり最小距離が 2 となるため良い復号性能が期待できない．列重みを 3 以上とし，任意の c_1, $c_2(1 \leq c_1,\ c_2 \leq n,\ c_1 \neq c_2)$ について $\boldsymbol{h}_{c_1} \neq \boldsymbol{h}_{c_2}$ となるようにパリティ検査行列を構成する必要がある．

[31] の構成方法は，different set と呼ばれる，パリティ検査行列内の要素が 1 で要素同士の間隔が，全て異なる行列となる．このようなパリティ検査行列を行数と列数が等しいように構成し，符号化率に応じて適宜行ベクトルを間引く方法である．

2.2.1 提案する C_1 の構成方法

提案する構成法では，重み 2 の符号語が出来ないように列重みを 3 とし，$p \times p$ の単位行列 I と巡回置換行列 α を用いて，双対符号を含むLDPC符号となるように C_1 を構成する．

ここで行重み k も決定する．C_1 の符号化率を r_1，C_2 の符号化率を r_2 とすると量子符号化率 $r_1 - r_2$ が大きければアリスとボブで共有できる鍵のサイズが大きくなる．符号 C の符号化率を 0.75 とすると $C^{\perp}$ の符号化率は 0.25 となり，C_1/C_2 の符号化率は 0.5 となる．本論文では $k = 12$ とし，C_1 の符号化率が約[1]0.75 となるようにパリティ検査行列 H を構成した．

行重み $k = 12$ より，サイズ $3p \times 12p$ のパリティ検査行列となる．サイズ $p \times p$ の単位行列 I，巡回置換行列 α は任意の 2 列において，列の重なりは 0 である．列重み 3 の場合を

[1]列重み j に対し $j-1$ 個の一次従属な行ベクトルを持つので，厳密な符号化率は $\frac{p\times(k-j)+(j-1)}{p\times k}$ となる．

考えているため，p 行ずつ分割した 3 列の組み合わせでのみ列の重なりを考えれば良いことになる．

また，ここで内径について考察する．パリティ検査行列 H をサイズ $p\times p$ の行列ごとに分割する．このとき，3×12 個のブロック成分に分割される．i 行 q 列目 $(1\le i\le 3,\ 1\le q\le 12)$ のブロック成分は，ある $t_{(i,\ q)}$ を用いて $\alpha^{t_{(i,\ q)}}$（$\alpha^0 = I$ とする.）と表せる．格子状の位置関係にある 4 つの要素

$$\begin{bmatrix} \cdots & \alpha^{t_{(i_1,\ q_1)}} & \cdots & \alpha^{t_{(i_1,\ q_2)}} & \cdots \\ & \vdots & & \vdots & \\ \cdots & \alpha^{t_{(i_2,\ q_1)}} & \cdots & \alpha^{t_{(i_2,\ q_2)}} & \cdots \end{bmatrix}$$

において，$t_{(i_1,\ q_1)} - t_{(i_2,\ q_1)} = t_{(i_1,\ q_2)} - t_{(i_2,\ q_2)}$ が成り立つとき，$\begin{bmatrix}\alpha^{t_{(i_1,\ q_2)}}\\ \alpha^{t_{(i_2,\ q_2)}}\end{bmatrix}$ は $\begin{bmatrix}\alpha^{t_{(i_1,\ q_1)}}\\ \alpha^{t_{(i_2,\ q_1)}}\end{bmatrix}$ を $t_{(i_1,\ q_2)} - t_{(i_1,\ q_1)}$ 巡回させたものと等しくなり，長さ 4 のサイクルが p 個出来る．逆に $t_{(i_1,\ q_1)} - t_{(i_2,\ q_1)} \ne t_{(i_1,\ q_2)} - t_{(i_2,\ q_2)}$ の時は 4 つの要素内に限ればサイクルは存在しない．

ここで簡単のため，整数 $t_{i,\ q}$ を成分に持つ行列 $H_{3,\ 12}$ を，

$$H_{3,\ 12} = (t_{i,\ q})_{1\le i\le 3,\ 1\le q\le 12}$$

で定義し，$H_{3,\ 12}$ をパリティ検査行列と同一視する．以下に C_1 のパリティ検査行列 $H_{3,\ 12}$ の構成手順を述べる．

(a) 1 行目のブロックは， α^{q-1} とした．

(b) 2 行目のブロックは， q 列と $q+6$ 列のブロックで長さ 4 のサイクルが出来るように α^t を設定した．

(c) 3 行目のブロックは，1，2 行目と q 列と $q+6$ 列のブロック以外のペアでサイクルが出来るように α^t を設定する．例としては，2 行目と q 列と $q+3$ 列で長さ 4 のサイクルが出来るようにペアを作り，1 行目と q 列と $q+1$ 列で長さ 4 のサイクルが出来るようにペアを順々に作る等が挙げられる．本論文でもこの例に基づく形で実際に構成している．

この構成法によって構成されパリティ検査行列は，任意の行ベクトルの重なりが 0 または 2 であり，双対符号を含む LDPC 符号である．実際に構成したパリティ検査行列 $H_{3,\ 12}$ は

$$H_{3,\ 12} = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 3 & 5 & 7 & 9 & 11 & 7 & 9 & 11 & 13 & 15 & 17 \\ 11 & 12 & 15 & 12 & 10 & 12 & 13 & 13 & 17 & 22 & 20 & 21 \end{bmatrix} \tag{2.2}$$

である．

2.2.2 C_1 の最小距離と内径

C_1 の最小距離について考える．線形符号なので最小重みの符号語を考えれば良い．

定理 2.1 C_1 の最小距離は 4，内径は 4 である．

証明 符号語 $\boldsymbol{u}$ について $H_{3,\ 12}\boldsymbol{u}^T = \boldsymbol{0}$ が成り立つ．つまり，$\boldsymbol{u}$ の成分が 1 に対応する列ベクトルについての重みの和が 0 となる必要があり，列ベクトルの要素が 1 である要素の総数は偶数である．これより，列重みが 3 なので最小距離は偶数であることがわかる．C_1 のパリティ検査行列 $H_{3,\ 12}$ には同じ列ベクトルは存在しないので，最小距離は 4 以上であることがわかる．また一例として，符号語 $\boldsymbol{u}$ の n 列の成分の内，164, 574, 740, 822 列の 4 つの成分が 1 のとき $H_{3,\ 12}\boldsymbol{u}^T = \boldsymbol{0}$ を満たしている．よって最初距離は 4 である．内径に関しては長さ 4 のサイクルが存在するので 4 である． ■

2.2.3 C_2 の最小距離

C_2 の最小距離は，C_1/C_2 の復号性能と関係するため．ここでは C_2 の最小距離を考える．

定理 2.2 C_2 の最小距離は 12 である．

証明 $C_1 = C_2^{\perp}$ より，C_1 のパリティ検査行列 $H_{3,\ 12}$ の行ベクトルが C_2 の生成元となる．つまり，任意の $\boldsymbol{u} \in C_2$ は パリティ検査行列 $H_{3,\ 12}$ の行ベクトルの線形和で表現できる．また，$H_{3,\ 12}$ の行ベクトルは C_2 の符号語であることから，C_2 の最小距離は 12 以下である．C_2 の最小距離が 12 未満だと仮定すると， $\boldsymbol{u}$ を p 列ずつに分割した 12 列のブロックの中に，全ての要素が 0 となる列ブロックが存在する．全ての要素が 0 となる列ブロックが存在すると重みが必ず 12 以上となることを示す．

$3 \times p$ 行を p 行ずつ分割した 3 つの行ブロックを，それぞれ $A,\ B,\ C$ とすると，次の 3 つの場合に分けることが出来る．

(i) $\boldsymbol{u}$ が $A,\ B,\ C$ のうち 1 つの行ブロックだけで表現できるときは，全ての要素が 0 となる列ブロックが明らかに存在せず，$\boldsymbol{u}$ の重みは 12 以上となる．

(ii) $\boldsymbol{u}$ が $A,\ B,\ C$ のうち 2 つの行ブロックで表現できるときは，全ての要素が 0 の列ブロックと，パリティ検査行列 $H_{3,12}$ でペアとした列ブロック以外は 0 とならず，必ず 2 個以上 1 を含むため $\boldsymbol{u}$ の重みは 20 以上となる．

(iii) $\boldsymbol{u}$ が $A,\ B,\ C$ の 3 つのブロックで表現できるとき，パリティ検査行列 $H_{3,\ 12}$ に同じ列ベクトルは存在しないことから，全ての要素が 0 の列ブロックが存在するときに，別の列ブロックも全て 0 となる列ブロック数は，$A,\ B,\ C$ のブロックを入れ替えても全ての要素が 0 となる場合の総数である．よって最大の場合でも $3! = 6$ 列であり， $\boldsymbol{u}$ の重みは 12 以上となる．

以上から，いずれの場合も全ての要素が 0 の列ブロックが存在すると必ず重みが 12 以上となるため，C_2 の最小距離は 12 である．■

2.3 実験結果

本節では，C_1, C_1/C_2 の復号性能についてそれぞれ考察する．

2.3.1 C_1 の復号性能

2.1 節で説明した (2.1) の LDPC 符号と，本論文で提案した双対符号を含む LDPC 符号 (2.2) の復号性能比較を行った．(2.2) の LDPC 符号は，素数 p を それぞれ $p = 83,\ 419$ とし，符号長は 996，5028，(2.1) の LDPC 符号は $p = 83,\ j = 3,\ k = 12$, 符号長 996 とした．符号化率は全て約 0.75 である．通信路は BSC とし，最大反復回数は 100 回とした．横軸は BSC 誤り率，縦軸はブロック誤り率としている．量子鍵配送では鍵共有が目的であり，ブロック誤り率が盗聴者の情報量と関係しているため，復号性能をブロック誤り率で評価する．実験結果は図 2.1 である．

C_1 の復号性能は，符号長が長くなるほど良くない．符号長を長くしても p を大きくしているだけで最小距離は変わらないためである．このため符号長と BSC 誤り率の積が小さくなる場合でないと正しく復号出来ない．また，同じ符号長 (2.1) の LDPC 符号と比較しても，復号性能が悪いことがわかる．まず長さ 4 のサイクルがあることが原因として考えられる．また，別の符号語に誤復号された符号語を調べてみると，(2.1) の LDPC 符号は重み 6 の符号語に誤復号されることが多く，(2.2) の LDPC 符号は重み 4 の符号語に誤復号されることが多かった．(2.1) と (2.2) の LDPC 符号において，最小距離の下界は同じであるが，(2.1) の符号の最小距離は 6 であるか，重み 4 の符号語が少ないことが考えられる．以上の理由により (2.2) の方が復号性能が良くないと考えられる．

2.3.2 C_1/C_2 の復号性能

C_1 単体の復号性能が良くなくても，C_2 を含めた C_1/C_2 の復号性能が良ければよい．ここでは C_1/C_2 の復号性能について考察する．

符号長約 1000，BSC の誤り率 0.005 の条件で実験を行うと，ブロック誤り 10000 回のうち，別の符号語に誤復号されたのは約 4 割の 3943 回となった．この誤復号された符号語が C_2 の符号語となっていれば，C_1 単体での復号性能に比べて C_1/C_2 の復号性能は良くなる．誤復号された符号語の結果を表 2.1 に示す．

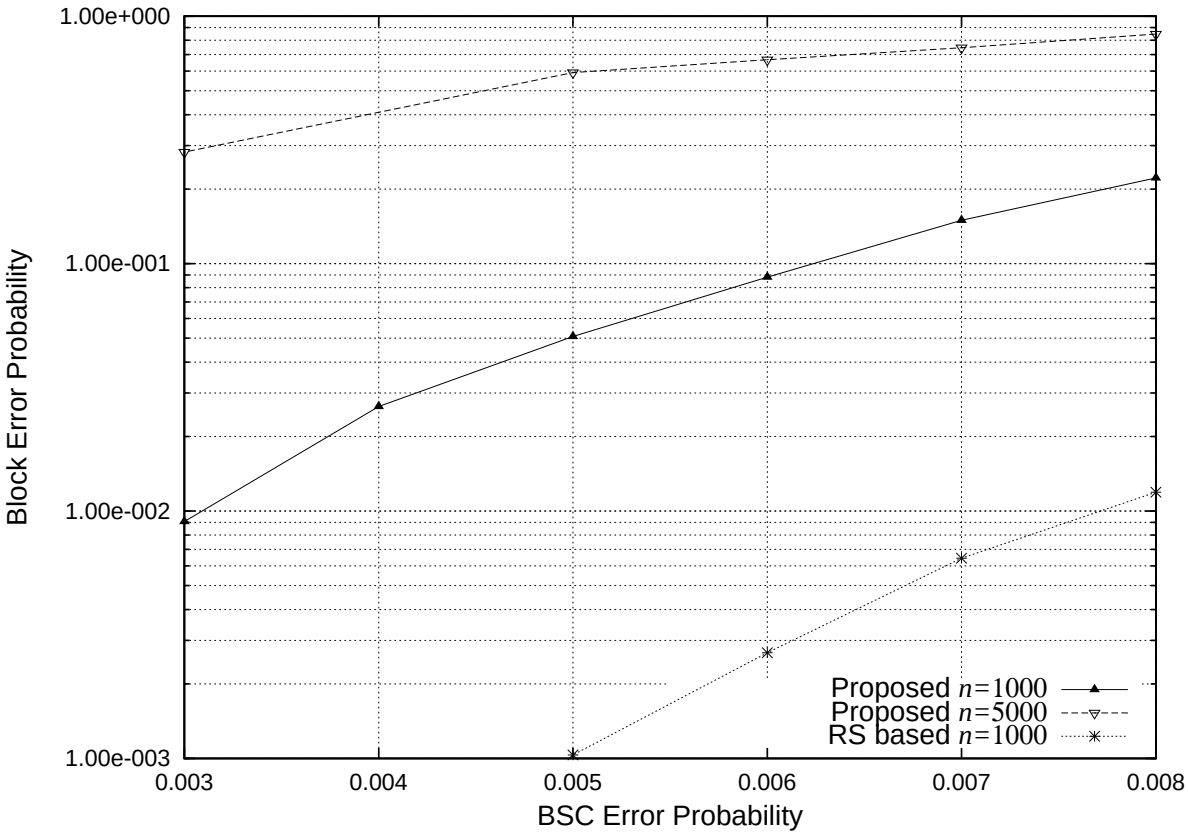

図 2.1: 双対符号を含む LDPC 符号 $C_1(C_2^{\perp})$ の復号性能実験結果

重み 4 の符号語に誤復号されたのが，全体の 93%にも及んだ．C_2 の最小距離は定理 2 より 12 であり，重み 4 の符号語は C_2 の符号語ではない．重み 12 以上の符号語についても，この条件下では C_2 の符号語に誤復号されることはなく，C_1 と C_1/C_2 の復号性能は同じとなった．

