2019 年度　修士学位請求論文

菱形サブセルを用いる電子署名を追加した安全なＱＲコードの研究

明治大学大学院先端数理科学研究科

先端メディアサイエンス専攻

WANG　CAN

Master's Thesis

Research on secure QR code with digital signature using rhombus subcell

Frontier Media Science Program,

Graduate School of Advanced Mathematical Sciences,

Meiji University

Can Wang

概要

ＱＲコードは代表的な二次元シンボルとして，製造業，流通業に使い始め，またインタネットの普及によって，ウェブサイトの情報のキャリアーとして使われている．最近では，モバイルの普及によって，ＱＲ決済も広がっている．しかし，ＱＲコードの作成は簡易な一方で，不正利用される可能性もある．本稿ではＱＲコードの安全性を高まるため，菱形サブセルを用いる電子署名を追加したＱＲコードの実装及び新方式のＱＲコードの耐性実験を行う．

Abstract

QR(Quick Response) code, a represent of two-dimensional symbols, has begun to be used in the manufacturing and distribution industries. And since the extensively spread of Internet, QR code also begin to be used as a carrier for location of websites. Recently, with the popularization of mobile payment, QR payment has also been spreading. However, since QR codes are easy to be generated, they can also be used in a malicious way. In this paper, in order to enhance the security of the QR code, we implement a new type of QR code with a digital signature in its rhombus subcells and conduct some experiments about the robustness of the new type of QR code.

目次

第１章　　はじめに

1.1. 研究背景

バーコードには一次元シンボルと二次元シンボルがあり，最初は流通業界におけるデータキャリアとして使われていた．一元シンボルは 1970 年代に買い物の精算を省力化するため開発された共通商品コードである．コード 39, コーダーバーなどのコードは今も製造業, 物流，流通業界に使われている．1980 年代になって，もっと多くの情報を表現したいという市場ニーズを応じて，二次元シンボルの開発が活発になった．代表的なシンボルは，PDF417，コード 49，データマトリックス，ＱＲ（Quick Response）コードである．二次元シンボルは一次元シンボルより，横と縦軸に情報を格納できるため，符号化できる情報量が増える．ＱＲコードは二次元シンボルの一つとして，様々な商品の生産・運送・保管・販売などに使われるようになった．近年，中国市場を中心に普及したキャッシュレス決済においても，ＱＲコードが普及してきた．

しかし，ＱＲコードの生成方式は公用されており，多くのオープンソースがあり，中心には情報を入力するだけで，すぐ簡単にＱＲコードを作れるウェブサイトもある．この特性を悪用し，不正者がＱＲコードを偽造する可能性がある．例えば，利用者がリーダーで読み取ったデータをきちんと確認せずウェブサイトにアクセスすることが多いため，ＱＲ コードを介して不正な偽のウェブサイトに誘導される危険性が存在している．

2011 年 10 月にＱＲコードを介して，料金を発生させるマルウェアが発見した[13]．某ウェブサイトに掲載されたＱＲコードを撮影すると，約 6 ドルが自動的に課金されるという．また，中国の IT 大手・騰訊（テンセント）のセキュリティ部門が発表した「2017 年上半期インターネットセキュリティ報告書」では，ＱＲコードは主な携帯電話ウイルス感染ルートになる[14]と言われている．

このようなＱＲコードの潜在的な脅威に対して，電子署名を導入することが提案されている．

1.2. 研究目的

本研究は，ＱＲコードの安全性を高まるため，ＱＲコードに電子署名を埋め込む方法を検討する．

ＱＲコードに署名を追加するのは解決必要がある問題は三つである．一つは署名を埋め込む空間を確保すること．二つは署名に対する誤り訂正が必要である．三つはデータと区別できる区間を作ること．菱形サブセルを用いる電子署名を追加したＱＲコードを通じて，この三つの問題を解決できる．

本研究は菱形サブセルを用いる電子署名を追加したＱＲコードの実現手法及び新方式のＱＲコードの耐性実験を行う．

1.3. 本稿構成

本稿は８章で構成される．第１章では研究の背景と目的を述べる．第２章では本稿の基礎定義と既存研究について述べる．第 3 章では電子署名を埋め込む予備実験を行う．第 4 章では，新型ＱＲコードの互換性の必要性について述べる．第 5 章では，菱形サブセルを用いるＱＲコードの設計手法を述べる．第 6 章では開発システムを述べる．第７章では生成された菱形サブセルを用いるＱＲコード誤り耐性実験を行う．第８章は今後の課題とまとめを行う．

第２章　　基礎定義と既存研究

2.1. 基礎定義

2.1.1. ＱＲコード

ＱＲコードは二次元シンボルの一種であり，機能パターンと符号化領域の二つの部分から構成される．日本工業規格が基本の構造方法を定めた基準[1]を作った．ＱＲコードの構造は図 2.1 が示されている．機能パターンは位置検出パターン，形式情報，タイミングパターンと位置合わせパターンから構成される．符号化領域にはデータコードと誤り訂正コードの二つの部分が含まれている．符号化された情報とその誤り訂正コードが書かれている．

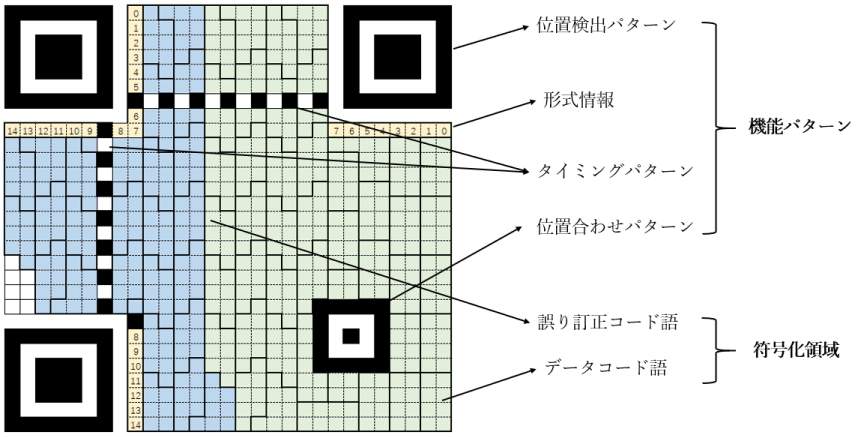


図 2.1: ＱＲコードの構造
（3-M 型，[1]で書かれている内容を元に自分で作図した）

ＱＲコードの型番(バージョン)は１～40 型まで存在し，型番が大きくなるにつれて格納できるデータ量が増加する．型番はＮ型のＱＲコードコードの一辺のセル数は 17＋４Ｎである．ＱＲコードの型番を問わず，以下の機能パターンが存在している．

位置検出パターンはＱＲコードの位置を検出ために，ＱＲコードの三つの隅に配置され，形式が固定されたパターンである．

形式情報には，シンボルに使用されている誤り訂正レベルとマスクパターンと構成される．その長さは 15bit に固定され，位置検出パターン周囲の二箇所に書かれている．誤り訂正レベルはＬ（７％），Ｍ（１５％），Ｑ（２５％），Ｈ（３０％）の４階段がある．マスク処理はシンボルの符号化領域の明暗バランスを取るために人為的にマスクをかける．

タイミングパターンは，セルの座標を確定するため，明，暗交互のビット列である．

位置合わせパターンはＱＲコードの画像に歪みが生じた時にセルの座標を修正するため各バージョンにより指定されたパターンで固定位置に設置された機能パターンの一部である．

また，型番は 7 以上の場合，ＱＲコード右上と左下の二箇所に位置検出パターンと隣接している場所で 15bits の固定長で記入されている．

それ以外の部分は符号化領域として使われている．型番，誤り訂正レベルによりデータコード数と誤り訂正コード数が異なる．ＱＲコードに格納するデータの長さがデータコード数より小さい時，データの後に終端コードと呼ばれる二進数ビット列 0000 を付け，データコード数の長さまで交互に二進数ビット列 11101100，00010001 を埋め込む．その部分を埋め草部と呼ぶ．

2.1.2. 電子署名

電子署名は，紙文書における印鑑や署名に相当するデジタルデータに対するものである．電子署名は電子取引，保存文書，業務記録などに使える．

電子署名は署名と検証二つの手順が含まれる．図 2.2 に電子署名のモデルを示している．

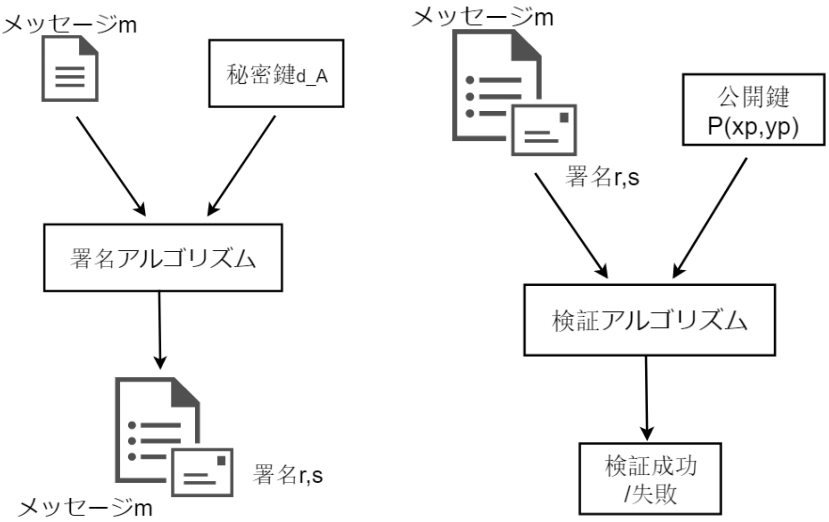


図 2.2: 電子署名のモデル

電子署名を実現する公開鍵暗号方式を応用したデジタル署名方式が提案されている． 一般的に広く使用されている電子署名方式は RSA（Rivest–Shamir–Adleman）署名[6]，ElGamal 署名[7]，DSA(Digital Signature Algorithm)署名[8]，Schnorr 署名[9]などである．

本論文で使う電子署名は ECC 楕円曲線を基づいて作る．この方式は ECDSA(Elliptic Curve Digital Signature Algorithm)と呼ばれる，楕円曲線を用いるデジタル署名の一種であ

