追跡可能性を持つ名簿システムにおける結託耐性と分割耐性

木下 盾[1,3]　上原 哲太郎[2,4]

概要：多くの企業が扱う顧客個人情報は事業活動のためには不可欠のものだが、同時に安全管理の責任が求められるため、利活用できる状態でいかに漏洩を防止するかという課題がある．特に内部不正による漏洩は，事故防止が困難である．そこで，対策として，従業員ごとにパターンを変えて名簿へダミー個人情報を埋め込むことが考えられるが、これも結託攻撃によるダミー個人情報の除去が考えられるため、結託耐性符号の利用が考えられている．本研究では，結託耐性符号を名簿用ダミーデータに利用すると名簿の分割持ち出し等により符号が失われるという問題に対し、誤り訂正符号を活用して攻撃耐性を高めたダミー個人情報埋め込みシステムを提案する．

キーワード：個人情報保護，結託耐性符号，追跡可能性，誤り訂正符号，情報ハイディング

1. はじめに

情報化社会が進む今日，多くの企業の事業活動が情報化によって支えられている．大量の顧客個人情報を管理，分析できるようになったのも情報化の恩恵の一つである．しかし，デジタル処理可能な名簿は複製や持ち出しが容易に行えるため，顧客名簿の漏洩対策が必要となる．

特に，内部不正による漏洩事件は一件あたりの漏洩件数が多い傾向 [1] にある．さらに，漏洩を検知することも難しい．一般的な内部不正の対策として，監視の強化や，名簿の持ち出し禁止などが挙げられるが，これらは業務効率の低下を伴う．そこで，業務効率の低下を伴わない対策が必要になる．

業務効率の低下を伴わない対策として，名簿へのダミー個人情報の埋め込みが考えらえれる．これは，従業員とダウンロードした日時ごとに異なるダミー個人情報を名簿に埋め込む手法である．ここで，ダミー個人情報は実在しない偽の個人情報であり，その住所やメールアドレスは，名簿の所有者である企業が管理しているとする．名簿が漏洩した場合，他の企業がその名簿を利用し DM（ダイレクトメール）や E メール等を送信すると，ダミー個人情報にも DM や E メールが届く．どのダミー個人情報に DM や E メールが届いたかを調べることで，漏洩に関わった従業員及びその名簿をダウンロードした日時を特定できる．しかし，これには複数の名簿を比較し異なる個人情報を削除する結託攻撃を行われる可能性がある．結託攻撃を行われるとダミー個人情報を削除されてしまい，漏洩に関わった従業員を特定できなくなる．そこで，結託耐性符号の利用が考えられる．結託耐性符号とは，冗長な bit を付加することで一定数までの結託による結託攻撃に耐性を持つ符号である．

不正に持ち出された名簿は，名簿業者に売却される可能性がある．名簿業者は入手した名簿を名寄せと業者間での転売を繰り返すことで，名簿の質と量を高める．そして，膨大な量となった名簿から一部を抜き出し，企業等に販売する [2]．すなわち，売却された名簿にダミー個人情報が埋め込まれていても，名簿業者が販売する名簿にダミー個人情報が含まれているとは限らない．ダミー個人情報が販売される名簿に含まれない限り，漏洩を検知することもできない．これには，誤り訂正符号の利用が考えられる．誤り訂正符号はランダムな誤りを検出し訂正することが可能な符号である．誤り訂正符号を利用することで，名簿にダミー個人情報が一部しか含まれていなくても，そのダミー個人情報から含まれていないダミー個人情報を復元することができる．

ダミー個人情報の埋め込みは，削除することが困難なダミー個人情報が埋め込まれていることを組織内部の人間に周知することで，顧客名簿漏洩の心理的抑止力になる．また，企業は名簿が漏洩したことを検知することが可能となる．さらに，名簿がどのような経路で名簿業者から企業へ渡ったか追跡することもできる．このような効果をもたらすには，ダミー個人情報ができるだけ削除されないことが

[1] 立命館大学大学院 情報理工学研究科
[2] 立命館大学 情報理工学部
[3] kinoshita@cysec.cs.ritsumei.ac.jp
[4] uehara@cs.ritsumei.ac.jp

表 1 内部不正による情報漏洩対策

対策	効果	業務効率
手続きの強化	小さい	低下
アクセス権制限	小さい	低下
情報持ち出し制限	小さい	低下
ID・パスワード管理	小さい	不変
ログの記録	大きい	不変
ダミー個人情報埋め込み	大きい	不変