C_1 の最小距離が 4 で，C_2 の最小距離が 12 とその差が大きいため，C_1/C_2 の復号性能は，C_1 単体での復号性能と変わらない．BSC の誤り率を大きくすれば，重み 12 以上の符号語に誤復号される割合は増えるが，C_1 単体での性能は劣化し，最大反復回数内で符号語を推定出来ない場合も増加するため，C_1/C_2 の復号性能も良くない．全体として，C_1 と C_1/C_2 とでの復号性能において差は生じにくいと考えられる．

2.4 考察

本論文では，双対符号を含む LDPC 符号を用いて C_1 を構成した．提案手法で C_1, C_2 を構成すると，C_2 の最小距離が C_1 のパリティ検査行列の列重みと等しくなり，C_1 の最小距

符号語の重み	4	6〜10	12〜
個数	3663	183	97
割合	**93%**	4.5%	2.5%

表 2.1: 誤復号された符号語に関する結果

離と C_2 の最小距離に大きな差が生じる．よって C_1 単体での復号性能に依存せざるを得ない．その為，提案手法を基に C_1, C_2 を構成する場合は，C_1 単体で良い復号特性を持つように構成する必要がある．そこで問題となるのが最小距離と内径である．提案手法では長さ 4 のサイクルが生じるが，[23] により長さ 4 のサイクルを除去することは可能である．C_1 の最小距離を大きく方法としては，列重みをさらに大きくして構成する方法が挙げられる．また，本論文と異なる手法で双対符号を含む LDPC 符号を構成すれば，C_1 の最小距離が小さく復号性能が悪くても，小さい符号語を C_2 に集めることで C_1 の復号性能を補える可能性がある．しかし，双対符号を含む構成法では，LDPC 符号の復号特性を良くする条件と相反するため，もし双対符号を含まないような構成法が可能であれば，双対符号を含まない構成法の方が良い．

第3章

任意LDPC符号による構成

本章では，良い復号特性を持つ任意 LDPC 符号を C_1 に適用した場合を考察する．任意の LDPC 符号が使用可能なため，実装面での利点があり，復号性能と実装面の両立が可能となる構成法である．また，前章と異なり，符号が双対符号を含まない場合に相当するため，符号が双対符号を含む場合のような，LDPC 符号の良い特性と相反する必要条件を持つ必要性がない．

3.1 使用する非正則LDPC符号

まず実験の際に C_1 に適用する LDPC 符号を考慮する．C_1 は実用的な復号法で良い復号特性を持つことが望まれるため，LDPC 符号を適用するのが有用である．本章では，[17] の非正則 LDPC 符号を C_1 に適用する．以下，その非正則 LDPC 符号として有用な [17] の LDPC 符号について説明を行う．

素数 p に対し，サイズ $p \times p$ の巡回置換行列を

$$P := \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

とし，サイズ $p \times p$ の行列を

$$T := \begin{bmatrix} 1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

とする．サイズ $M \times N$ のパリティ検査行列 $H := [H^{(p)} \mid H^{(d)}]$ において，$H^{(p)}$, $H^{(d)}$ をそれぞれ

$$H^{(p)} := \begin{bmatrix} T & I & O & \cdots & O & O \\ O & I & I & \cdots & O & O \\ O & O & I & \cdots & O & O \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ O & O & O & \cdots & I & I \\ O & O & O & \cdots & O & I \end{bmatrix}$$

$$H^{(d)} := \begin{bmatrix} I & I & \cdots & I \\ P & P^2 & \cdots & P^k \\ P^2 & P^4 & \cdots & P^{2k} \\ \vdots & \vdots & \vdots & \vdots \\ P^{j-1} & P^{2(j-1)} & \cdots & P^{k(j-1)} \end{bmatrix}$$

と定義する．さらに，$H^{(d)}$ の 要素 I, $P^{k(j-1)}$ をマスク手法 [10] により，適宜零行列 O に置き換え，$H^{(d)}$ を $H_W^{(d)}$ と変形することで，非正則化を行う．マスク手法とは，H^d を $j \times k$ のブロック行列 $H^d = (H^d_{j,\,k})$ とし，$j \times k$ の行列 $W = (w_{j,\,k})$ との演算 $W \circ H^d$ を $w_{j,\,k} = 0$ のときは $w_{j,\,k} H^d_{j,\,k} = 0$，$w_{j,\,k} = 1$ のときは $w_{j,\,k} H^d_{j,\,k} = H^d_{j,\,k}$ とする手法であり，正則 LDPC 符号を零行列と置換することで，非正則 LDPC 符号に変換する手法として，近年注目されている手法である．

[17] の LDPC 符号の符号化は非常に単純なので，ハードウェアで実装するのに適している．その符号化アルゴリズムを説明する．

符号語を $\boldsymbol{u} := (\boldsymbol{u}_p,\ \boldsymbol{u}_d)$ とする．$\boldsymbol{u}_p$ は長さ jp，$\boldsymbol{u}_d$ は長さ kp のベクトルで，$\boldsymbol{u}_p$ を $\boldsymbol{u}_d$ から計算する．いま 2 元で考えているので，$H_Y \boldsymbol{u}^T = \boldsymbol{0}^T$ から，$H^p \boldsymbol{u}_p^T = H^d \boldsymbol{u}_d^T$ が成り立つ．

符号化のアルゴリズムは，以下のステップで表現される．

Step1: $\boldsymbol{u}_d$ を $(\boldsymbol{u}_{d,\,0},\ \boldsymbol{u}_{d,\,1}, \cdots,\ \boldsymbol{u}_{d,\,k-1})$ と分解する．$\boldsymbol{u}_i (0 \le i \le k-1)$ は長さ p の 2 元ベクトルである．

Step2: $\boldsymbol{v}_i (0 \le i \le j-1)$ を長さ p の 2 元ベクトル，$(\boldsymbol{v}_{d,\,0},\ \boldsymbol{v}_{d,\,1}, \cdots,\ \boldsymbol{v}_{d,\,j-1})^T := H^d \boldsymbol{u}_d^T$ とすると，

$$\boldsymbol{v}^T_{d,\,b} = \sum_{i=0}^{k-1} P^{bi} \boldsymbol{u}^T_{d,\,i},\ (0 \le b \le j-1)$$

となり，この計算は Horner 法[1]を使用することで，図 3.1 のような簡単な回路で計算可能となる．(図中の p は p ビットずつの意.)

[1] $a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$ を $((\cdots((a_n x + a_{n-1})x + a_{n-2})x + \cdots)x + a_1)x + a_0$ と変形して計算することで，加算 n 回，乗算 $n(n+1)/2$ 回の計算を加算 n 回，乗算 n 回で行える．このような変形を行い，演算回数を減らす手法のことを，Horner 法と呼ぶ.

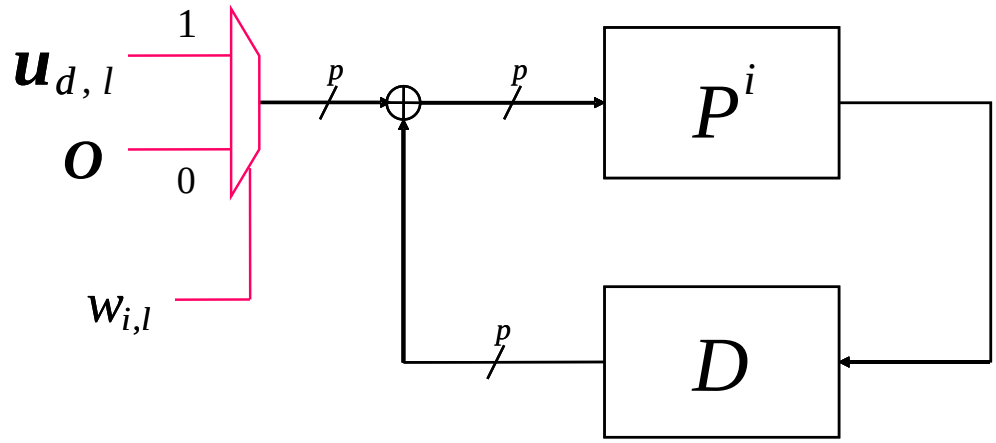

図 3.1: 回路図

Step3: $\boldsymbol{u}_p$ は $H^p \boldsymbol{u}_p^T = (\boldsymbol{v}_{d,\,0},\ \boldsymbol{v}_{d,\,1}, \cdots,\ \boldsymbol{v}_{d,\,j-1})^T$ で演算可能である.

詳細は [17] を参照されたい.

3.1.1 LDPC 符号の復号実験結果

まず, [17] の LDPC 符号の復号性能実験を行った. パラメータ設定として, 素数 $p = 139$, 23 とし, 符号化率 0.5, 符号長はそれぞれ 5004, 828 である. 非正則化に用いたマスク行列 W を以下に示す.

$$W = \begin{bmatrix}
0&0&1&0&0&0&0&0&0&0&0&1&0&0&1&0&1&0\\
0&0&0&0&0&0&1&0&0&0&1&0&1&0&1&0&0&1\\
0&0&0&0&1&0&0&1&0&0&0&1&0&0&0&0&1&0\\
1&0&0&0&0&0&0&0&1&0&0&0&1&1&0&1&0&0\\
0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&1&1\\
0&0&0&0&0&1&0&0&0&1&0&1&0&1&0&1&0&0\\
0&1&0&0&1&0&0&0&0&0&1&0&0&1&0&0&0&1\\
1&0&0&0&0&0&1&0&0&0&0&1&0&0&1&0&1&0\\
0&0&0&1&0&0&0&0&0&1&0&1&0&1&0&1&0&0\\
0&1&0&0&0&0&0&0&0&0&1&0&1&0&1&0&0&1\\
1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&1\\
0&0&0&0&1&0&1&0&1&0&0&0&0&1&0&0&1&0\\
0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&1&0&1\\
0&0&0&0&0&1&0&1&0&1&0&0&0&1&0&0&1&0\\
0&1&0&0&0&0&0&0&0&0&1&0&1&0&1&1&0&0\\
0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&1\\
0&0&0&0&0&1&0&0&0&0&0&1&0&1&0&1&0&0\\
0&0&0&1&0&0&0&0&1&0&1&0&1&0&1&0&1&0
\end{bmatrix}$$

本研究で仮定する通信路は 2 元対称通信路とし, 最大反復回数は 100 回とした. 横軸は 2 元対称通信路のビット誤り率, 縦軸はブロック誤り率としている. 量子鍵配送では鍵共有が