る． RSA 方式[9]より，同じセキュリティレベルを満たすには署名サイズがより短い．そのアルゴリズムを以下に簡単に示す．

楕円曲線とは，一般的に，以下のような方程式で表される．

$$y^2 = x^3 + ax + b$$

ECDSA による署名生成では(r, s)という 2 つの値を生成する．署名の生成は順に次の作業を行う．

(1)$[1, n-1]$からランダムに秘密鍵d_Aを生成する．nは素数である．

(2)$\mathrm{P_A} = \mathrm{d_A G}$から公開鍵ペア$\mathrm{P_A}(\mathrm{x_P}, \mathrm{y_P})$を生成する． Gはベースポイントである．

(3) $[1, n-1]$からランダムに kを生成する．

(4)$(x_1, y_1) = k\mathrm{G}$を計算する．

(5)$r = x_1 \bmod n$を計算する． $r = 0$の場合(1)に戻って，改めて乱数kを選ぶ．

(6)$s = k^{-1}(z + rk) \bmod \mathrm{n}$を計算する．$s = 0$の場合(1)に戻って，改めて$k$を選ぶ．ここで，$z$は$\mathrm{Hash}(m)$で求められたメッセージのハッシュ値．

(7) (r, s)はメッセージmに対する署名となり，メッセージmと一緒送信される．

署名の検証する手順は以下となる．

(1) r, sは$[1, n-1]$内の整数であることを確認する．

(2)$u_1 = s^{-1} z \bmod n$を計算する．

(3)$u_2 = s^{-1} r \bmod n$を計算する．

(4)$(x_1, y_1) = u_1 \mathrm{G} + u_2 \mathrm{P}_A$を計算する． $r \equiv x_1 \bmod n$であれば，署名は正当なものが確認できる．

2.2. 既存研究

柏井ら[2]がＱＲコードの下に URL(Uniform Resource Locator)を記載し，ＱＲコードの埋め草部に URL に対する RSA 署名を埋め込む方式を提案している，検証の時，署名を復号し，ＱＲコードに格納されたデータを検証する．OCR（光学文字認識）で読み取ったＵＲＬの情報とＱＲコードのデータを比較し，誤りが否かを検証する．

柏井らが使用した RSA 署名に用いる鍵のサイズは 2048bits であり，ＱＲコードの型式は 11-L である．作成したＱＲコードは図 2.3 で示している．柏井らの提案はＱＲコードの下に URL を書き付けることで，オフラインでの署名認証ができるになった．しかし，情報データに対して，署名に使わられた容量が 2048bits になってしまう．そのため，データコードに使える部分が減らされる．また，署名に埋め込める容量を保つため，ＱＲコードのバージョンを上げる必要がある．

図 2.3: 柏井ら作成したＱＲコード[2]

先名[3]の研究は ECDSA を用い，オフラインにおいてＱＲコードの真正性判定と個人認証の両機能を備えるＱＲコードを提案した。ECDSA により生成された署名をＱＲコードのコード語領域の誤り訂正コード語部分に排他的論理和（ＸＯＲ）によって埋め込む.

この手法では，容量を増えずに署名を埋め込むことが可能である．しかし，誤り訂正部分が署名を埋め込むために使われ，誤り訂正機能が無くなり，誤りが発生した場合，復号ができなくなる可能性がある.

図 2.4: 先名作成したＱＲコード[3]

寺浦[4]は電子署名の実装には互換性，読み取り性，分離性を満たす必要があることを提言した．そのため，セルの多色化と領域分割を提案した．ＱＲコードがサンプリングを行う時，周辺のセルからの影響をできるだけ削減するため，セルの中心位置を使っている．つま

り，セルの周辺にサブセルを挿入しても，ＱＲコードに埋め込んだ情報を読み取れる．しかし，セルの大きさにより，中心セルが周囲セルに侵食され，新たな誤りが発生する恐れがある．

そこで，本論文では，寺浦が提案した菱形サブセルを基づいて実装し，サブセルが元セルに対する影響を実験的に明らかにする．

表 2.1: 先行研究との比較

	柏井(2012)	先名(2017)	本研究
暗号化方式	RSA 方式	ECDSA 方式	ECDSA 方式
公開鍵長	2048 ビット	160 ビット	最小 160 ビット 任意長可能
電子署名記憶方式	埋め草領域	誤り訂正部	別空間
課題	電子署名データ量が大きい ＱＲコードが大きい	誤り訂正能力がなくなる	現実的な環境で評価する

第３章　　予備実験

ＱＲコードに変更を加えることなく，署名を埋め込めるかというか判断する為に，以下な予備実験を行った．

本実験では，ＱＲコード符号化領域のデータコードに ECDSA による生成された署名をデータの後に格納する．

図 3.1 は電子署名を入れてないＱＲコードであり，図 3.2 はデータ部に電子署名を入れたＱＲコードである．ここで，ECDSA による生成された署名，ＱＲコードのデータ格納手順に従い，二進数に符号化して格納されている．

図 3.1: 電子署名を入れてないＱＲコード

図 3.1 のＱＲコードに格納された内容は：
'testing for ＱＲ code with ecdsa'である．

図 3.2:　データ部に電子署名を入れたＱＲコード

図 3.2 のＱＲコードに格納された内容は：
'testing for ＱＲ code with ecdsa',
'044314341612070294005925385127063243568885011050406315595163999768718267 14

7349',
'037423242994442122752135984209996994684205671970801121173642220243278014613997'である．

ＱＲコードに格納された情報は二つの部分に構成される．前半はデータ部分，後半はデータを基づいて生成された署名である．署名は十進数で表示されている．

柏井らの RSA 署名方式と同じ，ECDSA 手法による生成された署名をＱＲコードに埋め込んだ結果，ＱＲコードの型番が大きくなった．RSA 手法による生成された 2048bit 署名と同じ安全性を保つために,ECDSA による生成された署名の大きさは長さ 160bits の 2 つの整数である署名ペアが必要である．柏井らの手法は 11 型のＱＲコードが必要であるに対し，ECDSA 手法による生成された署名のサイズが小さくなったにもかかわらず，ＱＲコードの型番は 3 型から 7 型になるころが分かった．

従って，単純に既存のＱＲコードに署名を埋め込む方式は，ＱＲコードのサイズを増大させてしまい，データコードの大きさによってはオーバフローを引き起こす恐れがある．既存のアプリやインフラに互換性を持たせるためには，サイズを維持した新しい方式が求められる．

第４章　　互換性の必要性

柏井の提案と予備実験を基づいて，以下の拡張案を考える．

ECDSA による生成された署名を柏井方式のように，署名を符号化領域の埋め草部に埋め込んでみると，電子署名の格納とデータとの分離性が実現可能になる．

しかし，この方式では空間上の拡張は実現してない．

現在使われているＱＲコードを徐々に置換するには，互換性が必要である．そこで，セルの不寄与領域を使う．

ＱＲコードがサンプリングを行う時，周辺セルの侵食を削減するため，セルの中心位置で画素をサンプリングしている．サンプリングとして採用されてないセル周辺の部分を不寄与領域と呼ぶ．図 4.1 が示して灰色の部分が不寄与領域である．

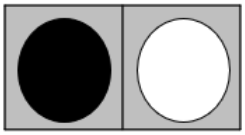


図 4.1: セルの不寄与領域

不寄与領域を利用すると，ほかの情報を埋め込むことが可能である．不寄与領域を使うことによって，空間上の領域拡張ができる．そのため，サブセルの形は菱形にこだわる必要がなく，円形，正方形にしても実現できる．

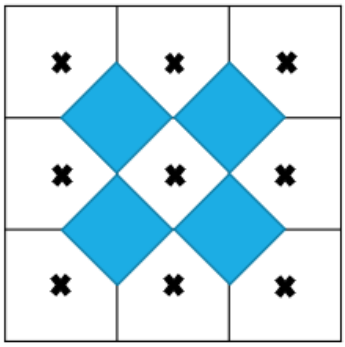


図 4.2: ＱＲコードセルのサンプリング

また，空間上領域拡張にはロバスト性に貢献する．例を上げて説明しよう．ＱＲコードの一辺の長さが固定され，菱形サブセルを用いるバージョン３のＱＲコードと通常のバージョン６のＱＲコードのセルの長さはほぼ同一になる．

各バージョンのＱＲコードの誤り訂正能力を議論する．

同じく誤り訂正レベルをＬ（7%）にする時を例としてあげる．菱形サブセルを用いるバージョン 3 の誤り訂正能力は元セルの 7 符号語と菱形サブセルの 7 符号語であり，合わせて 14 符号語の誤り訂正できる．一方，誤り訂正レベルがＬレベルのバージョン 6 のＱＲコード（以下 6 -L 型ＱＲコードと呼ぶ）の誤り訂正できる符号語が 18 である．菱形サブセルを付けることで，誤り訂正できるコード数が減少する．

また， 6 -L 型ＱＲコードのデータコード語数は 136 である．それに対して，3-L 型ＱＲコードのデータコード語数は 55 である． 6 -L 型ＱＲコードは 3-L 型ＱＲコードより，81 コード数多く格納できるため，48bytes の署名を直接埋め草部に入れることは可能であり，3-L 型ＱＲコードより多いデータを格納することもできる．

図 4.3: 菱形サブセルを用いる 3-L 型ＱＲコード

図 4.4: 6 - L 型ＱＲコード

菱形サブセルを用いるバージョン 3 の時，周辺の位置合わせパターン，タイミングパターンなどに使われている領域が必要なため，前述のような「無駄遣い」が発生した．しかし，バージョン 3 のセルのサイズがバージョン 6 より大きいため，サンプリングする時の間隔も大きくなる．そのため，ＱＲコードの画像が小さくなるほど，バージョン 3 はバージョン 6 よりロバスト性がよい．