重要である．そのためには，ダミー個人情報の埋め込みに対する名簿への結託攻撃の問題と分割の問題を解決する必要がある．本研究では，結託耐性符号と誤り訂正符号を利用したダミー個人情報を埋め込む名簿システムを作成した．これを用いて符号を利用したダミー個人情報の埋め込みが，どの程度の結託耐性と分割耐性を持つか検証を行う．

2. 研究背景

2.1 内部不正対策

独立行政法人情報処理推進機構が行った経営者，システム管理者に向けた調査 [3] によれば，現在多くの企業では内部不正対策として ID・パスワードの管理やアクセス権の制限が行なわれていることがわかる．しかし，従業員に向けた同調査では，システム内に証拠が残ることやアクセスが監視されていることが内部不正への動機の低下につながるという結果が出ている．すなわち，捕まる可能性を高めることが従業員の内部不正への動機を低下させ，内部不正の対策として効果が高いと考えらる．**表 1** に内部不正による個人情報漏洩への対策と，その有用性，業務効率低下の有無をまとめる．ログの記録はシステム内に証拠が残り，ダミー個人情報の埋め込みはダウンロードした名簿に証拠が残るため，捕まる可能性を高め，内部不正対策として効果が大きいと言える．

2.2 ダミー個人情報埋め込み

ダミー個人情報の埋め込みは，実在しない偽の個人情報（ダミー個人情報）を名簿に埋め込む手法である．これは，ダミー個人情報が埋め込まれていることを組織内部の人間に周知することで，顧客名簿漏洩の心理的抑止力になる．また，企業は漏洩を検知することが可能になる．さらに，持ち出された名簿がどの名簿業者を経由し，他企業へ販売されたかも明確にできる．このような考え方は電子透かしのコンテンツ追跡やソーシャル DRM と類似している．ある情報（画像，動画，文書，名簿などのコンテンツ）をカバーデータとし，そこに識別情報（著作権，購入者情報）を埋め込む．電子透かしにおいて，識別情報を埋め込む際には以下の 3 点が重要となる [4].

- 識別情報を埋め込まれても，カバーデータの品質を損なわない
- カバーデータの編集や圧縮等，各種改変が行われても識別情報が消えない
- 識別情報の読み書きは許可されたもののみが可能であること

ダミー個人情報の埋め込みが一般的な電子透かしと異なる点は，識別情報の埋め込みによりカバーデータの内容が変化する点である．名簿にダミー個人情報が加わることで別の名簿になってしまう．しかし，個人情報は年月とともに引っ越しや出産，結婚など様々な要因で変更される．その変更が必ず顧客名簿に反映されるとは考え難い．このような点を踏まえると，ダミー個人情報が数パーセント埋め込まれても名簿としての利用価値は大きく損なわれることはないと考えられる．また，どれがダミー個人情報であるか利用者（攻撃者）が気づかなければ，ダミー個人情報を取り除かれることはない．ここでは，個人情報を見ただけではダミー個人情報とは判定できないと仮定する．

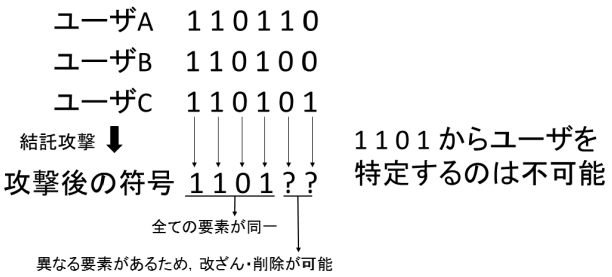

図 1 結託攻撃

表 2 Marking Assumption

仮定	検出した要素への操作	仮定の強さ
narrow-sense	攻撃者の符号要素	弱い
expanded narrow-sense	攻撃者の符号要素か除去	強い
wide-sense	任意の要素	強い
expanded wide-sense	任意の要素か除去	最も強い

2.3 結託耐性符号

結託耐性符号とは，電子透かしに対する有効な攻撃の一つである結託攻撃に耐性のある符号である．結託攻撃とは，異なる識別情報が埋め込まれたカバーデータを比較し，異なる部分を改ざん, 削除することで，埋め込まれた識別情報を読み取り不可能にする攻撃である．結託耐性符号は,Marking Assumption[5] の成立（図 1）を前提に，ある一定数で結託攻撃を行われても，全ての攻撃者を特定することができる．Marking Assumption は攻撃者が結託攻撃を行う際に，符号の要素が異なる部分しか改ざんできないという仮定である．検出した要素を改ざんする際，任意の要素へ改ざんできるのか，攻撃者の符号要素へしか改ざんできないのかで仮定の強さが変わる．また検出した要素を削除できるかどうかでも仮定の強さが変わり，4 つに分類することができる（**表 2**）.