目的であり，上述したように，ブロック誤り率がイヴの相互情報量と関係しているため，復号性能をブロック誤り率で評価する．

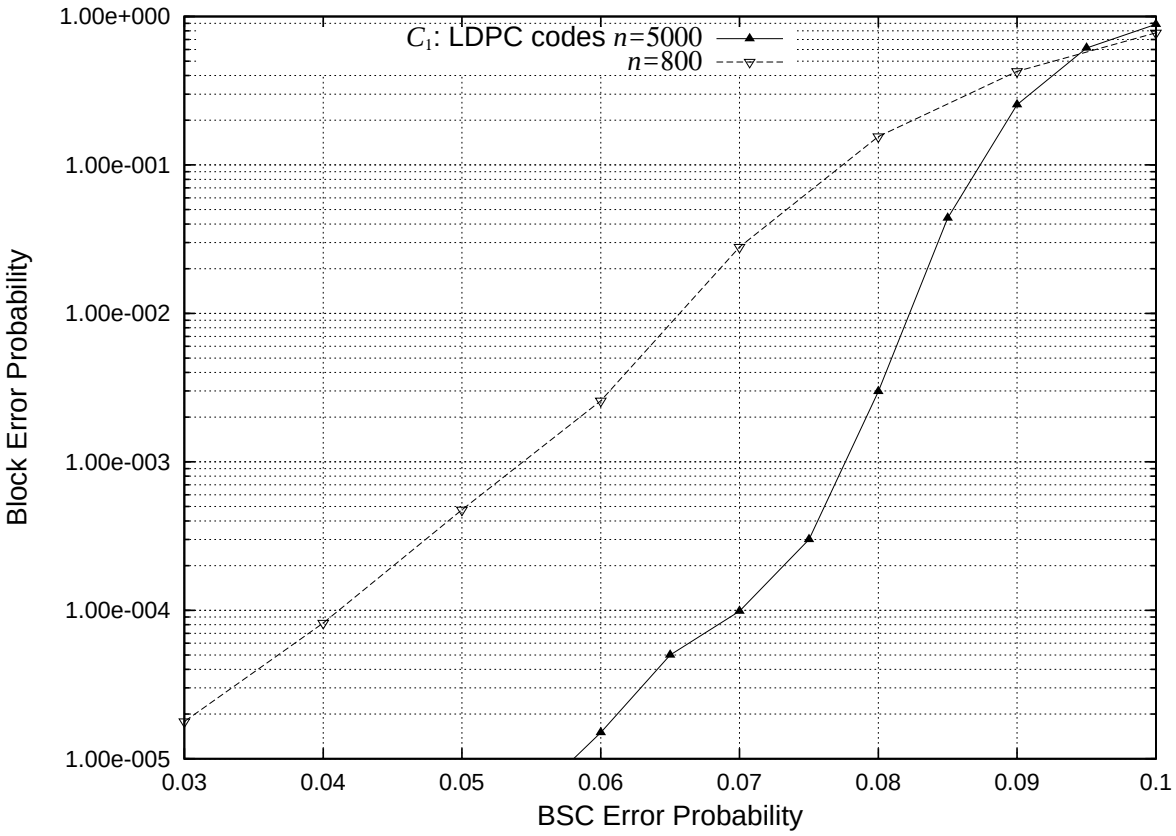

図 3.2: LDPC 符号による復号性能実験結果

復号実験結果を図 3.2 に示す．横軸が 2 元対称通信路の誤り率，縦軸がブロック誤り率である．符号長約 5000 の LDPC 符号は，7 %の雑音に対してもブロック誤り率約 10^{-4} の復号性能を示しており，イヴの相互情報量を十分小さく，CSS 符号の符号化率にも依存するが，最悪の場合でも情報量を 1 以下に抑えることが可能である．この LDPC 符号に限ったことではないが，符号長が長くなればなるほど，復号性能は向上している．

3.2 CSS 符号構成

符号 C_1 のパリティ検査行列 $H_1 = [H^{(p)} \mid H_W^{(d)}]$，符号 C_2 のパリティ検査行列 H_2 において，

$$H_2 = \begin{bmatrix} H_1 \\ H_s \end{bmatrix}$$

の関係が成り立つ.

3.2.1 符号構成の困難性

行列 H_s の構成方法が $C_2^{\perp}$ の復号性能を決定する．しかし，$C_2^{\perp}/C_1^{\perp}$ の復号性能の評価は困難である．その原因としては，$C_2^{\perp}/C_1^{\perp}$ は理論的に復号性能が良ければ十分であるため，最尤復号法等による理論的な復号性能評価で良い．しかし，最尤復号法は計算量の問題で，実際に復号して評価を行うのは困難である．また，符号における，符号語の重み分布が既知であれば復号性能の評価は可能となるが，具体的な符号において，重み分布が知られている符号は一部の符号のみである．また，重み分布の計算も一般に困難である．LDPC 符号との組み合わせで用いられる sum-product 復号法は，サイクルが存在しない場合には最大事後確率復号法と等価である．したがって，Tanner グラフ（すなわちパリティ検査行列）が疎であり，Tanner グラフ中にサイクル数が少ないとき効果を発揮する復号法である．LDPC 符号の場合，双対符号のパリティ検査行列は一般に高密度となり，Tanner グラフ中にサイクルが避けられない状態となる．つまり，sum-product 復号法を用いても，最尤復号法での復号性能評価に比べて大きく劣化し，復号性能評価に不適である．これらの要因から $C_2^{\perp}/C_1^{\perp}$ の復号性能評価の困難となり，C_1, C_2 の符号構成を困難にしている起因と言える．したがって，C_1 に良い復号特性を持つ非正則 LDPC 符号を適用した場合における，$C_2^{\perp}/C_1^{\perp}$ の符号語に対し，実験により考察を与える.

3.3 $C_2^{\perp}/C_1^{\perp}$ の符号語について

線形符号は，最小ハミング重みを持つ符号語のハミング重みが最小距離となる．よって，ハミング重みの小さい符号語が少ない，もしくは存在しないことが良い復号特性を持つ条件となる.

3.3.1 生成される符号語の分類

符号 $C_2^{\perp}/C_1^{\perp}$ の復号性能は，H_2 を生成行列として定義される符号 $C_2^{\perp}$ から，H_1 を生成行列として定義される $C_1^{\perp}$ の符号語を差分した符号の復号性能と同一視できる．つまり，H_1 を生成行列として生成される符号語は，$C_2^{\perp}/C_1^{\perp}$ の復号性能に影響しない．$C_2^{\perp}/C_1^{\perp}$ の復号性能に影響を与える符号語は，以下の 2 種類存在する.

(i) H_s を生成行列として定義される符号語.

(ii) H_1 を生成行列として生成される符号語と H_s を生成行列として生成される符号語との重ね合わせで生成される符号語．ただし，それぞれの符号語は，重みは 0 ではない符号語とする．

行列 H_s を生成行列として定義される符号を良い符号に構成するには，良い符号の生成行列を適用すれば良く，実際に構成するのは容易である．つまり，$C_1,\ C_2$ を構成する上で問題となるのは，(ii) の場合の，H_1 を生成行列として生成される符号語と H_s を生成行列として生成される符号語との重ね合わせで生成される符号語である．$C_2^{\perp}/C_1^{\perp}$ の復号性能を良くするためには，この H_1 を生成行列として生成される符号語と H_s を生成行列として生成される符号語の重ね合わせの符号語に，重みの小さい符号語が存在しないことが望まれる．

3.3.2 実験方法

最小距離等の理論的な考察を与えるのは困難なため，実験を行うことで，重みの小さい符号語に対する評価を行う．本節では，その実験方法について説明する．

まず今回の実験では， H_s として H_1 の双対符号の生成行列を適用した場合を考察する．その理由としては，任意の符号語と直交しているため，H_s と H_1 を生成行列としたときの重ね合わせた符号語に，重みの小さい符号語ができる可能性が低いと推測したためである．

パリティ検査行列 H_1 で定義される LDPC 符号の符号語を $\boldsymbol{u}_l$ ，その LDPC 符号の双対符号の符号語を $\boldsymbol{u}_d$ とすると，重ね合わせた符号語は，$\boldsymbol{u}_l + \boldsymbol{u}_d$ である．$\boldsymbol{u}_a = \boldsymbol{u}_l + \boldsymbol{u}_d$ とすると，2 元であるから $\boldsymbol{u}_l + \boldsymbol{u}_a = \boldsymbol{u}_d$ と変形でき，重ね合わせた符号語の重みが小さいことは，$\boldsymbol{u}_a$ の重みが小さいことを意味する．実験において $\boldsymbol{u}_a$ は雑音に対応し， $\boldsymbol{u}_a$ の重みが小さいことは，雑音が小さいことに対応する．したがって，重ね合わせた符号語に重みの小さい符号語が数多く存在すれば，H_1 による復号を行えば，復号成功の確率が高くなることが容易に予想できる．

以下に実験手順を簡単に説明する．

(i) H_1 で定義される LDPC 符号の双対符号からランダムに選択した，重み 0 ではない符号語 $\boldsymbol{u}_d$ を受信語とする．

(ii) H_1 で定義される LDPC 符号，sum-product 復号法の組み合わせにより，受信語 $\boldsymbol{u}_d$ の復号を実行する．

3.3.3 符号語の実験結果

図 3.2 に示した [17] の LDPC 符号において，符号長約 5000 と約 800 の LDPC 符号の違いは，素数 p の違いだけである．復号計算時間を比較すると，符号長が短い方が，復号にか

かる計算処理時間が短い．したがって，本論文では，符号長約 800 の LDPC 符号を実験に使用する．

実験結果を表 3.1 に示す．復号施行回数は仮定したビット誤り率ごとに，それぞれ約 200 万回復号実験を行った．

sum-product 復号法では復号時に，2 元対称通信路の場合なら通信路のビット誤り率を用いて復号を行う．そのビット誤り率を変化させて実験を行った．本実験において，ビット誤り率は，LDPC 符号の符号語と，双対符号の符号語との異なるビットの割合に対応する．

仮定した誤り率（%）	復号後のビット変化率（%）
1	6.73
2.5	5.14
3.75	3.94
5	3.08
7.5	2.06
10	1.05
15	0.0696

表 3.1: 復号結果

実際に復号を行った結果，復号成功回数は 0 回となった．しかし，仮定したビット誤り率が原因とも考慮できるため，詳細な考察を行う．

復号後のビット変化率，すなわち，受信語と復号後に出力された推定語のビットを比較したときに，ビットの変化の割合を確率で表現したものである．仮定したビット誤り率が，1 %の場合と 7.5 %以上の場合，復号後のビット変化率は，仮定したビット誤り率との誤差が大きい．このことから，1 %の場合は，LDPC 符号の誤り訂正能力から考慮しても，仮定したビット誤り率に符号語が存在する確率は低い．7.5 %以上の場合は，実験では得られていないが，もし存在したとしても，重ね合わせで生成される符号語の重みも大きいため，復号性能への影響は少ない．

仮定したビット誤り率が，2.5 %から 5 %の場合は，復号後のビット変化率と仮定したビット誤り率との誤差が小さい．しかし，符号長約 800 の LDPC 符号は，2.5 %から 5 %の雑音に対しては十分な誤り訂正能力を有する．したがって，LDPC 符号の符号語と，その双対符号の符号語からランダムに選択した符号語の重ね合わせで生成される符号語の重みが，符号長の 2.5 %から 5 %である符号語が存在すれば，復号成功の確率は高い．実験結果から復号は成功していないため，LDPC 符号の符号語と，その双対符号の符号語からランダムに選択した符号語の重ね合わせで生成される符号語の重みが，符号長の 2.5 %から 5 %である符号語が存在する確率は低い．符号長約 5000 の場合の同様な結果が期待される．

以上から，$C_2^{\perp}/C_1^{\perp}$ の復号性能に影響を及ぼす，重みの小さい符号語が存在する確率は低いことが示された．

3.4 提案 CSS 符号構成法 1

実験結果（表 3.1）より，任意の LDPC 符号を用いても，良い復号性能を持つ CSS 符号が構成可能である確率が高いことが示されているため，本節では，LDPC 符号を用いて実際に C_1, C_2 を構成する手法を提案する．以下にその構成方法を記述する．

(1) C_1 に $m_1 \times n$ のパリティ検査行列 H_1 により定義される LDPC 符号を選択する．
(2) $n - m_1$ ビットを無造作に構成する．
(3) $n - m_1$ ビットを情報ビットとし，C_1 の n ビット符号語を m_2 個生成する．
(4) m_2 個の符号語を用いて，$m_2 \times n$ のパリティ検査行列 H_2 を構成する．この H_2 により，$C_2^{\perp}$ を定義する．ただし，情報ビットで生成される H_2 の組織部が疎になるように情報ビットを構成する．

実装の際，状況に応じて C_1 は任意の LDPC 符号を選択可能である．また，C_1 を固定し，C_1 の部分符号 C_2 を無造作に選択すれば，$C_2^{\perp}/C_1^{\perp}$ の復号誤り率が高い確率で低い [51]．したがって本論文でも，情報ビットが 1 である要素を無造作に構成している．また，復号する際を考慮し，H_2 の組織部を疎に構成している．

3.4.1 復号性能評価

イヴの相互情報量の上界を決定するためには，C_1/C_2, $C_2^{\perp}/C_1^{\perp}$ の復号性能を評価する必要がある．復号性能に関し，C_1/C_2 の復号性能は，C_1 に良い復号特性を持つ LDPC 符号を選択可能なため問題とならない．つまり，sum-product 復号法により良い復号特性を示す．問題となるのは $C_2^{\perp}/C_1^{\perp}$ の復号性能を如何に評価するかである．$C_2^{\perp}/C_1^{\perp}$ は理論的に良い復号特性を持てば良いが，具体的な LDPC 符号に対し，最尤復号を行うことは困難であるため，本論文では，計算機による復号性能評価を行う．

上述した CSS 符号構成法は，$C_2^{\perp}$ のパリティ検査行列 H_2 が，一部が疎，一部が密な行列となる．行列が密であれば Tanner グラフに短いサイクルの存在が避けられないため，sum-product 復号法では良い復号性能が期待できない．そのため，復号性能評価法として，sum-product + OSD-i による復号法 [16] を用いた．