第５章　　菱形サブセルを用いるＱＲコードの設計

5.1. 菱形サブセルの仕様

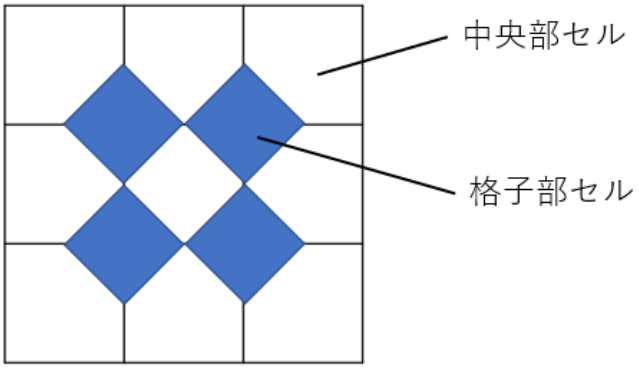


図 5.1: 中央部セルと格子部セル

菱形サブセルの形式は図 5.1 で示されている．通常なＱＲコードのセルを中央部セルと呼び，サブセルとなっている部分を格子部セルと呼ぶ．菱形セルの頂点は周辺の中央部セルの中心位置である．格子部セルの辺長は中央部セルの辺長の$\frac{\sqrt{2}}{2}$倍である．

5.2. 生成部分の構造

5.2.1. 生成方法

菱形サブセルを用いるＱＲコードを生成する要点は三つある．一つは，埋め草部にサブセルのタグを入れる長さを確保すること．二つは，サブセルのタグを埋め草部に埋めること．三つは，サブセルを生成すること．この順番で菱形サブセルを用いるＱＲコードを生成する．

図 5.2，図 5.3 にＱＲコード中央部セル，格子部セル符号化領域の構造を示す．本論文の実装ではサブセルのタグを 2bytes に規定し，その 1 byte はサブセルの存在を示す指示子であり，0xE0 と指定する．もう 1 byte には，公開鍵の ID を入れる．公開鍵の ID の定義域は 0x01~0xEE に指定することができる．読み取る時は，指示子でサブセルの存在を確認し，公開鍵の ID を読み取る．読み取った公開鍵の ID から，対応表から対応する公開鍵を取り出す．

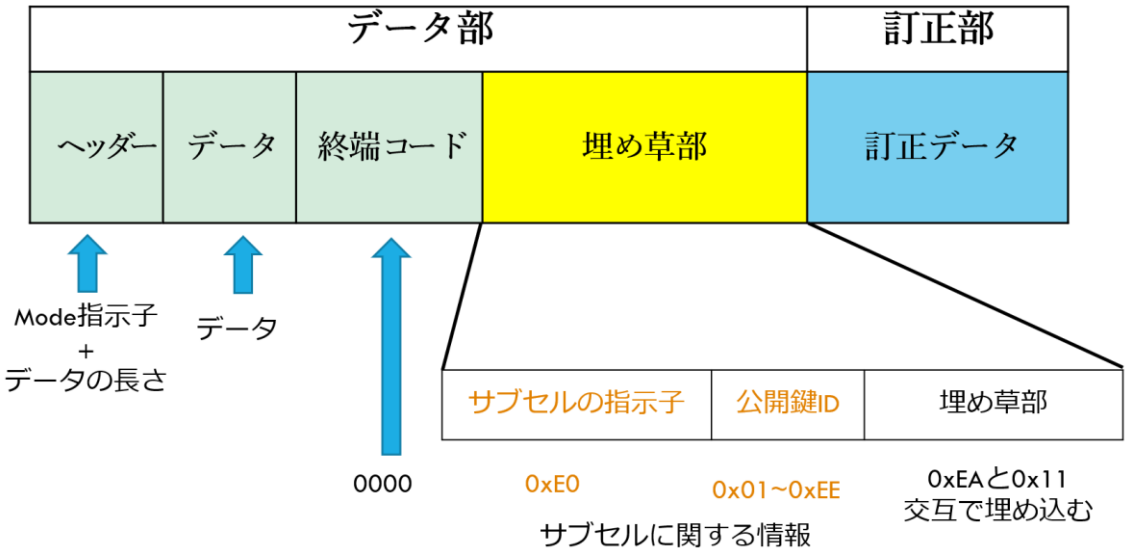


図 5.2: ＱＲコード中央部セル符号化領域の構造

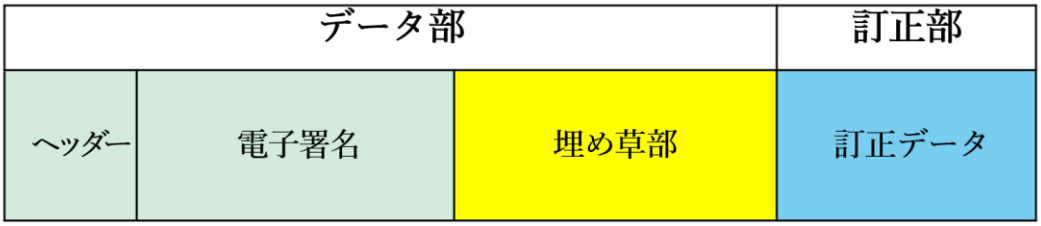


図 5.3: ＱＲコード格子部セル符号化領域の構造

次に，格子部サブセルに埋めこむ電子署名の生成方法を述べる．
ECDSA による生成された電子署名の仕様は以下となる．

0x30 + LEN1 + 0x02 + LEN2 + 00 (optional) + r + 0x02 + LEN3 + 00(optional) + s

そのうち，LEN3 は 0x00(1byte)と署名 s の長さの和であり，LEN2 は 0x00(1byte) と署名 r の長さの和であり，LEN1 は LEN2 と LEN3 の和の元に 4 を加えた値である．160 ビットの署名を例とすると，生成された署名の総長さは 46～48bytes になる．2bytes の変動の原因は，署名 r と s の一番目の byte は 0x80 より大きな時のみ追加の 0x00 を付けるためである．

生成された署名はバイナリで格納されているため，ＱＲコードのデータ部のような符号化をする必要がない．署名の長さの生成された時と同様になる．また，署名の長さの制限のため，バージョンが低いＱＲコードは菱形サブセルの生成はできない．そのため，バージョン１のＱＲコード，バージョン２のＱＲコード，誤り訂正レベルが M,Q,H のバージョン３のＱＲコード，誤り訂正レベルが Q,H のバージョン４のＱＲコード，誤り訂正レベルが H

のバージョン 5 のＱＲコードは適用されない.

4 章拡張案(埋め草部に署名を埋め込むこと)と提案方式が適用されるＱＲコードのバージョン，誤り訂正レベルは同一である．しかし，署名を埋め込むことで拡張案に埋めこむ可能なデータ量が削られる．例えば，バージョン 3 の場合，4 章拡張案のデータ符号数が 7 になる．つまり，56bit になる．そのうちに，ヘッダーなどに 12〜14bit が使われ，実際に埋め込む可能なデータは 8 ビットバイトモードの場合 5 文字，英数字モードの場合，7 文字になる．埋め込む可能なデータ量の変化を表 5.1 に示す.

表 5.1: 埋め込む可能なデータ符号数※

型番	誤り訂正レベル	4 章拡張案	提案方式	原データ量
1	L	/	/	19
	M			16
	Q			13
	H			9
2	L	/	/	34
	M			28
	Q			22
	H			16
3	L	7	53	55
	M	/	/	44
	Q	/	/	34
	H	/	/	26
4	L	32	78	80
	M	16	62	64
	Q	/	/	48
	H	/	/	36
5	L	60	106	108
	M	38	84	86
	Q	16	62	64
	H	/	/	46
6	L	88	134	136
	M	60	106	108
	Q	28	74	76
	H	12	58	60

※注： 1 符号数=8bit

次はサブセルに対する誤り訂正部分を説明する．

サブセルは元のセルと違うサイズ，大きさを持っているため，それに対する独自な誤り訂正仕組みを作る必要がある．それに関して，二つ異なる基準のサブセル誤り訂正特性表を定義する．一つは通常なＱＲコードの誤り訂正レベルと一致させ，コード数に対する相対な誤り訂正率を同じにする．もう一つは誤り訂正率と関係なく独立に，各バージョン，誤り訂正レベル元の誤り訂正できるコード数を基づいて作成する．

一つ目の誤り訂正特性に関しては，サブセルの数は常に通常なＱＲコードのセル数より小さいため，誤り訂正できるサブセル数は通常なＱＲコードのセル数より小さくなる．二つ目の誤り訂正特性については，サブセルの誤り訂正できるコード数は通常なＱＲコードの誤り訂正できるコード数を同じに設定したため，サブセルの誤り訂正率は同じバージョンの通常なＱＲコードより高くなり，誤り訂正レベルで定義された誤り訂正率を超える可能性がある．表 5.2 と表 5.3 はその二つの基準を基づいて求めた誤り訂正特性の一部である．

菱形サブセルのコード数は型番，一辺のセル数を基づいて，機能パターンの部分を除いた後に計算した結果である．例えば，型番は 3 の時，一辺のセル数は 29 である．菱形サブセルの一辺のセルが 28 になる．総セル数は 28×28=784 セルである．そのうち，ＱＲコードの識別が影響されないため，格子部にデータを埋め込む時には，中央部セルの機能パターンの部分を除く必要があるため，図 5.4 に示している「×」を付けている場合にはデータを埋め込ないにする．左上，右上と左下にある位置検出パターン，形式情報の位置 8×8=64 セルを三ヶ所に合わせて 192 セルを除く．また，タイミングパターンの周辺セル数 48 個，位置合わせパターンの周辺セル 36 個を除く．菱形サブセルに使えるセル数は 784-192-48-36=508 セルである．つまり，菱形サブセルには 508bit のデータを埋め込むことは可能である．また，誤り訂正を行う時は符号数(8bit)単位で訂正するため，実際に菱形サブセルに埋め込む可能な符号数は 63 である．このように，各型番下に菱形サブセルのコード数を計算し，また，誤り訂正レベル（L,M,Q,H）を基づいて，Reed-Solomon ブロック(以下 RS ブロックと呼ぶ)のコードを計算する．RS ブロック(*c,k,r*)の *c* は型番総コード数である．つまり，計算で求めた符号数である．*k* はデータコード数で，*r* は誤り訂正数であり，誤り訂正レベルと誤り訂正率で算出される．型番 3 以上のＱＲコードに対して，*c,k,r* は以下の関係式を満たす．