2.3.1 ACC 符号

ACC(Anti-Collusion Code) 符号 [6] は追跡性をもつ結託耐性符号のひとつであり，バイナリで構成される．ACC 符号は BIBD（Blanced Incomplete Block Design）と呼ばれる集合によって構成され，利用する BIBD によって符号の性能が異なる．BIBD は点と呼ばれる有限集合 X と，ブロックと呼ばれる X の部分集合族 B の組 (X, B) である．部分集合族 B は b 個のブロックで構成され，1 つのブロックには k 個の点 x が含まれる．

$$X = \{x_0, x_1, x_2, ..., x_{v-1}\}$$
$$B = \{B_0, B_1, B_2, ..., B_{b-1}\}$$

さらに以下の 2 つの性質 (均衡性) を持つ．

- 任意の点 x に対し,x を含むブロックは r 個存在する
- 任意の異なる 2 つの点を共に含むブロックは λ 個存在する

各パラメータは以下の関係を持ち [4],BIBD は $D(v, k, \lambda)$ と表される．

$$vr = bk$$
$$\lambda v\,(v-1) = bk\,(k-1)$$

Hou らは単位環を利用した BIBD とアフィン平面を利用した BIBD により, 従来よりも性能が良い ACC 符号 [7] を構築した．これらは最も強い Marking Assumption(expanded wide-sense) を満たしている．アフィン平面による ACC 符号は，結託人数が多い結託攻撃にも耐性を持つが，生成できる符号の数（以下符号数）が少ない．単位環による ACC 符号は，結託数ではアフィン平面による ACC 符号に劣るが，符号をより多く生成できる．

本研究では，符号長及び符号数を重視する．結託耐性符号をダミー個人情報の埋め込みパターンに利用すると，符号長が短いほど，ダミー個人情報の埋め込み量が減り，符号数が多いほど，従業員及びダンロード日時の識別数が増える．今回は同じ符号長でより多くの符号を生成できる，単位環による ACC 符号を利用する．

単位環を利用した BIBD で構成される ACC 符号は符号長 $p^3 + 1$bit で, $p^2(p^2 - p + 1)$ 個までの符号を生成でき, p 人までの結託を特定（以下結託耐数）できる（p は素数）．

2.4 名簿の分割

ダミー個人情報の埋め込みにおいて，ダミー個人情報は名簿全体に埋め込まれる．名簿から識別情報を正確に読み取るには，埋め込んだ全てのダミー個人情報を全て取得する必要がある．しかし，デジタル処理可能な名簿は容易に分割や結合ができる．もしダミー個人情報が埋め込まれた名簿が分割されると，識別情報を読み取ることが不可能となる（図 2）．

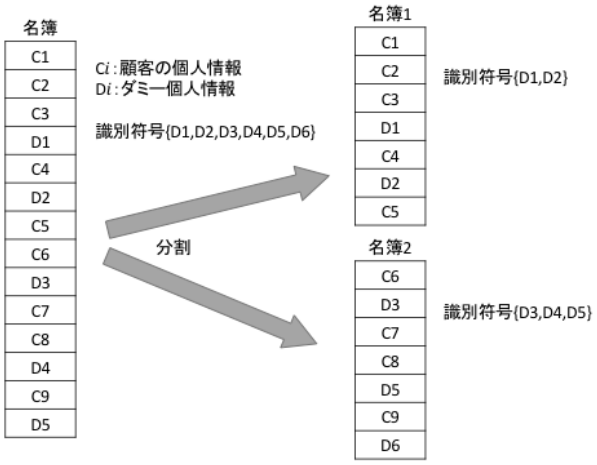

図 2 名簿の分割

2.5 誤り訂正符号

誤り訂正とは，通信路において生じた誤りを受信側で訂正するしくみである．特に，送信側のデータの再送に頼らない誤り訂正を前方誤り訂正 (Forward Error Correction, FEC) と呼ぶ．誤り訂正符号は，情報を符号化する際に冗長性を付加することで前方誤り訂正を実現している符号である．

一般的に誤り訂正符号は，復号誤り率，符号化率，計算量で評価される．しかし，本研究では事前に符号化を行い，復号に関してもリアルタイム処理を必要としていないため，計算量は考慮しない．

結託耐性と分割耐性を両立するには，結託耐性符号を誤り訂正符号で符号化する手法が考えられる．ただし，結託耐性符号を誤り訂正符号で符号化する際に，結託耐性を失ってはいけない．そのためには，入力値がそのまま出力に含まれる符号である必要がある．このような符号を組織的符号と呼ぶ．簡単な組織的符号にパリティチェックが挙げられる．