3.4.2 sum-product + OSD-iによる復号法

sum-product + OSD-iによる復号法 [16] の手順について簡単に説明する．

(1) 普通に sum-product 復号を行い，sum-product 復号により計算された信頼度情報から，信頼度の高い順に k ビット選択する．
(2) i 以下の全 ${}_nC_k$ 通りの組み合わせビット反転を行う．
(3) その全組み合わせ k ビットをに対し，それぞれ n ビット符号語を生成する．
(4) 生成された全符号語のうち，受信語とのハミング距離が一番小さい符号語を推定語とする．

よって，i を大きくすることで雑音のあるビットパターン全てを符号化可能なため，最尤復号に迫る復号が可能である．また，sum-product 復号法による反復復号を行う際に計算される尤度を用いることで，受信した時点でビット間に信頼度の差がない二元対称通信路においても有用な復号法である．

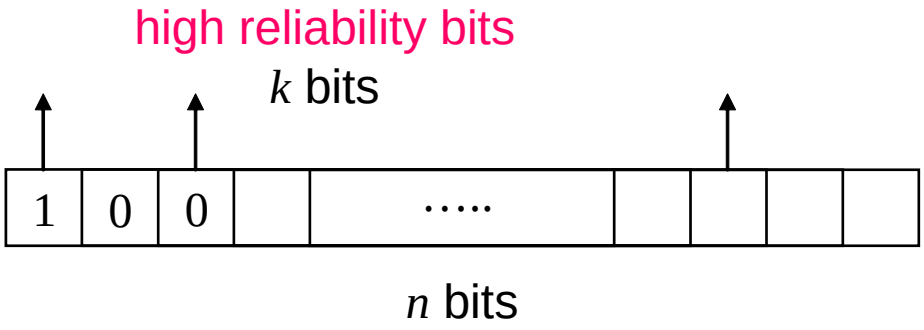

図 3.3: sum-product + OSD-iによる復号法

[16] の実験結果より，sum-product + OSD-iによる復号法も，Tanner グラフの長さが短いサイクルが復号性能劣化につながることが予想されたため，長さ 4 のサイクルを除去する手段として，[23] の Tanner グラフ変換手法を用いた．[23] の変換手法は，パリティ検査行列を一般化パリティ検査行列に変換する方法であり，符号等価に長さ 4 のサイクルを除去可能である．

3.4.3 長さ 4 のサイクルを除去アルゴリズム

本節では，パリティ検査行列 H によって定義される線形符号 C に対し，C に関する Tanner グラフから長さ 4 のサイクルを除去する方法 [23] を，例を用いて説明する．具体例

として，(7, 4, 3) ハミング符号を考える．

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

この H に対し，第 4 ビットと第 7 ビットに関して長さ 4 のサイクルを除去アルゴリズムを適用すると，

$$H' = \left[\begin{array}{ccccccc|c} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{array}\right]$$

となる．第 8 ビットはパンクチャビットである．パンクチャビットは sum-product 復号法による復号時には対数尤度比 0 として復号処理を行う．この変換アルゴリズムにより，長さ 4 のサイクルに関しては，最大事後確率と等価なアルゴリズムを sum-product 復号法にて実現可能である．また，任意のパリティ検査行列に適用可能であり，例として挙げた場合以外でも，同時に複数の長さ 4 のサイクルを除去することも可能である．極端な例として先程のハミング符号と似た例を挙げる．

$$H_s = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

この H_s に対し，第 5 ビットと第 6 ビットと第 7 ビットに関して長さ 4 のサイクルを除去アルゴリズムを適用すると，

$$H'_s = \left[\begin{array}{ccccccc|c} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right]$$

と同時に変形可能である．長さ 4 のサイクルを除去するアルゴリズムは他にも存在するが [55]，復号性能の向上具合が [23] のアルゴリズムの方が良かったため，[23] のサイクル除去アルゴリズムを採用した．

3.4.4 実験結果 1

本節では，計算機を用いて行った復号性能の実験結果を示す．アリスとボブのビット列間の雑音を，二元対称通信路による雑音と仮定できる．そのため，通信路は二元対称通信路を

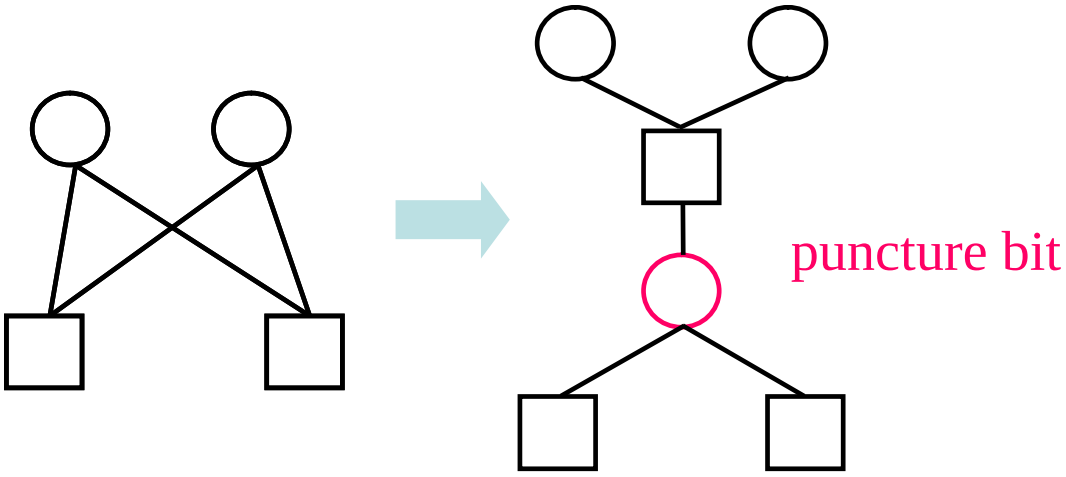

図 3.4: 長さ 4 のサイクル除去の例

仮定し実験を行った．また，量子鍵配送では鍵共有が目的であり，上述したように，正しく鍵を共有できる確率がイブの相互情報量の上界を決定するため，復号性能をブロック復号誤り率で評価する．

3.4.5 実験 1 に使用した LDPC 符号

本実験において，C_1 は符号長 400，列重み，行重みがそれぞれ 3，10 の符号化率 0.7 の正則 LDPC 符号を用いた．この LDPC 符号の内径（一番短いサイクル）は 6 である．復号の際の最大反復回数は 50 回とした．$C_2^{\perp}$ は，パリティ検査行列の組織部の列重みを 2 となるように情報ビットを選択し，200×400 のパリティ検査行列を構成した．符号化率は，一次従属な行ベクトルが存在したため，0.5025 である．つまり，CSS 符号としての符号化率は 0.2025 である．$C_2^{\perp}/C_1^{\perp}$ は，最大反復回数を 500 回として実験した．また，$C_2^{\perp}$ のパリティ検査行列に対し，Tanner グラフ変換処理を施した結果，パリティ検査行列は 2185×2385 の行列に拡張された．sum-product + OSD-i $(i \in \{1, 2\})$ による復号の際の停止条件（[16] における α）は，同じ符号語を 20 回推定語した場合としている．この実験結果を図 3.5 に示す．（図では，都合上 $C_2^{\perp}/C_1^{\perp}$ を C_2^d/C_1^d と表記している．）横軸は二元対称通信路の誤り率，縦軸はブロック復号誤り率である．

3.4.6 考察 1

Tanner グラフのサイクル，パリティ検査行列の行重みが均一でない等の影響で，$C_2^{\perp}/C_1^{\perp}$ を sum-product 復号法で復号した場合の復号性能は，雑音レベルに関わらず良くない．Tanner

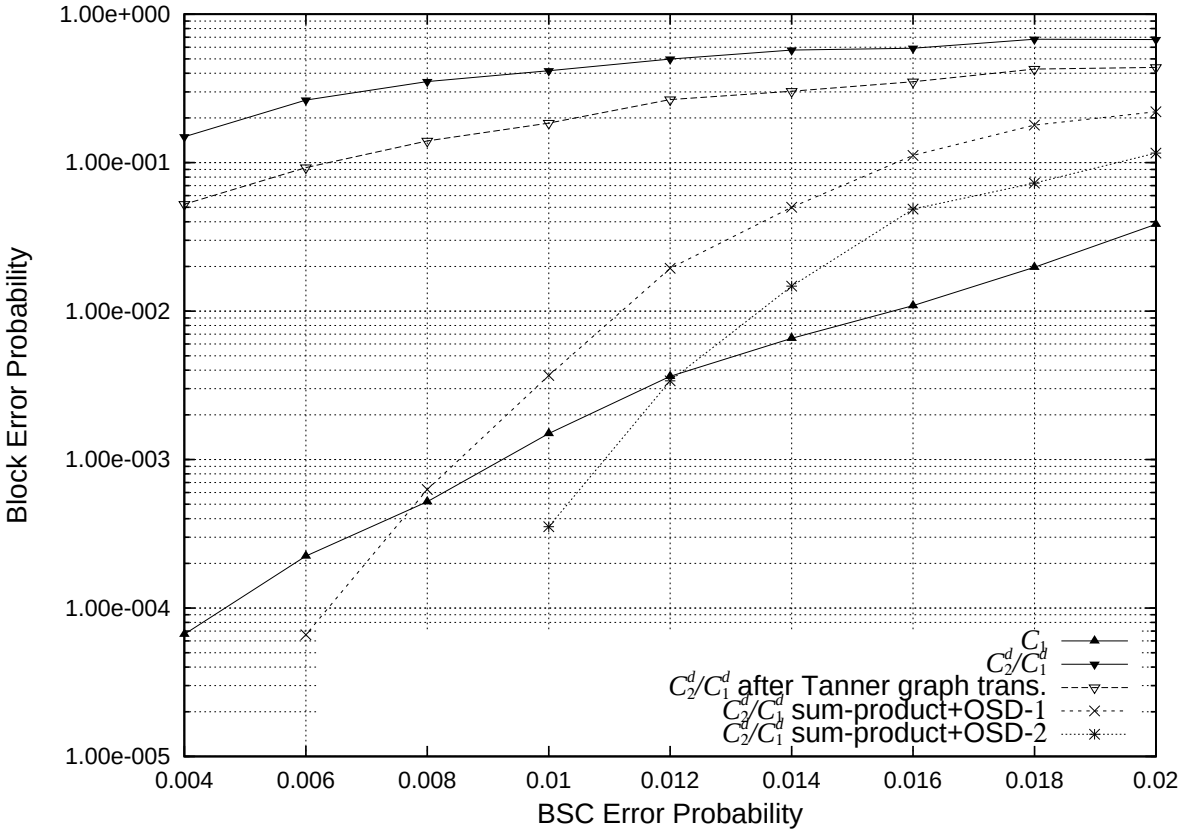

図 3.5: 提案 CSS 符号構成法 1 における C_1, $C_2^{\perp}/C_1^{\perp}$ の復号性能実験結果 1

グラフ変換後 sum-product 復号法で復号した場合も，変換前と比較して若干改善された程度となった．次に，Tanner グラフ変換を施した場合のみ sum-product + OSD-i ($i \in \{1, 2\}$) による復号実験を行った．sum-product + OSD-1 による復号の場合は 1.0 %，sum-product + OSD-2 による復号の場合は 1.2 %以下の雑音に対して，C_1 の LDPC 符号の復号誤り率より低くなった．また，BB84 量子鍵配送プロトコルでは，誤り訂正処理後，アリスとボブの保持しているビット列が異なっていても，C_2 による剰余類にて同じビット列を共有できればかまわない．実験においてこのような復号成功の割合，つまり $C_2^{\perp}$ では復号誤りとなるが，$C_2^{\perp}/C_1^{\perp}$ では復号成功となったのは，復号成功全体の 1〜5 %となった．

実験結果より，二元対称通信路の誤り率が 1 %程度であれば，イブの相互情報量を 1 ビット以下に抑制可能である．また，例えば C_1 を sum-product 復号法で復号し，Tanner グラフのサイクルの影響で符号語を推定できず最大反復回数に至った場合に，sum-product + OSD-i による復号を行えば，C_1 の復号誤り率はさらに低い．図 3.6 より，sum-product + OSD-1 による復号を行えば，C_1 の復号誤り率は，二元対称通信路の誤り率が 2 %において，10^{-3} 程度であり，sum-product 復号法のみの場合に比較して，約 2 倍の雑音に対して同じ復