$$c=k+2r \qquad ※$$

※3-L 型の場合が $c = k + 2r + 1$になる．

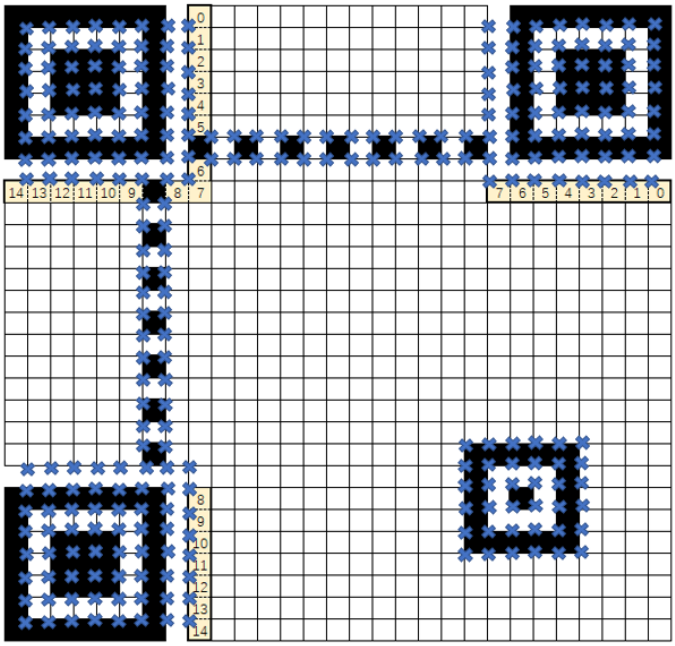


図 5.4: 菱形サブセルを埋め込む時避ける場所

表 5.2: 誤り訂正レベルによるサブセル誤り訂正特性（一部）

型番	コード数 (*c*)	データ数 (*k*)	誤り訂正コード数	残余セル数	誤り訂正レベル	RS ブロック数	RS ブロック※
1	/	/	/	/	/	/	/
2	/	/	/	/	/	/	/
3	63	48	15	4	L	1	(63,48,7)
4	91	77	14	4	L	1	(91,77,7)
		63	28		M	1	(45,31,7)
						1	(46,32,7)
5	123	105	18	4	L	1	(123,105,9)
		83	40		M	1	(61,41,10)
						1	(62,42,10)
		59	64		Q	1	(30,14,8)
						3	(31,15,8)
6	159	135	24	4	L	1	(159,135,12)
		107	52		M	1	(79,53,13)
						1	(80,54,13)
		79	80		Q	1	(39,19,10)
						3	(40,20,10)
		63	96		H	1	(39,15,12)
						3	(40,16,12)

※注:RS ブロック(*c,k,r*)　*c*:型番総コード数，*k*:データコード数，　*r*:誤り訂正数

表 5.3: 誤り訂正数によるサブセルの誤り訂正特性（一部）

型番	コード数 (*c*)	データ数 (*k*)	誤り訂正コード数	残余セル数	誤り訂正レベル	RS ブロック数	RS ブロック※	誤り訂正率 (%)
1	/	/	/	/	/	/	/	/
2	/	/	/	/	/	/	/	/
3	63	48	15	4	L	1	(63,48,7)	11.1
4	91	71	20	4	L	1	(91,71,10)	7.7
		55	36		M	1	(45,27,9)	20.0
						1	(46,28,9)	19.6
5	123	97	26	4	L	1	(123,97,13)	10.6
		75	48		M	1	(61,37,12)	19.7
						1	(62,38,12)	19.4
		51	72		Q	1	(30,12,9)	30.0
						3	(31,13,9)	29.0
6	159	123	36	4	L	1	(79,61,9)	11.4
						1	(80,62,9)	11.3
		95	64		M	1	(39,23,8)	20.5
						3	(40,24,8)	20.0
		63	96		Q	1	(39,15,12)	30.7
						3	(40,16,12)	30.0

※注:RS ブロック(*c,k,r*)　*c*: 型番総コード数，*k:*データコード数，　*r.*誤り訂正数

第 6 章　　実装実験

6.1. 生成部分

6.1.1. 生成手順

通常なＱＲコードの生成手順を以下に簡単に説明をする.

まずは情報データを文字種類別で符号化する. 次は, 符号化されたデータの長さ, ＱＲコードのバージョン, 誤り訂正レベルにより, 誤り訂正符号を生成する. そして, データ列と誤り訂正コードを最終的なビット列を構築する. その後, マトリックスに機能パターンを配置し, 残されてセルに最終的なビット列を一定な順序で埋め込む. 符号化領域に適切なマスクをかける. 最後に, 形式情報, 型番情報を埋め込む.

菱形サブセルを用いるＱＲコードの構築は, まずＱＲコードの生成する方法は変わらない. しかし, ＱＲコードの埋め草部にサブセルのタグを埋める必要があるため, タグを入れる長さを確保する必要がある.

格子部サブセル埋め込む手順について説明する. 格子部にデータを埋め込む時に, 中央部セルのＱＲコードを認識するために, 位置検出パターン, 形式情報, タイミングパターンと位置合わせパターンなどの機能パターンを避ける必要がある. それ以外には, データを埋め込む順番はこだわる必要がない. 本研究では, ＱＲコードの左上から右下まで順番に埋め込んでいる.

6.1.2. 開発システム

開発に用いた開発環境を表 6.1 に示す.

表 6.1: 開発システムの開発環境

ソフトウェア	Eclipse IDE (4.12.0)
ライブラリ	Github
オープンソース	Zxing(Google)
GUI ツールキット	Java Swing

本研究において開発システムはオープンソース Zxing を基づいて作成した. Zxing は言語 Java で書かれた一元シンボルと二次元シンボルの画像処理用ライブラリである.

開発システムに使われた主なクラスとメソッドを表 6.2〜6.10 に示す.

表 6.2: Uitop クラスに定義されている主なメソッド

メソッド	説明
void init()	Windows イニシャライズ
void EventListener()	入力の情報により QR コードを生成する

表 6.3: CodeDrawing クラスに定義されている主なメソッド

メソッド	説明
static void Drawing(ByteMatrix m,int dimension,int width,int height, int version)	通常のＱＲコードを描く
static void Drawingecdsa(ByteMatrix m, ByteMatrix ecdsa, int version, int width, int height)	ＱＲコードの中央部マトリクスと格子部マトリクスを描く

表 6.4: Encoder クラスに定義されている主なメソッド

メソッド	説明
QRCode encode(String cntent, ErrorCorrectionLevel ecLevel)	Zxing を参照する
QRCode encode(String cntent, ErrorCorrectionLevel ecLevel, Map hints)	入力した情報により QR コードのマトリクスを作成する
static Mode chooseMode(String content, String encoding)	Zxing を参照する
static int chooseMaskPattern(BitArray bits, ErrorCorrectionLevel ecLevel, Version version, ByteMatrix matrix)	Zxing を参照し，最優マスクを選ぶ
void terminateBits(int numDataBytes, BitArray bits)	QR コードの最終コードを作成する. 菱形サブセルのタグ，鍵 ID を埋め込む
byte[] generateECBytes(byte[] dataBytes, int numEcBytesInBlock)	誤り訂正ブロックの生成

表 6.5: ECDSAEncoder クラスに定義されている主なメソッド

メソッド	説明
static ByteMatrix ecdsaencoder(byte[] sign,ErrorCorrectionLevel ecLevel, VersionSubcell version)	中央部セル及び格子部セルのマトリクスを作成する
byte[] generateECBytes(byte[] dataBytes, int numEcBytesInBlock)	通常の QR コードの生成方法を参照する

表 6.6: MatrixUtilECDSA クラスに定義されている主なメソッド

メソッド・定数	説明
static final int[][] POSITION_DETECTION_PATTERN	位置検出パターンのブロック用定数
static final int[][] POSITION_ADJUSTMENT_PATTERN	位置合わせパターンのブロック用定数
static final int[][] POSITION_ADJUSTMENT_PATTERN_COORDINATE_TABLE	位置合わせパターンのブロック用定数
static void buildMatrix(BitArray dataBits, ErrorCorrectionLevel ecLevel, VersionSubcell version, ByteMatrix matrix)	菱形サブセルを生成する

表 6.7: ECDSAUtils クラスに定義されている主なメソッド

メソッド	説明
Map<String,String> getGenerateKey()	ECDSA 方式の鍵ペアを生成し，保存する
static byte[] ECDSAencrypt(String msg, String mark)	鍵 ID から秘密鍵を読出し, 入力した msg により署名を生成する
static void ECDSAdecrypt(String msg, String mark,byte[] sign)	鍵 ID から公開鍵を読出し, 取得された msg と署名 sign から署名を検証する

表 6.8: VersionSubcell クラスに定義されている主なメソッド

メソッド	説明
BitMatrix buildFunctionPattern()	機能パターン位置の初期化
static VersionSubcell[] buildVersions()	格子部セルデータ部と訂正部のブロックを設定する

表 6.9: SubcellBitMatrixParser　クラスに定義されている主なメソッド

メソッド	説明
byte[] readSubcellwords(int numVersion)	格子部マトリクスから格子部セルビット列を引き出す

表 6.10: DecodedSubcellBit クラスに定義されている主なメソッド

メソッド	説明
static BitArray decode(byte[] bytes, VersionSubcell version, ErrorCorrectionLevel ecLevel)	誤り訂正した格子部セルビット列から署名を切り取る

本研究において開発したＱＲコード生成ソフトの実行画面を図 6.1 に示す. 本ソフトにメッセージデータを入力し，ＱＲコードのバージョン(1 から 40)，誤り訂正レベル(L,M,Q,H)を指定する．電子署名を埋め込むかを設定し，埋め込む鍵 ID を選択し，電子署名を埋め込み可能か否かが判断され，ＱＲコードが生成される．