下記の 2 点を基準に本研究で利用する誤り訂正符号を検討すると，

- 組織的符号である
- できるだけ短い符号長で多くの誤りを訂正できる

ターボ符号 [8] と低密度パリティ検査 (Low Density Parity Check, LDPC) 符号 [9] が挙げられる．これらの符号は，誤り訂正符号の理論的限界であるシャノン限界に非常に近い性能を保持している組織的符号である．

特に，LDPC 符号は符号長や符号化率によってはターボ符号よりも性能が良く，様々な符号長や符号化率の符号をが容易に構成できる [10] ．よって本研究では LDPC 符号を検討する．

2.5.1 パリティチェック

パリティチェックとは，符号化されたバイナリの誤りを検出することができる最も簡単な誤り検出符号である．バイナリの符号にパリティビット（Blok Check Character）

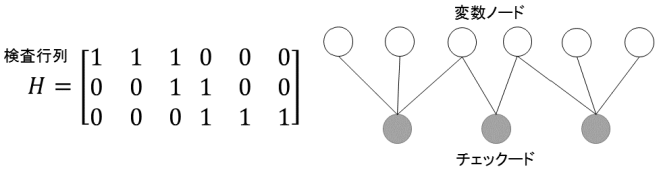

図 3 検査行列とタナーグラフ

を付加することで実現される．あらかじめ偶数か奇数を指定し，符号要素の 1 の数が偶数か奇数かでパリティビットの値（0，1）を決定する．

パリティビットを含めた符号要素に誤りが生じると，要素が 1 の数が変化し，パリティビットが正しくなくなることで誤りが生じたことを検知できる．ただし，誤った要素の位置を特定することはできない．また，複数の要素に誤りが生じ，要素の 1 に数に変化がなかった場合は，誤りを検出することができない．パリティビットは排他的論理和で計算することができる．

パリティビットは一定単位で要素を分割したブロックに 1 対 1 で対応している．1 つのブロックに 1 つのパリティビットを対させる方法を垂直パリティチェックと呼ぶ．複数ブロックの同じ位置にある要素を数個で 1 つのブロックにし，それに 1 つのパリティビットを対応させる方法を水平パリティチェックと呼ぶ．2 つを合わせた垂直水平パリティは，誤り検出だけなく誤り訂正能力も持つ．

2.5.2 LDPC 符号

LDPC 符号は疎な検査行列で定義される線形符号である．疎な行列とは，行列内の非零要素の数が非常に少ない行列のことを指す．検査行列からはタナーグラフと呼ばれる 2 部グラフが導かれる．タナーグラフは変数ノードとチェックノードから構成される．変数ノードは符号語の各要素に対応し，チェックノードは要素間のパリティチェックに対応する．チェックノードに接続する全ての変数ノードの排他的論理和は 0 となる．検査行列はタナーグラフの接続を表し，列が変数ノード，行がチェックノードを意味している（図 3）．

LDPC 符号は，正則と非正則で分類される．全ての列重みと全ての行重みが等しい検査行列で定義される LDPC 符号を正則 LDPC 符号と呼ぶ．正則 LDPC 符号でない LDPC 符号は，非正則 LDPC 符号と呼ばれる．

復号には sum-product 復号を用いる．sum-product 復号は変数ノードとチェックノードの間でメッセージを交互に送信し合い，それを繰り返すことで消失した要素を復元していく復号方式である．このような手法をメッセージパッシング方式と呼ぶ．送信するメッセージは周辺事後確率である．しかし本研究では，入力がバイナリであるため，無記憶二元消失通信路上の sum-product 復号となる．

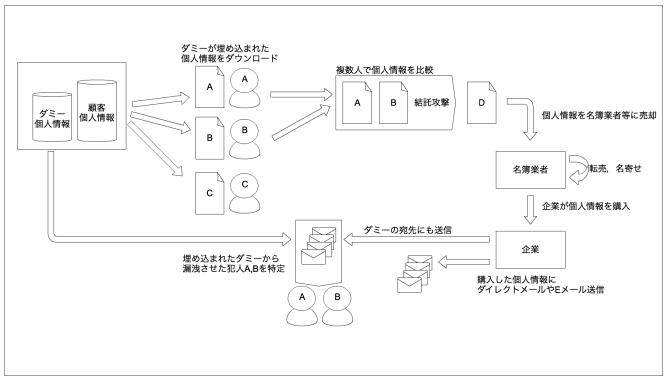

図 4 持ち出された名簿の流れ

3. 研究手法

3.1 攻撃シナリオ

不正に持ち出された名簿の流れを図 4 に図示する．攻撃者は従業員であり, 顧客の名簿をダウンロードする正当な権限を所持していることを前提とし，ダウンロードした名簿にはダミーの個人情報を埋め込まれていることを知っているものとする．また，ダミー個人情報の住所やメールアドレスは，顧客名簿を持つ企業が監視しているとする．