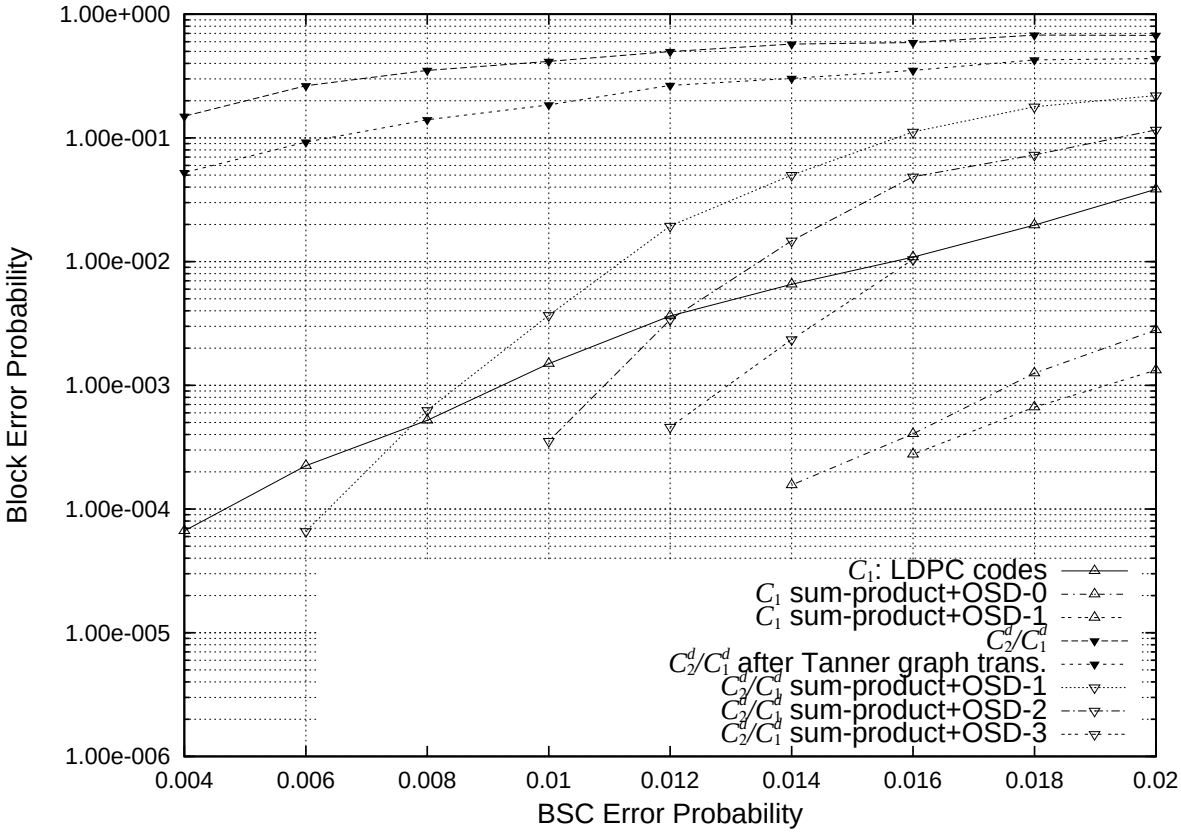

図 3.6: 提案 CSS 符号構成法 1 における C_1, $C_2^{\perp}/C_1^{\perp}$ の復号性能実験結果 2

号誤り率となった．sum-product + OSD-1 と i が 1 程度による復号であれば，実際に復号処理を行うことも十分可能である．$C_2^{\perp}/C_1^{\perp}$ は，理論的に復号性能が良いことを示せれば十分なため，i を 3 以上にて sum-product + OSD-iによる復号を行えば，sum-product + OSD-2 による復号よりも復号誤り率を低くできる．実際，sum-product + OSD-iによる復号を失敗した場合，受信語とのハミング距離が大きい符号語を推定語として失敗している．したがって，i を大きくすることで，$C_2^{\perp}/C_1^{\perp}$ の復号性能の向上を期待できる．つまり，耐雑音性に着目すれば，イブの相互情報量を 1 ビット以下に抑制したまま，符号を変えずにより多くの雑音に耐えることも可能である．

3.4.7 組織部の構成方法について

本論文の実験では，符号長が短い LDPC 符号を C_1 に選択したため，なるべくサイクルを少なくする目的で $C_2^{\perp}$ のパリティ検査行列の組織部の列重みを 2 として構成した．組織部の列重み以外は同条件で，列重みを 3 にしてパリティ検査行列を構成した場合は，列重み 2

の場合と比較して，復号性能が若干劣る結果となった．ただし，符号長が長い場合には，最小距離の観点から，列重みを 3 以上を交えて構成する方が良いと推測される．ただし，パリティ検査行列全体の密度が上がるため，sum-product 復号法には不向きなパリティ検査行列となることに注意しておく．

3.4.8 符号長と符号化率の関係

実装の際，鍵共有の成功確率，及びイブの相互情報量が問題となる．両者を満たす条件が同等であれば，無駄がないシステムと言える．具体的に言えば，符号長及び生成される鍵長が短い場合，イブの相互情報量の抑制は容易となるが，そのレベルで安全とすると，アリスとボブの鍵共有の成功確率は低くなる．

表 3.2 より，符号長がある程度あれば，雑音レベルに関わらず，鍵共有を高い確率で成功とし，イブの情報量を小さく抑制可能となる．したがって，実装面を考慮すれば，数千レベルの符号長が必要であり，一番適している．

符号長 / 符号化率	数百以上 (千以下)	数千以上
高（耐雑音少）	やや実用的	実用的
低（耐雑音多）	非実用的	実用的

表 3.2: 符号長と符号化率の関係

3.4.9 Suffuled BP 復号 (Bit Serial 型 sum-product 復号)

復号性能をさらに向上させる手段として，Suffuled BP 復号 [56] を用いた．(同様な復号法として，Bit Serial 型 sum-product 復号がある [54]．BP 復号法と sum-product 復号法は名称が異なるだけであり，本質的には変わらないものである．)

Bit Serial 型 sum-product 復号は，

Step2: $i = 1,\ 2,\ \cdots,\ M$ の順に，$h_{ij} = 1$ を満たす全ての組 $(i,\ j)$ に対して

$$\alpha_{ij} = \left(\prod_{n \in A(m)\backslash j} \mathrm{sign}(\lambda_n + \beta_{in}) \right) \times f\left(\sum_{n \in A(m)\backslash j} f(|\lambda_n + \beta_{in}|) \right)$$

を計算することで外部値対数比 α_{ij} を求める．

Step3: $i = 1,\ 2,\ \cdots,\ N$ の順に，$h_{ij} = 1$ を満たす全ての組 $(i,\ j)$ に対して

$$\beta_{ij} = \sum_{m \in b(n) \backslash i} \alpha_{mj}$$

を計算する．

この 2 つのステップを，1 ビットずつ行う．つまり，

Step2-3: $i = 1,\ 2,\ \cdots,\ N$ の順に，$h_{ij} = 1$ を満たす全ての組 $(i,\ j)$ に対し，$i = 1,\ 2,\ \cdots,\ M$ の順に，$h_{ij} = 1$ を満たす全ての組 $(i,\ j)$ に対して

$$\alpha_{ij} = \left(\prod_{n \in A(m) \backslash j} \mathrm{sign}(\lambda_n + \beta_{in}) \right) \times f \left(\sum_{n \in A(m) \backslash j} f(|\lambda_n + \beta_{in}|) \right)$$

を計算することで外部値対数比 α_{ij} を求め，

$$\beta_{ij} = \sum_{m \in b(n) \backslash i} \alpha_{mj}$$

を計算する．

したがって，一回の sum-product 復号操作毎に，行重み分だけ計算量は増加するが，信頼度の情報交換は，元の sum-product 復号法より多く行われ，復号性能は向上する．Suffuled BP 復号は計算量を削減する手法も考慮している．また，復号するビットの順序，制御を閾値を用いて復号する Bit Serial 型 sum-product 復号法も存在する．

3.4.10 実験結果 2

本節では，計算機を用いて行った復号性能の実験結果を示す．アリスとボブのビット列間の雑音を，二元対称通信路による雑音と仮定できる．また，量子鍵配送では鍵共有が目的であり，上述したように，正しく鍵を共有できる確率がイブの相互情報量の上界を決定するため，復号性能をブロック復号誤り率で評価する．

3.4.11 実験 2 に用いた LDPC 符号

実験に用いた符号は，符号長 480，$C_1,\ C_2^{\perp}$ の符号化率がそれぞれ 0.8，C_1 は列重み 3 の正則 LDPC 符号を用いた．CSS 符号化率は 0.6 である．復号の際の最大反復回数は，C_1 を 100 回，$C_2^{\perp}/C_1^{\perp}$ を 200 回として実験した．また，$C_2^{\perp}$ のパリティ検査行列に対し，Tanner

グラフ変換処理を施した結果，パリティ検査行列は 967×1351 の行列に拡張された．sum-product(BP) ＋ OSD-2，Shuffled BP ＋ OSD-2 による復号の際の停止条件（[16] における α）は，同じ符号語を 20 回推定語した場合としている．この実験結果を図 3.7 に示す.（以下実験結果の図中では，都合上 $C_2^{\perp}/C_1^{\perp}$ を C_2^d/C_1^d と表記している.）横軸は二元対称通信路の誤り率，縦軸はブロック復号誤り率である．

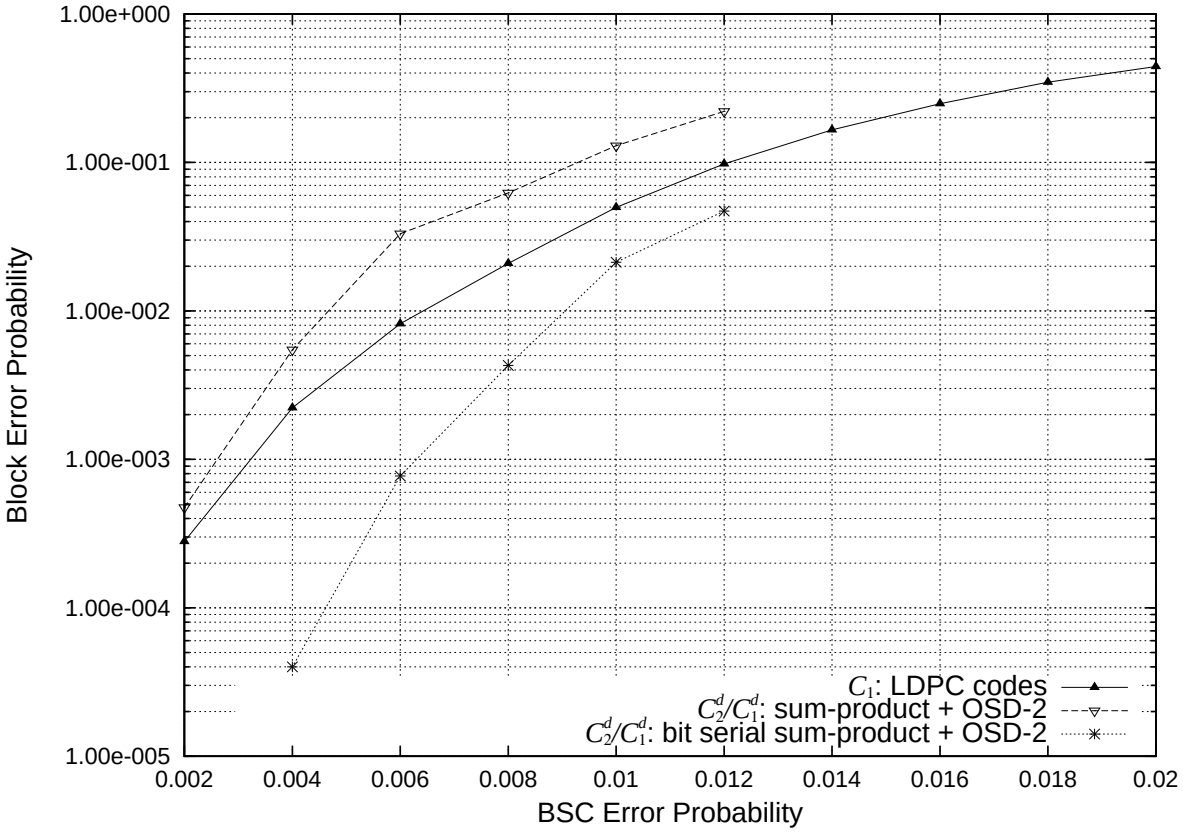

図 3.7: 提案 CSS 符号構成法 1 における C_1, $C_2^{\perp}/C_1^{\perp}$ の復号性能実験結果 3

3.4.12 考察 2

実験結果より，$C_1, C_2^{\perp}$ の符号化率を等しくした場合でも，$C_2^{\perp}/C_1^{\perp}$ の復号性能は，C_1 を sum-product 復号で復号した場合より復号性能が良い．次に，より良い復号性能を持つ LDPC 符号を用いて 2 つの符号を同様に構成する．構成方法もさらに簡単化する．

3.5 提案 CSS 符号構成法 2

まず本節では，CSS 符号構成法 1 をより詳細にした構成法を提案する．

(1) C_1 に $m \times n$ のパリティ検査行列 H_1 により定義される LDPC 符号を選択する．
(2) H_1 から $n-m$ 列を選択する．
(3) 選択した各行に対して $n-m$ ビットを情報ビットとし，C_1 の n ビット符号語を m 個生成する．
(4) m 個の符号語を用いて，$m \times n$ のパリティ検査行列 H_2 を構成する．この H_2 により，$C_2^{\perp}$ を定義する．

手順の通り，H_1 の一部を H_2 に適用させて構成している．したがって，構成の難易度としては，CSS 符号構成法 1 より容易になっている．手順 (2) で選択している列の決定方法によっては，符号の復号性能に差が生じる．一般にパリティ検査行列の密度が高いほど，符号の性能良い [43]．しかし，$C_2^{\perp}/C_1^{\perp}$ の復号性能が C_1 の復号性能より良くなれば十分なため，H_1 の列重みが小さい列を $n-m$ 列を選択した．また，この構成法では，それぞれの符号の符号化率は C_1, $C_2^{\perp}$ 共に等しくなる．したがって，その符号化率を r とすると，CSS 符号としての符号化率は，$r-(1-r)=2r-1$ となり，生成される鍵サイズは，$n(2r-1)$ ビットである．