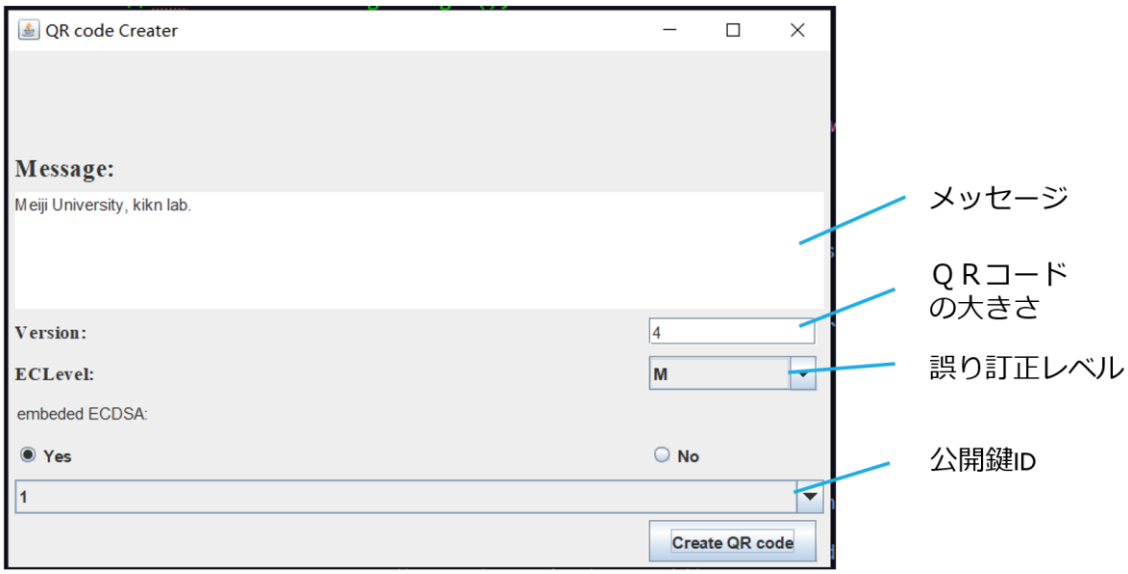


図 6.1: 菱形サブセルを用いるＱＲコード作成ソフト実行例

生成されたＱＲコード例を図 6.2 に示す．左は通常のＱＲコードであり，右は提案方式においてサブセルに電子署名を入れた例である．サブセルの部分を分かりやすく見せるため，サブセルのデータが 1 の時青色に，0 の時白色で表示している．

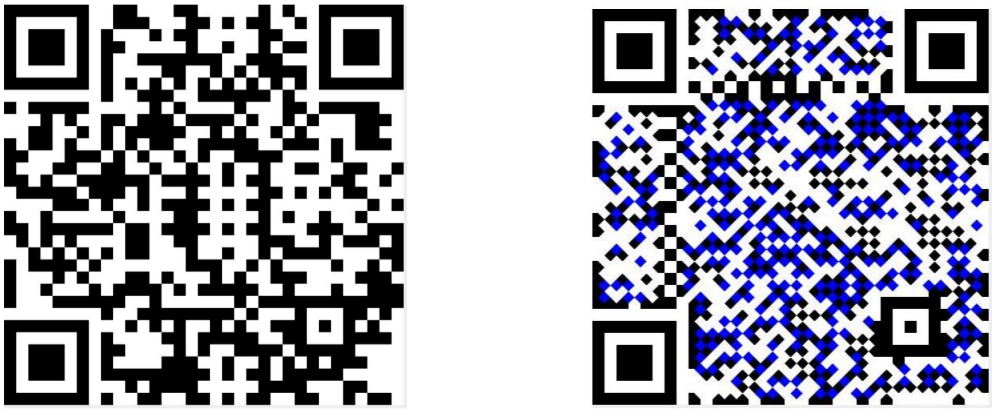


図 6.2: 生成されたＱＲコード例
左：電子署名なし　右：提案方式による電子署名あり

6.2. 読み取り部

ＱＲコードの読み取り部は生成されたＱＲコードに対して，ＱＲコードに格納されたデータ内容を読み取り，サブセルに格納された署名を確認する．

実用の際には，電子署名に使われる公開鍵と秘密鍵は認証局（CA）での認証した後に，公開鍵証明書を発行することが必要である．本実験に限り，認証局の処理を省略する．

もちろん，生成プログラミングは公開鍵と秘密鍵両方を持つのに対し，読み取りプログラミングは公開鍵のみを持っている．

6.2.1. 読み取る手順

ＱＲコードの読み取る方法は生成手法の逆手順である，しかし，一般的なＱＲコードリーダーは符号化領域データコード語の終端パターンまでしか読み取らない．サブセルの部分を読み取るために，まずは埋め草部に埋め込んだサブセルのタグを検測する．サブセルのタグが確認した場合，もう一度ＱＲコードを走査する．走査の位置はサブセルの位置である．その同時に，サブセルのタグ後に付いている公開鍵 ID を取得する．サブセルのタグが確認できなかった場合，署名が存在しない警告を提示する．

サブセルのデータを取得した後，ＱＲコードの符号化領域と同様に，誤り訂正を行う．

6.2.2. 識別例

識別結果の例を図 6.3，図 6.4 で示す．

```
numData is :428
mode is :BYTE
the tag is: 236
bitmatrix resultText is: Meiji University, kikn lab.
bitmatrix RawBytes is: [B@368102c8
bitmatrix NumBits is: 440
bitmatrix mark is: 0
warning: No signature QR code!!!!!!
```

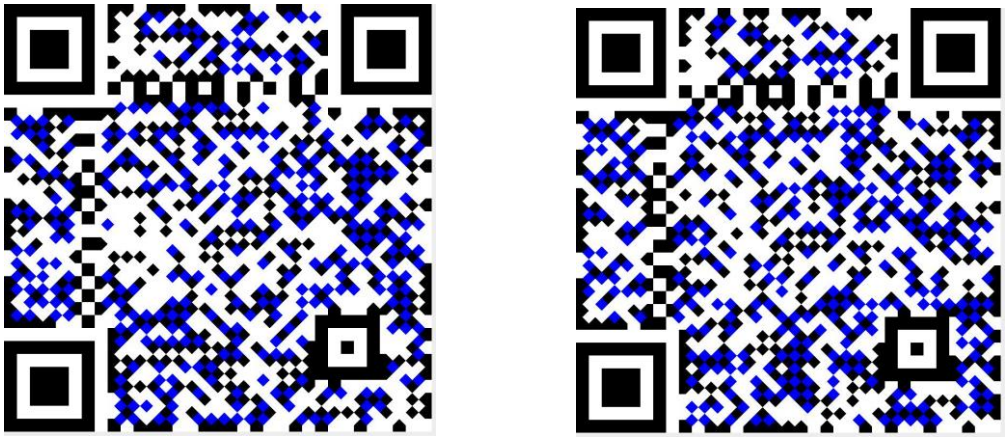
図 6.3:　電子署名なしＱＲコードの読み取り結果例

```
numData is :428
mode is :BYTE
bitmatrix resultText is: Meiji University, kikn lab.
bitmatrix RawBytes is: [B@368102c8
bitmatrix NumBits is: 440
bitmatrix mark is: 1
There is a signature in QR code's subcell
subcell data:[B@dcf3e99
whole subcell code word is:[B@dcf3e99
the subcell data(signature) is:
 ..XX.... ..X.XX.X ......X. ...X.X.. .XX...XX ....XX.X ...XXX.X .X.X.X.. .XXXXXX. .XXXX.XX .X.X.XXX .X.XX..X .XXXXX.. ..
signature is:[B@75a1cd57
signature ecdsa verification:true
PUBLIC_KEY:MD4wEAYHKoZIzj0CAQYFK4EEAAkDKgAEBI79Kf7Xt1jJElyZkOKabVPsl2gvK4uWrWvInIT7U5S2Z7BYMn9w1Q==
```

図 6.4: 電子署名があるＱＲコードの読み取り結果例

図 6.3，図 6.4 が示しているような，電子署名なしの場合，「No signature QR code!!!」という警告が出される．一方，電子署名がある場合，電子署名の公開鍵 ID と公開鍵の内容が表示されている．この例では，署名が正しく検証されていることを示している．

図 6.5 では，ＱＲコードの署名を偽造した例である．左は鍵 ID を用いてＱＲコードのメッセージに対する署名を埋め込んだ例である．右のＱＲコードには左と同じメッセージは入っているが，署名用鍵を持っていない偽造者が自ら自分の鍵で署名を埋め込んだ例である．その検証結果を図 6.6，図 6.7 で示す．



（正しい署名が埋め込んだＱＲコード）　（署名を偽造したＱＲコード）

図 6.5: ＱＲコードの署名を偽造する例

```
numData is :244
mode is :BYTE
bitmatrix resultText is: Meiji University, kikn lab.
bitmatrix RawBytes is: [B@368102c8
bitmatrix NumBits is: 512
bitmatrix mark is: 1
There is a signature in QR code's subcell
subcell data:[B@dcf3e99
whole subcell code word is:[B@dcf3e99
the subcell data(signature) is:
 ..XX.... ..X.XXX. ......X. ...X.X.X ........ XXX.XXXX XXX.X... .....X.. X..XXX.X X...X..X XX...X.. X.X....X XX..X.X. .X
signature is:[B@75a1cd57
signature ecdsa verification:true
PUBLIC_KEY:MD4wEAYHKoZIzj0CAQYFK4EEAAkDKgAEBI79Kf7Xt1jJElyZkOKabVPsl2gvK4uWrWvInIT7U5S2Z7BYMn9w1Q==
```

図 6.6: 正しい署名が埋め込んだＱＲコードの読み取り結果例

```
numData is :244
mode is :BYTE
bitmatrix resultText is: Meiji University, kikn lab.
bitmatrix RawBytes is: [B@368102c8
bitmatrix NumBits is: 512
bitmatrix mark is: 2
There is a signature in QR code's subcell
subcell data:[B@dcf3e99
whole subcell code word is:[B@dcf3e99
the subcell data(signature) is:
 ..XX.... ..X.XX.. ......X. ...X.X.. ...XX..X X.XX.... XXX..XX. .XXXX..X XX..XX.X XXX..... X.X..XX. .X..X.X. X...X.X. .X
signature is:[B@75a1cd57
signature ecdsa verification:false
PUBLIC_KEY:MD4wEAYHKoZIzj0CAQYFK4EEAAkDKgAEBI79Kf7Xt1jJElyZkOKabVPsl2gvK4uWrWvInIT7U5S2Z7BYMn9w1Q==
```

図 6.7:署名を偽造したＱＲコードの読み取り結果例

図 6.6，図 6.7 が示しているように，正しい署名，署名 ID を埋め込んだ場合，署名が正しく検証される．一方，偽造したＱＲコードは検証失敗になる．

通常なＱＲコードリーダーから，両方のＱＲコードは同じメッセージ読み取れるが，署名を埋め込み，署名を検証することで，ＱＲコードを偽造したことが分かる．

第７章　　耐性実験

本章では，菱形サブセルを用いるＱＲコードが通常なＱＲコードと比べ，誤りに対する耐性が劣化されたかについて検証する．

誤りに対する耐性はＱＲコードのサイズ，解像度，読み取り機械などの要素を関係している．菱形サブセルを生成する時，通常なＱＲコードセルに対する影響を明らかにするため，以下の実験を行う．