攻撃者は，顧客の名簿を他の従業員と結託するか, 異なる日時に自分で複数回名簿をダウンロードすることで, 複数の名簿を用意する．そして，攻撃者は用意した複数の名簿を使い結託攻撃を行う．

攻撃者は，結託攻撃を行った名簿を名簿業者に売却する．名簿業者は，購入した名簿を自社の膨大な量の名簿に追加する．そして，自社の名簿から一部を抜き出し，それを他の名簿業者や企業へ販売する．

企業は名簿業者から購入した名簿の宛先に DM や E メールを送信する．この際，名簿業者から購入した名簿にダミー個人情報が含まれていると，ダミー個人情報の宛先にも DM や E メールが送信される．そして，ダミー個人情報を監視している企業は名簿の漏洩を検知することができる．

3.2 提案システム

提案システムは顧客名簿を管理する CRM(Customer Relationship Management) である．従来のダミー個人情報埋め込みは，顧客名簿に直接ダミー個人情報が埋め込まれている場合があるが，提案システムは顧客名簿とダミー個人情報を異なるデータベースに保持する．そのため，顧客名簿を用いた調査や分析を行う際，ダミー個人情報の存在を考慮する必要がない．

提案システムの特徴的な機能は顧客名簿のダウンロードである．顧客名簿がダウンロードされる際，提案システムはその時のユーザと日時をあらかじめ生成しておいた符号の 1 つに対応させ，ログとして記録する．そして，その符

符号α	符号β	ダミー
1	0	宮元
1	1	田中
0	1	足立
1	1	山本

符号αなら {宮元, 田中, 山本} が埋め込まれる
符号βなら {田中, 足立, 山本} が埋め込まれる

図 5 埋め込み方法

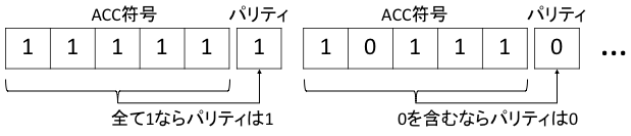

図 6 ACC 符号におけるパリティチェック

表 3 ACC 符号 $p = 11$ の性能

符号長	識別数	結託耐数	要素が 1 の数
1332bit	13431	11	1320

表 4 LDPC 符号の性能

符号長	パリティビット長	要素が 1 の数（平均）
2662bit	1330bit	2083

号から埋め込むダミー個人情報をデータベースから抜き出し，それをダウンロードされる名簿に埋め込む．

また，もう一つの機能としてダミー個人情報から，いつ誰が漏洩に加担したのかを特定することがきる．漏洩したダミー個人情報を提案システムに入力すると，記録されたログから名簿を漏洩した可能性のある従業員と日時を特定する．

3.3 埋め込み手法

ACC 符号やパリティビット，パリティビット含む LDPC 符号はバイナリで構成される．これらの符号を，ダミー個人情報が埋め込まれているかどうかのパターンで表現する．

まず，符号長の数だけダミー個人情報を用意する．符号の各 bit はダミー個人情報 1 件と対応させる．そして，符号のある bit が 1 である場合は，その bit に対応したダミー個人情報をダウンロードされる名簿に埋め込む．0 である場合には，ダミー個人情報は埋め込まない．これによって，符号の 1bit をダミー個人情報 1 件で表現している．例を図 5 に示す．埋め込まれるダミー個人情報の件数は，符号長よりも小さい数となる．

誤り訂正符号は一般的に誤りが独立に分布することを前提としているため，インターリーブを施すことが必要である．インターリーブとは，符号の要素の順番を並び替えることで，バースト誤り（連続して起きる誤り）の発生確率を減少させる手法である．名簿は年齢や地域で分割されることが考えられる．このとき，ダミー個人情報も連続して分割されてしまう可能性がある．そこで，ダミー個人情報を符号と対応させる際，ダミー個人情報をランダムな順序にすることでインタリーブを施すことができる．これによって，バースト誤りの発生確率を抑えられる．