3.5.1 実験結果 3

符号長が数千の良い復号特性を持つ非正則 LDPC 符号を用いる場合，符号長数百の正則 LDPC 符号を用いた場合と比較し，2 点問題が生じる．まず 1 点目は，良い復号特性を持つ非正則 LDPC 符号，つまり符号語の重みが全体的に見れば大きい符号を用いるため，符号 $C_2^{\perp}$ のパリティ検査行列 H_2 は，正則 LDPC 符号を用いた場合と比較して密になる．つまり，sum-product 復号法にさらに不向きな構造となる．もう 1 点は，符号長が大きくなることで，OSD による復号が最尤復号と差が生じてくる．これは，OSD による復号が最尤復号の一部を行っているためであり，計算量の問題で符号長が大きい場合どうしても差が生じてしまう．本論文では，以上の 2 点を考慮し，復号性能評価を行う．

定理 3.1 二元消失通信路では，雑音に対しパリティ検査行列を変形すれば，sum-product 復号で最尤復号が可能である．

証明 二元消失通信路では，sum-product 復号による反復復号で復号失敗するのは，消失したビットの集合が停止集合となるときである．Tanner グラフに換言すれば，消失している変数ノードに連結しているチェックノード全てが消失している変数ノードに 2 つ以上連結し

ているときである [11]．消失している変数ビットに対し，パリティ検査行列の変換を行う場合，消失したビットで符号語が構成されるビット以外，つまりその部分の行列のランクが列数と等しい場合はチェックノードとの連結を 1 に変換可能である．また消失したビットで符号語が構成されるビットは最尤復号でも復号できない．以上から， 雑音に対しパリティ検査行列を変形すれば，sum-product 復号で最尤復号が可能であることが示された． ■

定理 3.1 より，二元消失通信路ではパリティ検査行列の変形で最尤復号が可能である．したがって，二元対称通信路ではあるが，本実験でも雑音に対してパリティ検査行列の変形を行うことで最尤復号に近づけ，復号性能評価を行う．具体的に言えば，雑音のあるビットに対し，短いサイクルに影響されにくい状況，停止集合が小さくなるような状況にパリティ検査行列を変形する．そして，最尤復号が行いやすいようにパリティ検査行列を変形したあとで，sum-product 復号を行う．また，密な部分に対しては，[23] の Tanner グラフ変換を施した．つまり，密の部分の影響を抑えるためである．

雑音のあるビットを考慮しない場合でも，パリティ検査行列の密な部分に雑音があれば sum-product 復号法は，定理 3.1 内でも記述した停止集合の影響から全く符号語を推定できずに最大反復回数に至ってしまうため，密な部分と疎な部分が混合した多数のパリティ検査行列で同時に sum-product 復号法により復号を行い，符号語を推定できた符号語を選択すれば良い．

定理 3.1 の具体例として，(7, 4, 3) ハミング符号を考える．

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

5, 6, 7 ビットが消失した場合を考慮すると，sum-product 復号法では復号できない．まず行列を次のように変形してみる．変形は単純な行加算である．

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

このように変形すれば，5, 6, 7 ビットが消失した場合でも，sum-product 復号法で復号可能となった．つまり，消失したビットをパリティビットの部分に変更できれば，復号可能となる．

3.5.2 実験 3 に用いた非正則 LDPC 符号

実験に用いた非正則 LDPC 符号は，素数 $p = 59$，符号長 2183，C_1, $C_2^{\perp}$ の符号化率がそれぞれ約 0.8(0.783)，最大反復回数を，C_1 は 100 回，$C_2^{\perp}/C_1^{\perp}$ は 256 回として実験を行った．

実験結果 1 と同じ符号化率であるが，復号性能は実験結果 1 の LDPC 符号と比較してかなり良い復号実験結果を示している．この 2 つの符号の実験結果を図 3.8 に示す．非正則化に用いたマスク行列 W_H を以下に示す．

$$W_H = \begin{bmatrix}
0&0&0&1&0&1&0&1&0&1&1&0&0&0&1&1&1&0&1&0&1&1&0&1&1&0&1&0&1\\
0&0&1&0&0&1&0&0&1&1&0&1&1&1&1&1&1&0&1&1&0&1&0&0&1&0&1&1&0\\
1&0&0&0&1&0&1&0&0&1&0&1&0&0&1&1&0&1&1&1&0&1&1&0&1&1&1&1&0\\
0&1&1&0&0&0&1&0&0&1&1&0&1&0&0&1&1&1&0&1&1&0&1&1&0&1&0&0&1\\
1&0&0&1&1&0&0&1&0&0&1&1&1&1&1&0&1&0&0&0&1&0&1&0&1&1&1&1&0\\
0&1&0&0&1&0&1&0&0&0&1&1&0&1&0&1&0&1&1&1&1&1&0&1&0&1&1&0&1\\
0&1&0&1&0&0&0&1&1&1&0&1&1&1&0&0&0&1&1&1&0&1&1&1&0&0&0&1&1\\
1&0&1&0&0&1&0&0&1&0&1&0&1&1&1&0&1&1&0&0&1&0&1&1&1&1&0&1&1
\end{bmatrix}$$

また，高雑音に対し復号可能な低符号化率の場合における実験も行った．

素数 $p = 89$，符号長 7832，$C_1, C_2^{\perp}$ の符号化率がそれぞれ約 0.55(0.5454)，最大反復回数を，C_1 は 200 回，非正則化に用いたマスク行列 W_L を以下に示す．

$$W_L = \begin{bmatrix}
0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&1&0&0&0&1&0&0&0&0&0&0&1\\
0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&1&0&1&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0&0\\
0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
1&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&1&0&0&0&0&0&0&1&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{bmatrix}$$

$C_2^{\perp}/C_1^{\perp}$ は最大反復回数を 512 回として実験を行った．この 2 つの符号の実験結果を図 3.9 に示す．C_1 はともに [17] の非正則 LDPC 符号を用いている．

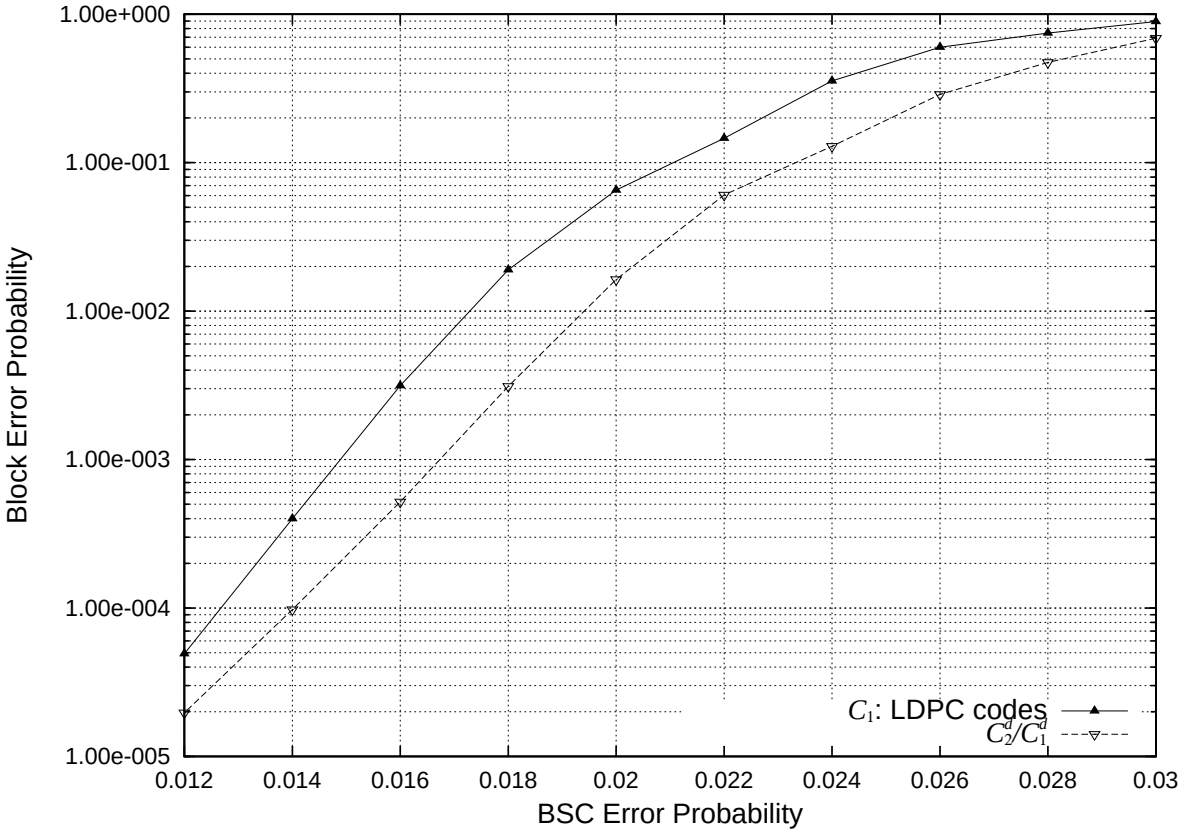

図 3.8: 提案 CSS 符号構成法 2 における C_1, $C_2^{\perp}/C_1^{\perp}$ の復号性能実験結果 1

3.5.3 考察 3

実験結果 2 より，CSS 符号構成法 2 で構成した場合，符号長，符号化率に関わらず $C_2^{\perp}/C_1^{\perp}$ の復号性能は，C_1 を sum-product 復号で復号した場合より復号性能が良い.. したがって，量子通信路の雑音レベルに応じて適切な LDPC 符号を決定し，その選択した LDPC 符号のパリティ検査行列の構造の一部をそのまま用いて $C_2^{\perp}$ のパリティ検査行列を構成するだけで良い．符号化率約 0.55 の場合，6 %程度の雑音に対しても，イブの相互情報量を 1 ビット以下に抑制可能である．また，BB84 量子鍵配送プロトコルでは，誤り訂正処理後，アリスとボブの保持しているビット列が異なっていても，C_2 による剰余類にて同じビット列，つまり鍵を共有できればかまわない．符号長 7832 の実験において，このような復号成功の割合，つまり $C_2^{\perp}$ では復号誤りとなるが，$C_2^{\perp}/C_1^{\perp}$ では復号成功となったのは，最大で復号成功全体の約 85 %以上になった（表 3.3）．傾向としては，雑音レベルが高いほど，このような状況が起こりやすかった．符号長 7832 の実験結果で多かった理由としては，まず雑音レベルが高いことで誤復号の復号領域に受信語が入りやすい，C_1 の行重みが小さいため $C_1^{\perp}$ に小さい符号語が多く存在することが理由として挙げられる．

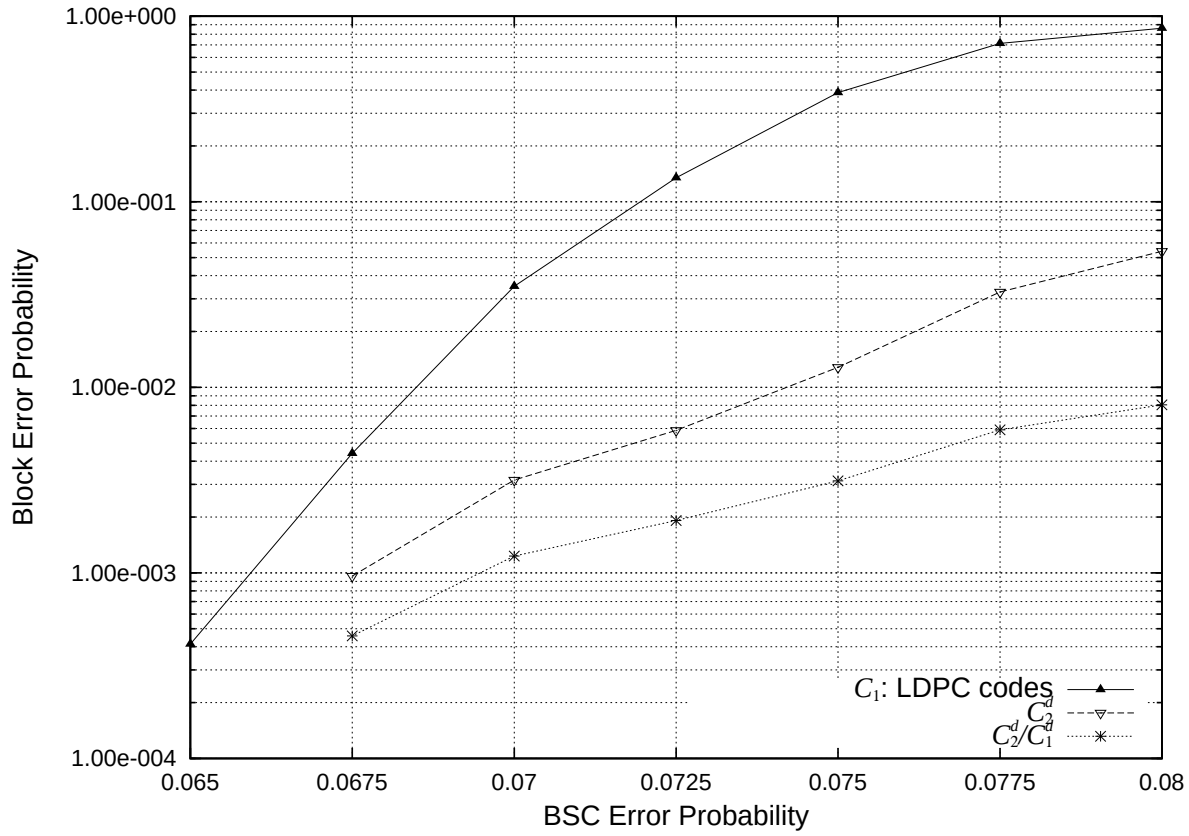

図 3.9: 提案 CSS 符号構成法 2 における C_1, $C_2^{\perp}/C_1^{\perp}$ の復号性能実験結果 2

3.6 様々な符号化率による非正則 LDPC 符号の復号性能評価

前節の実験結果より，$C_2^{\perp}/C_1^{\perp}$ の復号性能は C_1 の復号性能より良い．したがって，C_1 に用いる LDPC 符号の復号性能によりイヴの相互情報量は抑制される．