実験１　誤り量に対する評価
実験２　解像度に対する評価

7.1. 実験１：誤り増加の影響実験

誤りの耐性はＱＲコードのバージョン，誤り訂正レベルに関係している．

バージョン 3，誤り訂正レベルがＬのＱＲコードを例として説明する．

この時の格子部セルの誤り訂正コード数も中央部セルの誤り訂正コード数と同じである．

ＱＲコードの画像の解像度は 800×800 ピクセルとする．誤りに使われているマークのサイズを変更した時，誤りの耐性を評価する．誤りに使われているマークを図 7.1 に示す．マークのサイズはセル長で測る．

図 7.1: 誤り用マーク

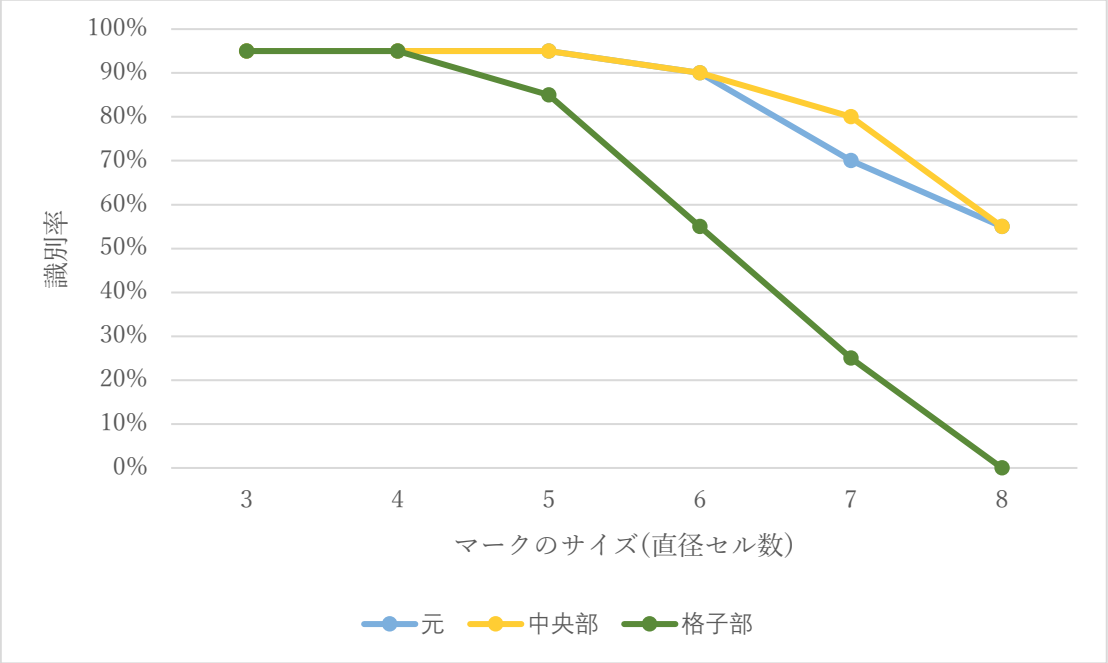


図 7.2: 誤りの大きさによる識別成功率

図 7.2 はマークのサイズにより，読み取り成功率が変化している．マークの位置をランダムで貼り付ける．各マークサイズに対して各二十回実験を行う．ここで通常のＱＲコードの読み取り成功率と菱形サブセルを用いるＱＲコードの中央部セルの読み取り成功率と格子部セルの読み取り成功率を示している．

円直径が 7 セル数の時の例を図 7.3 に示す．上列が通常のＱＲコード，下列が菱形サブセルの例である．

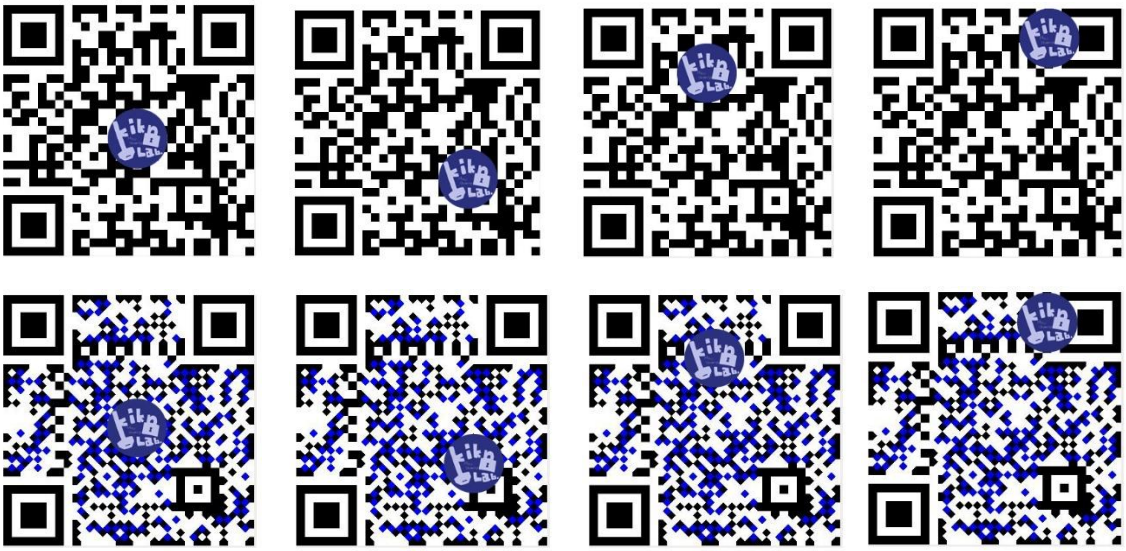


図 7.3: 誤り耐性実験の例(マーク直径 7 セル)

実験から読み取りの成否は誤りの位置，誤りの大きさに関わることが分かった．位置合わせパターンが被られた時，ＱＲコードとして認識されないことがあり，識別が失敗する．識別失敗した例を図 7.4 に示す．

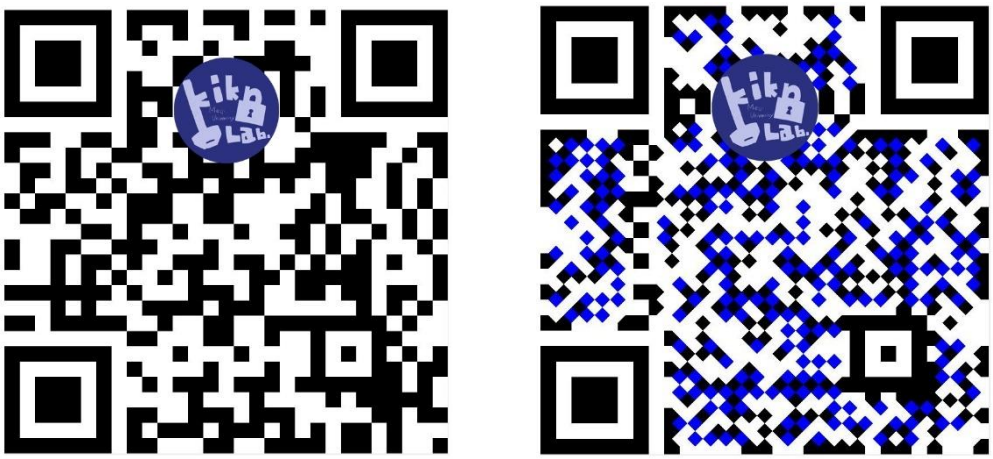


図 7.4: 識別失敗したＱＲコード
（マーク直径 7 セル）

位置合わせパターンなどが被られた時を除くと，菱形サブセルの追加により中央部セルの読み取りへの影響は生じていなかった．しかし，格子部セルに埋め込まれた電子署名の読み取り成功率は中央部セルの読み取り成功率より低い．その理由の一つは誤りに影響される格子部セルの数は中央部セルより多い場合があるからと考えられる．もう一つの理由は，中央部セルを埋め込む時常に二列ずつで埋め込んだに対し格子部セルを一行ずつで埋め込んでいる．図 7.5 で示しているように，一つの符合語を同じ色と見なす．青い円形がマークを示す．マークに影響される中央部コード数が 2 に対し，格子部コードの数は 5 になる．つまり，中央部には誤りが 2 つになる時，格子部に 4 つの誤りが発生されている可能性がある．もちろん，誤りは丁度良くセルの周辺に沿い発生しているわけではないから，中央部の誤り数は 2 に限らない．

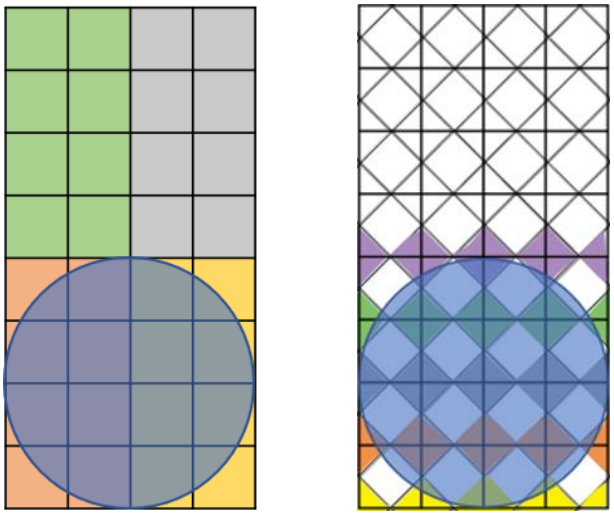


図 7.5: 誤りの影響
左：格子部がない場合　右：格子部がある場合

その対策として，格子部セルの誤り訂正能力を中央部セルの誤り訂正能力以上を設置すること．もう一つは，中央部セルのように，二列又は二行ずつで埋め込むことが考えられる．

7.2. 実験 2：解像度に関する実験

本実験では，解像度がＱＲコードの読み取り性能に及ぼす影響を明らかにする．

菱形サブセルの生成により，セルのサイズは小さくなったため，リーダーが遠所にある時，認識される画像の解像度が低下して認識誤りを引き起こす可能性が考えている．解像度が低くなると，菱形サブセルを用いるＱＲコードの識別が元のＱＲコードより劣ると予想する．