3.4 ダミー個人情報から符号の読み込み

符号はダミー個人情報の有無で表現されている．ダミー個人情報が存在していた場合は符号の要素が 1 であることがわかる．しかし，ダミー個人情報が存在していない場合，符号の要素が 0 であるか，符号の要素は 1 であるがダミー個人情報が削除されてたのかの判別は不可能である．よって，ダミー個人情報が存在していない場合は符号が消失したと考える．

3.5 パリティチェックの利用

ダミー個人情報の埋め込みは，符号要素の消失誤りであると考えることができる．ダミー個人情報が埋め込まれていない場合は，符号要素が 0 であるか要素が消失したかの区別はつかない．しかし，誤りの位置を検出できていると考えることはできる．

我々が今回使用した Hou らによる ACC 符号は要素のほとんどが 1 で構成されている．例えば、本研究で使用する $p = 11$ の ACC 符号は符号長 1332bit のうち，1320bit が 1 である．

これらを踏まえると，パリティチェックの利用が考えられる．符号要素が 0 である場合，消失かどうかの区別がつかないことから，1 である符号要素をできるだけ多く復元することを方針とする．

一般的なパリティチェックはブロックに含まれる 1 の数が偶数か奇数でパリティビットを決定する．しかし，ACC 符号の多くが 1 であることから，符号をブロックに分割すると，ほとんどのブロックの要素が全て 1 で構成される．ACC 符号のこの特徴を利用し，対応するブロックの要素が全て 1 ならばパリティビットを 1，そうでなければパリティビットを 0 にする（図 6）．この方式を用いると，ブロックを構成する要素が全て消失しても，パリティビットが 1 であればブロックを構成する要素全てを復元することができる．

4. 実験

4.1 使用する符号

使用する ACC 符号のパラメータは $p = 11$ である．本実験では数式処理システム MAGMA[11] を用い，**表 3** の性能を持つ ACC 符号を最大である 13431 個生成した．

この ACC 符号を，LDPC 符号で符号化する．LDPC 符号化には 2664×1332，行重み 6，列重み 3 の行列をガウス消去法により一次従属な行を削除した 2664×1332 行列を使用する．この検査行列の生成には LDPC 関係の公開プログラム [12] を利用した．検査行列により生成される

表 5 ACC 符号にパリティビットを付加した符号

符号長	パリティ長	要素が 1 の数
1599bit	267bit	1484〜1570

LDPC の性能を**表 4** に示す．LDPC 符号の要素が 1 の数は平均を計算した．符号長 2083 のうち 1320 は ACC 符号の 1 であるため，LDPC 符号によって付加されたパリティビット 1332 は約半分が 1 であることがわかる．

ACC 符号にパリティビットを付加した符号を生成する．ACC 符号を 5bit ごとにブロックにとし，1 ブロックに対してパリティビットを付加した符号（6 と同様，垂直パリティチェック）を用いる．**表 5** に ACC 符号にパリティビットを付加した符号の性能を示す．

要素が 1 の数の値は埋め込まれるダミー個人情報の量を意味する．

4.2 実装

実装には Web アプリケーションフレームワークである Ruby on Rails 4.2.5.1 を用いた．リレーショナルデータベース管理システムには SQLite 3.9.2 を使用した．

扱う個人情報は全て実在しない擬似個人情報である．擬似個人情報の生成には疑似個人情報データ生成サービス [13] を利用した．

作成した web アプリケーションは，ID・パスワードによるログイン機能，動的にダミー個人情報を埋め込んだ名簿出力しログを記録する機能，読み込んだ名簿から漏洩に関与した可能性のあるユーザと日時を特定する機能を持つ．入出力は csv ファイルである．データベースは以下のテーブルを持つ．

- 顧客の個人情報を保持するテーブル
- ダミー個人情報を保持するテーブル
- 符号とダウンロードしたユーザ・日時を保持したテーブル
- ユーザ名，パスワード，最終ログイン時間等のユーザ情報を保持したテーブル

4.3 実験内容

本実験では，実装したシステムを用いてダミー個人情報の結託耐性と分割耐性を検証する．

4.3.1 結託耐性の検証

結託耐性を検証するため，複数のを名簿をダウンロードし結託攻撃を行う．結託攻撃は名簿を比較し異なる個人情報を全て削除（ expanded wide-sense Marking Assumption）するものとする．結託攻撃を行った名簿から漏洩に関与したユーザを全て特定できるか，誤ったユーザの特定を特定しないかを調べる．