本節では [17] の非正則 LDPC 符号における符号長約 5000 において，様々な符号化率の状況での復号性能実験を行った．素数はそれぞれ p=73, 139, 127, 73，符号化率がそれぞれ約 0.82，0.75，0.67，0.55 である．最大反復回数を 100 回として実験を行った．この LDPC 符号の実験結果を図 3.10 に示す．非正則化に用いたマスク行列を以下の行列である．

$$W_{0.82} = \begin{bmatrix}
1000001000000000100010010000100100001000110011010110111 1 \\
1010000001000001000100000010001100010010001110101111011 0 \\
0000010010001000000100100010010001001000111011010101101 1 \\
0100010011000001000010000011000100010010001101101010100 1 \\
1001000010001001000100010001000000001000101001011101011 0 \\
0101001000001000100011000100010001000100011101000011101 1 \\
0000010101010010001000010100100000100000100101011101010 1 \\
0010000100010100110001001001000001000110011000101010111 0 \\
0000100000100110000000100100001010100001001110011110110 1 \\
0100001000100100011001000000100010001000010011100111101 0 \\
0001100000010000001000001000011000100001001111111001011 1 \\
0010100100100010010000101000000010000100010100111111110 1
\end{bmatrix}$$

二元対称通信路の雑音（%）	$C_2^{\perp}/C_1^{\perp}$ で補えた割合（%）
8.0	85.8
7.75	82.4
7.5	75.8
7.25	67.5
7.0	61.1
6.75	52.4

表 3.3: 復号成功における $C_2^{\perp}/C_1^{\perp}$ で補えた割合

$$W_{0.75} = \begin{bmatrix}
1&0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&1&0&0&0&1&1&1&1&1&1&1&1&1&1\\
0&1&0&1&0&0&1&0&0&1&1&0&0&0&0&0&0&1&0&0&0&1&1&1&1&1&1&1&1&1\\
0&0&1&0&1&0&0&1&0&0&0&0&0&1&0&1&0&0&1&0&0&1&1&1&1&1&1&1&1&1\\
1&0&0&0&1&0&0&0&0&0&0&0&1&0&0&1&0&1&1&0&0&1&1&1&1&1&1&1&1&1\\
0&0&1&0&0&0&1&0&0&1&0&0&1&0&0&0&0&0&1&0&1&1&1&1&1&1&1&1&1&1\\
1&0&0&1&0&0&0&0&1&0&0&1&1&0&0&0&1&0&0&0&1&1&1&1&1&1&1&1&1&1\\
0&0&1&0&0&1&0&1&0&0&1&0&0&0&1&0&0&0&0&1&0&1&1&1&1&1&1&1&1&1\\
0&1&0&0&1&1&0&0&0&0&0&1&0&1&0&0&1&0&0&0&0&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&1&0&1&1&0&1&0&0&0&1&0&0&0&0&1&0&1&1&1&1&1&1&1&1&1\\
0&1&0&1&0&0&1&0&0&1&0&0&0&1&0&1&0&1&0&1&0&1&1&1&1&1&1&1&1&1
\end{bmatrix}$$

$$W_{0.67} = \begin{bmatrix}
1&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&1&0&1&1&1&1&1&0\\
0&1&0&1&0&0&1&0&0&0&1&0&0&0&0&0&0&1&1&0&1&1&1&1\\
0&0&0&0&1&0&0&1&0&0&0&0&0&1&0&1&0&1&1&1&1&0&1&1\\
1&0&0&0&0&1&0&0&0&0&0&0&1&0&0&1&0&1&0&1&1&1&1&1\\
0&0&1&0&0&0&1&0&0&1&0&0&1&0&0&0&0&1&1&1&0&1&1&1\\
1&0&0&1&0&0&0&0&1&0&0&1&0&0&0&0&1&0&1&1&1&1&0&1\\
0&0&1&0&0&1&0&0&0&0&1&0&0&0&1&0&0&1&1&1&1&0&1&1\\
0&1&0&0&1&1&0&0&0&0&0&0&0&1&0&0&1&1&1&1&1&1&1&0\\
0&0&0&0&0&0&0&1&1&0&1&0&0&0&1&0&0&1&1&0&1&1&1&1\\
0&0&0&1&0&0&0&0&0&1&0&0&0&1&0&1&0&1&1&1&0&1&1&1\\
0&1&0&0&0&0&1&0&0&0&0&1&0&0&1&0&0&1&0&1&1&1&1&1\\
0&0&1&0&1&0&0&1&0&1&0&0&1&0&0&0&0&1&1&1&1&1&0&1
\end{bmatrix}$$

$$W_{0.55} = \begin{bmatrix}
0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&1&0&0&0&0&0&1\\
0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&1&0&0&0&1&0&1&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&1&0\\
1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&1&0\\
0&0&1&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&1&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0\\
1&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1\\
0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&1&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&1&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&1&1&0&0&0&0&0&1&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0\\
1&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&1&1&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&1&0&0&1&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0\\
0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&1&0&0&1&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&1&0&0&1&0&0\\
0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&1&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&1&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0\\
1&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&1&0&0&0&1&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1
\end{bmatrix}$$

3.6.1 考察

量子通信路の場合，現状では最低でも 3 %程度の雑音は不可避である．実験結果（3.10）より，符号化率 2/3 前後が適切であると考えられる．符号化率 2/3 であれば，CSS 符号としての符号化率は 1/3 となる．ただし，復号性能に関してはまだ改善の以下の余地が存在する．

- マスク行列を設計する際，用いる最大次数すなわち最大列重みを大きくする．
- マスク行列を設計する際，マスク行列のサイズを大きくする．
- パラメータである素数 p を大きく設定すなわち符号長を大きくする．

符号長を大きくする以外は，復号性能と設計の難易度，実装の容易さがトレードオフの関係になっている．また，ブロック復号誤り率 10^{-5} 付近において，曲線がゆるやかになっている領域が存在するが，これは次数 3 すなわち列重み 3 の分布を用いて設計したためであり，エラーフロア領域と言われている．一方，急激に誤り率が下がる曲線を描いている領域をウォーターフロー領域と呼ぶ．このエラーフロア領域とウォーターフロー領域もまたトレードオフになっており，ウォーターフロー領域を優先する場合には，小さい次数すなわち列重

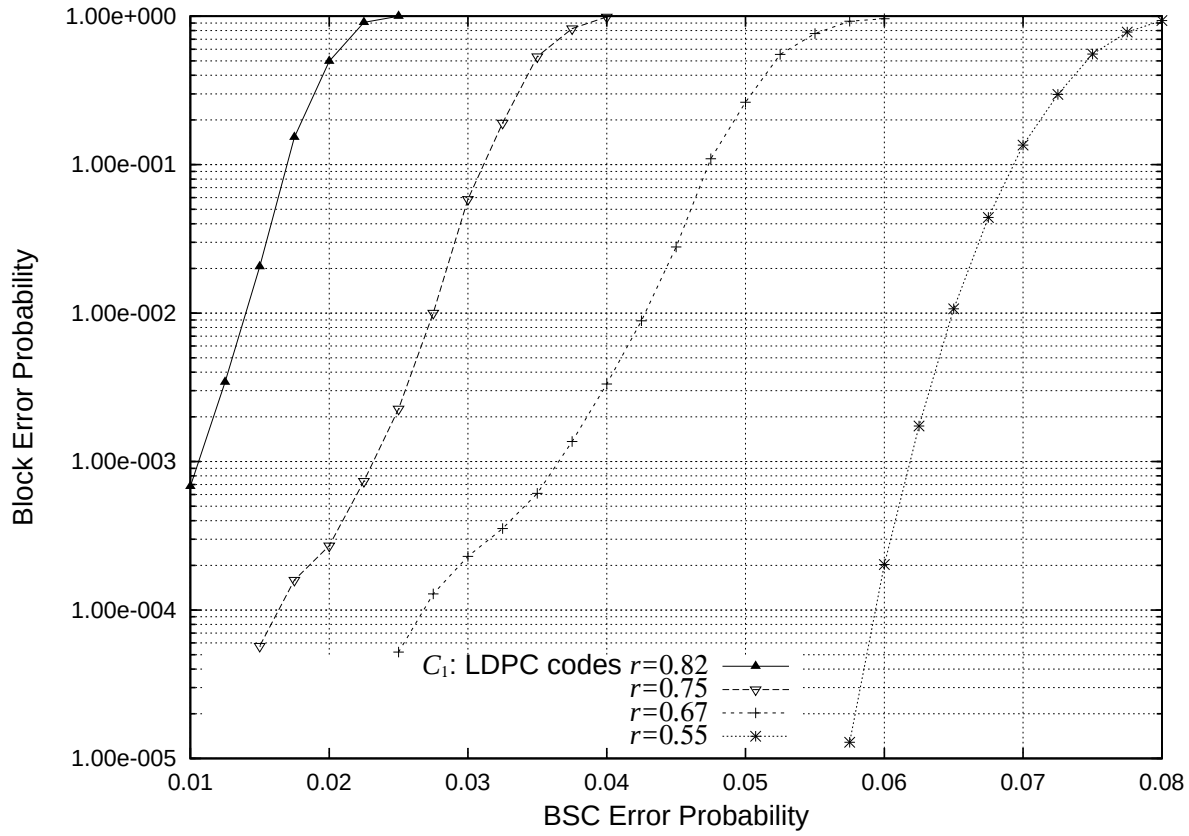

図 3.10: 符号長約 5000 における非正則 LDPC 符号の復号性能実験結果

みを多めにパリティ検査行列を構成すれば良く，エラーフロア領域を優先する場合には，大きい次数すなわち列重みを多めにパリティ検査行列を構成すれば良い．

本章では LDPC 符号を用いて C_1, C_2 を構成する，実用面でより柔軟で詳細な手法を提案した．したがって，量子通信路の雑音レベルに応じて符号長 n，符号化率 r の LDPC 符号を決定するだけで，一意に符号化率 $2r-1$ の CSS 符号が構成可能となった．符号長に関して言えば，数千以上であっても，良い CSS 符号を構成可能である．つまり，雑音レベルに応じて，符号を柔軟に構成可能となる．

第4章

結論

情報理論的に安全性が保障されている One-Time Pad 暗号の鍵配送の問題を解決する手段として，BB84 量子鍵配送プロトコルは有用である．雑音のある量子通信路においては，量子誤り訂正符号である CSS 符号を用いた誤り訂正及び秘匿性増強が有用であり，無条件安全性が保障されている．しかし，実際に符号構成するのは困難となっており，古典処理における BB84 プロトコルの中で一番の問題点となっている．

本論文では，次世代誤り訂正符号としても期待されており，実用的な sum-product 復号法にて Shannon 限界に迫る復号特性を持つ LDPC 符号を用いて，CSS 符号の 2 つの古典線形符号 C_1, C_2 を構成する手法を提案した．符号 C_1，C_2 は CSS 符号より $C_1 \supset C_2$ が必要条件となる．符号 C_1 は，$C_1 \supset C_1^{\perp}$ の場合と $C_1 \not\supset C_1^{\perp}$ の 2 通り存在するが，前者は 2 章でも説明したように，LDPC 符号が sum-product 復号法にて良い復号特性を持つ条件と，$C_1 \supset C_1^{\perp}$ となる条件が相反する関係になるため，LDPC 符号の良さを十分に発揮できる構成法ではない．一方，$C_1 \not\supset C_1^{\perp}$ となる場合は，$C_1 \supset C_2$ の条件を満たすだけで良いため，任意の LDPC 符号を C_1 に使用可能であり，LDPC 符号の良い復号性能を最大限に生かした符号構成が可能である．また，実装面を考慮し，符号化が容易に行えるような LDPC 符号を選択可能であり，符号構成に不必要な条件が存在しないため符号構成の自由度が大きい．

量子通信路では，通信路による雑音だけでなく，盗聴による雑音も含まれる．よって，様々な状況下でも対応可能な，自由度の大きい符号構成法が望ましい．本論文では，実用性の高い状況下，つまり適した符号長の選択，良い復号特性を持つ非正則 LDPC 符号を選択し，実際に 2 つの線形符号 C_1，C_2 を構成した．また，構成方法も，LDPC 符号を選択するだけで一意に決定される構成法になっており，符号化率 r の LDPC 符号を決定するだけで，符号化率 $2r-1$ の CSS 符号が構成される．したがって，誤り訂正処理に用いる LDPC 符号 C_1 を決定するだけで，容易に符号構成が可能であり，実装にも適した構成方法である．

参考文献

[6] S. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Trans. on Inf. Theory*, vol. 46, pp. 325-343, 2000.