本実験は，800×800 ピクセルのＱＲコードから，解像度を下げ，通常のＱＲコードと菱形サブセルを用いるＱＲコードの読み取り精度を評価する．

ＱＲコードの例を図 7.6 に示す．

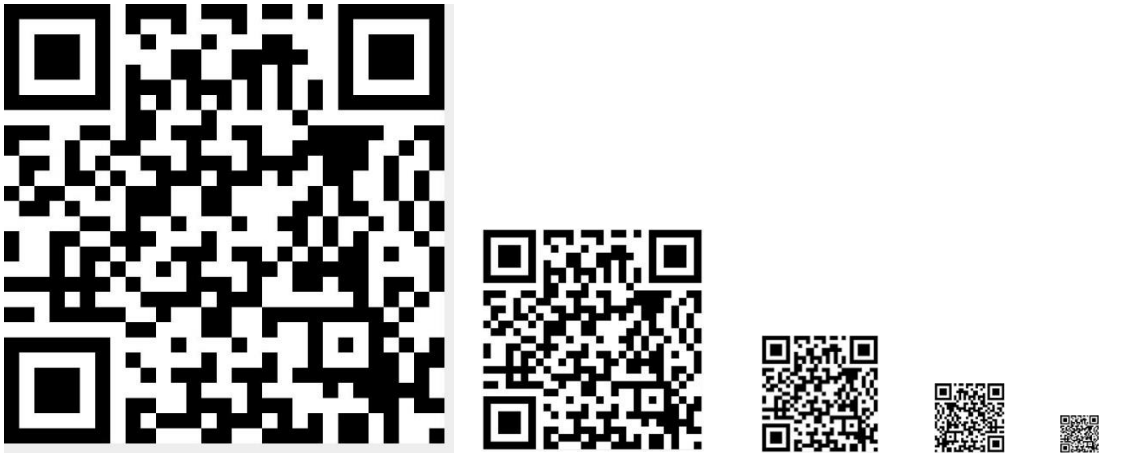


図 7.6: 解像度によるＱＲコードの例
（上列：通常なＱＲコード
下列：菱形サブセルを用いるＱＲコード）

読み取り結果を表 7.1 に示す．各解析度下に，メッセージのサイズを変わって，実験を 10 回行った．メッセージのサイズは 5 文字から 50 文字まで 5 文字長さを増やす．予想通り，菱形サブセルを用いるＱＲコードは通常なＱＲコードより先に周辺セルの侵食による識別ができなくなる．

表 7.1: 解像度による識別結果

解像度 （ピクセル）	通常なＱＲコード 識別率(%)	菱形サブセルを用いる ＱＲコード識別率(%)
800×800	100	100
600×600	100	100
400×400	100	100
300×300	100	100
250×250	100	100
200×200	100	100
150×150	100	0
100×100	100	0
80×80	100	0
60×60	0	0

第８章　　結論

本研究では菱形サブセルを用いるＱＲコードの生成方法を設計し，生成ソフトウェアを開発した．誤りに対する耐性実験を行った．

菱形サブセルの追加により，中央部セルの読み取りへの影響は生じていなかった．１例だけ，通常のＱＲコードは識別出来なかったが，菱形サブセルの追加したＱＲコードの中央部セルの識別が成功した．また，格子部セルは誤りに対する耐性が中央部セルより弱いことが分かった．マークのサイズ(直径)は６セルの時，通常のＱＲコードの識別率は 90%，中央部セルの識別率は 90%の対し，格子部セル識別率は 55%に下がった．その原因は，誤りは格子部セルに対する影響は中央部セルより大きいと考えられる．

全ての実験は画像を人工的に操作して模擬的に行っており，実機でカメラを用いて評価したわけではない．今後はアプリケーションの開発し，より現実的な環境での評価が必要であろう．

電子署名をＱＲコードに追加することによって，偽造，データの改ざんは防ぐことが可能である．菱形サブセルを用いるＱＲコードを使うことで，現在使われているＱＲコードを徐々に置換していくことが可能である．

耐性実験からＱＲコードのサイズが十分であれば，格子部セルは中央部セルのデータを読み取ることに影響が小さいことが分かった．また，格子部セルの誤りに対する耐性を増やすには，格子部セルの構造を変えるか，格子部セル誤り訂正能力を上がるかの二つの方法が考えられる．ＱＲコードのサイズが小さい時，格子部セルは中央部セルのデータを読み取ることに影響がある．格子部セルが周辺セルへの侵食があったと考えられる．

謝辞

本稿をまとめるにあたって多くの方のご指導・ご協力を賜りました.

指導教員である明治大学総合数理学部の菊池浩明教授には, 研究に対する指導や, 多くの成長する機会を与えていただき感謝いたします. テララコード研究所の寺浦信之様から研究に対する指導や助言に感謝いたします。本研究に様々な助言をしていただいた菊池研の皆様に感謝いたします.

最後に著者の学生生活を支えてくれた家族と友人に感謝の意を表すると共に, 謝辞にかえさせていただきます.

参考文献

[1] 日本工業規格.JIS X0510:2004 二次元コードシンボル-ＱＲコード-基本仕様

[2] 柏井祐樹，渡辺優平，森井昌克．オフラインサイト認証可能なＱＲコード．情報科学技術フォーラム(FIT), pp.107-112, 2012

[3] 先名健一. 個人認証機能を有するデジタル署名型ＱＲコード. 電子情報通信学会, pp.61-66, 2017

[4] 寺浦 信之，越前 功，岩村 恵市．菱形サブセルを用いたＱＲシンボルの互換性を保つ領域分割による容量拡大と電子署名の実装検討．コンピュータセキュリティシンポジウム 2019(CSS2019), pp.17-24, 2019

[5] 寺浦信之, 櫻井幸一．多値セル型二次元コードでの多分割領域への複数ユーザのアクセス制御．情報処理学会論文誌，Vol.57, No.9, pp.1965–1973, 2015.

[6] R.L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM. 21 (2): 120–126, 1978

[7] Taher Elgamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. IEEE Transactions on Information Theory: 469-472, 1985

[8] National Institute of Standards and Technology. Digital Signature Standard. Federal Information Processing Standard Publication 186, 1991

[9] C. P. Schnorr. Efficient signature generation by smart cards. International Association for Cryptologic Research: 161-174, 1991

[10] Mohsen Bafandehkar, Sharifah Md Yasin, Ramlan Mahmod, Zurina Mohd Hanapi . Comparison of ECC and RSA Algorithm in Resource Constrained Devices. International Conference on IT Convergence and Security (ICITCS), 2013

[11] Pranoti Panchal, Savitri Patil. Android Mobile Security using Secure Hash Algorithm, International Journal of Computer Science and Mobile Computing (IJCSMC), Vol. 5, Issue. 1 pp.226 – 232, 2016.

[12] Vaidhyesh P.S, SECURING IoT DEVICES BY GENERATING Q R CODES, International Journal of Pure and Applied Mathematics Volume 119 No. 12, pp.13743-13749, 2018.

[13] https://scan.netsecurity.ne.jp/article/2011/10/13/27439.html(参照 2019.7.25)

[14] https://spc.jst.go.jp/news/170802/topic_2_05.html(参照 2019.7.25)

付録

サブセルの誤り訂正特性

型番	コード数	データ数	誤り訂正コード数	残余セル数	誤り訂正レベル	RS ブロック数	RS ブロック	誤り訂正率
1	/	/	/	/	/	/	/	/
2	/	/	/	/	/	/	/	/
3	63	48	15	4	L	1	(63,48,7)	11.1%
4	91	77	14	4	L	1	(91,77,7)	7.7%
		63	28		M	1	(45,31,7)	15.6%
						1	(46,32,7)	15.2%
5	123	105	18	4	L	1	(123,105,9)	7.3%
		83	40		M	1	(61,41,10)	16.4%
						1	(62,42,10)	16.1%
		59	64		Q	1	(30,14,8)	26.7%
						3	(31,15,8)	25.8%
6	159	135	24	4	L	1	(159,135,12)	7.5%
		107	52		M	1	(79,53,13)	16.5%
						1	(80,54,13)	16.25%
		79	80		Q	1	(39,19,10)	25.6%
						3	(40,20,10)	25.0%
		63	96		H	1	(39,15,12)	30.8%
						3	(40,16,12)	30.0%
7	180	154	26	0	L	1	(180,154,13)	7.2%
		124	28		M	2	(90,62,14)	15.5%
		84	48		Q	4	(45,21,12)	26.7%
		68	56		H	4	(45,17,14)	31.1%
8	224	192	32	0	L	2	(112,96,8)	7.1%
		152	72		M	4	(56,38,9)	16.1%
		112	112		Q	4	(56,28,14)	25%
		84	140		H	1	(44,16,14)	31.8%
						4	(45,17,14)	31.1%

9	272	232	40	0	L	2	(136,116,10)	7.4%
		184	44		M	4	(68,46,11)	16.2%
		128	144		Q	4	(45,21,12)	26.7%
						2	(46,22,12)	26.1%
		96	176		H	8	(34,12,11)	32.4%
10	324	276	48	0	L	2	(162,138,12)	7.4%
		220	104		M	4	(81,55,13)	16.0%
		148	176		Q	4	(40,18,11)	27.5%
						4	(41,19,11)	26.8%
		116	208		H	4	(40,14,13)	32.5%
						4	(41,15,13)	31.7%
11	380	324	56	0	L	2	(190,162,14)	7.4%
		260	120		M	4	(95,65,15)	15.8%
		172	208		Q	4	(47,21,13)	27.7%
						4	(48,22,13)	27.1%
		140	240		H	4	(47,17,15)	31.9%
						4	(48,18,15)	31.3%
12	440	376	64	0	L	4	(110,94,8)	7.3%
		300	140		M	5	(88,60,14)	15.9%
		216	224		Q	8	(55,27,14)	25.5%
		160	280		H	10	(44,16,14)	31.8%
13	504	424	80	0	L	4	(126,106,10)	7.9%
		344	160		M	8	(63,43,10)	15.9%
		244	260		Q	6	(50,24,13)	26.0%
						4	(51,25,13)	25.5%
		166	338		H	3	(38,12,13)	34.2%
						10	(39,13,13)	33.3%
14	543	463	80	4	L	1	(135,115,10)	7.4%
						3	(136,116,10)	7.4%
		363	180		M	6	(60,40,10)	16.7%
						3	(61,41,10)	16.4%
		263	280		Q	7	(54,26,14)	25.9%
						3	(55,27,14)	25.5%
		213	330		H	7	(49,19,15)	30.6%
						4	(50,20,15)	30.0%