4.3.2 分割耐性の検証

分割耐性を検証するために，10 万件の個人情報が含まれた名簿（ダミー個人情報含む）をダウンロードする．そこから個人情報を抜き出し新たに名簿を作成する．今回は抜き出し件数は 1 万件単位で，各 10 個の異なる名簿を用意した．新たに作成した名簿から，元の名簿をダウンロードしたユーザと日時を特定できるか，誤ったユーザを特定しないかを調べる．同時に LDPC 符号復号前と復号後で特定精度の違いを調べる．ACC 符号にパリティビットを付加した符号についても同様の実験を行う．

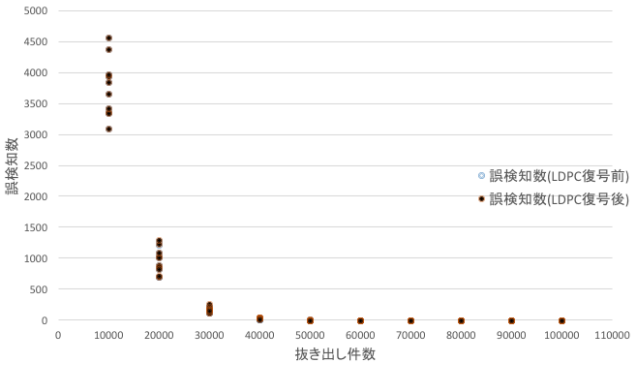

図 7 誤検知散布図

5. 結果

5.1 結託耐性

結託耐性の実験の結果，結託数 11 までの結託攻撃に対しては、結託したユーザを全て特定することができた．また，結託していないユーザを誤って特定することもなかった．

5.2 LDPC 符号利用時の分割耐性

分割耐性の実験の結果を分布図にまとめたものを図 7 に示す．縦軸は誤ったユーザや日時を特定した誤検知の件数を意味し，横軸はダウンロードした 10 万件の個人情報から抜き出した件数を意味する．誤検知は ACC 符号の数 13431 が母数となる．

図 7 において，誤検知数（LDPC 復号前）と誤検知数（LDPC 復号後）の点は全てが重なっている．（厳密には点は重なっておらず LDPC 符号による復号で誤検知は減少しているが，かなり効果は低い）これは LDPC 符号による復号がほとんど誤検知数に影響していないことを意味している．そこで，LDPC 符号による復号で復元できた bit 数を調べた．縦軸に復元できた bit 数意味し，横軸は削除されたダミー個人情報の数を意味するグラフを図 8 に示す．図 8 で，抜き出し件数が多い場合でも多く復元しているのは，ダミー個人情報が埋め込まれていない場合を消失とみなしているためである．

5.3 パリティチェック利用時の分割耐性

10 万件のうち 1 万件抜き出した名簿は，LDPC 符号を用いた実験では 10 回中 8 回が 1bit も復元することができ

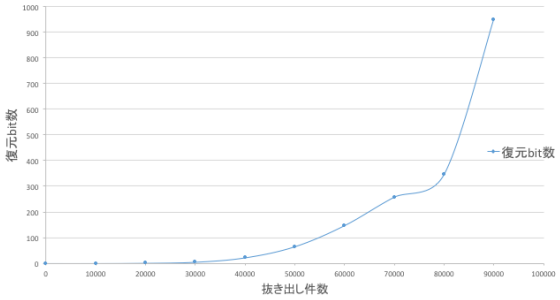

図 8 復元 bit 数

なかった．

ACC 符号にパリティビットを付加した符号では，10 回行った実験で，最大 32 個，最小でも 14 個のパリティビットをダミー個人情報から読み取ることができた．この符号のパリティ長は 267bit であり，そのうち 1 であるものは 255〜264 個である．よって，10 万件の名簿には ACC 符号を意味するダミー個人情報とは別に，約 260 個のパリティビットを意味するダミー個人情報が埋め込まれている．そこから 1 万件抜き出したものに含まれるパリティビットを意味するダミー個人情報は約 $260 \times \frac{1}{10}$ 個となり，本実験の値は確率的に妥当である．

パリティビットが 1 だとわかると，対応するブロックが全て 1 であることがわかる．つまり，30 個のパリティビットをダミー個人情報から読み取れた場合，ACC 符号を最大 150bit 復元できる．ただし，対応するブロックがすでに 1 と判明している場合もある．