[7] A. Amraoui and R. Urbanke, "LpdcOpt," EPFL, Switzerland, renewal of at any time. Available at `http://lthcwww.epfl.ch/research/ldpcopt/`.

[8] C. H. Bennett and G. Brassard, "Quantum cryptography: public key distribution and coin tossing," *Proc. of IEEE Int. Conf. on Comp., Syst. and Sig. Proc.*, pp. 175-179, Bangalore, India, Dec. 10-12, 1984.

[9] A. R. Calderbank and P. W. Shor, "Good quantum error correcting codes exist," *Phys. Rev. A*, vol. 54, pp. 1098-1105, 1996, *arXiv e-Print* quant-ph/9512032; A. M. Steane, "Multiple particle interference and error correction," *Proc. of the Royal Soc. of London A*, vol. 452, pp. 2551-2577, 1996, *arXiv e-Print* quant-ph/9601029.

[10] L. Chen, I. Djurdjevic, J. Xu, S. Lin and K. Abdel-Ghaffar,"Construction of quasi-cyclic LDPC codes based on the minimum weight codewords of Reed-Solomon codes," *Proc. of 2004 IEEE ISIT*, pp. 239, Chicago, IL USA, June 27-July 2, 2004.

[11] C. Di, D. Proietti, T. Richardson, E. Telatar and R. Urbanke,"Finite length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. IT-48, no. 6, pp. 1570-1579, Jun. 2002.

[12] W. Diffie and M. E. Hellman, "New directions in dryptography," *IEEE Trans. on Inf. Theory*, vol. IT-22, no. 6, pp. 644-654, Nov. 1976.

[13] E. Eleftheriou and S. Ölçer, "G.gen: LDPC codes for G.dmt.bis and G.lite.bis.," ITU - Telecomm. Standardization Sec., Study Group 15, Temp. Doc. CF-060, Jan. 2001.

[14] J. L. Fan, "Array codes as low-density parity-check codes," *Proc. of the 2nd Int. Symp. on Turbo Codes and Related Topics*, pp. 543-546, Brest, France, Sep. 2000.

[15] M. P. C. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics, " *IEEE Tran. Inf. Theory*, pp.1379-1396, Sept. 1995.

[16] M. P. C. Fossorier, "Iterative reliability-based decoding of low density parity check codes," *IEEE Jour. on Selected Areas in Comm.,* vol. 19, no. 5, pp. 908-917, May 2001.

[17] H. Fujita, M. Ohata and K. Sakaniwa, “An algebraic method for constructing efficiently encodable irregular LDPC codes,” *Proc. of 2005 IEEE ISIT*, pp. 855-859, Adelaide, Australia, Sep. 4-9, 2005; preliminary version in *IEICE Tech. Rep.*, vol. 104, no. 729, pp. 67-72, Mar. 17-18, 2005.

[18] H. Fujita and K. Sakaniwa, “An efficient encoding method for LDPC codes based on cyclic shift,” *Proc.of 2004 IEEE ISIT*, pp. 276, Chicago, IL USA, June 27-July 2, 2004.

[19] R. G. Gallager, “Low-density parity-check codes,” Cambridge, MA: MIT Press, 1963; preliminary version in *IRE Trans. on Inf. Theory*, vol. 8, pp. 21-28, Jan. 1962.

[20] 萩原学, 今井秀樹，“復号誤りを利用した量子暗号の為の CSS 型 LDPC 符号構成,” 第 28 回情報理論とその応用シンポジウム (SITA2005)，vol. 1，pp. 415-418，2005.

[21] 萩原学，今井秀樹, “ 短距離量子暗号システムにおける安全な誤り訂正符号の構成,” 暗号と情報セキュリティシンポジウム 2006 (SCIS2006), 2006.

[22] A. Imamoglu, D. D. Awschalom, G. Burkard, D. P. DiVincenzo, D. Loss, M. Sherwin and A. Small, “Quantum information processing using quantum dot spins and cavity QED,” *Phys. Rev. Lett.*, vol. 83, pp. 4204, 1999.

[23] K. Kasai, T. Shibuya and K. Sakaniwa, “A code-equivalent transfomation removing cycles of length four in Tanner graphs,” *IEICE Tech. Rep.*, vol. 104, no. 302, pp. 25-28, Sep. 2004.

[24] F. R. Kschischang, B. J. Frey and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. on Inf. Theory*, vol. 47, pp. 498-519, Feb. 2001.

[25] H.-K. Lo and H. F. Chau, “Unconditional security of quantum key distribution over arbitrarily long distances,” *Science*, vol. 283, pp. 2050-2056, 1999, *arXiv e-Print* quant-ph/9803006.

[26] H. Lou and J. Garcia-Frias, “Quantum error-correction using codes with low-density generator matrix,” *2005 IEEE 6th Workshop on Signal Proc. Advances in Wireless Comm.*, pp. 1043-1047, June 2005.

[27] B. Lounis and W. E. Moerner, “Single photons on demand from a single molecule at room temperature,” *Nature*, vol. 407, pp. 491-493, 2000.

[28] M. Luby, M. Mitzenmacher, A. Shokrollahi and D. Spielman, “Improved low-density parity-check codes using irregular graphs and belief propagation,” *Proc. of 1998 IEEE ISIT*, pp. 117, Cambridge, MA USA, Aug. 16-21, 2001.

[29] R. Lucas, M. Bossert and M. Breitbach, “On iterative soft-decision decoding of linear binary block codes and product codes,” *IEEE Jour. Selected Areas in Comm.*, vol.16, pp. 276-296, Feb. 1998.

[30] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, pp. 399-431, 1999.

[31] D. J. C. MacKay, G. Mitchison and P. L. McFadden, "Sparse-graph codes for quantum error-correction," *IEEE Trans. on Inf. Theory*, vol. 50, no. 10, Oct. 2004, *arXiv e-Print* quant-ph/0304161.

[32] G. A. Margulis, "Explicit constructions of graphs without short cycles and low-density codes," Combinatorica, vol. 2, no. 1, pp. 71-78, 1982.

[33] W. Matsumoto and H. Imai,"A study on rate-compatible LDPC codes," *Proc. of 2004 ISITA*, pp. 529-534, Parma, Italy, Oct. 10-13, 2004.

[34] D. Mayers, "Unconditional security in quantum cryptography," *Jour. of the ACM*, vol. 48, no. 3, pp. 351-406, May 2001, *arXiv e-Print* quant-ph/9802025; preliminary version in D. Mayers, "Quantum key distribution and string oblivous transfer in noisy channels," *Adv. in Cryptology - Proc. of Crypto 1996*, pp. 343-357.

[35] P. Michler, A. Kiraz, C. Becher, W. V. Schoenfeld, P. M. Petroff, Lidong Zhang, E. Hu and A. Imamoglu, "A quantum dot single-photon turnstile device," *Science*, vol. 290, pp. 2282, 2000.

[36] T. Mittelholzer, A. Dholakia and E. Eleftheriou, "Reduced-complexity decoding of low density parity check codes for generalized partial response," *IEEE Trans. on magnetics*, vol. 37, no. 2, pp. 721-728, Mar. 2001.

[37] T. M. N. Ngatched, M. Bossert and A. Fahrner, "Two decoding algorithms for low-density parity-check codes," *2005 IEEE Int. Conf. on Comm.*, vol.1, pp. 637-677, May 16-20, 2005.

[38] D. Stinson, "Cryptography: Theory and Practice," CRC Press, Florida, 1995.

[39] T. Richardson, M. Shokrollahi and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. on Inf. Theory*, vol.47, pp. 619-637, Feb. 2001.

[40] T. Richardson and R. Urbanke, "modern coding theory," EPFL, renewal of at any time. Available at `http://lthcwww.epfl.ch/`.

[41] R. L. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Comm. of the ACM*, vol. 21, no. 2, pp. 120-126, Feb. 1978.

[42] J. Rosenthal and P. O. Vontobel, "Constructions of LDPC codes using Ramanujan graphs and ideas from Margulis," *in Proc. of the 38-th Annual Allerton Conf. on Comm., Control, and Comp.*, pp. 248-257, 2000.

[43] I. Sason and R. Urbanke, "Parity-check density versus performance of binary linear block codes over memoryless symmetric channels," *IEEE Trans.*, vol. 49, no. 7, pp. 1611-1635, July 2003.

[44] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. Jour.*, vol. 28, no. 4, pp.656-715, 1949.

[45] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Jour. on Computing*, vol. 26, no. 5, pp. 1484-1509, 1997.

[46] P. W. Shor and J. Preskill, "Simple proof of security of the BB84 quantum key distribution protocol," *Phys. Rev. Lett.*, vol. 85, no. 2, pp. 441-444, July 2000.

[47] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. on Inf. Theory*, vol. IT-27, pp. 533-547, Sep. 1981.

[48] T. Tian, C. Jones , J. Villasenor and R. Wesel, "Constructions of irregular ldpc codes with low error floors," *Proc. of IEEE ICC 2003*, vol. 5, pp. 3125-3129, May 2003.

[49] N. Varnica and M. P. C. Fossorier, "Belief propagation with information correction: imporved near maximum-likelihood decoding of low-density parity-check codes," *Proc. of 2004 IEEE ISIT*, Chichago, USA, pp. 343, July 2004.

[50] T. Wadayama, "Average coset weight distributions of Gallager's LDPC code ensemble," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2715-2721, July 2005.

[51] S. Watanabe, R. Matsumoto and T. Uyematsu, "Noise tolerance of the BB84 protocol with random privacy," *Proc. of 2005 IEEE ISIT*, Adelaide, Australia, pp. 1013-1017, Sep. 4-9, 2005, *arXiv e-Print* quant-ph/0412070; preliminary version in *Proc. of SITA 2004*, vol. 2, pp. 767-770, Gifu, Dec. 14-17, 2004.

[52] N. Wiberg, "Codes and decoding on general graphs," Ph. D thesis, Dept. Elec. Eng., Linkoping Univ., 1996.

[53] H. Xiao-Yu Hu, E. Eleftheriou and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. on Inf. Theory*, vol. 51, no. 1, pp. 386-398, Jan. 2005.

[54] K. Yamaguchi, Y. Kurihara, M. Yabe and K. Kobayashi, "Studies on bit serial sum-product decoding based on controlling decoding order," *Proc. of ISITA*, Parma, Italy, pp. 1001-1006, 2004.

[55] J. S. Yedidia, J. Chen and M. P. C. Fossorier, "Generating code representations suitable for belief propagation decoding," *Proc. of the 40th Annual Allerton Conf. on Comm., Control, and Comp.*, 2002.

[56] J. Zhang and M. P. C. Fossorier, "Shuffled belief propagation decoding, "*Proc. Asilomar Conf. on Signals, Systems and Comp.*, vol. 1, pp. 8-15, Nov. 2002.

[57] V. V. Zyablov and M. S. Pinsker, "Estimation of the error-correction complexity for Gallager low-density codes," *Problems of Inf. Trans.*, vol. 11, no. 1, pp. 18-28, 1976.

研究業績

口頭発表論文

[1] 大畑真生，萩原学，松浦幹太，今井秀樹, “BB84 量子鍵配送プロトコルの為の双対符号を含む LDPC 符号構成法,” 第 28 回情報理論とその応用シンポジウム (SITA2005)，vol. 1，pp. 411-414，2005.

[2] 大畑真生，萩原学，松浦幹太，今井秀樹, “ BB84 量子鍵配送プロトコルの為の非正則LDPC 符号と双対符号の符号語解析,” 暗号と情報セキュリティシンポジウム 2006 (SCIS2006), 2006.

[3] 大畑真生，松浦幹太，“BB84 量子鍵配送プロトコルの為の LDPC 符号を用いた CSS 符号構成法,” 第 29 回情報理論とその応用シンポジウム (SITA2006)，vol. 2，pp. 675-678，2006.

[4] 大畑真生，松浦幹太，“BB84 量子鍵配送プロトコルの為の任意 LDPC 符号を用いた CSS 符号構成法,” 暗号と情報セキュリティシンポジウム 2007 (SCIS2007), 2007.

論文誌等

[5] M. Ohata and K. Matsuura, “A Construction Method of CSS Codes Using LDPC Codes for BB84 Quantum Key Distribution Protocol,” *IEEE Trans. on Inf. Theory*, 2008.（投稿予定）

謝辞

研究を進めるにあたり，貴重な意見をくださり，そしてさまざまな場面で指導してくださった松浦幹太助教授に心より感謝いたします．研究を進める力や，プレゼンテーション能力，ディスカッション能力など，これから社会で生きていく上でかならず必要となる資質をこの2年間で培うことができたと思っております．また研究室やミーティングで，さまざまな意見や考え方を教えてくださった研究室の皆さま，本当にありがとうございました．様々な場面で私が困難に陥ったとき，みなさまのご意見や心使いは大変ありがたく，困難に挫けずに前進する力をいただいた気がします．

修士として過ごした2年間，皆さまとともに過ごせたことをとてもうれしく思います．今後ともますますのご自愛，ご活躍のほどを心からお祈り申し上げます．