15	615	525	90	4	L	5	(123,105,9)	7.3%
		423	192		M	1	(76,52,12)	15.8%
						7	(77,53,12)	15.6%
		307	308		Q	1	(43,21,11)	25.6%
						13	(44,22,11)	25.0%
		231	384		H	9	(38,14,12)	31.6%
						7	(39,15,12)	30.8%
16	691	591	100	4	L	4	(138,118,10)	7.2%
						1	(139,119,10)	7.2%
		471	220		M	9	(69,47,11)	15.9%
						1	(70,48,11)	15.7%
		331	360		Q	5	(57,27,15)	26.3%
						7	(58,28,15)	25.9%
		259	432		H	11	(38,14,12)	31.6%
						7	(39,15,12)	30.8%
17	771	661	110	4	L	4	(154,132,11)	7.1%
						1	(155,133,11)	7.1%
		531	240		M	9	(77,53,12)	15.6%
						1	(78,54,12)	15.4%
		355	416		Q	13	(48,22,13)	27.1%
						3	(49,23,13)	26.5%
		291	480		H	13	(48,18,15)	31.3%
						3	(49,19,15)	30.6%
18	855	735	120	4	L	5	(171,147,12)	7.0%
		595	260		M	5	(85,59,13)	15.3%
						5	(86,60,13)	15.1%
		405	450		Q	15	(57,27,15)	26.3%
		315	540		H	9	(47,17,15)	31.9%
						9	(48,18,15)	31.3%
19	943	799	144	4	L	5	(157,133,12)	7.6%
						1	(158,134,12)	7.6%
		643	300		M	7	(94,64,15)	16.0%
						3	(95,65,15)	15.8%
		463	480		Q	1	(58,28,15)	25.9%
						15	(59,29,15)	25.4%

		343	600		H	17	(47,17,15)	31.9%
						3	(48,18,15)	31.3%
20	1035	879	156	4	L	3	(172,146,13)	7.6%
						3	(173,147,13)	7.5%
		699	336		M	9	(86,58,14)	16.3%
						3	(87,59,14)	16.1%
		495	540		Q	9	(57,27,15)	26.3%
						9	(58,28,15)	25.9%
		405	630		H	15	(49,19,15)	30.6%
						6	(50,20,15)	30.0%
21	1094	940	154	0	L	5	(156,134,11)	7.1%
						2	(157,135,11)	7.0%
		742	352		M	10	(68,46,11)	16.2%
						6	(69,47,11)	15.9%
		534	560		Q	6	(54,26,14)	25.9%
						14	(55,27,14)	25.5%
		434	660		H	6	(49,19,15)	30.6%
						16	(50,20,15)	30.0%
22	1194	1018	176	0	L	6	(149,127,11)	7.4%
						2	(150,128,11)	7.3%
		810	384		M	6	(74,50,12)	16.2%
						10	(75,51,12)	16.0%
		594	600		Q	6	(59,29,15)	25.4%
						14	(60,30,15)	25.0%
		474	720		H	6	(49,19,15)	30.6%
						18	(50,20,15)	30.0%
23	1298	1106	192	0	L	6	(162,138,12)	7.4%
						2	(163,139,12)	7.4%
		882	416		M	14	(81,55,13)	16.0%
						2	(82,56,13)	15.9%
		638	660		Q	22	(59,29,15)	25.4%
		518	780		H	2	(49,19,15)	30.6%
						24	(50,20,15)	30.0%
24	1406	1208	198	0	L	7	(156,134,11)	7.1%
						2	(157,135,11)	7.0%

		958	448		M	2	(87,59,14)	16.1%
						14	(88,60,14)	15.9%
		686	720		Q	10	(58,28,15)	25.9%
						14	(59,29,15)	25.4%
		536	870		H	15	(48,18,15)	31.3%
						14	(49,19,15)	30.6%
25	1518	1298	220	0	L	2	(151,129,11)	7.3%
						8	(152,130,11)	7.2%
		1038	480		M	2	(75,51,12)	16.0%
						18	(76,52,12)	15.8%
		734	784		Q	22	(54,26,14)	25.9%
						6	(55,27,14)	25.5%
		558	960		H	18	(47,17,15)	31.9%
						14	(48,18,15)	31.3%
26	1634	1394	240	0	L	10	(136,116,10)	7.4%
						2	(137,117,10)	7.3%
		1130	504		M	4	(90,62,14)	15.6%
						14	(91,63,14)	15.4%
		802	832		Q	30	(51,25,13)	25.5%
						2	(52,26,13)	25.0%
		626	1008		H	22	(45,17,14)	31.1%
						14	(46,18,14)	30.4%
27	1754	1490	264	0	L	10	(146,124,11)	7.5%
						2	(147,125,11)	7.5%
		1222	532		M	13	(92,64,14)	15.2%
						6	(93,65,14)	15.1%
		870	884		Q	14	(51,25,13)	25.5%
						20	(52,26,13)	25%
		674	1080		H	10	(48,18,15)	31.3%
						26	(49,19,15)	30.6%
28	1831	1571	260	4	L	9	(183,157,13)	7.1%
						1	(184,158,13)	7.1%
		1271	560		M	9	(91,63,14)	15.4%
						11	(92,64,14)	15.2%
		895	936		Q	5	(50,24,13)	26%
						31	(51,25,13)	25.5%

		711	1120		H	9	(45,17,14)	31.1%
						31	(46,18,14)	30.4%
29	1959	1671	288	4	L	9	(163,139,12)	7.4%
						3	(164,140,12)	7.3%
		1359	600		M	1	(97,67,15)	15.5%
						19	(98,68,15)	15.3%
		951	1008		Q	21	(54,26,14)	25.9%
						15	(55,27,14)	25.5%
		759	1200		H	1	(48,18,15)	31.3%
						39	(49,19,15)	30.6%
30	2091	1791	300	4	L	9	(139,119,10)	7.2%
						6	(140,120,10)	7.1%
		1431	660		M	21	(95,65,15)	15.8%
						1	(96,66,15)	15.6%
		969	1122		Q	51	(41,19,11)	26.8%
		803	1288		H	25	(45,17,14)	31.1%
						21	(46,18,14)	30.4%
31	2227	1875	352	4	L	13	(139,117,11)	7.9%
						3	(140,118,11)	7.9%
		1507	720		M	23	(74,50,12)	16.2%
						7	(75,51,12)	16.0%
		1107	1120		Q	13	(55,27,14)	25.5%
						27	(56,28,14)	25.0%
		877	1350		H	23	(49,19,15)	30.6%
						22	(50,20,15)	30.0%
32	2367	2007	360	4	L	9	(197,167,15)	7.6%
						3	(198,168,15)	7.6%
		1647	720		M	3	(78,54,12)	15.4%
						27	(79,55,12)	15.2%
		1167	1200		Q	7	(59,29,15)	25.4%
						33	(60,30,15)	25.0%
		911	1456		H	25	(45,17,14)	31.1%
						27	(46,18,14)	30.4%
33	2511	2151	360	4	L	9	(167,143,12)	7.2%
						6	(168,144,12)	7.1%

		1731	780		M	9	(83,57,13)	15.7%
						21	(84,58,13)	15.5%
		1251	1260		Q	9	(59,29,15)	25.4%
						33	(60,30,15)	25.0%
		951	1560		H	37	(48,18,15)	31.3%
						15	(49,19,15)	30.6%
34	2659	2269	390	4	L	11	(177,151,13)	7.3%
						4	(178,152,13)	7.3%
		1795	864		M	5	(73,49,12)	16.4%
						31	(74,50,12)	16.2%
		1309	1350		Q	41	(59,29,15)	25.4%
						4	(60,30,15)	25.0%
		1039	1620		H	41	(49,19,15)	30.6%
						13	(50,20,15)	30.0%
35	2756	2366	390	0	L	4	(183,157,13)	7.1%
						11	(184,158,13)	7.1%
		1916	840		M	4	(91,63,14)	15.4%
						26	(92,64,14)	15.2%
		1378	1378		Q	53	(52,26,13)	25.0%
		1076	1680		H	4	(45,17,14)	31.1%
						56	(46,18,14)	30.4%
36	2912	2472	440	0	L	8	(145,123,11)	7.6%
						12	(146,124,11)	7.5%
		2012	900		M	28	(97,67,15)	15.5%
						2	(98,68,15)	15.3%
		1456	1456		Q	56	(52,26,13)	25.0%
		1112	1800		H	28	(48,18,15)	31.3%
						32	(49,19,15)	30.6%
37	3072	2632	440	0	L	8	(153,131,11)	7.2%
						12	(154,132,11)	7.1%
		2112	960		M	32	(96,66,15)	15.6%
		1536	1536		Q	64	(48,24,12)	25.0%
		1212	1860		H	28	(49,19,15)	30.6%
						34	(50,20,15)	30.0%
38	3236	2768	468	0	L	4	(179,153,13)	7.3%

						14	(180,154,13)	7.2%
		2216	1020		M	28	(95,65,15)	15.8%
						6	(96,66,15)	15.6%
		1616	1620		Q	4	(59,29,15)	25.4%
						50	(60,30,15)	25.0%
		1286	1950		H	14	(49,19,15)	30.6%
						51	(50,20,15)	30.0%
39	3404	2900	504	0	L	16	(189,161,14)	7.4%
						2	(190,162,14)	7.4%
		2324	1080		M	16	(94,64,15)	16.0%
						20	(95,65,15)	15.8%
		1694	1710		Q	16	(59,29,15)	25.4%
						41	(60,30,15)	25.0%
		1214	2190		H	27	(46,16,15)	32.6%
						46	(47,17,15)	31.9%
40	3576	3056	520	0	L	4	(178,152,13)	7.3%
						16	(179,153,13)	7.3%
		2456	1120		M	24	(89,61,14)	15.7%
						16	(90,62,14)	15.6%
		1776	1800		Q	24	(59,29,15)	25.4%
						36	(60,30,15)	25.0%
		1416	2160		H	24	(49,19,15)	30.6%
						48	(50,20,15)	30.0%