6. 考察

分割耐性（LDPC 符号）の実験結果より，LDPC 符号により消失した要素が復元できていることがわかる．しかし，誤検知の数を減らすことはできなかった．例えば，ダミー個人情報が比較的削除されなかった場合，LDPC 符号によって，消失したほぼ全ての要素を復元することができる．ただし，これはダミー個人情報が比較的削除されなかった場合であり，要素を復元しなくとも誤検知はあまり発生しない．逆に，ダミー個人情報が大量に削除されてしまった場合，多くの誤検知が発生し，消失した要素も多いため LDPC 符号でも要素の復元をすることは難しい．以上より，LDPC 符号を利用することで分割耐性を持たせることができたとは言い難い．また，本実験の結果ならば LDPC 符号を利用するよりも，単純にダミー個人情報を 2 回埋め込んだ方が分割耐性がある可能性が高い．

LDPC 符号を利用したのは，ダミー個人情報の削除が一般的な符号の誤りと同様であると考えたためである．しかし，上記のとおり予想した結果は得られなかった．ACC 符号の分割耐性を向上させるためには，ACC 符号及びダミー個人情報の埋め込みの性質を利用することが重要である．

分割耐性（パリティチェック）の実験結果より，ダミー個人情報の埋め込みにおいては，LDPC 符号よりもパリティチェックの方が優れた結果となった．LDPC 符号は，複数のパリティビットから消失した要素を復元する方式である．これによって，一定数までならば要素の消失を完全に復元することができる．一方でパリティチェックは 1 つのパリティビットから複数の要素を復元する方式である．そのため，多くの要素が消失してもパリティビットが 1 つ分かれば少しでも復元することができる．ダミー個人情報の埋め込みでは符号の完全な復元は必要なく，わずかな要素から少しでも多くの要素を復元することが求められる．よって，ダミー個人情報の埋め込みの分割耐性においては，パリティチェックが有効である．

7. まとめと今後の課題

本研究では，結託耐性符号と誤り訂正符号用い，結託耐性と分割耐性を持つダミー個人情報の埋め込み手法を提案した．実験により，ACC 符号を LDPC 符号で符号化するよりも，ACC 符号にパリティチェックを利用したほうが分割耐性が高いことを示すことができた．

パリティチェックは本実験の垂直パリティチェックでなく，水平垂直パリティチェックを採用することでさらに精度を上げることができる．さらに，本研究では要素として扱わなかった 0 にもダミー個人情報を割り当てることで，0 も復元する手法が考えらる．符号要素の中で極めて少ない 0 を 1 つでも復元できれば，符号全体の中から大幅に絞り込むことが可能となる．

よって，ACC 符号の特徴を生かしたパリティチェック方式の利用を今後の課題として挙げる．

参考文献

[1] Symantec Corporation 2014 Internet Security Threat Report, Volume 19 (2014)
[2] 消費者庁 名簿販売事業者における個人情報の提供等に関する実態調査報告書 (2016)
[3] 独立行政法人情報処理推進機構 組織内部者の不正行為によるインシデント調査 (2012)
[4] 田中 賢一, 小松 尚久　電子透かし技術―ディジタルコンテンツのセキュリティ pp3 pp498 東京電機大学出版局 (2004)
[5] Boneh, D., Shaw, J.: Collusion-Secure Fingerprinting for Digital Data, IEEE Transactions on Information Theory, pp44(5) (1998)
[6] Trappe, W., Wu, M., Jane Wang, Z., Ray Liu, K.J.: Anti-collusion Fingerprintingfor Multimedia. IEEE Transactions on Signal Processing pp51(4) (2003)
[7] Shuhui Hou, Tetsutaro Uehara, Yoshitaka Morimura, Michihiko Minoh.: Fingerprinting Codes for Live Pay-Television Broadcast Via Internet, Multimedia Content Analysis and Mining pp258(5) (2007)
[8] C. Berrou, A. Glavieux and P. Thitimajshima, ”Near Shannon limit error-correcting coding and decoding:

Turbo Codes," in Proc. IEEE Int. Conf. Commun. ICC'93, Geneva, Switzerland, May 1993, pp. 1064-1070
[9] R.G. Gallager, Low-Density Parity-Check Codes MIT Press, Cambridge, (1963)
[10] 和田山正　低密度パリティ検査符号とその復号法 LDPC(Low Density Parity Check)/sum-product 復号法 トリケップス　pp. 24（2012）
[11] Key, J.D.: Some applications of Magma in designs and codes: Oval designs, Hermitian unital and generalized Reed-Muller codes. J. Symbolic Computation pp31(1/2), pp37-53 (2001)
[12] 和田山正　LDPC 符号関係の公開プログラム https://dl.dropboxusercontent.com/u/1820240 /supportpages/LDPCpublic.html （参照日：2016 年 8 月 8 日）
[13] 疑似個人情報データ生成サービス http://hogehoge.tk/personal/（参照日：2015 年 12 月 6 日）